Оглавление

[1. Введение 1](#_Toc138148705)

[2. Технологические аспекты разработки программного комплекса 2](#_Toc138148706)

[2.1. Исследование области 2](#_Toc138148707)

[2.2. Архитектура программного комплекса 2](#_Toc138148708)

[2.2.1 Общие сведения о программном комплексе 2](#_Toc138148709)

[2.2.2 Композитор сюжетов (КС) 3](#_Toc138148710)

[2.2.3 ТШ 3](#_Toc138148711)

[2.2.4 ВИА 4](#_Toc138148712)

[2.3. Использованные технологии, библиотеки и архитектурные шаблоны 5](#_Toc138148713)

[2.3.1. C#, .NET 5](#_Toc138148714)

[2.3.2 WPF 5](#_Toc138148715)

[2.3.3 MVVM 6](#_Toc138148716)

[2.3.4. MongoDB 7](#_Toc138148717)

[2.3.5 VkNet 8](#_Toc138148718)

[2.3.6 FuzzySharp 8](#_Toc138148719)

[2.3.7. Extended WPF Toolkit 9](#_Toc138148720)

[2.4 Организация хранения информации в базе данных 10](#_Toc138148721)

[2.4.1. Концептуальная модель базы данных 10](#_Toc138148722)

[2.4.2. Логическая модель базы данных 11](#_Toc138148723)

[3. Разработка и внедрение программного комплекса 12](#_Toc138148724)

[3.1 Разработка программного комплекса 12](#_Toc138148725)

[3.1.1. Разработка физической модели базы данных 12](#_Toc138148726)

[3.1.2 Разработка проекта NfModels 14](#_Toc138148727)

# 1. Введение

Современные телеканалы сталкиваются с множеством бизнес-задач, которые необходимо эффективно решать для успешной работы и предоставления качественных услуг. Управление процессами, контроль персонала, анализ данных - все эти аспекты играют важную роль в оперативности и эффективности работы телеканала. Для решения этих задач разработка соответствующего программного комплекса становится неотъемлемой необходимостью. Основными целями такого комплекса являются автоматизация бизнес-процессов, повышение производительности и эффективности, упрощение управления и улучшение предоставления услуг.

Целью дипломной работы является проектирование, разработка и тестирование программного комплекса "Фабрика новостей" с использованием технологий WPF и СУБД MongoDB.

Данный комплекс направлен на решение бизнес-задач редакции телеканала, предоставляя средства для систематизации, каталогизации и хранения оперативных материалов, организации архивного хранения и аналитики данных.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* Проанализировать бизнес-процессы, сформировавшиеся в редакции телеканала, для определения основных потребностей и требований к программному комплексу;
* Спроектировать архитектуру программного комплекса "Фабрика новостей", учитывая особенности работы и потребности редакции телеканала;
* Разрабатывать базу данных в СУБД MongoDB, обеспечивая надежное хранение информации о процессах редакции;
* Создавать пользовательский интерфейс программного комплекса с использованием технологии WPF и языка программирования C#, чтобы обеспечить удобство и интуитивно понятный доступ к функциональности комплекса;
* Провести тестирование и отладку программного комплекса для обеспечения его стабильной работы, и соответствия требованиям редакции телеканала;
* Оформить необходимую документацию к дипломной работе, включая техническое описание, инструкции и руководства пользователя.

Перейдём к рассмотрению технологических аспектов разработки программного комплекса.

# 2. Технологические аспекты разработки программного комплекса

# 2.1. Исследование области

Словарь терминов:

Люфт (нем. Luft: 1) воздух; 2) зазор, просвет) — включение в сюжет живой реальной сценки, «картинки» с соответствующим интершумом, но без закадрового текста. Нередко люфт дается до стендапа для того, чтобы зритель сразу погрузился в атмосферу события.

Синхрон – синоним интервью.

Интершум – синоним фоновый шум.

Закадровый голос – голос корреспондента, идущий за кадром во время новостного сюжета.

# 2.2. Архитектура программного комплекса

# 2.2.1 Общие сведения о программном комплексе

Программного комплекс "Фабрика новостей" состоит из нескольких программ:

* *База данных*: Хранит информацию о проектах, материалах, съемках, метках и работниках, используя СУБД MongoDB. База данных обеспечивает доступ, сохранение и управление данными, необходимыми для работы программного комплекса.
* *Композитор Сюжетов (КС):* Редактор сюжетов, предоставляющий блочный редактор для создания и редактирования сюжетов.
* *Телевизионный Штаб (ТШ):* Менеджер проектов, обеспечивающий контроль над телевизионными проектами, съемками и материалами.
* *Виртуальный Интерактивный Агент (ВАИ)*: Бот, предоставляющий доступ к запланированным съемкам, ходу работы над проектами и отчетности через социальную сеть ВКонтакте.

Архитектура программного комплекса "Фабрика новостей" представляет собой распределенную систему, где каждый компонент выполняет определенные функции и взаимодействует с другими компонентами. Модульность комплекса позволяет легко масштабировать и модифицировать систему, а также облегчает поддержку и разработку каждого модуля независимо от остальных. Взаимодействие между компонентами осуществляется посредством стандартного для всех интерфейса, что стандартизирует и упрощает связи между ними. Централизованное хранилище данных обеспечивает целостность и доступность информации.

# 2.2.2 Композитор сюжетов (КС)

Композитор Сюжетов (КС) играет ключевую роль в комплексе "Фабрика новостей" и является неотъемлемой частью процесса создания и редактирования телевизионных сюжетов. Он предоставляет мощный инструментарий для структурирования, организации и визуализации информации, связанной с сюжетами.

Работа композитора сюжетов включает в себя различные аспекты, начиная от создания новых сюжетов и заканчивая редактированием уже существующих. Композитор сюжетов предоставляет блочный редактор, который позволяет создавать сюжеты с использованием разных элементов и компонентов. Он предоставляет возможность структурировать информацию в виде блоков, которые можно легко перемещать, редактировать и комбинировать.

Композитор сюжетов облегчает процесс создания сюжетов путем предоставления гибкого и интуитивно понятного интерфейса, который позволяет редакторам легко манипулировать контентом и расставлять акценты в сюжетах. Он также обеспечивает визуализацию сюжетов, позволяя редакторам видеть структуру и последовательность событий в сюжете.

Роль композитора сюжетов в комплексе заключается не только в создании и редактировании сюжетов, но и в обеспечении эффективной работы всей системы "Фабрика новостей". Он является инструментом, который помогает редакторам телевизионных сюжетов в их творческом процессе, упрощает организацию информации и повышает эффективность работы всей команды.

# 2.2.3 ТШ

Телевизионный Штаб (ТШ) в программном комплексе "Фабрика новостей" играет важную роль в организации и управлении телевизионными проектами. Он выполняет функции центра управления и координации различных процессов, связанных с созданием сюжетов и управлением съемками.

ТШ является основным инструментом для планирования и контроля хода работы над телевизионными проектами. Он обеспечивает возможность управлять проектами, назначать задачи и контролировать сроки выполнения. Таким образом, ТШ способствует эффективной организации работы и соблюдению установленных сроков.

Кроме того, ТШ обеспечивает централизованный доступ к материалам, связанным с телевизионными проектами. Он предоставляет возможность управлять базой данных, где хранятся съемки, материалы и другая информация, необходимая для создания сюжетов. ТШ позволяет удобно организовать доступ к этим материалам для работников, связанных с проектами.

Таким образом, роль Телевизионного Штаба в комплексе заключается в организации и управлении телевизионными проектами, планировании и контроле хода работы, а также в обеспечении удобного доступа к материалам базы данных проекта.

# 2.2.4 ВИА

Виртуальный Интерактивный Агент (ВИА) является одним из компонентов программного комплекса "Фабрика новостей". Он представляет собой бота, интегрированного с социальной сетью ВКонтакте, который обеспечивает доступ к запланированным съемкам, проектам, а также информации о материалах.

Роль ВИА в комплексе заключается в предоставлении удобного и интерактивного интерфейса для сотрудников телеканала. Благодаря интеграции с ВКонтакте, ВИА позволяет работникам получать информацию о запланированных съемках, следить за ходом работы над проектами и получать оперативные уведомления о новых материалах и съемках.

Использование платформы ВКонтакте в качестве основной среды общения с ВИА позволяет сотрудникам телеканала взаимодействовать с программным комплексом через уже существующую популярную платформу социальной сети. Это означает, что не требуется разработка и поддержка дополнительного веб-сайта или мобильного приложения, что в свою очередь экономит время и ресурсы команды разработчиков.

ВИА упрощает и ускоряет процессы взаимодействия с программным комплексом. Работники телеканала могут получать доступ к необходимой информации и выполнять определенные задачи, используя функциональность ВИА через интерфейс социальной сети ВКонтакте, находясь на выездных мероприятиях или дома. Это позволяет им работать удаленно и обмениваться данными с фабрикой новостей даже вне рабочего места.

# 2.3. Использованные технологии, библиотеки и архитектурные шаблоны

# 2.3.1. C#, .NET

Выбор языка программирования C# и платформы .NET для разработки программного комплекса "Фабрика новостей" был обусловлен несколькими факторами и потребностями проекта. Эти технологии предоставляют широкий набор инструментов и функциональности, позволяющих создать мощное и масштабируемое приложение для управления телевизионными сюжетами и проектами.

C# является объектно-ориентированным языком программирования, который обладает мощными средствами для разработки сложных и структурированных систем. Он предоставляет разработчикам удобный и понятный синтаксис, обширную стандартную библиотеку классов и инструменты для управления памятью и выполнения операций. Это позволяет эффективно создавать и поддерживать код, улучшая его читаемость, надежность и производительность.

Одним из основных преимуществ использования C# и .NET является существование огромной базы пользовательских пакетов, благодаря которым была обеспечена совместимость и интеграция с сервисами ВКонтакте и базой данных MongoDB.

Платформа .NET предлагает обширный набор инструментов и функций для разработки приложений любой сложности. Она обеспечивает удобное развертывание, масштабируемость и безопасность приложений.

Выбор C# и .NET для разработки "Фабрики новостей" обусловлен их возможностью создать мощное, гибкое и эффективное программное решение, которое способно управлять сложными процессами создания и управления телевизионными сюжетами. Эти технологии предоставляют все необходимые инструменты и функциональность для разработки программного обеспечения любого уровня.

# 2.3.2 WPF

Для разработки программного комплекса "Фабрика новостей" был выбран Windows Presentation Foundation (WPF) как технология для создания пользовательского интерфейса. Одной из основных причин выбора WPF была его способность обеспечить богатый и гибкий пользовательский интерфейс с возможностью создания сложных и красивых визуальных элементов.

WPF предлагает мощный инструментарий для разработки интерактивных приложений, основанных на привлекательных и интуитивно понятных пользовательских интерфейсах. Он позволяет разработчикам создавать интерфейсы с привлекательными графическими эффектами, анимацией, стилями и адаптивным дизайном. Такой подход помогает создавать современные и профессионально выглядящие приложения, что является важным аспектом в разработке программного комплекса для телеканала.

Другим важным фактором при выборе WPF была его интеграция с языком программирования C#. WPF предоставляет богатый набор инструментов и возможностей для разработки пользовательского интерфейса с использованием C#. Благодаря этому, разработчики могут использовать мощные функциональные возможности языка программирования и легко интегрировать их в интерфейс приложения.

WPF также обладает хорошей поддержкой стилизации и шаблонизации элементов пользовательского интерфейса, что позволяет создавать переиспользуемые компоненты и упрощает процесс разработки и поддержки интерфейса. Это особенно полезно при создании сложных редакторов и управляющих элементов, которые используются в программном комплексе "Фабрика новостей".

Таким образом, выбор WPF для разработки программного комплекса "Фабрика новостей" был обусловлен его способностью предоставить богатый и гибкий пользовательский интерфейс, интеграцию с языком программирования C#, поддержку стилизации и шаблонизации элементов. Эти возможности позволяют создать профессионально выглядящее и удобное в использовании приложение.

# 2.3.3 MVVM

При разработке программного комплекса "Фабрика новостей" был использован архитектурный шаблон MVVM (Model-View-ViewModel) для обеспечения модульности, удобства сопровождения и разделения ответственности между компонентами системы.

Основной причиной выбора шаблона MVVM является его способность эффективно разделять логику приложения и пользовательский интерфейс. Модель представляет собой состояние данных и бизнес-логику, которая управляет этими данными. View (представление) отвечает за отображение данных пользователю и обработку пользовательского ввода. ViewModel служит связующим звеном между моделью и представлением, обеспечивая передачу данных и команд между ними.

Выбор MVVM позволяет создать гибкую и масштабируемую архитектуру, где каждый компонент выполняет свою специфическую функцию. Модель содержит основную логику приложения и не зависит от конкретного представления, что обеспечивает ее повторное использование и тестируемость. Представление отвечает только за отображение данных и реагирует на изменения в ViewModel, что облегчает разработку пользовательского интерфейса. ViewModel обеспечивает связь между моделью и представлением, предоставляет данные для отображения и обрабатывает пользовательский ввод.

Выбор шаблона MVVM для разработки программного комплекса "Фабрика новостей" был обусловлен его преимуществами в разделении ответственности, обеспечении модульности и удобства сопровождения, что способствует эффективной разработке и поддержке системы на протяжении всего ее жизненного цикла.

# 2.3.4. MongoDB

При выборе СУБД для разработки программного комплекса "Фабрика новостей" было принято решение в пользу MongoDB в связи с рядом факторов.

Во-первых, MongoDB обладает высокой масштабируемостью и производительностью, что является важным аспектом при работе с большим объемом данных и высокими нагрузками. Фабрика новостей предполагает обработку и хранение большого количества информации о проектах и съемках, а MongoDB позволяет эффективно масштабировать систему и обеспечивать высокую производительность при выполнении запросов к базе данных.

Во-вторых, MongoDB предлагает гибкую схему данных, что означает, что структура данных может меняться со временем без необходимости проведения миграций схемы, что значительно упрощает разработку и сопровождение системы. Это особенно актуально для проекта "Фабрика новостей", где возможны изменения и доработки функционала со временем.

И наконец, MongoDB имеет обширную документацию и активное сообщество пользователей, что облегчает внедрение и поддержку СУБД. Разработчики программного комплекса могут полагаться на ресурсы и опыт сообщества для решения вопросов и получения поддержки.

В целом, выбор MongoDB для разработки "Фабрики новостей" обусловлен его способностью эффективно работать с гибкими структурами данных, обеспечивать высокую масштабируемость и производительность, а также гибкую схему данных и поддержку открытого сообщества.

# 2.3.5 VkNet

При разработке программного комплекса "Фабрика новостей" было принято решение выбрать библиотеку VkNet для интеграции с социальной сетью ВКонтакте. Это решение было обусловлено несколькими факторами и требованиями проекта.

Прежде всего, VkNet предоставляет удобный и простой интерфейс для взаимодействия с API ВКонтакте. Библиотека обеспечивает высокий уровень абстракции и инкапсуляции, что позволяет разработчикам легко и эффективно осуществлять различные операции, такие как отправка сообщений, получение информации о пользователях, работа с группами и многое другое.

Кроме того, VkNet активно поддерживается сообществом разработчиков и обновляется с учетом изменений в API ВКонтакте. Это позволяет быть уверенным в том, что библиотека будет соответствовать текущим требованиям и функциональности социальной сети.

Еще одним преимуществом выбора VkNet является его распространенность и популярность среди разработчиков. Существует множество документации, учебных материалов и сообществ, где можно получить поддержку и помощь в случае возникновения проблем или вопросов.

В результате, выбор VkNet для разработки программного комплекса "Фабрика новостей" был обусловлен его удобством, поддержкой, совместимостью и широкими возможностями для работы с API ВКонтакте. Это позволило эффективно реализовать функциональность Виртуального Интерактивного Агента и обеспечить комфортное взаимодействие с социальной сетью ВКонтакте в рамка программного комплекса.

# 2.3.6 FuzzySharp

FuzzySharp был использован в процессе разработки с целью реализации нечеткого сравнения и поиска текстовых данных. Он предоставил функциональность и алгоритмы, которые позволили сравнивать и оценивать схожесть между строками текста на основе различных метрик и показателей. Использование FuzzySharp помогло решить задачи, связанные с распознаванием и классификацией текстовых данных, а также определением степени схожести между ними. Это обеспечило более гибкую и точную обработку текстовых элементов в проекте, улучшая точность поиска и анализа данных.

# 2.3.7. Extended WPF Toolkit

В разработке программного комплекса "Фабрика новостей" я использовал Extended WPF Toolkit для расширения возможностей пользовательского интерфейса. Эта библиотека предоставляет дополнительные элементы управления, которые не входят в стандартный набор WPF, позволяя создавать более интерактивные и функциональные пользовательские интерфейсы.

Один из примеров использования элементов Extended WPF Toolkit в проекте был связан с созданием формы редактирования проектов телевизорного штаба. Для удобства пользователей и более эффективного взаимодействия с фабрикой новостей, было необходимо включить дополнительный функциональный элемент – DateTimePicker, который позволяет выбрать на календаре конкретную дату и время.

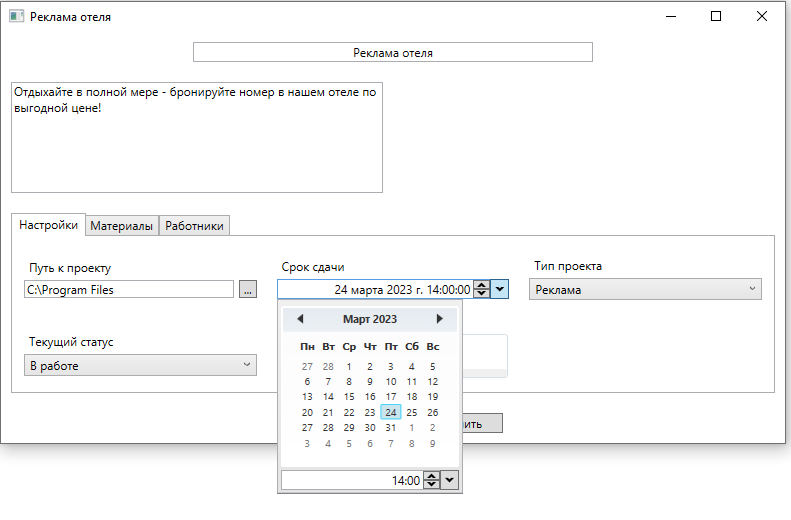


Рис. {N}

Благодаря использованию Extended WPF Toolkit, удалось значительно расширить возможности пользовательского интерфейса и создать более удобное и функциональное окружение для работы с фабрикой новостей.

# 2.4 Организация хранения информации в базе данных

# 2.4.1. Концептуальная модель базы данных

Концептуальная модель базы данных при использовании NoSQL СУБД MongoDB отличается от модели реляционных баз данных, так как MongoDB является документноориентированной СУБД, использующий использует формат документов BSON (Binary JSON), который позволяет хранить документы в виде иерархических структур. При проектировании базы данных необходимо учитывать эту структуру и определять иерархию данных.

MongoDB не требует определения схемы данных заранее, что позволяет гибче работать с данными и вносить изменения в структуру базы данных без необходимости пересоздания схемы. Также, это может приводить к неожиданным ошибкам в работе программного комплекса из-за отсутствия жесткой структуры данных.

Таблица {N}.

|  |  |
| --- | --- |
| **Сущности** | **Атрибуты сущностей** |
| Съёмка | Название, описание  Адрес  Время съёмки  Информация о работниках и выполняемых ими обязанностях на съёмке  Присвоенные метки |
| Материал | Название, описание  Путь к материалу в файловой системе  Тип материала  Присвоенные метки  Информация о работниках ответственных за материал |
| Проект | Название, описание  Использованные материалы  Путь к проекту в файловой системе  Срок сдачи проекта  Тип проекта  Информация о работниках и выполняемых ими обязанностях в проекте  Присвоенные метки |
| Работник | Имя, Фамилия  Должность |
| Метка (Тег) | Название, описание  Тег-родитель  Цвет |

# 2.4.2. Логическая модель базы данных

В MongoDB для хранения данных используется документоориентированная модель, что позволяет хранить связанные данные внутри одного документа или в нескольких документах с использованием ссылок. В случае использования нескольких коллекций, для программной реализации соединения таблиц в MongoDB можно использовать ссылки на документы.

В данном случае, связанные данные хранятся в разных коллекциях, а в документе коллекции, который ссылается на другой документ, хранится только идентификатор (ObjectId) этого документа.

Таблица {N}.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Связь | Главный документ | Дочерний документ | Поле для связи главного документа | Поле для связи дочернего документа |
| Метки съёмки | Filmings | Tags | Элемент поля «Tags» | \_id |
| Метки материала | Materials | Tags | Элемент поля «Tags» | \_id |
| Метки проекта | Projects | Tags | Элемент поля «Tags» | \_id |
| Работники съёмки | Filmings | Persons | Элемент поля «MyCrew» | \_id |
| Работники материала | Materials | Persons | Элемент поля «MyCrew» | \_id |
| Работники проекта | Projects | Persons | Элемент поля «MyCrew» | \_id |
| Материалы проекта | Projects | Materials | Элемент поля «Materials» | \_id |
| Родитель тега | Tags | Tags | «ParentTagId» | \_id |

# 3. Разработка и внедрение программного комплекса

# 3.1 Разработка программного комплекса

# 3.1.1. Разработка физической модели базы данных

В случае MongoDB, данные хранятся в виде BSON-документов (Binary JSON), каждый из которых содержит поля и значения, а также ссылки на другие документы.

Физическая модель базы данных программного комплекса "Фабрика новостей" по автоматизации бизнес-процессов редакции телеканала может быть описана следующим образом:

Таблица {N}. Коллекции базы данных «NewsFactory»

|  |  |
| --- | --- |
| Коллекция БД | Индексы |
| Filmings | \_id\_ - уникальность поля «\_id» |
| Materials | \_id\_ - уникальность поля «\_id» |
| Persons | \_id\_ - уникальность поля «\_id» |
| Projects | \_id\_ - уникальность поля «\_id» |
| Tag | \_id\_ - уникальность поля «\_id»  Title\_ - уникальность поля «Title» |

Таблица {N}. Структура документов коллекции «Filmings»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| \_id | ObjectId | Идентификатор съёмки |
| Title | String | Название съёмки |
| Description | String | Описание съёмки |
| CreationTime | Date | Дата создания объекта |
| Tags | Array | Метки |
| Status | Int32 | Текущий статус съёмки |
| MyTime | Date | Время съёмки |
| Address | String | Адрес съёмки |

Таблица {N}. Структура документов коллекции «Materials»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| \_id | ObjectId | Идентификатор материала |
| Title | String | Название материала |
| Description | String | Описание материала |
| CreationTime | Date | Дата создания объекта |
| Tags | Array | Метки |
| MyCrew | Array | Работники и выполняемые ими обязанности для материала |
| Path | String | Путь к материалу в файловой системе |
| MyType | Int32 | Тип материала |

Таблица {N}. Структура документов коллекции «Persons»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| \_id | ObjectId | Идентификатор сотрудника |
| FirstName | String | Имя сотрудника |
| LastName | String | Фамилия сотрудника |
| CreationTime | Date | Дата создания объекта |
| Post | String | Должность |

Таблица {N}. Структура документов коллекции «Projects»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| \_id | ObjectId | Идентификатор проекта |
| Title | String | Название проекта |
| Description | String | Описание проекта |
| CreationTime | Date | Дата создания объекта |
| Tags | Array | Метки |
| MyCrew | Array | Работники и выполняемые ими обязанности в проекте |
| Path | String | Путь к проекту в файловой системе |
| MyType | Int32 | Тип проекта |
| DeadLine | Date | Срок сдачи проекта |
| Status | Int32 | Текущий статус работы над проектом |
| Materials | Array | Массив идентификаторов материалов, использованных в проекте |

Таблица {N}. Структура документов коллекции «Tags»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| \_id | ObjectId | Идентификатор метки |
| Title | String | Название метки |
| Description | String | Описание метки |
| CreationTime | Date | Дата создания объекта |
| ParentTagId | ObjectId | Метка-родитель |
| MyColor | Object | Цвет метки |

# 3.1.2 Разработка проекта NfModels

Проект NfModels (News Factory Models) является общим для всех программ в комплексе "Фабрика новостей". Он содержит код, который обеспечивает основные функциональности, необходимые для работы с базой данных, реализации шаблона MVVM (Model-View-ViewModel), заранее заготовленных диалогов работы с операционной системой и содержит некоторую общую логику, используемую другими программами комплекса.

В рамках проекта NfModels реализована связь с базой данных, позволяющая программам получать доступ к данным, хранящимся в базе данных MongoDB. Код, отвечающий за взаимодействие с базой данных, обеспечивает чтение, запись и обновление данных, а также выполнение запросов и фильтрацию данных с использованием MongoDB .NET Driver.

Для реализации шаблона MVVM в программном комплексе используется модели представления (ViewModel) различных объектов, которые отвечают за представление данных и обработку логики взаимодействия с пользователем. Проект NfModels предоставляет как базовые классы и интерфейсы для создания моделей представления, так и классы объектов фабрики, обеспечивая единообразную структуру и методы для связи данных с пользовательским интерфейсом.

Также проект NfModels содержит заранее заготовленные диалоги работы с файловой системой (FileDialogs), которые используются программами для предоставления пользователю возможности выбора места сохранения, загрузки файлов или экспорта данных.

Одна из важных особенностей проекта NfModels заключается в том, что он содержит часть общей логики, которая используется различными программами комплекса. Это позволяет избежать дублирования кода и обеспечивает единообразие функциональности между программами.

В целом, проект NfModels играет неотъемлемую роль в комплексе "Фабрика новостей", обеспечивая общую функциональность, связь с базой данных, модели представления для шаблона MVVM и другие необходимые компоненты, способствуя более эффективной разработке и поддержке программного комплекса.

Рассмотрим основу файловую структуры данного проекта.

Таблица {N}. Основа файловой структуры проекта NfModels

|  |  |
| --- | --- |
| Файл / Директория | Описание |
| /Infrastructure | Дополнительные файлы инфраструктуры решения |
| /Infrastructure/Commands | Директория, которая содержит классы команд, используемые для обработки пользовательских действий, таких как нажатие кнопки или выбор элемента из списка |
| /Infrastructure/Converters | директории размещаются классы конвертеров, которые позволяют преобразовывать данные из одного формата в другой |
| /Services | Директория, где располагаются классы, предоставляющие различные службы и функциональности, которые могут быть использованы в моделях представления |
| /Services/DBProvider.cs | Класс, провайдера статических методов работы с базой данных |
| /Services/FileDialogs.cs | Класс, провайдера статических методов работы с файловой системы |
| /ViewModels | Директория, где размещаются классы моделей представления |
| /ViewModels/NewsFactoryObjects | Директория, моделей представления объектов фабрики |
| /Views | Директория, содержащая XAML-файлы, которые определяют внешний вид пользовательского интерфейса |
| /NfResources.xaml | Файл разметки, в которой содержатся ресурсы, наследуемые в другие проекты решения |

# 3.1.3 Разработка КС

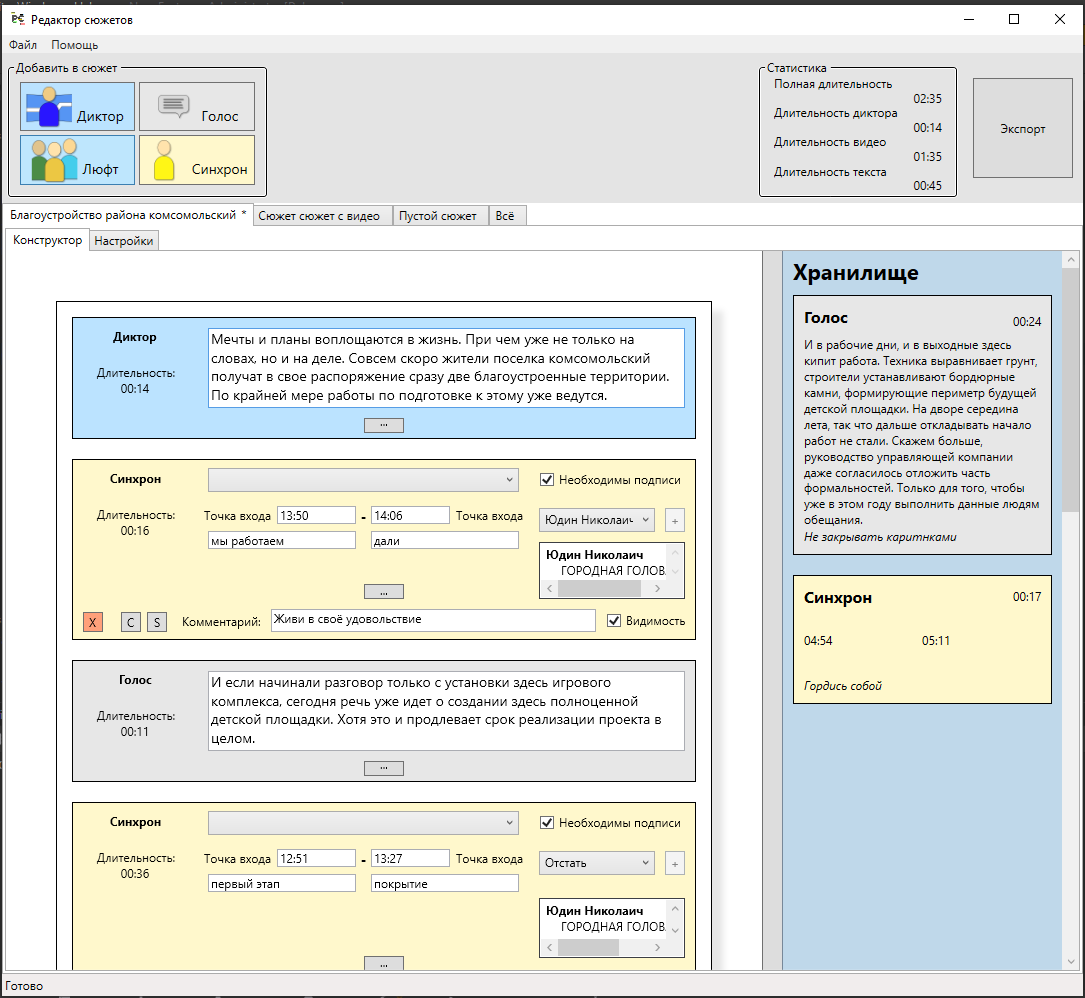
Разработка композитора сюжетов, как части программного комплекса "Фабрика новостей", была одним из ключевых этапов проекта. Целью разработки было создание мощного инструмента для создания и редактирования сюжетов новостных программ. Композитор сюжетов должен предоставлять удобный интерфейс для визуального составления сюжетов из блоков информации, а также обеспечивать возможность редактирования и сохранения сюжетов.

Для реализации шаблона проектирования MVVM были созданы соответствующие классы модели представления (ViewModel) и представления (View). Модели представления отображены в следующей таблице.

Таблица {N}. Модели представления композитора сюжетов

|  |  |
| --- | --- |
| Модель представления | Описание |
| StartupViewModel | Модель представления приветственного диалога |
| MainWindowViewModel | Модель представления главного окна |
| SettingsViewModel | Модель представления окна настроек |
| DictorViewModel | Модель представления части новостного сюжета «Диктор» |
| LuftViewModel | Модель представления части новостного сюжета «Люфт» |
| SynchroneViewModel | Модель представления части новостного сюжета «Синхрон» |
| VoiceViewModel | Модель представления части новостного сюжета «Закадровый голос» |
| ScriptViewModel | Модель представления новостного сюжета |
| MaterialViewModel | Модель представления материала сюжета |
| MaterialPlayerViewModel | Модель представления проигрывателя материалов сюжета |

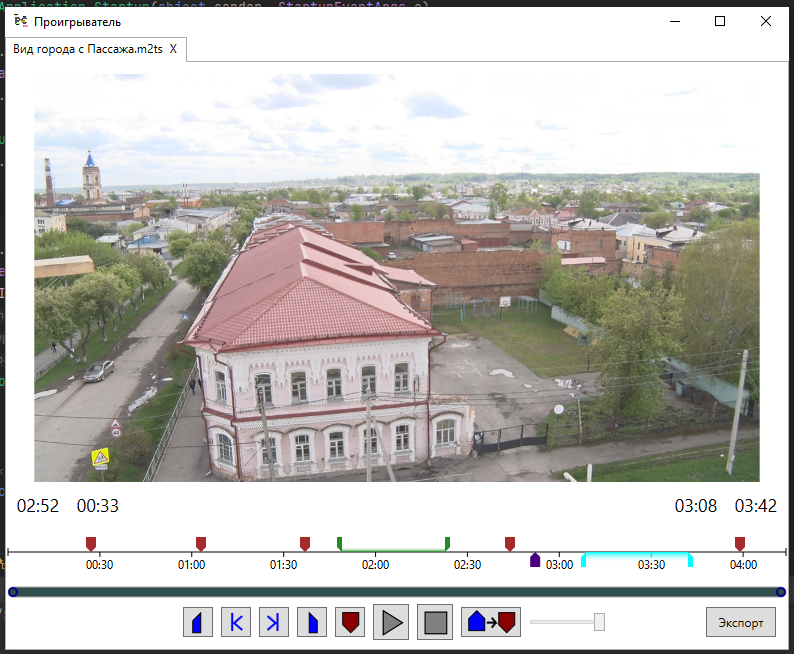
В композиторе сюжетов реализована возможность создания и редактирования сюжетов с использованием блочного редактора. Он позволяет пользователям визуально компоновать и структурировать информацию, представленную в виде блоков. Каждый блок содержит определенную часть сюжета, такую как дикторская программа, закадровый текст, люфт и снихрон.



Пользователи могут свободно перемещать блоки, а также изменять их содержимое. Композитор сюжетов предоставляет различные инструменты и функции для работы с блоками. Так части сюжета, которым пока нет места в сюжете можно отложить в хранилище (Stash), а при необходимости вернуть обратно.

Также в рамках композитора сюжетов был реализован проигрыватель материалов, который предоставляет возможность вставлять видеофрагменты прямо в сюжеты. Этот проигрыватель значительно облегчает работу редакторов сюжетов, позволяя им непосредственно просматривать и выбирать нужные фрагменты видеоматериалов для включения в сюжеты.

Одной из важных функций проигрывателя является возможность ставить маркеры на таймлайне. Маркеры позволяют редакторам отметить интересующие моменты видеофрагментов, которые они хотят использовать в сюжете. Редакторы могут легко устанавливать маркеры на нужных временных позициях в проигрывателе, чтобы запомнить эти моменты и быстро вернуться к ним в дальнейшем.



Интересно то, что уже использованные видеофрагменты отображаются в проигрывателе как маркеры. Это позволяет редакторам легко отслеживать и визуально представлять, какие фрагменты уже были использованы в составлении сюжета. Благодаря этому, редакторы могут легко контролировать и организовывать свою работу, избегая повторного использования одних и тех же фрагментов.

Кроме того, композитор сюжетов обеспечивает возможность сохранения и загрузки созданных сюжетов в собственный файл с расширением «.nscr» (скор. от news script – новостной сюжет), а также экспорта на печать их в формате HTML. Это позволяет пользователям легко обмениваться и распространять свои сюжеты с коллегами и другими заинтересованными сторонами.

Важным аспектом разработки композитора сюжетов была также обеспечение удобного пользовательского интерфейса с интуитивно понятными элементами управления. Были разработаны различные функциональные панели, инструменты и контекстные меню, чтобы пользователи могли легко осуществлять необходимые действия и настройки при работе с композитором сюжетов.

В итоге, разработка композитора сюжетов позволила создать мощный инструмент, способствующий эффективной и креативной работе редакторов сюжетов телевизионных программ. Благодаря его функциональности и удобному интерфейсу, редакторы могут быстро и легко создавать сюжеты, а также вносить необходимые изменения в уже существующие сюжеты, улучшая качество и привлекательность телевизионных программ.

# 3.1.4 Разработка ТШ

В начале процесса разработки были проведены анализ и определение требований к функциональности ТШ. Определены основные задачи, которые должна выполнять программа, такие как контроль над телевизионными проектами, управление съемками и материалами, планирование и отслеживание выполнения задач.

Затем был разработан дизайн и архитектура ТШ, определены основные компоненты и модули, необходимые для реализации требуемой функциональности. В рамках архитектуры были определены различные классы, интерфейсы и связи между ними.

Для реализации функциональности ТШ использовались различные технологии и инструменты. Был использован язык программирования, такой как C#, и фреймворк для разработки пользовательского интерфейса WPF (Windows Presentation Foundation).

В процессе разработки ТШ были реализованы основные функциональные возможности, включая создание и редактирование проектов, управление съемками, контроль за материалами, планирование задач и другие операции, необходимые для эффективной работы телевизионного штаба.

Также в процессе разработки ТШ были проведены тестирование и отладка программы для обеспечения ее стабильной работы и соответствия требованиям.

В итоге разработки Телевизионного штаба получилась программа, которая позволяет управлять телевизионными проектами, контролировать съемки и материалы, планировать задачи и обеспечивать эффективную работу штаба.

4. Руководства оператора и программиста

4.1. Руководство оператора

4.2. Руководство программиста