Содержание

1. Бизнес-анализ 5

1.1. Интервьюирование заказчика и пользователей 5

1.2. Описание текущей ситуации 6

1.3. Описание будущей ситуации 8

1.4. Описание заинтересованных лиц и конечных пользователей 9

1.5. Бизнес-требования 10

2. Анализ существующих решений 12

2.1. Обзор существующих решений 12

2.1.1. Решение 1 «1С: Военкомат» 12

2.2. Сравнительные анализ решений 12

2.3. Выводы по анализу 13

3. Концепция решения 15

3.1. Техническое задание 15

3.1.1. Общие сведения 15

3.1.2. Назначение и цели создания системы 15

3.1.2.1. Назначение системы 15

3.1.2.2. Цели системы 15

3.1.3. Требования к системе 16

3.1.3.1. Требования к структуре и функционированию системы. 16

3.1.3.2. Требования к численности и квалификации пользователей 16

3.1.3.3. Требования к составу и параметрам технических средств 17

3.1.3.4. Требования к информационной и программной совместимости 17

Требования перечислены в Таблице 3.1. 17

3.1.3.5. Требования к защите, безопасности и сохранности данных. 17

3.1.4. Состав и содержание работ по созданию системы 17

3.2. Модель прецедентов 18

3.2.1. Акторы 18

3.2.2. Диаграмма прецедентов 18

3.2.3. Сценарии использования 19

3.3. Customer Journey Mapping 21

3.4. User Story Mapping 22

3.5. Анализ факторов риска проекта 23

4. Проектирование решения 26

4.1. Описание архитектуры системы 26

4.2. Описание возможных технических решений 26

4.3. Проектирование интерфейса пользователя 30

4.4. Проектирование базы данных 31

5. Программная реализация системы 33

Приложение 1 – Листинг базы данных 34

Приложение 2 – Листинг программной реализации системы 39

1. Бизнес-анализ
   1. Интервьюирование заказчика и пользователей

Для успешного проектирования и разработки системы учета военнообязанных было проведено интервьюирование заказчика. Целью интервью было выявление ключевых потребностей, проблем и ожиданий, связанных с процессом хранения информации о военнообязанных, формирования повесток, а также анализ текущих процессов и их недостатков.

Вопросы и ответы заказчика представлены в Таблице 1.1.

Таблица 1.1 – Вопросы и ответы заказчика.

|  |  |
| --- | --- |
| Вопрос | Ответ |
| Какие основные проблемы вы видите в текущем процессе учета военнообязанных? | Основные проблемы включают сложность управления записями военнообязанных, отсутствие автоматизации в процессе формирования повесток и ручное ведение учета призывных комиссий, что затрудняет принятие управленческих решений. |
| Какие функции вы считаете обязательными для новой системы? | Обязательными функциями являются: управление записями военнообязанных, автоматизация формирования повесток, возможность добавления и редактирования данных о призывных комиссиях. |
| Какие требования к производительности системы вы предъявляете? | Система должна работать стабильно даже в пиковые периоды (например во время призыва) и поддерживать одновременную работу нескольких пользователей. |

Вопросы и ответы пользователей представлены в Таблице 1.2.

Таблица 1.2 – Вопросы и ответы пользователей

|  |  |
| --- | --- |
| Вопрос | Ответ |
| Какие функции новой системы облегчили бы вашу работу? | Удобный интерфейс для работы с записями военнообязанных, автоматизация формирования повесток и быстрый доступ к информации о призывных комиссиях. |

На основе проведенного интервьюирования были выявлены ключевые проблемы и потребности заказчика и пользователей. Основные требования к системе включают:

Упрощение процесса учета военнообязанных.

* автоматизация формирования повесток и учета призывных комиссий;
* интеграция с государственными базами для актуализации информации;
* удобный интерфейс для сотрудников.

Эти данные будут использованы для проектирования функциональных и нефункциональных требований к системе, а также для разработки архитектуры модуля учета военнообязанных.

* 1. Описание текущей ситуации

Для работы военкомата и удовлетворенность клиентов. На основе проведенного интервьюирования и анализа бизнес-процессов были выявлены ключевые трудности и их последствия.

Основные проблемы:

1. Частые сбои в работе системы:

* Проблема: Существующая система учета военнообязанных часто зависает или выходит из строя, что приводит к остановке работы.
* Воздействует на: Администрацию военкомата и сотрудников.
* Результатом чего является: Потеря времени, недовольство сотрудников, снижение эффективности работы.
* Сейчас решается: Перезагрузка системы или временный переход на ручное оформление записей.
* Имеющиеся решения не: обеспечивают стабильность и надежность работы системы.

1. Отсутствие интеграции с другими системами:

* Проблема: Текущая система не интегрирована с системами учета военнообязанных, управления расписанием повесток и платежными системами.
* Воздействует на: Администрацию военкомата и сотрудников.
* Результат: Необходимость ручного ввода данных, ошибки при оформлении записей, сложность генерации отчетов.
* Текущее решение: Ручной перенос данных между системами, что занимает много времени и приводит к ошибкам.
* Имеющиеся решения не: автоматизируют процесс и не обеспечивают точность данных.
* Текущая ситуация демонстрирует необходимость внедрения информационной системы, которая устранит существующие проблемы и повысит эффективность процессов.
  1. Описание будущей ситуации

После успешного внедрения модуля информационной системы учета военнообязанных ожидается повышение эффективности сотрудников. Будущая ситуация будет характеризоваться следующими изменениями:

Основные улучшения:

1. Повышение стабильности и надежности системы:

* Как будет: Система будет разработана с учетом высокой нагрузки и пиковых периодов, что обеспечит ее стабильную работу без сбоев.
* Результат: Снижение количества простоев, повышение доверия пользователей к системе.

1. Удобный интерфейс:

* Как будет: Система будет оснащена интуитивно понятным интерфейсом, который упростит работу кассиров и позволит сотрудникам быстро находить нужную информацию.
* Результат: Снижение времени обучения сотрудников, повышение удобства для клиентов.

Внедрение новой информационной системы учет военнообязанных позволит существенно улучшить ключевые показатели эффективности работы военкомата.

Основные ожидаемые результаты включают:

* Повышение стабильности и надежности системы.
* Автоматизация процессов.
* Эффективное управление данными.
* Удобный интерфейс.

Эти изменения обеспечат повышение эффективности сотрудников. Описание будущей ситуации служит мотивацией для проектной команды и заказчика, а также обеспечивает контекст для принятия решений на всех этапах проекта.

* 1. Описание заинтересованных лиц и конечных пользователей

Для успешного проектирования и разработки модуля информационной системы необходимо определить всех заинтересованных лиц и конечных пользователей системы. В таблице ниже представлены ключевые заинтересованные лица, их роли, ответственность и вовлеченность в проект.

Таблица 1.5 – Описание заинтересованных лиц и конечных пользователей.

|  |  |  |
| --- | --- | --- |
| Представитель | Администрация военкомата | Сотрудник |
| Описание | |  |  | | --- | --- | | Руководство военкомата, отвечающее за общее управление и эффективность работы. |  | | Сотрудники, непосредственно работающие с системой. |
| Тип | Эксперт | Профессионал |
| Ответственность | Определение требований к системе, контроль за внедрением и эксплуатацией системы. | Использование системы для управления военнообязанными, повестками. |
| Критерий успеха | Повышение эффективности работы сотрудников, сокращение времени ввода и поиска данных, увеличение выручки. | Удобство работы с системой, минимизация ошибок, сокращение времени на оформление. |
| Вовлеченность | Рецензирование требований, утверждение архитектурных решений, контроль за внедрением. | Тестирование системы, предоставление обратной связи по удобству интерфейса. |
| Поставляемые артефакты | Аналитика по эффективности работы системы. | Удобный интерфейс для работы, быстрый доступ к данным. |
| Комментарии / Проблемы | Недостаточная техническая подготовка для оценки сложных технических решений. | Необходимость обучения работе с новой системой. |

Определение заинтересованных лиц и их ролей в проекте позволяет четко понять, кто будет использовать систему, какие требования они предъявляют и как их вовлечь в процесс разработки. Основные заинтересованные лица включают администрацию военкомата и сотрудников военкомата. Каждая группа имеет свои критерии успеха и вовлеченность в проект, что необходимо учитывать при проектировании и разработке системы.

* 1. Бизнес-требования

Бизнес-требования к модулю информационной системы автоматизации процесса учета военнообязанных формируются на основе ключевых бизнес-целей заказчика (администрации военкомата) и ожиданий других заинтересованных лиц. Эти требования отражают основные направления развития проекта и определяют, какие функции системы будут реализованы для достижения поставленных целей.

Ключевые бизнес-цели:

1. Повышение эффективности работы:

* Цель: Увеличение эффективности за счет сокращения времени на заполнение и поиск данных.
* Бизнес-требование: Система должна обеспечивать быструю запись военнообязанных с минимальным временем отклика (не более 2 секунд).
* Ключевые функции (features):
* Быстрый поиск нужных данных.
* Генерация документов.

1. Обеспечение стабильности и надежности системы:

* Цель: Минимизация простоев и сбоев в работе системы, особенно в пиковые периоды.
* Бизнес-требование: Система должна быть устойчивой к высоким нагрузкам и обеспечивать стабильную работу 24/7.
* Ключевые функции (features):
* Поддержка работы в пиковые периоды (например, при осеннем призыве).
* Автоматическое резервирование данных для предотвращения потери информации.
* Мониторинг производительности системы в реальном времени.

Для наглядного представления связи бизнес-целей, требований и ключевых функций может быть использована карта воздействий (Impact Mapping (Рисунок 1.1)).

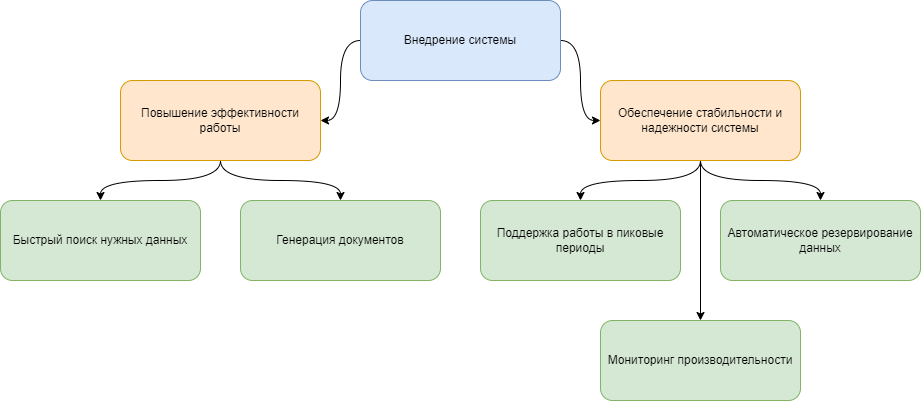


Рисунок 1.1 - ImpactMapping

1. Анализ существующих решений

Для решения проблемы учета военнообязанных было проведено исследование существующих приложений, предназначенных для этих целей. В результате были выделены их достоинства и недостатки.

* 1. Обзор существующих решений

В открытом доступе нет информации о программном обеспечении для военкоматов, поэтому проведем сравнение с предполагаемой системой: 1С: Военкомат.

* + 1. Решение 1 «1С: Военкомат»

Конфигурация 1С для управления призывниками. Продукт предназначен для отделов военкомата, занимающихся призывами и учетом призывников. Решение позволяет повысить эффективность работы сотрудников военкомата.

* + 1. Решение 2 «ВоенУчет»

Продукт предназначен для отделов военкомата, занимающихся призывами и учетом призывников. Решение позволяет повысить эффективность работы сотрудников военкомата.

* + 1. Решение 3 «Армия РФ»

Система для управления военкомата. Включает в себя все слои, в том числе отдел призывов. Решение позволяет повысить эффективность работы сотрудников военкомата.

* 1. Сравнительные анализ решений

Для сравнения рассмотренных решений были выделены следующие критерии, которые отражают ключевые бизнес-требования заказчика, описанные в разделе 1, а также наиболее приоритетные детализированные требования и ограничения:

* интуитивно понятный интерфейс — важный критерий для обеспечения удобства использования;
* стоимость лицензии — критичный фактор для мелких и новых компаний, которые стремятся минимизировать затраты;
* формирование договоров — функция, которая позволяет автоматизировать формирование документов;
* легкость интеграции — критерий, который определяет, насколько быстро и просто можно внедрить решение на предприятии;

На основе выделенных критериев было проведено сравнение рассмотренных решений. Результаты представлены в Таблице 2.1.

Таблица 2.1 – Сравнение решений

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критерии сравнения | 1С: Военкомат | ВоенУчет | Армия РФ | Разрабатываемая программа |
| Наличие интуитивно понятного интерфейса | - | + | - | + |
| Высокая стоимость лицензии | + | + | + | - |
| Формирование договоров | + | - | + | + |
| Легкость интеграции на предприятие | - | + | + | + |

* 1. Выводы по анализу

Проведенный анализ существующих решений позволил выявить их ключевые достоинства и недостатки, а также определить направления для разработки собственного решения, которое полностью удовлетворит требования заказчика. Было принято решение разработать собственное решение «с нуля». Разрабатываемое решение будет сочетать в себе лучшие черты, предлагая уникальные функции, которые полностью удовлетворят потребности заказчика. Это позволит создать конкурентоспособный продукт, оптимизирующий процессы хранения данных о призывниках, призывных комиссиях и повесток в военкомате.

1. Концепция решения
   1. Техническое задание
      1. Общие сведения

Полное наименование системы – Military Conscription System (Система военкомата)

Плановые сроки работы по созданию системы: 12.12.2024 – 31.01.2025.

* + 1. Назначение и цели создания системы
       1. Назначение системы

Система предназначена для автоматизации учета военнообязанных. Она позволит хранить и управлять данными о призывниках, повестках и призывных комиссиях, автоматизировать формирование повесток.

* + - 1. Цели системы

Основной целью создания системы является повышение эффективности работы сотрудников военкомата путем:

* ведения базы призывников;
* ведения базы призывов;
* автоматизации формирования повесток;
* повышения качества выполняемой работы;

Система позволит повысить эффективность и точность своей деятельность, предоставляя централизованный контроль над всеми аспектами.

* + 1. Требования к системе
       1. Требования к структуре и функционированию системы.

Программа должна обеспечивать возможность выполнения следующих функций:

* добавление пользователей (высокий уровень);
* авторизация пользователей (высокий уровень);
* просмотр призывников с возможностью поиска (высокий уровень);
* детальный просмотр призывников (высокий уровень);
* добавление и редактирование призывников (высокий уровень);
* просмотр повесток (высокий уровень);
* детальный просмотр и внесение доп. Данных в повестки (средний уровень);
* просмотр, добавление, удаление документов (средний уровень);
* формирование повестки (высокий уровень);
  + - 1. Требования к численности и квалификации пользователей

Минимальное количество персонала, требуемого для работы программы – 3 человека: системный администратор, член комиссии, врач.

Системный администратор должен иметь среднее специальное техническое образование. В перечень задач должны входить:

* поддержание работоспособности технических средств;
* задача установки программы;
* задача создания резервных копий базы данных.

Член комиссии и врач должны уметь работать с компьютером на уровне пользователя.

* + - 1. Требования к составу и параметрам технических средств

В состав технических средств должен входить IBM-совместный персональный компьютер (ПЭВМ), выполняющий роль сервера, включающий в себя:

* процессор с частотой не менее 2.0 hz;
* оперативная память не менее 4Гб;
* жесткий диск объемом не менее 256Гб.
  + - 1. Требования к информационной и программной совместимости

Требования перечислены в Таблице 3.1.

Таблица 3.1 – Требования к совместимости

|  |  |
| --- | --- |
| База данных | MSSQL |
| Язык программирования | C# (12 версия) |
| SDK | Net core 8.0 |
| IDE | VS 2022 |
| ОС | Windows (7,8,10,11) |

* + - 1. Требования к защите, безопасности и сохранности данных.

Программа должна обеспечивать защиту персональных данных пользователей и призывников, обрабатываемых в системе, в соответствии с требованиями действующего законодательства Российской Федерации.

* + 1. Состав и содержание работ по созданию системы

Состав и содержание работ представлены в Таблице 3.2.

Таблица 3.2 – Состав работ

|  |  |  |
| --- | --- | --- |
| Стадия | Сроки | Исполнитель |
| Анализ | 5 дня | Аналитик |
| Планирование | 5 дня | Аналитик |
| Проектирование и дизайн | 10 дня | Аналитик, программист |
| Разработка | 20 дней | Программист |
| Тестирование | 8 дней | Тестировщик, программист |
| Развертывание | 1 день | Программист |

* 1. Модель прецедентов
     1. Акторы

Для проектирования и разработки модуля автоматизации взаимодействия риэлторов и клиентов необходимо определить акторов, которые будут взаимодействовать с системой. Акторы — это пользователи, внешние системы или объекты, которые инициируют выполнение сценариев или играют пассивную роль в системе. В Таблице 3.3 представлены ключевые акторы, их типы и описательные характеристики.

Таблица 3.3 – Акторы системы

|  |  |  |
| --- | --- | --- |
| Наименование актора | Тип актора | Описательная характеристика |
| Администратор | Первичный (пользователь) | Сотрудник военкомата, который имеет доступ к управлению другими сотрудниками, призывными комиссиями, просмотр и управление повестками, а также военнообязанных. |
| Врач | Первичный (пользователь) | Сотрудник системы, имеет доступ к просмотру созданным повесткам, а также добавлению информации о медицинском обследовании. |
| Член комиссии | Первичный (пользователь) | Сотрудник системы, который имеет доступ к управлению призывниками, а также вынесению вердикта по вопросу призыва. |

* + 1. Диаграмма прецедентов

Диаграмма прецедентов представлена на Рисунке 3.1.

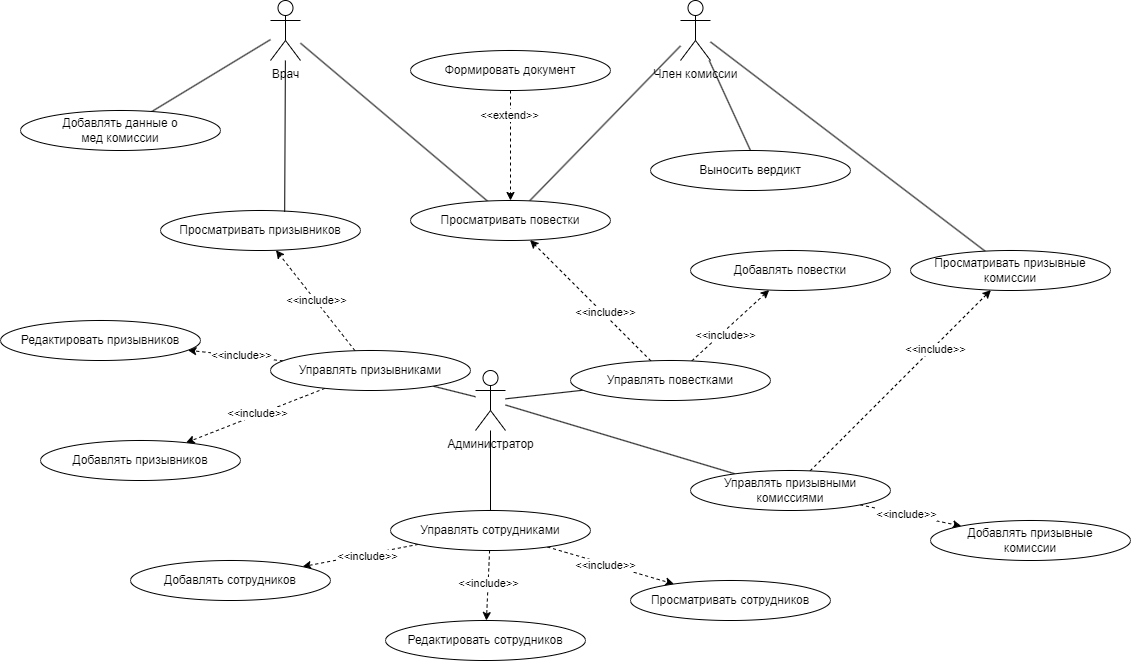


Рисунок 3.1 –Диаграмма прецендентов.

* + 1. Сценарии использования

Таблица 3.4 – Сценарий 1. «Авторизация»

|  |  |
| --- | --- |
| Описание | Авторизация пользователя |
| Идентификатор прецедента | UC 1.1 |
| Список акторов | Администратор, врач, член комиссии |
| Предусловия | Пользователь имеет учетную запись. |
| Действия пользователя | Реакция системы |
| Ввод данных | Отображение данных в полях. |
| Нажатие кнопки «Войти» | Поиск пользователя в базе. |
| Постусловие | Пользователь найден. |
| Включение | Отсутствует |
| Расширение | Отсутствует |
| Обобщение | Отсутствует |

Таблица 3.5 – Сценарий 2. «Добавление сотрудников»

|  |  |
| --- | --- |
| Описание | Добавление сотрудников |
| Идентификатор прецедента | UC 1.2 |
| Список акторов | Администратор. |
| Предусловия | Пользователь авторизирован. |
| Действия пользователя | Реакция системы |

Продолжение Таблицы 3.5.

|  |  |
| --- | --- |
| В меню нажатие кнопки «Сотрудники» | Переход на страницу со списком сотрудников. |
| В меню нажатие кнопки «Добавить» | Переход на страницу с формой добавления сотрудника. |
| Ввод данных | Отображение данных |
| Нажатие кнопки «Сохранить» | Сохранение данных в базу |
| Постусловие | Сотрудник с таким логином отсутствует в системе |
| Включение | Отсутствует |
| Расширение | Отсутствует |
| Обобщение | Отсутствует |

Таблица 3.6 – Сценарий 3. «Просмотр сотрудников»

|  |  |
| --- | --- |
| Описание | Просмотр сотрудников |
| Идентификатор прецедента | UC 1.3 |
| Список акторов | Администратор. |
| Предусловия | Пользователь авторизирован. |
| Действия пользователя | Реакция системы |
| В меню нажатие кнопки «Сотрудники» | Переход на страницу со списком сотрудников. |
| Постусловие | Отсутствует |
| Включение | Отсутствует |
| Расширение | Искать по полю |
| Обобщение | Отсутствует |

Таблица 3.7 – Сценарий 4. «Добавление мед. комиссии»

|  |  |
| --- | --- |
| Описание | Добавление данных о мед. комиссии |
| Идентификатор прецедента | UC 1.4 |
| Список акторов | Врач. |
| Предусловия | Пользователь авторизирован. |
| Действия пользователя | Реакция системы |
| В меню нажатие кнопки «Повестки». | Переход на страницу со списком повесток. |
| Выбор повестки | Переход на страницу с подробной информацией |

Продолжение Таблицы 3.7

|  |  |
| --- | --- |
| Нажатие кнопки «Добавить мед. комиссию» | Отображение формы для заполнения |
| Ввод данных | Отображение введенных данных. |
| Нажатие кнопки «Сохранить» | Сохранение данных в базу. |
| Постусловие | В повестке еще не должно быть мед комиссий |
| Включение | Отсутствует |
| Расширение | Отсутствует |
| Обобщение | Отсутствует |

* 1. Customer Journey Mapping

Customer Journey Mapping представлен на Рисунке 3.2.



Рисунок 3.2 – Customer Journey Mapping.

* 1. User Story Mapping

User Story Mapping представлен на Рисунке 3.3.



Рисунок 3.3 – User Story Mapping

* 1. Анализ факторов риска проекта

Анализ факторов риска проекта представлен в Таблице 3.8.

Таблица 3.8 – Анализ факторов риска

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Тип | Риск | Последствия | Меры по предотвращению | Меры по минимизации последствий | Вероятность свершения риска [P] | Степень ущерба [L] | Влияние  [R] = [P] x [L] |
| Технический | Сбои в работе системы из-за недостаточной производительности. | Задержки в работе системы, недовольство клиентов и сотрудников. | Проведение нагрузочного тестирования на этапе разработки. | Оптимизация кода и использование масштабируемых технологий. | 6 | 9 | 54 |
| Организационный | Недостаточная квалификация сотрудников для работы с новой системой. | Ошибки при использовании системы, снижение эффективности работы. | Проведение обучения сотрудников до внедрения системы. | Создание подробной документации и поддержка пользователей на этапе внедрения. | 7 | 5 | 35 |
| Организационный | Недостаточная вовлеченность заказчика в процесс разработки. | Несоответствие системы ожиданиям заказчика, задержки в согласовании решений. | Регулярное проведение встреч с заказчиком и предоставление промежуточных результатов. | Назначение ответственного представителя заказчика для взаимодействия. | 4 | 7 | 28 |

Продолжение Таблицы 3.8

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Тип | Риск | Последствия | Меры по предотвращению | Меры по минимизации последствий | Вероятность свершения риска [P] | Степень ущерба [L] | Влияние  [R] = [P] x [L] |
| Коммерческий | Превышение бюджета проекта из-за неправильной оценки трудозатрат. | Финансовые потери, задержка сроков реализации. | Проведение детальной оценки трудозатрат и создание резервного бюджета. | Регулярный мониторинг расходов и корректировка плана работ. | 5 | 7 | 35 |
| Проектный | Неполное или некорректное понимание требований заказчика. | Разработка системы, не соответствующей ожиданиям заказчика. | Проведение детальных интервью с заказчиком и пользователями, создание прототипов. | Регулярное согласование требований и внесение изменений на ранних этапах разработки. | 5 | 8 | 40 |

1. Проектирование решения
   1. Описание архитектуры системы

Для наглядного представления архитектуры системы используется диаграмма развертывания (Рисунок 4.1). На диаграмме отображены основные компоненты системы и их взаимосвязи.

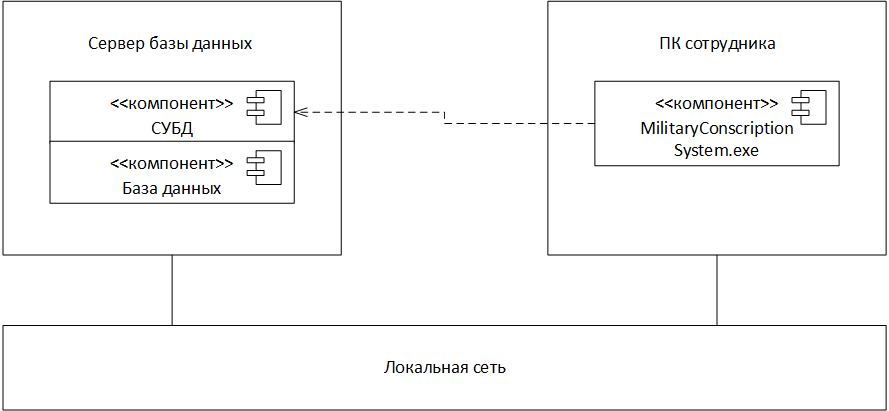


Рисунок 4.1 – Диаграмма развертывания.

* 1. Описание возможных технических решений

В качестве инструментальных средств разработки была выбрана IDE Visual Studio 2022, СУБД MS SQL Server 2020 ввиду своей современности, надежности и удобства использования. В Таблицах 4.1 – 4.2 представлены обоснования выбора.

Таблица 4.1 – Обоснование выбора средств разработки

|  |  |  |
| --- | --- | --- |
| Критерии сравнения | Visual Studio | Rider |
| Удобство | Интуитивно понятный пользовательский интерфейс, который позволяет легко создавать и отлаживать приложения.  Встроенная поддержка отладки и профилирования для упрощения разработки приложений. | Понятный пользовательский интерфейс, позволяющий создавать, оперировать и отслеживать приложения. Встроенный анализ кода для всех поддерживаемых языков с отображением ошибок и проблем с качеством кода. |
| Языковая поддержка | Широкий спектр поддерживаемых языков программирования, включая C++, C#, F#, Visual Basic и другие. | Отличное решение для разработки на ASP.NET, XAML, JavaScript, TypeScript, SASS, LESS, HTML, SQL, XSLT, YAML и другие. |
| Производительность | Имеет высокую производительность благодаря множеству оптимизаций и улучшений, которые Microsoft регулярно внедряет в свою среду разработки. | Высокая производительность и поддержка современных технологий, таких как Unreal Engine, Mercurial, SVN, и многих других. |
| Наличие  инструментов  разработки | Интеграция с множеством инструментов для управления проектами, контроля версий и развертывания приложений. | Мощные инструменты для автоматической проверки кода и обеспечения высокого качества разработки, встроенные шаблоны проектов и возможность рефакторинга проекта. |

Выбор среды разработки зависит от ваших индивидуальных предпочтений и потребностей. Visual Studio часто используется для работы с технологиями Microsoft, такими как .NET и C#, в то время как Rider который является альтернативой используется в большем спектре языков. Обе среды разработки обладают мощными инструментами, интегрированной средой разработки и возможностью расширения функциональности через плагины, но в отличии от Rider Visual Studio хоть и имеет меньше инструментов разработки, но обладает более простым интерфейсом с встроенным отображением ошибок.

В качестве средств разработки клиентской части была выбрана Visual Studio 2022 так как Visual Studio часто используется для работы с технологиями Microsoft, такими как .NET и C#.

Таблица 4.2 – Обоснование выбора СУБД

|  |  |  |
| --- | --- | --- |
| Критерии сравнения | СУБД MS SQL Server | СУБД PostgreSQL |
| Удобство | Высокая производительность и масштабируемость.  Обширные инструменты для администрирования и мониторинга баз данных. | Бесплатное и открытое программное обеспечение с активным сообществом разработчиков.  Поддержка распределенных транзакций и многопоточности. |
| Совместимость | Широкие возможности для разработки приложений и интеграции с другими технологиями Microsoft. | Обладает отличной совместимостью с различными приложениями и инструментами, а также поддерживает стандарт SQL (Structured Query Language). |

Продолжение Таблицы 4.2

|  |  |  |
| --- | --- | --- |
| Критерии сравнения | СУБД MS SQL Server | СУБД PostgreSQL |
| Поддержка | Официальные каналы поддержки Microsoft, включая документацию, форумы сообщества, онлайн-ресурсы и техническую поддержку. Кроме того, существует множество сторонних поставщиков услуг поддержки, которые могут предоставить консультации, обучение и техническую поддержку для MS SQL Server. | PostgreSQL осуществляется сообществом разработчиков, а также коммерческими компаниями, которые предоставляют услуги поддержки, консультирования, обучения и разработки для этой системы управления базами данных. |
| Надежность | Считается надежной и стабильной. Она имеет встроенные механизмы высокой доступности, отказоустойчивости и аварийного восстановления, такие как Always On Availability Groups, механизмы резервного копирования и восстановления, и многие другие. | Обладает высокой надежностью и отказоустойчивостью. Она широко используется в различных критически важных сферах, таких как финансы, здравоохранение, государственное управление и многие другие. |

Выбор между MS SQL Server и PostgreSQL зависит от потребностей и требований конкретного проекта. Обе СУБД имеют свои преимущества и недостатки. В качестве системы управления базами данных была выбрана СУБД MS SQL Server 2020 так как MS SQL Server разработан Microsoft и обычно используется в корпоративной среде. Кроме того, так как мы выбрали Visual Studio в качестве среды разработки, то MS SQL Server будет более удобен, так как он поддерживается Visual Studio по умолчанию, в отличии от PostgreSQL.

* 1. Проектирование интерфейса пользователя

Результат проектирования интерфейса представлен на Рисунке 4.2.

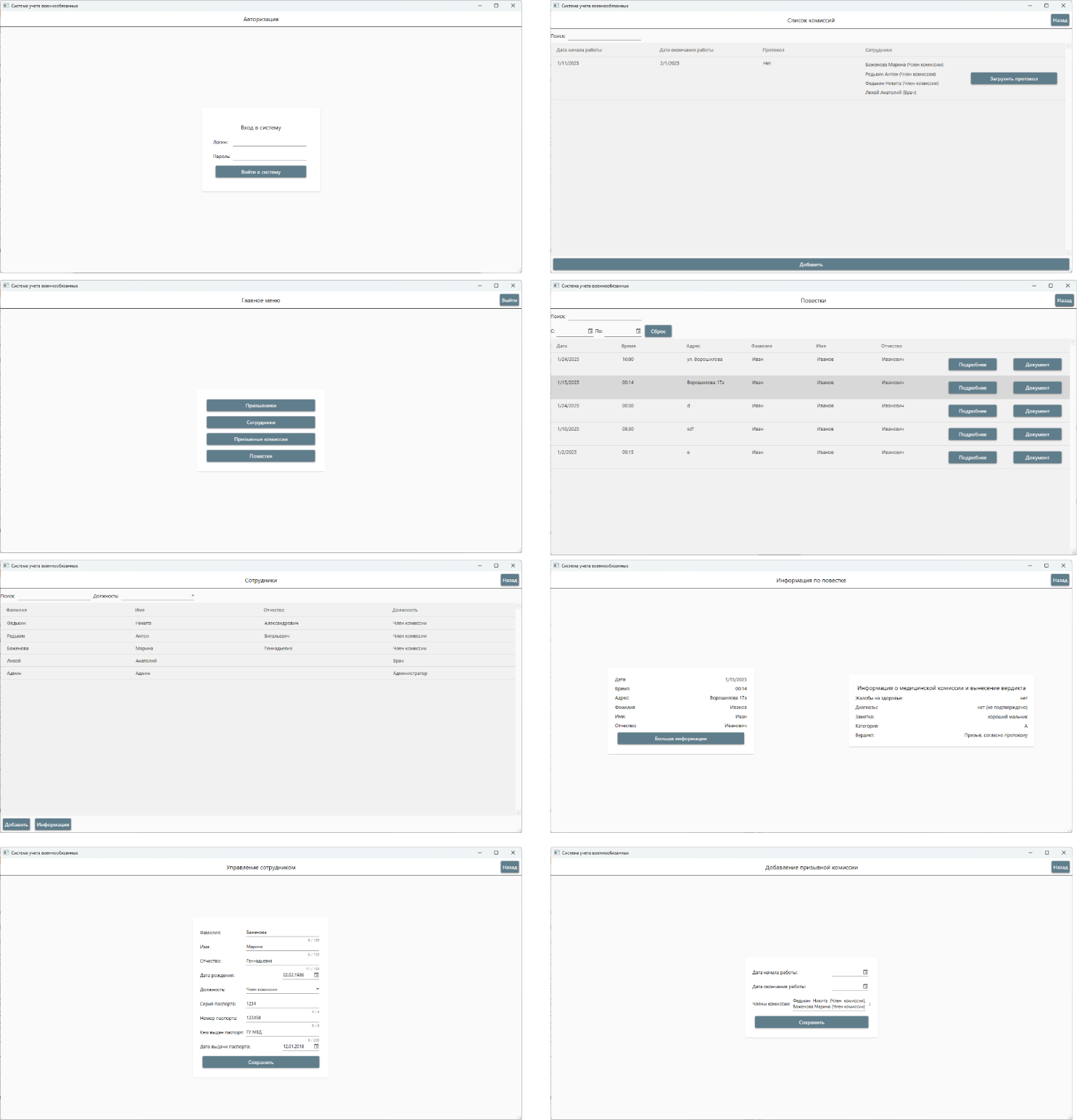


Рисунок 4.2 – Интерфейс приложения

* 1. Проектирование базы данных

Диаграмма базы данных представлена на Рисунке 4.3. Листинг базы данных представлен в Приложении 1.

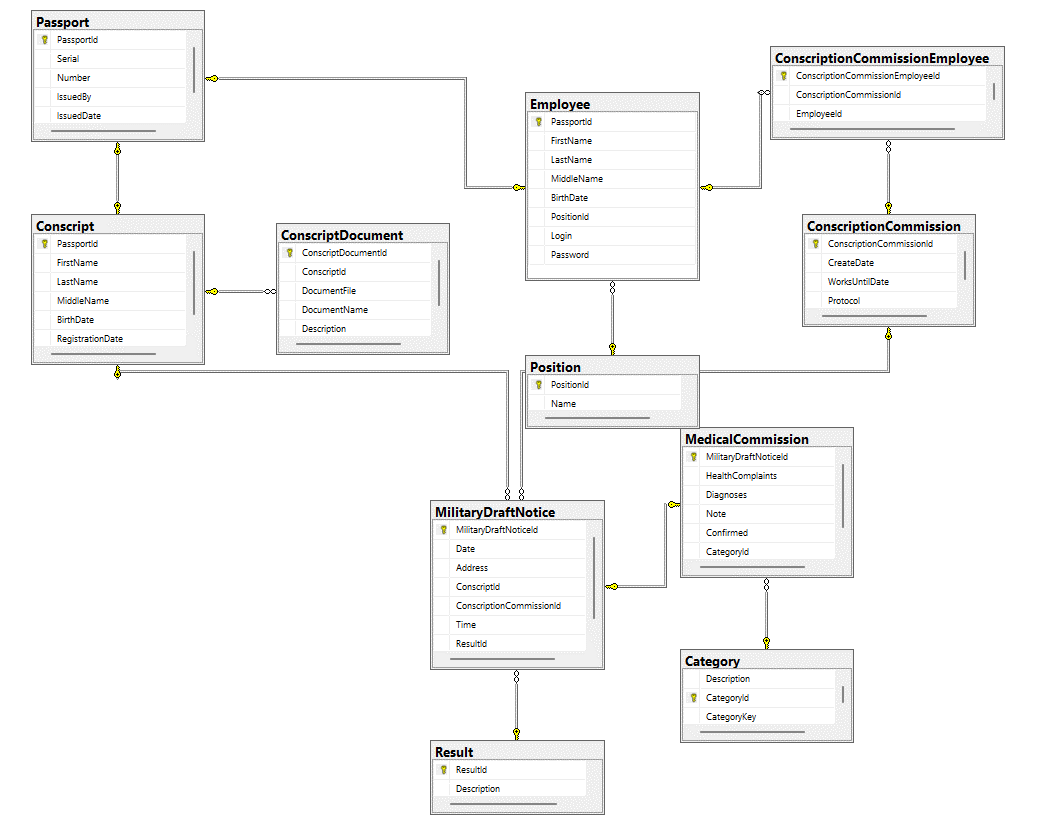


Рисунок 4.3 – Диаграмма базы данных

1. Программная реализация системы

Программная реализация программы представлена в листингах в Приложении 2.

1. Тестирование системы

Приложение 1 – Листинг базы данных

USE [master]

GO

CREATE DATABASE [MilitaryConscriptionDatabase]

GO

USE [MilitaryConscriptionDatabase]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Category] Script Date: 31.01.2025 10:32:52 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Category](

[Description] [nvarchar](200) NULL,

[CategoryId] [int] NOT NULL,

[CategoryKey] [varchar](1) NULL,

PRIMARY KEY CLUSTERED

(

[CategoryId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Conscript] Script Date: 31.01.2025 10:32:52 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Conscript](

[PassportId] [int] NOT NULL,

[FirstName] [nvarchar](100) NULL,

[LastName] [nvarchar](100) NULL,

[MiddleName] [nvarchar](100) NULL,

[BirthDate] [date] NULL,

[RegistrationDate] [date] NULL,

PRIMARY KEY CLUSTERED

(

[PassportId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ConscriptDocument] Script Date: 31.01.2025 10:32:52 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ConscriptDocument](

[ConscriptDocumentId] [int] IDENTITY(1,1) NOT NULL,

[ConscriptId] [int] NULL,

[DocumentFile] [varbinary](max) NULL,

Рисунок 1.1.1 – Листинг базы данных

продолжение приложения 1

[DocumentName] [nvarchar](200) NULL,

[Description] [nvarchar](300) NULL,

PRIMARY KEY CLUSTERED

(

[ConscriptDocumentId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ConscriptionCommission] Script Date: 31.01.2025 10:32:52 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ConscriptionCommission](

[ConscriptionCommissionId] [int] IDENTITY(1,1) NOT NULL,

[CreateDate] [date] NULL,

[WorksUntilDate] [date] NULL,

[Protocol] [varbinary](max) NULL,

PRIMARY KEY CLUSTERED

(

[ConscriptionCommissionId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[ConscriptionCommissionEmployee] Script Date: 31.01.2025 10:32:52 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ConscriptionCommissionEmployee](

[ConscriptionCommissionEmployeeId] [int] IDENTITY(1,1) NOT NULL,

[ConscriptionCommissionId] [int] NULL,

[EmployeeId] [int] NULL,

PRIMARY KEY CLUSTERED

(

[ConscriptionCommissionEmployeeId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Employee] Script Date: 31.01.2025 10:32:52 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Employee](

Рисунок 1.1.2 – Продолжение листинга базы данных

Продолжение приложения 1

[PassportId] [int] NOT NULL,

[FirstName] [nvarchar](100) NULL,

[LastName] [nvarchar](100) NULL,

[MiddleName] [nvarchar](100) NULL,

[BirthDate] [date] NULL,

[PositionId] [int] NULL,

[Login] [nvarchar](100) NULL,

[Password] [nvarchar](20) NULL,

PRIMARY KEY CLUSTERED

(

[PassportId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[MedicalCommission] Script Date: 31.01.2025 10:32:52 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[MedicalCommission](

[MilitaryDraftNoticeId] [int] NOT NULL,

[HealthComplaints] [nvarchar](500) NULL,

[Diagnoses] [nvarchar](500) NULL,

[Note] [nvarchar](200) NULL,

[Confirmed] [bit] NULL,

[CategoryId] [int] NULL,

PRIMARY KEY CLUSTERED

(

[MilitaryDraftNoticeId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[MilitaryDraftNotice] Script Date: 31.01.2025 10:32:52 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[MilitaryDraftNotice](

[MilitaryDraftNoticeId] [int] IDENTITY(1,1) NOT NULL,

[Date] [date] NULL,

[Address] [nvarchar](200) NULL,

[ConscriptId] [int] NULL,

[ConscriptionCommissionId] [int] NULL,

[Time] [time](7) NULL,

[ResultId] [int] NULL,

PRIMARY KEY CLUSTERED

(

[MilitaryDraftNoticeId] ASC

Рисунок 1.1.3 – Продолжение листинга базы данных

Продолжение приложения 1

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Passport] Script Date: 31.01.2025 10:32:52 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Passport](

[PassportId] [int] IDENTITY(1,1) NOT NULL,

[Serial] [int] NULL,

[Number] [int] NULL,

[IssuedBy] [nvarchar](200) NULL,

[IssuedDate] [date] NULL,

PRIMARY KEY CLUSTERED

(

[PassportId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Position] Script Date: 31.01.2025 10:32:52 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Position](

[PositionId] [int] IDENTITY(1,1) NOT NULL,

[Name] [nvarchar](150) NULL,

PRIMARY KEY CLUSTERED

(

[PositionId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Result] Script Date: 31.01.2025 10:32:52 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Result](

[ResultId] [int] IDENTITY(1,1) NOT NULL,

[Description] [nvarchar](200) NULL,

PRIMARY KEY CLUSTERED

(

[ResultId] ASC

Рисунок 1.1.4 – Продолжение листинга базы данных

Продолжение приложения 1

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[MedicalCommission] ADD DEFAULT ((0)) FOR [Confirmed]

GO

ALTER TABLE [dbo].[Conscript] WITH CHECK ADD FOREIGN KEY([PassportId])

REFERENCES [dbo].[Passport] ([PassportId])

GO

ALTER TABLE [dbo].[ConscriptDocument] WITH CHECK ADD FOREIGN KEY([ConscriptId])

REFERENCES [dbo].[Conscript] ([PassportId])

GO

ALTER TABLE [dbo].[ConscriptionCommissionEmployee] WITH CHECK ADD FOREIGN KEY([ConscriptionCommissionId])

REFERENCES [dbo].[ConscriptionCommission] ([ConscriptionCommissionId])

GO

ALTER TABLE [dbo].[ConscriptionCommissionEmployee] WITH CHECK ADD FOREIGN KEY([EmployeeId])

REFERENCES [dbo].[Employee] ([PassportId])

GO

ALTER TABLE [dbo].[Employee] WITH CHECK ADD FOREIGN KEY([PassportId])

REFERENCES [dbo].[Passport] ([PassportId])

GO

ALTER TABLE [dbo].[Employee] WITH CHECK ADD FOREIGN KEY([PositionId])

REFERENCES [dbo].[Position] ([PositionId])

GO

ALTER TABLE [dbo].[MedicalCommission] WITH CHECK ADD FOREIGN KEY([CategoryId])

REFERENCES [dbo].[Category] ([CategoryId])

GO

ALTER TABLE [dbo].[MedicalCommission] WITH CHECK ADD FOREIGN KEY([MilitaryDraftNoticeId])

REFERENCES [dbo].[MilitaryDraftNotice] ([MilitaryDraftNoticeId])

GO

ALTER TABLE [dbo].[MilitaryDraftNotice] WITH CHECK ADD FOREIGN KEY([ConscriptId])

REFERENCES [dbo].[Conscript] ([PassportId])

GO

ALTER TABLE [dbo].[MilitaryDraftNotice] WITH CHECK ADD FOREIGN KEY([ConscriptionCommissionId])

REFERENCES [dbo].[ConscriptionCommission] ([ConscriptionCommissionId])

GO

ALTER TABLE [dbo].[MilitaryDraftNotice] WITH CHECK ADD FOREIGN KEY([ResultId])

REFERENCES [dbo].[Result] ([ResultId])

GO

USE [master]

GO

ALTER DATABASE [MilitaryConscriptionDatabase] SET READ\_WRITE

GO

Рисунок 1.1.5 – Продолжение листинга базы данных

Приложение 2 – Листинг программной реализации системы

<Page x:Class="MilitaryConscriptionSystem.AllPages.AddConsCommissionPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:design="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:kit="http://schemas.xceed.com/wpf/xaml/toolkit"

xmlns:local="clr-namespace:MilitaryConscriptionSystem.AllPages"

mc:Ignorable="d"

Title="Добавление призывной комиссии">

<design:Card HorizontalAlignment="Center"

VerticalAlignment="Center"

Padding="20">

<StackPanel Orientation="Vertical"

VerticalAlignment="Center">

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Дата начала работы:" />

<DatePicker Margin="7"

Width="100"

HorizontalAlignment="Right"

x:Name="StartDatePicker" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Дата окончания работы:" />

<DatePicker Margin="7"

Width="100"

HorizontalAlignment="Right"

x:Name="EndDatePicker" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Члены комиссии:" />

<DockPanel>

<Button x:Name="ShowButton"

Click="ShowButton\_OnClick"

Content="↓"

Padding="0"

DockPanel.Dock="Right"

Style="{DynamicResource MaterialDesignFlatButton}" />

<TextBox x:Name="EmployeesTextBox"

HorizontalAlignment="Right"

Width="200"

Margin="7"

TextWrapping="Wrap"

DockPanel.Dock="Left"

IsReadOnly="True" />

Рисунок 2.1.1 – Листинг AddConsCommissionPage.xaml

Продолжение приложения 2

</DockPanel>

</DockPanel>

<DockPanel>

</DockPanel>

<Popup x:Name="Popup"

MinWidth="{Binding ElementName=EmployeesTextBox, Path=ActualWidth}"

StaysOpen="False"

Placement="Bottom"

PlacementTarget="{Binding ElementName=EmployeesTextBox}">

<ItemsControl x:Name="EmployeeCheckListView"

Margin="1"

Padding="5"

Background="#FAFAFA">

<ItemsControl.ItemTemplate>

<DataTemplate>

<DockPanel>

<CheckBox Checked="CheckBox\_StateChanged"

Margin="5"

Unchecked="CheckBox\_StateChanged"

IsChecked="{Binding IsChecked}"

Content="{Binding Employee.ComboBoxField}" />

</DockPanel>

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

</Popup>

<Button Margin="7"

Padding="5"

x:Name="SaveButton"

Click="SaveButton\_OnClick"

Content="Сохранить" />

</StackPanel>

</design:Card>

</Page>

Рисунок 2.1.2 – Продолжение листинга AddConsCommissionPage.xaml

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

using Microsoft.EntityFrameworkCore;

using MilitaryConscriptionSystem.Database;

namespace MilitaryConscriptionSystem.AllPages;

public partial class AddConsCommissionPage : Page

{

private List<EmployeeCheck> \_checks;

public AddConsCommissionPage()

{

Рисунок 2.1.2 – Листинг AddConsCommissionPage.xaml.cs

Продолжение приложения 2

InitializeComponent();

\_checks = new();

foreach (var employee in Db.Context.Employees.Include(c => c.Position).Where(c => c.PositionId != 5).ToList())

\_checks.Add(new(employee));

EmployeeCheckListView.ItemsSource = \_checks;

}

private void SaveButton\_OnClick(object sender, RoutedEventArgs e)

{

try

{

if (StartDatePicker.SelectedDate != null && EndDatePicker.SelectedDate != null &&

\_checks.Any(c => c.IsChecked))

{

List<Employee> selected = \_checks.Where(c => c.IsChecked).Select(c => c.Employee).ToList();

if (selected.Any(c => c.PositionId == 1) &&

selected.Any(c => c.PositionId == 4))

{

ConscriptionCommission commission = new()

{

CreateDate = DateOnly.FromDateTime(StartDatePicker.SelectedDate.Value),

WorksUntilDate = DateOnly.FromDateTime(EndDatePicker.SelectedDate.Value)

};

List<ConscriptionCommissionEmployee> commissionEmployees = new();

foreach (var employee in selected)

{

commissionEmployees.Add(new()

{

EmployeeId = employee.PassportId,

ConscriptionCommission = commission

});

}

Db.Context.ConscriptionCommissions.Add(commission);

Db.Context.ConscriptionCommissionEmployees.AddRange(commissionEmployees);

Db.Context.SaveChanges();

MessageBox.Show("Данные добавлены!", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Asterisk);

NavigationService.GoBack();

}

else

MessageBox.Show("В комиссии должен быть минимум 1 сотрудник каждой должности!", "Сообщение",

MessageBoxButton.OK,

Рисунок 2.1.3 – Продолжение листинга AddConsCommissionPage.xaml.cs

Продолжение приложения 2

MessageBoxImage.Information);

}

else

MessageBox.Show("Поля не могут быть пустыми!", "Сообщение", MessageBoxButton.OK,

MessageBoxImage.Information);

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void ShowButton\_OnClick(object sender, RoutedEventArgs e)

{

Popup.IsOpen = true;

}

private void CheckBox\_StateChanged(object sender, RoutedEventArgs e)

{

try

{

List<string> selected = \_checks.Where(c => c.IsChecked)

.Select(c => c.Employee.ComboBoxField).ToList();

if (selected.Count != 0)

{

string text = String.Empty;

foreach (var se in selected)

text += se + ", ";

EmployeesTextBox.Text = text.Substring(0, text.Length - 2);

}

else

EmployeesTextBox.Text = string.Empty;

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

Рисунок 2.1.4 – Продолжение листинга AddConsCommissionPage.xaml.cs

<Page x:Class="MilitaryConscriptionSystem.AllPages.AddNoticePage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:design=<http://materialdesigninxaml.net/winfx/xaml/themes>

Рисунок 2.1.5 – Листинг AddNoticePage.xaml

Продолжение приложения 2

xmlns:kit="http://schemas.xceed.com/wpf/xaml/toolkit"

xmlns:local="clr-namespace:MilitaryConscriptionSystem.AllPages"

mc:Ignorable="d"

Title="Добавление повестки">

<design:Card HorizontalAlignment="Center"

VerticalAlignment="Center"

Padding="20">

<StackPanel Orientation="Vertical">

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Призывная комиссия:" />

<ComboBox Margin="7"

Width="200"

HorizontalAlignment="Right"

x:Name="CommissionComboBox"

DisplayMemberPath="Name" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Дата:" />

<DatePicker x:Name="DatePicker"

Margin="7"

Width="100"

HorizontalAlignment="Right" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Время:" />

<ComboBox x:Name="TimeComboBox" Margin="7"

Width="100"

HorizontalAlignment="Right"/>

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Адрес:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

MaxLength="200"

x:Name="AddressTextBox" />

</DockPanel>

<Button Margin="10"

Padding="5"

x:Name="SaveButton"

Content="Сохранить"

Click="SaveButton\_OnClick" />

</StackPanel>

</design:Card>

</Page>

Рисунок 2.1.5 – Продолжение листинга AddNoticePage.xaml

Продолжение приложения 2

using System.Windows;

using System.Windows.Controls;

using MilitaryConscriptionSystem.Database;

namespace MilitaryConscriptionSystem.AllPages;

public partial class AddNoticePage : Page

{

private MilitaryDraftNotice \_notice;

public AddNoticePage(Conscript conscript)

{

\_notice = new()

{

ConscriptId = conscript.PassportId

};

InitializeComponent();

List<string> times = new();

TimeOnly start = new(8, 0);

TimeOnly end = new(17, 15);

while (start < end)

{

times.Add(start.ToString("t"));

start = start.AddMinutes(15);

}

TimeComboBox.ItemsSource = times;

CommissionComboBox.ItemsSource = Db.Context.ConscriptionCommissions.ToList();

}

private void SaveButton\_OnClick(object sender, RoutedEventArgs e)

{

try

{

if (TimeComboBox.SelectedItem != null && !String.IsNullOrEmpty(AddressTextBox.Text) &&

CommissionComboBox.SelectedItem != null)

{

string[] time = ((String)TimeComboBox.SelectedItem).Split(":");

\_notice.Date = DateOnly.FromDateTime(DatePicker.SelectedDate.Value);

\_notice.Time = new TimeOnly(int.Parse(time[0]),int.Parse(time[1]));

\_notice.Address = AddressTextBox.Text;

\_notice.ConscriptionCommissionId =

((ConscriptionCommission)CommissionComboBox.SelectedItem).ConscriptionCommissionId;

Db.Context.MilitaryDraftNotices.Add(\_notice);

Рисунок 2.1.6 – Продолжение листинга AddNoticePage.xaml.cs

Продолжение приложения 2

Db.Context.SaveChanges();

MessageBox.Show("Данные сохранены!", "Сообщение", MessageBoxButton.OK,

MessageBoxImage.Information);

NavigationService.GoBack();

}

else

MessageBox.Show("Поля не могут быть пустыми!", "Сообщение", MessageBoxButton.OK,

MessageBoxImage.Information);

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

Рисунок 2.1.7 – Продолжение листинга AddNoticePage.xaml.cs

<Page x:Class="MilitaryConscriptionSystem.AllPages.EditConscriptPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:design="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:MilitaryConscriptionSystem.AllPages"

mc:Ignorable="d"

Title="Управление призывником"

Loaded="ConscriptEditPage\_OnLoaded"

d:DesignHeight="450"

d:DesignWidth="800">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<design:Card HorizontalAlignment="Center"

VerticalAlignment="Center"

Padding="20">

<StackPanel x:Name="DataPanel"

Orientation="Vertical"

Grid.Column="0">

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Фамилия:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

Рисунок 2.1.8 – Листинг EditConscriptPage.xaml

Продолжение приложения 2

MaxLength="100"

x:Name="LastNameTextBox"

Text="{Binding LastName}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Имя:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

MaxLength="100"

x:Name="FirstNameTextBox"

Text="{Binding FirstName}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Отчество:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

MaxLength="100"

x:Name="MiddleNameTextBox"

Text="{Binding MiddleName}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Дата рождения:" />

<DatePicker Margin="7"

Width="100"

HorizontalAlignment="Right"

x:Name="BirthDatePicker"

SelectedDate="{Binding BirthDateTime}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Дата становления на учет:" />

<DatePicker Margin="7"

Width="100"

HorizontalAlignment="Right"

x:Name="RegistrationDateDatePicker"

SelectedDate="{Binding RegistrationDateTime}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Рисунок 2.1.9 – Продолжение листинга EditConscriptPage.xaml

Продолжение приложения 2

Text="Серия паспорта:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

x:Name="PassSTextBox"

MaxLength="4"

Text="{Binding Passport.Serial}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Номер паспорта:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

x:Name="PassNTextBox"

MaxLength="6"

Text="{Binding Passport.Number}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Кем выдан паспорт:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

MaxLength="200"

x:Name="IssuedByTextBox"

Text="{Binding Passport.IssuedBy}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Дата выдачи паспорта:" />

<DatePicker Margin="7"

Width="100"

HorizontalAlignment="Right"

x:Name="IssuedDatePicker"

SelectedDate="{Binding Passport.IssuedDateTime}" />

</DockPanel>

<Button Margin="7"

Padding="5"

x:Name="SaveButton"

Content="Сохранить"

Click="SaveButton\_OnClick" />

</StackPanel>

</design:Card>

<design:Card HorizontalAlignment="Center"

VerticalAlignment="Center"

Рисунок 2.1.10 – Продолжение листинга EditConscriptPage.xaml

Продолжение приложения 2

Grid.Column="1"

Padding="20">

<StackPanel Orientation="Vertical">

<TextBlock Style="{DynamicResource MaterialDesignSubtitle1TextBlock}"

HorizontalAlignment="Center"

Margin="5"

Text="Документы" />

<ListView x:Name="FileListView">

<ListView.ItemTemplate>

<DataTemplate>

<StackPanel Orientation="Vertical">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="{Binding DocumentName}" />

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="{Binding Description}" />

<Button Margin="7"

Padding="5"

x:Name="DeleteButton"

Click="DeleteButton\_OnClick"

Content="Удалить" />

</StackPanel>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<design:Card HorizontalAlignment="Center"

VerticalAlignment="Center"

Padding="10">

<StackPanel Orientation="Vertical">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

HorizontalAlignment="Center"

Text="Новый файл" />

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Описание:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

x:Name="DescriptionTextBox" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<Button Margin="7"

Padding="5"

Content="Загрузить файл"

x:Name="LoadFile"

Рисунок 2.1.11 – Продолжение листинга EditConscriptPage.xaml

Продолжение приложения 2

Click="LoadFile\_OnClick" />

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

x:Name="LoadedFileName" />

</DockPanel>

<Button Margin="7"

Padding="5"

x:Name="SaveFileButton"

Visibility="Collapsed"

Click="SaveFileButton\_OnClick"

Content="Добавить" />

</StackPanel>

</design:Card>

</StackPanel>

</design:Card>

</Grid>

</Page>

Рисунок 2.1.12 – Продолжение листинга EditConscriptPage.xaml

using System.IO;

using System.Windows;

using System.Windows.Controls;

using Microsoft.Win32;

using MilitaryConscriptionSystem.Database;

namespace MilitaryConscriptionSystem.AllPages;

public partial class EditConscriptPage : Page

{

private ConscriptDocument? \_document;

private Conscript \_conscript;

public EditConscriptPage()

{

\_conscript = new()

{

Passport = new() { IssuedDate = DateOnly.FromDateTime(DateTime.Now) },

BirthDate = DateOnly.FromDateTime(DateTime.Now),

RegistrationDate = DateOnly.FromDateTime(DateTime.Now)

};

InitializeComponent();

}

public EditConscriptPage(Conscript conscript)

{

if (conscript.Passport == null)

conscript.Passport = Db.Context.Passports.Find(conscript.PassportId);

\_conscript = conscript;

InitializeComponent();

}

Рисунок 2.1.13 – Листинг EditConscriptPage.xaml.cs

Продолжение приложения 2

private void SaveButton\_OnClick(object sender, RoutedEventArgs e)

{

try

{

if (!String.IsNullOrEmpty(\_conscript.FirstName) &&

!String.IsNullOrEmpty(\_conscript.LastName) &&

\_conscript is { BirthDate: not null, RegistrationDate: not null } &&

\_conscript.Passport.Serial.ToString().Length == 4 &&

\_conscript.Passport.Number.ToString().Length == 6 &&

\_conscript.Passport.IssuedDate != null &&

!String.IsNullOrEmpty(\_conscript.Passport.IssuedBy))

{

if (\_conscript.PassportId == 0)

{

Db.Context.Passports.Add(\_conscript.Passport);

Db.Context.Conscripts.Add(\_conscript);

}

else

{

Db.Context.Passports.Update(\_conscript.Passport);

Db.Context.Conscripts.Update(\_conscript);

}

Db.Context.SaveChanges();

MessageBox.Show("Данные сохранены", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Asterisk);

}

else

MessageBox.Show("Заполните все поля! Номер и серия паспорта должны быть указаны в правильном формате.",

"Сообщение", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void LoadFiles()

{

try

{

FileListView.ItemsSource = null;

FileListView.ItemsSource = Db.Context.ConscriptDocuments.Where(c => c.ConscriptId == \_conscript.PassportId)

.ToList();

}

catch (Exception e)

{

Console.WriteLine(e);

Рисунок 2.1.14 – Продолжение листинга EditConscriptPage.xaml.cs

Продолжение приложения 2

MessageBox.Show(e.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void SaveFileButton\_OnClick(object sender, RoutedEventArgs e)

{

try

{

if (\_conscript.PassportId != 0)

{

\_document.Description = DescriptionTextBox.Text;

\_document.ConscriptId = \_conscript.PassportId;

Db.Context.ConscriptDocuments.Add(\_document);

Db.Context.SaveChanges();

\_document = new();

SaveFileButton.Visibility = Visibility.Collapsed;

DescriptionTextBox.Text = String.Empty;

LoadFiles();

MessageBox.Show("Файл добавлен!", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Asterisk);

}

else

MessageBox.Show("Призывник еще не добавлен!", "Сообщение", MessageBoxButton.OK,

MessageBoxImage.Information);

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void ConscriptEditPage\_OnLoaded(object sender, RoutedEventArgs e)

{

try

{

DataPanel.DataContext = \_conscript;

LoadFiles();

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void LoadFile\_OnClick(object sender, RoutedEventArgs e)

{

try

{

Рисунок 2.1.15 – Продолжение листинга EditConscriptPage.xaml.cs

Продолжение приложения 2

OpenFileDialog dialog = new() { Filter = "documents | \*.docx; \*.doc;\*.pdf" };

if (dialog.ShowDialog() == true)

{

\_document = new()

{ DocumentFile = File.ReadAllBytes(dialog.FileName), DocumentName = dialog.SafeFileName };

SaveFileButton.Visibility = Visibility.Visible;

}

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void DeleteButton\_OnClick(object sender, RoutedEventArgs e)

{

try

{

ConscriptDocument document = ((Button)sender).DataContext as ConscriptDocument;

Db.Context.ConscriptDocuments.Remove(document);

Db.Context.SaveChanges();

MessageBox.Show("Файл удален!", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Asterisk);

LoadFiles();

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

Рисунок 2.1.16 – Продолжение листинга EditConscriptPage.xaml.cs

<Page x:Class="MilitaryConscriptionSystem.AllPages.EditEmployeePage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:design="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:MilitaryConscriptionSystem.AllPages"

mc:Ignorable="d"

Loaded="EmployeeEditPage\_OnLoaded"

Title="Управление сотрудником">

<design:Card HorizontalAlignment="Center"

VerticalAlignment="Center"

Padding="20">

Рисунок 2.1.17 – Листинг EditEmployeePage.xaml

Продолжение приложения 2

<StackPanel x:Name="DataPanel"

VerticalAlignment="Center"

Orientation="Vertical">

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Фамилия:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

MaxLength="100"

x:Name="LastNameTextBox"

Text="{Binding LastName}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Имя:" />

<TextBox HorizontalAlignment="Right"

MaxLength="100"

Width="200"

Margin="7"

x:Name="FirstNameTextBox"

Text="{Binding FirstName}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Отчество:" />

<TextBox HorizontalAlignment="Right"

MaxLength="100"

Width="200"

Margin="7"

x:Name="MiddleNameTextBox"

Text="{Binding MiddleName}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Дата рождения:" />

<DatePicker Margin="7"

Width="100"

HorizontalAlignment="Right"

x:Name="BirthDatePicker"

SelectedDate="{Binding BirthDateTime}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Должность:" />

<ComboBox Margin="7"

Width="200"

HorizontalAlignment="Right"

x:Name="PositionComboBox"

Рисунок 2.1.18 – Продолжение листинга EditEmployeePage.xaml

Продолжение приложения 2

SelectedItem="{Binding Position}"

DisplayMemberPath="Name" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Серия паспорта:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

x:Name="PassSTextBox"

MaxLength="4"

Text="{Binding Passport.Serial}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Номер паспорта:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

x:Name="PassNTextBox"

MaxLength="6"

Text="{Binding Passport.Number}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Кем выдан паспорт:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

MaxLength="200"

x:Name="IssuedByTextBox"

Text="{Binding Passport.IssuedBy}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Дата выдачи паспорта:" />

<DatePicker Margin="7"

Width="100"

HorizontalAlignment="Right"

x:Name="IssuedDatePicker"

SelectedDate="{Binding Passport.IssuedDateTime}" />

</DockPanel>

<Button Margin="7"

Padding="5"

x:Name="SaveButton"

Content="Сохранить"

Click="SaveButton\_OnClick" />

</StackPanel>

</design:Card>

</Page>

Рисунок 2.1.19 – Продолжение листинга EditEmployeePage.xaml

Продолжение приложения 2

using System.Windows;

using System.Windows.Controls;

using MilitaryConscriptionSystem.Database;

namespace MilitaryConscriptionSystem.AllPages;

public partial class EditEmployeePage : Page

{

private Employee \_employee;

public EditEmployeePage()

{

\_employee = new()

{

Passport = new() { IssuedDate = DateOnly.FromDateTime(DateTime.Today) },

BirthDate = DateOnly.FromDateTime(DateTime.Today)

};

InitializeComponent();

}

public EditEmployeePage(Employee employee)

{

\_employee = employee;

InitializeComponent();

if (\_employee.PositionId == 5) PositionComboBox.IsEnabled = false;

}

private void EmployeeEditPage\_OnLoaded(object sender, RoutedEventArgs e)

{

try

{

PositionComboBox.ItemsSource = Db.Context.Positions.ToList();

DataContext = \_employee;

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void SaveButton\_OnClick(object sender, RoutedEventArgs e)

{

try

{

if (!String.IsNullOrEmpty(\_employee.FirstName) &&

!String.IsNullOrEmpty(\_employee.LastName) &&

\_employee.BirthDate != null &&

\_employee.Position != null &&

\_employee.Passport.Serial.ToString().Length == 4 &&

\_employee.Passport.Number.ToString().Length == 6 &&

\_employee.Passport.IssuedDate != null &&

!String.IsNullOrEmpty(\_employee.Passport.IssuedBy))

Рисунок 2.1.20 – Листинг EditEmployeePage.xaml.cs

Продолжение приложения 2

{

if (\_employee.PassportId == 0)

{

Db.Context.Passports.Add(\_employee.Passport);

Db.Context.Employees.Add(\_employee);

}

else

{

Db.Context.Passports.Update(\_employee.Passport);

Db.Context.Employees.Update(\_employee);

}

Db.Context.SaveChanges();

MessageBox.Show("Данные сохранены", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Asterisk);

}

else

MessageBox.Show("Заполните все поля! Номер и серия паспорта должны быть указаны в правильном формате.",

"Сообщение", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

Рисунок 2.1.21 – Продолжение листинга EditEmployeePage.xaml.cs

<Page x:Class="MilitaryConscriptionSystem.AllPages.LoginPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:design="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:MilitaryConscriptionSystem.AllPages"

mc:Ignorable="d"

Title="Авторизация">

<design:Card Padding="32"

Margin="16"

VerticalAlignment="Center"

HorizontalAlignment="Center">

<StackPanel Orientation="Vertical">

<TextBlock VerticalAlignment="Center"

Style="{DynamicResource MaterialDesignSubtitle1TextBlock}"

HorizontalAlignment="Center"

Margin="10"

Text="Вход в систему" />

Рисунок 2.1.22 – Листинг LoginPage.xaml

Продолжение приложения 2

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Логин:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

x:Name="LoginTextBox" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Пароль:" />

<PasswordBox x:Name="PasswordBox"

Width="200"

HorizontalAlignment="Right"

Margin="7" />

</DockPanel>

<Button Margin="7"

Padding="5"

x:Name="LoginButton"

Content="Войти в систему"

Style="{DynamicResource MaterialDesignRaisedButton}"

Click="LoginButton\_OnClick" />

</StackPanel>

</design:Card>

</Page>

Рисунок 2.1.23 – Продолжение листинга LoginPage.xaml

using System.Windows;

using System.Windows.Controls;

using MilitaryConscriptionSystem.Database;

namespace MilitaryConscriptionSystem.AllPages;

public partial class LoginPage : Page

{

public LoginPage()

{

InitializeComponent();

}

private void LoginButton\_OnClick(object sender, RoutedEventArgs e)

{

try

{

string login = LoginTextBox.Text;

string password = PasswordBox.Password;

if (!String.IsNullOrEmpty(login) && !String.IsNullOrEmpty(password))

{

Employee? employee =

Рисунок 2.1.24 – Листинг LoginPage.xaml.cs

Продолжение приложения 2

Db.Context.Employees.FirstOrDefault(c => c.Login == login && c.Password == password);

if (employee != null)

{

App.Employee = employee;

NavigationService.Navigate(new MenuPage());

}

else

MessageBox.Show("Пользователь не найден", "Ошибка", MessageBoxButton.OK, MessageBoxImage.None);

}

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

Рисунок 2.1.25 – Продолжение листинга LoginPage.xaml.cs

<Page x:Class="MilitaryConscriptionSystem.AllPages.MenuPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:design="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:MilitaryConscriptionSystem.AllPages"

mc:Ignorable="d"

Title="Главное меню">

<design:Card Padding="20"

VerticalAlignment="Center"

HorizontalAlignment="Center"

Width="350">

<StackPanel Orientation="Vertical">

<Button Margin="7"

Padding="5"

x:Name="ConscriptButton"

Click="ConscriptButton\_OnClick"

Content="Призывники" />

<Button Margin="7"

Padding="5"

x:Name="EmployeeButton"

Visibility="Collapsed"

Click="EmployeeButton\_OnClick"

Content="Сотрудники" />

<Button Margin="7"

Padding="5"

x:Name="CommissionButton"

Visibility="Collapsed"

Click="CommissionButton\_OnClick"

Content="Призывные комиссии" />

<Button Margin="7"

Рисунок 2.1.26 –Листинг MenuPage.xaml

Продолжение приложения 2

Padding="5"

x:Name="NoticesButton"

Click="NoticesButton\_OnClick"

Content="Повестки" />

</StackPanel>

</design:Card>

</Page>

Рисунок 2.1.27 –Продолжение листинга MenuPage.xaml

using System.Windows;

using System.Windows.Controls;

namespace MilitaryConscriptionSystem.AllPages;

public partial class MenuPage : Page

{

public MenuPage()

{

InitializeComponent();

if (App.Employee.PositionId == 5)

{

EmployeeButton.Visibility = Visibility.Visible;

CommissionButton.Visibility = Visibility.Visible;

}

}

private void ConscriptButton\_OnClick(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new ViewConscriptsPage());

}

private void EmployeeButton\_OnClick(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new ViewEmployeesPage());

}

private void CommissionButton\_OnClick(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new ViewConsCommissionsPage());

}

private void NoticesButton\_OnClick(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new ViewNoticesPage());

}

}

Рисунок 2.1.28 –Листинг MenuPage.xaml.cs

<Page x:Class="MilitaryConscriptionSystem.AllPages.NoticeInfoPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

Рисунок 2.1.29 –Листинг NoticeInfoPage.xaml

Продолжение приложения 2

xmlns:design="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:MilitaryConscriptionSystem.AllPages"

mc:Ignorable="d"

Title="Информация по повестке">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<design:Card HorizontalAlignment="Center"

VerticalAlignment="Center"

MinWidth="400"

Padding="20">

<StackPanel Orientation="Vertical"

x:Name="MainInfoPanel">

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Дата:" />

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

HorizontalAlignment="Right"

Text="{Binding Date}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Время:" />

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

HorizontalAlignment="Right"

Text="{Binding Time, StringFormat='{}{0:HH:mm}'}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Адрес:" />

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

HorizontalAlignment="Right"

Text="{Binding Address}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

Рисунок 2.1.30 –Продолжение листинга NoticeInfoPage.xaml

Продолжение приложения 2

VerticalAlignment="Center"

Text="Фамилия:" />

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

HorizontalAlignment="Right"

Text="{Binding Conscript.LastName}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Имя:" />

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

HorizontalAlignment="Right"

Text="{Binding Conscript.FirstName}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

HorizontalAlignment="Right"

Text="Отчество:" />

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

HorizontalAlignment="Right"

Text="{Binding Conscript.MiddleName}" />

</DockPanel>

<Button Margin="7"

Padding="5"

x:Name="ConscriptButton"

Content="Больше информации"

Click="ConscriptButton\_OnClick" />

</StackPanel>

</design:Card>

<design:Card HorizontalAlignment="Center"

VerticalAlignment="Center"

Grid.Column="1"

Padding="20">

<StackPanel Orientation="Vertical">

<TextBlock Style="{DynamicResource MaterialDesignSubtitle1TextBlock}"

HorizontalAlignment="Center"

Margin="5"

Text="Информация о медицинской комиссии и вынесение вердикта" />

<StackPanel x:Name="ViewMedCommissionPanel"

Visibility="Collapsed"

Orientation="Vertical">

<DockPanel VerticalAlignment="Center">

Рисунок 2.1.31 –Продолжение листинга NoticeInfoPage.xaml

Продолжение приложения 2

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Жалобы на здоровье:" />

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

HorizontalAlignment="Right"

Text="{Binding HealthComplaints}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Диагнозы:" />

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

HorizontalAlignment="Right">

<TextBlock.Text>

<MultiBinding StringFormat="{}{0} {1}">

<Binding Path="Diagnoses" />

<Binding Path="DiagnosesApproved" />

</MultiBinding>

</TextBlock.Text>

</TextBlock>

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Заметка:" />

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

HorizontalAlignment="Right"

Text="{Binding Note}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Категория:" />

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

HorizontalAlignment="Right"

Text="{Binding Category.CategoryKey}" />

</DockPanel>

<DockPanel VerticalAlignment="Center"

x:Name="AddResultPanel"

Visibility="Collapsed">

<TextBlock Style="{DynamicResource

Рисунок 2.1.32 –Продолжение листинга NoticeInfoPage.xaml

Продолжение приложения 2

MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Вердикт:" />

<ComboBox Margin="7"

Width="200"

HorizontalAlignment="Right"

x:Name="ResultComboBox"

SelectedItem="{Binding Result}"

DisplayMemberPath="Description" />

</DockPanel>

<DockPanel VerticalAlignment="Center"

x:Name="ViewResultPanel"

Visibility="Collapsed">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Вердикт:" />

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

HorizontalAlignment="Right"

Text="{Binding Result.Description}" />

</DockPanel>

</StackPanel>

<StackPanel x:Name="AddMedCommissionPanel"

Visibility="Collapsed"

Orientation="Vertical">

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Жалобы на здоровье:" />

<TextBox HorizontalAlignment="Right"

Width="200"

MaxLength="500"

Margin="7"

TextWrapping="Wrap"

Text="{Binding HealthComplaints}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Диагнозы:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

MaxLength="500"

TextWrapping="Wrap"

Text="{Binding Diagnoses}" />

<CheckBox Content="Подтверждено справками"

IsChecked="{Binding Confirmed}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

Рисунок 2.1.33 –Продолжение листинга NoticeInfoPage.xaml

Продолжение приложения 2

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Заметка:" />

<TextBox HorizontalAlignment="Right"

Width="200"

MaxLength="200"

Margin="7"

TextWrapping="Wrap"

Text="{Binding Note}" />

</DockPanel>

<DockPanel VerticalAlignment="Center">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Категория:" />

<ComboBox Margin="7"

Width="200"

HorizontalAlignment="Right"

x:Name="CategoryComboBox"

DisplayMemberPath="CategoryKey"

SelectedItem="{Binding Category}" />

</DockPanel>

<Button Margin="7"

Padding="5"

x:Name="SaveMedButton"

Content="Сохранить"

Click="SaveMedButton\_OnClick" />

</StackPanel>

<Button Margin="7"

Padding="5"

x:Name="AddMedCommissionButton"

Visibility="Collapsed"

Content="Добавить информацию о мед. комиссии"

Click="AddMedCommissionButton\_OnClick" />

<Button Margin="7"

Padding="5"

x:Name="SaveResultButton"

Visibility="Collapsed"

Content="Сохранить"

Click="SaveResultButton\_OnClick" />

</StackPanel>

</design:Card>

</Grid>

</Page>

Рисунок 2.1.34 –Продолжение листинга NoticeInfoPage.xaml

using System.Windows;

using System.Windows.Controls;

using MilitaryConscriptionSystem.Database;

namespace MilitaryConscriptionSystem.AllPages;

Рисунок 2.1.35 –Листинг NoticeInfoPage.xaml.cs

Продолжение приложения 2

public partial class NoticeInfoPage : Page

{

private MilitaryDraftNotice \_notice;

public NoticeInfoPage(MilitaryDraftNotice notice)

{

\_notice = notice;

InitializeComponent();

CategoryComboBox.ItemsSource = Db.Context.Categories.ToList();

ResultComboBox.ItemsSource = Db.Context.Results.ToList();

MainInfoPanel.DataContext = \_notice;

ViewResultPanel.DataContext = \_notice;

AddResultPanel.DataContext = \_notice;

LoadData();

}

private void LoadData()

{

try

{

MedicalCommission? commission = Db.Context.MedicalCommissions.Find(\_notice.MilitaryDraftNoticeId);

if (commission == null)

{

if (App.Employee.PositionId == 4)

AddMedCommissionButton.Visibility = Visibility.Visible;

}

else if (\_notice.Result == null)

{

ViewMedCommissionPanel.Visibility = Visibility.Visible;

ViewMedCommissionPanel.DataContext = commission;

AddResultPanel.Visibility = Visibility.Visible;

SaveResultButton.Visibility = Visibility.Visible;

}

else

{

ViewMedCommissionPanel.Visibility = Visibility.Visible;

ViewMedCommissionPanel.DataContext = commission;

ViewResultPanel.Visibility = Visibility.Visible;

}

}

catch (Exception e)

{

Console.WriteLine(e);

MessageBox.Show(e.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void ConscriptButton\_OnClick(object sender, RoutedEventArgs e) =>

NavigationService.Navigate(new EditConscriptPage(\_notice.Conscript));

Рисунок 2.1.36 –Продолжение листинга NoticeInfoPage.xaml.cs

Продолжение приложения 2

private void AddMedCommissionButton\_OnClick(object sender, RoutedEventArgs e)

{

try

{

AddMedCommissionButton.Visibility = Visibility.Collapsed;

AddMedCommissionPanel.Visibility = Visibility.Visible;

MedicalCommission commission = new() { MilitaryDraftNoticeId = \_notice.MilitaryDraftNoticeId };

AddMedCommissionPanel.DataContext = commission;

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void SaveMedButton\_OnClick(object sender, RoutedEventArgs e)

{

try

{

MedicalCommission commission = (MedicalCommission)AddMedCommissionPanel.DataContext;

Db.Context.MedicalCommissions.Add(commission);

Db.Context.SaveChanges();

MessageBox.Show("Данные сохранены!", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Asterisk);

AddMedCommissionPanel.Visibility = Visibility.Collapsed;

LoadData();

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void SaveResultButton\_OnClick(object sender, RoutedEventArgs e)

{

try

{

Db.Context.MilitaryDraftNotices.Update(\_notice);

Db.Context.SaveChanges();

MessageBox.Show("Данные сохранены!", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Asterisk);

SaveResultButton.Visibility = Visibility.Collapsed;

AddResultPanel.Visibility = Visibility.Collapsed;

ViewResultPanel.Visibility = Visibility.Visible;

Рисунок 2.1.37 –Продолжение листинга NoticeInfoPage.xaml.cs

Продолжение приложения 2

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

Рисунок 2.1.38 –Продолжение листинга NoticeInfoPage.xaml.cs

<Page x:Class="MilitaryConscriptionSystem.AllPages.ViewConsCommissionsPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:MilitaryConscriptionSystem.AllPages"

mc:Ignorable="d"

Title="Список комиссий"

d:DesignHeight="450"

d:DesignWidth="800">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="\*" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<StackPanel Orientation="Horizontal">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Поиск:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

x:Name="SearchTextBox"

TextChanged="SearchTextBox\_OnTextChanged" />

</StackPanel>

<ScrollViewer Grid.Row="1">

<DataGrid x:Name="DataGrid"

IsReadOnly="True"

AutoGenerateColumns="False">

<DataGrid.Columns>

<DataGridTextColumn Header="Дата начала работы"

Width="\*"

Binding="{Binding Commission.CreateDate}" />

<DataGridTextColumn Header="Дата окончания работы"

Width="\*"

Binding="{Binding Commission.WorksUntilDate}" />

<DataGridTextColumn Header="Протокол"

Width="\*"

Рисунок 2.1.39 –Листинг ViewConsCommissionsPage.xaml

Продолжение приложения 2

Binding="{Binding Commission.ContainsProtocol}" />

<DataGridTemplateColumn Header="Сотрудники"

Width="\*">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<ItemsControl ItemsSource="{Binding Employees}">

<ItemsControl.ItemTemplate>

<DataTemplate>

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="{Binding ComboBoxField}" />

</DataTemplate>

</ItemsControl.ItemTemplate>

</ItemsControl>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Width="\*">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Margin="7"

Padding="5"

Content="Загрузить протокол"

x:Name="ProtocolButton"

Click="ProtocolButton\_OnClick" />

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</ScrollViewer>

<Button Margin="7"

Padding="5"

x:Name="AddButton"

Click="AddButton\_OnClick"

Grid.Row="2"

Content="Добавить" />

</Grid>

</Page>

Рисунок 2.1.40 –Продолжение листинга ViewConsCommissionsPage.xaml

using System.IO;

using System.Windows;

using System.Windows.Controls;

using Microsoft.EntityFrameworkCore;

using Microsoft.Win32;

using MilitaryConscriptionSystem.Database;

Рисунок 2.1.41 –Листинг ViewConsCommissionsPage.xaml.cs

Продолжение приложения 2

namespace MilitaryConscriptionSystem.AllPages;

public partial class ViewConsCommissionsPage : Page

{

class ViewConscriptionCommission()

{

public ConscriptionCommission Commission { get; set; }

public List<Employee> Employees { get; set; }

}

public ViewConsCommissionsPage()

{

InitializeComponent();

LoadData();

}

private void SearchTextBox\_OnTextChanged(object sender, TextChangedEventArgs e)

{

LoadData();

}

private void LoadData()

{

try

{

string search = SearchTextBox.Text;

List<ViewConscriptionCommission> commissions = new();

foreach (var commission in Db.Context.ConscriptionCommissions.ToList())

{

commissions.Add(new()

{

Commission = commission,

Employees = Db.Context.ConscriptionCommissionEmployees

.Include(c => c.Employee)

.Include(c => c.Employee.Position)

.Where(c => c.ConscriptionCommissionId == commission.ConscriptionCommissionId)

.Select(c => c.Employee).ToList()

});

}

commissions = commissions.Where(c => c.Commission.CreateDate.Value.ToString().Contains(search) ||

c.Commission.WorksUntilDate.Value.ToString().Contains(search) ||

c.Employees.Any(e => e.FirstName.ToLower().Contains(search)) ||

c.Employees.Any(e => e.LastName.ToLower().Contains(search)))

.ToList();

DataGrid.ItemsSource = null;

Рисунок 2.1.42 –Продолжение листинга ViewConsCommissionsPage.xaml.cs

Продолжение приложения 2

DataGrid.ItemsSource = commissions;

}

catch (Exception e)

{

Console.WriteLine(e);

MessageBox.Show(e.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void ProtocolButton\_OnClick(object sender, RoutedEventArgs e)

{

try

{

ConscriptionCommission commission = ((ViewConscriptionCommission)((Button)sender).DataContext).Commission;

OpenFileDialog dialog = new() { Filter = "documents | \*.docx; \*.doc;\*.pdf" };

if (dialog.ShowDialog() == true)

{

commission.Protocol = File.ReadAllBytes(dialog.FileName);

Db.Context.SaveChanges();

MessageBox.Show("Данные сохранены", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Asterisk);

LoadData();

}

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void AddButton\_OnClick(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new AddConsCommissionPage());

}

}

Рисунок 2.1.43 –Продолжение листинга ViewConsCommissionsPage.xaml.cs

<Page x:Class="MilitaryConscriptionSystem.AllPages.ViewConscriptsPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:MilitaryConscriptionSystem.AllPages"

mc:Ignorable="d"

Title="Список призывников"

d:DesignHeight="450"

d:DesignWidth="800">

<Grid>

Рисунок 2.1.44 –Листинг ViewConscriptsPage.xaml

Продолжение приложения 2

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="\*" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<StackPanel Orientation="Horizontal">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Поиск:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

x:Name="SearchTextBox"

TextChanged="SearchTextBox\_OnTextChanged" />

</StackPanel>

<ScrollViewer Grid.Row="1">

<DataGrid x:Name="DataGrid"

IsReadOnly="True"

AutoGenerateColumns="False">

<DataGrid.Columns>

<DataGridTextColumn Header="Фамилия"

Width="\*"

Binding="{Binding LastName}" />

<DataGridTextColumn Header="Имя"

Width="\*"

Binding="{Binding FirstName}" />

<DataGridTextColumn Header="Отчество"

Width="\*"

Binding="{Binding MiddleName}" />

<DataGridTextColumn Header="Дата рождения"

Width="\*"

Binding="{Binding BirthDate}" />

</DataGrid.Columns></DataGrid>

</ScrollViewer>

<StackPanel Orientation="Horizontal"

Grid.Row="2">

<Button Margin="7"

Padding="5"

x:Name="AddButton"

Click="AddButton\_OnClick"

Content="Добавить" />

<Button Margin="7"

Padding="5"

x:Name="ViewInfoButton"

Click="ViewInfoButton\_OnClick"

Content="Информация" />

<Button Margin="7"

Padding="5"

x:Name="CreateNoticeButton"

Visibility="Collapsed"

Click="CreateNoticeButton\_OnClick"

Content="Создать повестку" />

</StackPanel>

</Grid>

</Page>

Рисунок 2.1.45 –Продолжение листинга ViewConscriptsPage.xaml

Продолжение приложения 2

using System.Windows;

using System.Windows.Controls;

using Microsoft.EntityFrameworkCore;

using MilitaryConscriptionSystem.Database;

namespace MilitaryConscriptionSystem.AllPages;

public partial class ViewConscriptsPage : Page

{

public ViewConscriptsPage()

{

InitializeComponent();

LoadData();

if (App.Employee.PositionId == 5) CreateNoticeButton.Visibility = Visibility.Visible;

}

private void LoadData()

{

try

{

string search = SearchTextBox.Text.ToLower();

List<Conscript> conscripts =

Db.Context.Conscripts

.Include(c => c.Passport)

.Where(c =>

c.FirstName.ToLower().Contains(search) ||

c.LastName.ToLower().Contains(search) ||

(c.MiddleName != null && c.MiddleName.ToLower().Contains(search)))

.ToList();

DataGrid.ItemsSource = null;

DataGrid.ItemsSource = conscripts;

}

catch (Exception e)

{

Console.WriteLine(e);

MessageBox.Show(e.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void SearchTextBox\_OnTextChanged(object sender, TextChangedEventArgs e)

{

LoadData();

}

private void AddButton\_OnClick(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new EditConscriptPage());

}

Рисунок 2.1.46 –Листинг ViewConscriptsPage.xaml.cs

Продолжение приложения 2

private void ViewInfoButton\_OnClick(object sender, RoutedEventArgs e)

{

try

{

Conscript? conscript = DataGrid.SelectedItem as Conscript;

if (conscript != null)

{

NavigationService.Navigate(new EditConscriptPage(conscript));

}

else

{

MessageBox.Show("Выберите сначала призывника из таблицы", "Сообщение", MessageBoxButton.OK,

MessageBoxImage.Information);

}

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void CreateNoticeButton\_OnClick(object sender, RoutedEventArgs e)

{

Conscript? conscript = DataGrid.SelectedItem as Conscript;

if (conscript != null)

{

NavigationService.Navigate(new AddNoticePage(conscript));

}

else

{

MessageBox.Show("Выберите сначала призывника из таблицы", "Сообщение", MessageBoxButton.OK,

MessageBoxImage.Information);

}

}

}

Рисунок 2.1.47 –Продолжение листинга ViewConscriptsPage.xaml.cs

<Page x:Class="MilitaryConscriptionSystem.AllPages.ViewEmployeesPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:MilitaryConscriptionSystem.AllPages"

mc:Ignorable="d"

Title="Сотрудники"

d:DesignHeight="450"

d:DesignWidth="800">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

Рисунок 2.1.48 –Листинг ViewEmployeesPage.xaml

Продолжение приложения 2

<RowDefinition Height="\*" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<StackPanel Orientation="Horizontal">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Поиск:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

x:Name="SearchTextBox"

TextChanged="SearchTextBox\_OnTextChanged" />

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Должность:" />

<ComboBox Margin="7"

Width="200"

HorizontalAlignment="Right"

x:Name="PositionComboBox"

SelectionChanged="PositionComboBox\_OnSelectionChanged"

DisplayMemberPath="Name" />

</StackPanel>

<ScrollViewer Grid.Row="1">

<DataGrid x:Name="DataGrid"

IsReadOnly="True"

AutoGenerateColumns="False">

<DataGrid.Columns>

<DataGridTextColumn Header="Фамилия"

Width="\*"

Binding="{Binding LastName}" />

<DataGridTextColumn Header="Имя"

Width="\*"

Binding="{Binding FirstName}" />

<DataGridTextColumn Header="Отчество"

Width="\*"

Binding="{Binding MiddleName}" />

<DataGridTextColumn Header="Должность"

Width="\*"

Binding="{Binding Position.Name}" />

</DataGrid.Columns>

</DataGrid>

</ScrollViewer>

<StackPanel Orientation="Horizontal"

Grid.Row="2">

<Button Margin="7"

Padding="5"

x:Name="AddButton"

Click="AddButton\_OnClick"

Content="Добавить" />

<Button Margin="7"

Рисунок 2.1.49 –Продолжение листинга ViewEmployeesPage.xaml

Продолжение приложения 2

Padding="5"

x:Name="ViewInfoButton"

Click="ViewInfoButton\_OnClick"

Content="Информация" />

</StackPanel>

</Grid>

</Page>

Рисунок 2.1.50 –Продолжение листинга ViewEmployeesPage.xaml

using System.Windows;

using System.Windows.Controls;

using Microsoft.EntityFrameworkCore;

using MilitaryConscriptionSystem.Database;

namespace MilitaryConscriptionSystem.AllPages;

public partial class ViewEmployeesPage : Page

{

public ViewEmployeesPage()

{

InitializeComponent();

List<Position> positions = new() { new() { Name = "Все" } };

positions.AddRange(Db.Context.Positions.ToList());

PositionComboBox.ItemsSource = positions;

LoadData();

}

private void LoadData()

{

try

{

string search = SearchTextBox.Text.ToLower();

Position? position = (Position)PositionComboBox.SelectedItem;

List<Employee> employees =

Db.Context.Employees

.Include(c => c.Passport)

.Include(c => c.Position)

.Where(c =>

c.FirstName.ToLower().Contains(search) ||

c.LastName.ToLower().Contains(search) ||

(c.MiddleName != null && c.MiddleName.ToLower().Contains(search)))

.ToList();

if (position != null && position.PositionId != 0)

employees = employees.Where(c => c.PositionId == position.PositionId).ToList();

DataGrid.ItemsSource = null;

DataGrid.ItemsSource = employees;

}

Рисунок 2.1.51 –Листинг ViewEmployeesPage.xaml.cs

Продолжение приложения 2

catch (Exception e)

{

Console.WriteLine(e);

MessageBox.Show(e.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void SearchTextBox\_OnTextChanged(object sender, TextChangedEventArgs e)

{

LoadData();

}

private void AddButton\_OnClick(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new EditEmployeePage());

}

private void ViewInfoButton\_OnClick(object sender, RoutedEventArgs e)

{

try

{

Employee? employee = DataGrid.SelectedItem as Employee;

if (employee != null)

{

if (employee.PositionId != 6)

NavigationService.Navigate(new EditEmployeePage(employee));

else

MessageBox.Show("Нельзя редактировать администратора", "Сообщение", MessageBoxButton.OK,

MessageBoxImage.Information);

}

else

{

MessageBox.Show("Выберите сначала сотрудника из таблицы", "Сообщение", MessageBoxButton.OK,

MessageBoxImage.Information);

}

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void PositionComboBox\_OnSelectionChanged(object sender, SelectionChangedEventArgs e)

{

LoadData();

}

}

Рисунок 2.1.52 –Продолжение листинга ViewEmployeesPage.xaml.cs

Продолжение приложения 2

<Page x:Class="MilitaryConscriptionSystem.AllPages.ViewNoticesPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:MilitaryConscriptionSystem.AllPages"

mc:Ignorable="d"

Title="Повестки"

d:DesignHeight="450"

d:DesignWidth="800">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="\*" />

</Grid.RowDefinitions>

<StackPanel Orientation="Vertical">

<StackPanel Orientation="Horizontal">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="Поиск:" />

<TextBox HorizontalAlignment="Right"

Width="200"

Margin="7"

x:Name="SearchTextBox"

TextChanged="SearchTextBox\_OnTextChanged" />

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="С:" />

<DatePicker x:Name="StartDatePicker"

SelectedDateChanged="DatePicker\_OnSelectedDateChanged"

Width="100"

Margin="5" />

<TextBlock Style="{DynamicResource MaterialDesignTextBlock}"

VerticalAlignment="Center"

Text="По:" />

<DatePicker x:Name="EndDatePicker"

SelectedDateChanged="DatePicker\_OnSelectedDateChanged"

DisplayDateStart="{Binding ElementName=StartDatePicker, Path=SelectedDate}"

Width="100"

Margin="5" />

<Button x:Name="ResetButton"

Content="Сброс"

VerticalAlignment="Center"

Margin="5"

Click="ResetButton\_OnClick"/>

Рисунок 2.1.53 –Листинг ViewNoticesPage.xaml

Продолжение приложения 2

</StackPanel>

</StackPanel>

<ScrollViewer Grid.Row="1">

<DataGrid x:Name="DataGrid"

IsReadOnly="True"

AutoGenerateColumns="False">

<DataGrid.Columns>

<DataGridTextColumn Header="Дата"

Width="\*"

Binding="{Binding Date}" />

<DataGridTextColumn Header="Время"

Width="\*"

Binding="{Binding Time, StringFormat='{}{0:HH:mm}'}" />

<DataGridTextColumn Header="Адрес"

Width="\*"

Binding="{Binding Address}" />

<DataGridTextColumn Header="Фамилия"

Width="\*"

Binding="{Binding Conscript.FirstName}" />

<DataGridTextColumn Header="Имя"

Width="\*"

Binding="{Binding Conscript.LastName}" />

<DataGridTextColumn Header="Отчество"

Width="\*"

Binding="{Binding Conscript.MiddleName}" />

<DataGridTemplateColumn Width="\*">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Margin="7"

Padding="5"

Content="Подробнее"

Click="InfoButton\_OnClick" />

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Width="\*">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Margin="7"

Padding="5"

Content="Документ"

Click="DocumentButton\_OnClick" />

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</ScrollViewer>

</Grid>

</Page>

Рисунок 2.1.54 –Продолжение листинга ViewNoticesPage.xaml

Продолжение приложения 2

using System.Windows;

using System.Windows.Controls;

using Microsoft.EntityFrameworkCore;

using MilitaryConscriptionSystem.Database;

using Xceed.Document.NET;

using Xceed.Words.NET;

namespace MilitaryConscriptionSystem.AllPages;

public partial class ViewNoticesPage : Page

{

public ViewNoticesPage()

{

InitializeComponent();

LoadData();

}

private void SearchTextBox\_OnTextChanged(object sender, TextChangedEventArgs e)

{

LoadData();

}

private void LoadData()

{

try

{

string search = SearchTextBox.Text.ToLower();

DateTime? startDate = StartDatePicker.SelectedDate;

DateTime? endDate = EndDatePicker.SelectedDate;

List<MilitaryDraftNotice> notices = Db.Context.MilitaryDraftNotices

.Include(c => c.Conscript)

.Include(c => c.Conscript.Passport)

.Where(c => (c.Conscript.FirstName.ToLower().Contains(search) ||

c.Conscript.LastName.ToLower().Contains(search) ||

(c.Conscript.MiddleName != null && c.Conscript.MiddleName.ToLower().Contains(search))))

.ToList();

notices = notices.Where(c =>

((startDate != null && c.Date.Value >= DateOnly.FromDateTime(startDate.Value)) ||

startDate == null) &&

((endDate != null && c.Date.Value <= DateOnly.FromDateTime(endDate.Value)) || endDate == null))

.ToList();

if (App.Employee.PositionId != 5)

{

Рисунок 2.1.55 –Листинг ViewNoticesPage.xaml.cs

Продолжение приложения 2

List<ConscriptionCommission> commissions = Db.Context.ConscriptionCommissionEmployees

.Where(c => c.EmployeeId == App.Employee.PassportId).Select(c => c.ConscriptionCommission).ToList();

notices = notices.Where(c =>

commissions.Any(a => a.ConscriptionCommissionId == c.ConscriptionCommissionId))

.ToList();

}

DataGrid.ItemsSource = null;

DataGrid.ItemsSource = notices;

}

catch (Exception e)

{

Console.WriteLine(e);

MessageBox.Show(e.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void InfoButton\_OnClick(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new NoticeInfoPage(((Button)sender).DataContext as MilitaryDraftNotice));

}

private void DatePicker\_OnSelectedDateChanged(object? sender, SelectionChangedEventArgs e)

{

LoadData();

}

private void ResetButton\_OnClick(object sender, RoutedEventArgs e)

{

SearchTextBox.Text = String.Empty;

StartDatePicker.SelectedDate = null;

EndDatePicker.SelectedDate = null;

}

private void DocumentButton\_OnClick(object sender, RoutedEventArgs e)

{

try

{

MilitaryDraftNotice notice = ((Button)sender).DataContext as MilitaryDraftNotice;

Document document = DocX.Load("шаблон.docx");

string conscriptFullName =

$"{notice.Conscript.LastName} {notice.Conscript.FirstName} {notice.Conscript.MiddleName}";

string commissarFullName =

$"{App.Employee.LastName} {App.Employee.FirstName}

Рисунок 2.1.56 – Продолжение листинга ViewNoticesPage.xaml.cs

Продолжение приложения 2

{App.Employee.MiddleName}";

document.ReplaceText("<<гражданин>>", conscriptFullName);

document.ReplaceText("<<номер>>", notice.MilitaryDraftNoticeId.ToString());

document.ReplaceText("<<день>>", notice.Date.Value.Day.ToString());

document.ReplaceText("<<месяц>>", GetMonthName(notice.Date.Value.Month));

document.ReplaceText("<<год>>", notice.Date.Value.Year.ToString());

document.ReplaceText("<<час>>", notice.Time.Value.Hour.ToString());

document.ReplaceText("<<адрес>>", notice.Address);

document.ReplaceText("<<фио комиссара>>", commissarFullName);

document.SaveAs($"Повестка\_#{notice.MilitaryDraftNoticeId}\_{notice.Conscript.LastName}.docx");

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private string GetMonthName(int month)

{

return month switch

{

1 => "Января",

2 => "Февраля",

3 => "Марта",

4 => "Апреля",

5 => "Мая",

6 => "Июня",

7 => "Июля",

8 => "Августа",

9 => "Сентября",

10 => "Октября",

11 => "Ноября",

12 => "Декабря",

\_ => ""

};

}

}

Рисунок 2.1.57 – Продолжение листинга ViewNoticesPage.xaml.cs

<Window x:Class="MilitaryConscriptionSystem.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>

Рисунок 2.1.58 – Продолжение листинга MainWindow.xaml

Продолжение приложения 2

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:MilitaryConscriptionSystem"

xmlns:design="http://materialdesigninxaml.net/winfx/xaml/themes"

mc:Ignorable="d"

ResizeMode="CanResizeWithGrip"

Background="#fafafa"

Title="Система учета военнообязанных"

MinHeight="350"

MinWidth="500">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="3\*" />

</Grid.RowDefinitions>

<Border BorderBrush="Black"

BorderThickness="0,0,0,1">

<design:Card>

<Grid>

<TextBlock Style="{DynamicResource MaterialDesignSubtitle1TextBlock}"

VerticalAlignment="Center"

Margin="10"

Text="{Binding ElementName=Frame, Path=Content.Title}"

MaxWidth="300"

HorizontalAlignment="Center"

TextWrapping="Wrap" />

<Button x:Name="BackButton"

Grid.Row="0"

Margin="7"

Padding="5"

HorizontalAlignment="Right"

Visibility="Collapsed"

Click="BackButton\_OnClick"

Content="Назад" />

<Button Margin="7"

Padding="5"

x:Name="ExitButton"

HorizontalAlignment="Right"

Grid.Row="0"

Click="ExitButton\_OnClick"

Visibility="Collapsed"

Content="Выйти" />

</Grid>

</design:Card>

</Border>

<Frame x:Name="Frame"

Grid.Row="1"

Navigated="Frame\_OnNavigated"

Source="../AllPages/LoginPage.xaml"

NavigationUIVisibility="Hidden" />

</Grid>

</Window>

Рисунок 2.1.59 – Продолжение листинга MainWindow.xaml

Продолжение приложения 2

using System.Text;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using MilitaryConscriptionSystem.AllPages;

namespace MilitaryConscriptionSystem;

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Frame\_OnNavigated(object sender, NavigationEventArgs e)

{

try

{

if (Frame.CanGoBack)

{

if (Frame.Content.GetType() == typeof(MenuPage))

{

BackButton.Visibility = Visibility.Collapsed;

ExitButton.Visibility = Visibility.Visible;

}

else if (Frame.CanGoBack)

{

BackButton.Visibility = Visibility.Visible;

ExitButton.Visibility = Visibility.Collapsed;

}

}

else

{

BackButton.Visibility = Visibility.Collapsed;

ExitButton.Visibility = Visibility.Collapsed;

}

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

Рисунок 2.1.60 – Листинг MainWindow.xaml.cs

Продолжение приложения 2

private void BackButton\_OnClick(object sender, RoutedEventArgs e) => Frame.GoBack();

private void ExitButton\_OnClick(object sender, RoutedEventArgs e)

{

try

{

App.Employee = null;

Frame.GoBack();

}

catch (Exception exception)

{

Console.WriteLine(exception);

MessageBox.Show(exception.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

Рисунок 2.1.61 – Продолжение листинга MainWindow.xaml.cs