**РОСЖЕЛДОР**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

|  |  |  |  |
| --- | --- | --- | --- |
| **К защите:** |  |  | |
| **Заведующий кафедрой** | **Информационные** | |
| **технологии транспорта** | | |
|  | д-р техн. наук, проф. | |
|  |  | В. И. Хабаров | |
| *подпись* |  | *инициалы, фамилия* | |
|  |  |  | |
| *дата* |  |  | |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Тема:** | Разработка информационной системы для отдела трудоустройства | | | | | |
|  | управления занятости населения | | | | | |
|  | |  | БР.БИСТ.26.2022 |  |  |
|  | |  | *шифр документа* |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Выполнил** |  |  |  | **Руководитель** |
|  |  | К. Э. Чамзы |  |  |  | канд. техн. наук, доц.  С. П. Сарычев |
| *подпись* |  | *инициалы, фамилия* |  | *подпись* |  | *инициалы, фамилия* |
|  |  |  |  |  |  |  |
| *дата* |  |  |  | *дата* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Нормоконтролер работы |  |  |  | ст. преп.  Т. А. Распопина |
|  |  | *подпись* |  |  |
|  |  |  |  |  |
|  |  | *дата* |  |  |

**2022 г.**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

Факультет: Бизнес-информатики

Кафедра: Информационные технологии транспорта

Направление: 09.03.02 «Информационные системы и технологии»

Профиль: Интеллектуальные транспортные системы

|  |  |
| --- | --- |
|  | ***УТВЕРЖДАЮ****: зав. кафедрой «Информационные технологии транспорта»*  д-р техн. наук, проф.  В. И. Хабаров |
|  | *«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2022 г.* |

**З А Д А Н И Е**

**на выполнение выпускной квалификационной работы**

|  |  |  |  |
| --- | --- | --- | --- |
| студенту | Чамзы Кыргыс Эресовичу | | |
|  |  | | |
| 1. Тема «Разработка информационной системы для отдела трудоустройства управления занятости населения» утверждена приказом № 203/с от «30» мая 2022 г. | | | |
| 2. Задание выдано «12» мая 2022 г. | | | |
| 3. Срок сдачи законченной работы на кафедру «17» июня 2022 г. | | | |
| 4. Исходные данные: данные, полученные в ходе прохождения преддипломной практики | | | |
| 5. Содержание расчетно-пояснительной записки | | | |
| Наименование разделов и вопросов | | Примерное количество страниц | График (сроки) выполнения |
| Введение | | 2 | 14.05.2022 |
| Аналитический обзор предметной области | | 14 | 14.05.2022 |
| Разработка проекта автоматизации формирования отчет | | 17 | 16.05.2022 |
| Разработка приложения | | 17 | 28.05.2022 |
| Заключение | | 1 | 28.05.2022 |

6. Содержание и объемы графической части

|  |  |  |
| --- | --- | --- |
| Наименование графического документа (чертежа, схемы, графика) | Количество  листов  формата А1 | График  (сроки)  выполнения |
| Презентация PowerPoint | 19 | 01.06.2022 |
| Рисунки | 56 | 28.05.2022 |
| Таблицы | 1 | 22.05.2022 |

|  |  |  |
| --- | --- | --- |
| Руководитель |  | С. П. Сарычев |
|  | *(подпись, фамилия, И.О.)* |  |
| Задание к использованию принял |  | К. Э. Чамзы |
|  | *(подпись студента)* |  |

УДК 004.432.2

**АННОТАЦИЯ**

В работе 58 страниц, 56 рисунков, 1 таблица, 8 источников, 2 приложения.

Ключевые слова: *генератор отчетов, автоматизация отчетности, Excel, WPF, C#*.

В данной работе проводится теоретический анализ предметной области, в частности темы автоматизации отчетности и генераторов отчетов, а также разработка программы работы с файлами, с дружелюбным пользовательским интерфейсом.

Результатом данной работы является информационная система по автоматизации составления сводного отчета в отделе трудоустройства центра занятости населения о предоставлении государственной услуги сопровождения.

**ABSTRACT**

The work contains 58 pages, 56 figures, 1 table, 8 sources, 2 applications.

Keywords: report generator, reporting automation, Excel, WPF, C#.

This paper provides a theoretical analysis of the subject area, in particular the topics of automation of reporting and report generators, as well as the development of a program for working with files, with a friendly user interface.

The result of this work is an information system for automating the compilation of a summary report in the employment department of the employment center on the provision of state support services.

**ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

Фреймворк – программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

«C#» – объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании «Microsoft» под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы «Microsoft .NET Framework» и «.NET Core».

ЦЗН – центр занятости населения.

ГКУ – государственное казённое учреждение.

ПО – программное обеспечение.

Сводный отчет – документ, составленный из отчетов, присланных ЦЗН районов и городов.

Пользователь – должностное лицо, участвующее в функционировании автоматизированной системы или использующее результаты её функционирования, наделенное определенным объемом прав.

Система / Приложение / Программа – информационная система способная решить определенную проблему.

Библиотека / Пакет – сборник подпрограмм или объектов, используемых для разработки программного обеспечения.

**СОДЕРЖАНИЕ**

[Введение 7](#_Toc106299949)

[1 Аналитический обзор предметной области 9](#_Toc106299950)

[1.1 Отчетность в отделе трудоустройства управления занятости населения 9](#_Toc106299951)

[1.2 Cводная отчетность 9](#_Toc106299952)

[1.3 Автоматизация отчетности 10](#_Toc106299953)

[1.4 Генераторы отчетов 11](#_Toc106299954)

[1.5 Средства разработки приложений, формирующих отчеты из формата «Excel» 14](#_Toc106299955)

[1.6 Обзор библиотеки «EPPlus» 15](#_Toc106299956)

[1.7 Обзор модуля для «Python» «Xlsxwriter» 16](#_Toc106299957)

[1.8 Обзор библиотеки «NPOI» 17](#_Toc106299958)

[1.9 Обзор языка программирования «VBA» 18](#_Toc106299959)

[1.10 Изучение существующего бизнес-процесса, подлежащего автоматизации 19](#_Toc106299960)

[1.11 Описание бизнес-процесса с использованием средств автоматизации 21](#_Toc106299961)

[1.12 Вывод по первому разделу 22](#_Toc106299962)

[2 Разработка проекта по автоматизации формирования отчета 23](#_Toc106299963)

[2.1 Постановка задачи 23](#_Toc106299964)

[2.2 Моделирование бизнес-процессов разрабатываемой системы 24](#_Toc106299965)

[2.3 Выбор и обоснование средств разработки 27](#_Toc106299966)

[2.4 Чтение и запись данных 34](#_Toc106299967)

[2.5 Вывод по второму разделу 38](#_Toc106299968)

[3 Разработка приложения 40](#_Toc106299969)

[3.1 Разработка архитектуры приложения 40](#_Toc106299970)

[3.2 Разработка функционала приложения 42](#_Toc106299971)

[3.3 Разработка графического интерфейса 43](#_Toc106299972)

[3.4 Руководство пользователя 45](#_Toc106299973)

[3.5 Вывод по третьему разделу 55](#_Toc106299974)

[Заключение 57](#_Toc106299975)

[Список использованных источников 58](#_Toc106299976)

[Приложение А Код функциональной части программы 59](#_Toc106299977)

[Приложение Б Код графического интерфейса программы 60](#_Toc106299978)

ВВЕДЕНИЕ

По мере развития информационных технологий, выходные документы средств накопления информации постоянно усложняются и увеличиваются в объеме. Кроме того, нужна автоматизация самого ресурсоемкого и трудоемкого этап деятельности подавляющего большинства организаций – работы с печатными документами. Изменяется и само представление о печатном документе, он может содержать в себе сложные вычисления, изображения, графики, диаграммы и так далее.

В связи с усложнением форм и количества печатных документов особо остро встает задача создания электронных комплексов, позволяющих автоматизировать процесс «производства» печатных документов, используя информацию, представленную в самых разных видах. Такие электронные комплексы называются генераторами отчетов. Такие системы как правило являются неотъемлемой частью программных комплексов, созданных для автоматизации бумажного документооборота компаний - программ электронного документооборота.

Подготовка отчётов может быть сложной задачей. Неудивительно, что их создание занимает много времени, сил и делает сотрудников в конце недели ещё более уставшими. Автоматизация составления и распространения отчетов позволяет экономить время, благодаря тому, что процессы, не требующие непосредственного внимания человека, будут выполнены программными алгоритмами.

Цель выпускной квалификационной работы заключается в автоматизации процесса составления отчета о предоставлении государственной услуги сопровождения при содействии занятости инвалидов ГКУ ЦЗН.

Для достижения поставленной цели необходимо решить следующие задачи:

* проведение аналитического обзора предметной области;
* осуществление анализа генераторов отчетов, которые уже имеются на рынке в текущий момент;
* формирование требований к разрабатываемому программному продукту;
* моделирование бизнес-процессов;
* обоснование выбора среды и средств разработки;
* разработка информационной системы;
* составление руководства пользователя;
* анализ результатов работы.

Предмет исследования: приложение составляющее сводный отчет в отделе трудоустройства управления занятости.

Объект исследования: генерация отчетов формата «Excel» программными средствами.

1. Аналитический обзор предметной области
   1. Отчетность в отделе трудоустройства управления занятости населения

В каждом субъекте Российской Федерации имеется главный отдел ЦЗН уровня субъекта и подконтрольные ему ЦЗН районов и городов. Из дочерних отделов каждый месяц отправляются отчеты о результатах работы за месяц, в главном же отделе результаты по каждому ЦЗН района или города собираются в отдельные сводные отчеты, для дальнейшего анализа, визуализации, передачи и хранения данных.

В ЦЗН НСО каждый месяц формируется документ с информацией о предоставлении государственной услуги сопровождения при содействии занятости инвалидов ГКУ ЦЗН НСО. Формирование вышеупомянутого документа является трудоемкой задачей, так как заполнение всех данных происходит вручную.

* 1. Cводная отчетность

Сводная отчетность – это комплекс отчетной документации, отражающей деятельность нескольких организаций, связанных между собой. Ее составление характерно для крупных компаний, имеющих дочерние структуры [1].

Как видно из определения, главный отчет формируется в центральном отделении организации и для составления используются уже созданные отчеты в дочерних структурах.

Как показано на рисунке 1.1, дочерние структуры составляют отчет по какой-либо деятельности, используя свои внутренние данные, которые у каждого свой.

Сводный отчет по сути тоже является обычным отчетом, только данными в данном случае являются отчеты дочерних структур.

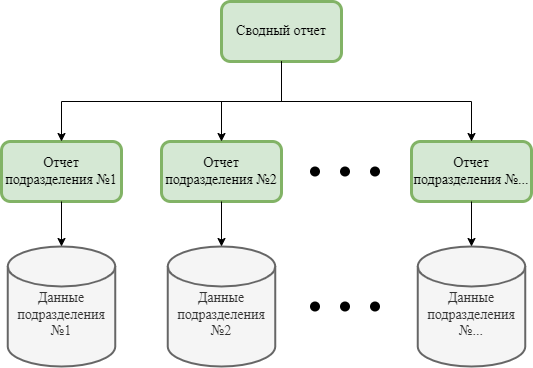


Рисунок 1.1 – Составные части сводного отчета

* 1. Автоматизация отчетности

Способы автоматизации отчетности зависят от того в каком формате хранятся исходные данные и в каком виде следует получить итоговый файл.

В исходных данных могут быть:

* таблицы;
* графики;
* рисунки;
* числовые показатели;
* текстовые поля;
* и любые другие возможные способы представления информации.

К форматам хранения и передачи данных можно отнести:

* табличные файлы. например, «Excel»;
* текстовые файлы, например, «Word»;
* файлы с расширением «.pdf»;
* базы данных;
* интернет-источники;
* сигналы информационной системы и другие.

Для автоматического формирования отчетов в организации с уже имеющейся системой, в большинстве случаев используют специальные программы, которые называются генераторами отчетов.

Генератор отчётов — программа или библиотека, позволяющая представить информацию в удобочитаемом структурированном виде [2].

Типичные представители программ-генераторов отчётов:

* JasperReports;
* Crystal Reports;
* FastReport;
* GenRep;
* CuteReport;
* Exwog;
* LimeReport.

Несмотря на обилие программ по составлению отчетов, большинство из них в основном разработаны для использования в процессе разработки.

* 1. Генераторы отчетов

Наиболее популярным представителем средства автоматизации отчетности в системах, имеющих базы данных является библиотека «JasperReports».

«JasperReports» — это «Java-библиотека» от группы «Jaspersoft Community», которая позволяет на основе «jrxml-шаблонов» и набора данных из различных источников, включая «JDBC», создавать отчёты в различных форматах. Отчёты могут быть представлены как на экране, так и выведены на принтер или в файл.

Возможности библиотеки «JasperReports»:

* поддержка различных источников данных: «JDBC», «XML», «CSV», «JavaBeans», «Hibernate», интерфейс «JRDataSource»;
* экспорт в различные форматы данных: «PDF», «HTML», «XHTML», «RTF», «ODT», «CSV», «XML», «XLS», «XLSX», «DOCX»;
* использование скриплетов, которые могут быть вызваны до или после определённых этапов генерации отчётов;
* использование динамических языков «JavaScript» и «Groovy» при формировании отчёта;
* формирование диаграмм с использованием библиотеки «JFreeChart»;
* формирование подотчётов с неограниченной глубиной вложенности;
* реализация кросстаблиц [3].

«JasperReports» используется специалистами IT-сферы для нужд разработки.

Другим примером генератора отчета является отдельный программный модуль «GenRep», который с недавнего начал поддерживать старый формат «XLS».

Возможности «GenRep»:

* поддержка различных источников данных: «DBF», «TXT», «ADO»;
* экспорт данных в форматы: «MS Excel», «MS Word» и «HTML»;
* генерация отчетов без установки компонентов «Microsoft Office».

«GenRep» позволяет генерировать отчеты из различных форматов, не являясь компонентом какой-либо среды разработки.

Примером генератора отчета для простых офисных задач является приложение «Exwog», интерфейс данного программного продукта отображен на рисунке 1.2. Генератор отчетов «Exwog» позволяет хранить данные в форматах «XLSX» и «XLS», используя эти данные приложение по шаблону текстового файла генерирует текстовый отчет.

В табличных файлах хранятся данные по столбцам, программа, ориентируясь по этим столбцам высчитывает количество строк, например, для пяти строк программа сгенерирует пять отдельных отчетов, каждый из которых заполняется по ранее заданному шаблону. В шаблоне текстового файла фигурными скобками помечаются места вставки данных по столбцам.

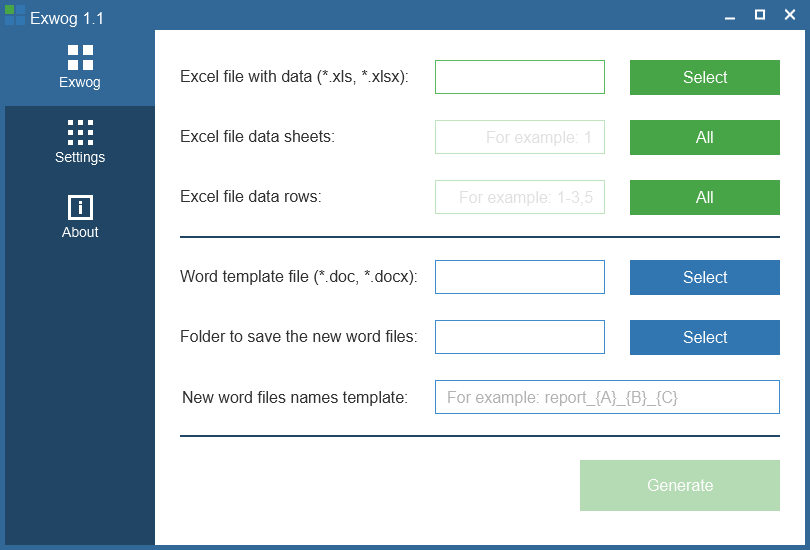


Рисунок 1.2 – Приложение «Exwog»

По представленным примерам видно, что существует большое количество разновидностей генераторов отчетов.

Применение того или иного решения зависит от поставленных задач и ожидаемого результата. Некоторые из них могут применятся как компонент при разработке программного обеспечение, а некоторые поставляются в качестве отдельного продукта.

Каждый генератор отчетов специфичен и может использоваться только в узкой направленности.

Для автоматизации отчетности можно воспользоваться уже существующими решениями, если они удовлетворяют специфике предметной области, в ином же случае необходима реализация нового генератора отчетов отвечающей специфике, заданной требованиями.

* 1. Средства разработки приложений, формирующих отчеты из формата «Excel»

В большинстве организаций, не пользующихся автоматизированными системами, отчетность осуществляется с использованием обычных табличных файлов, таких как «Excel».

«Microsoft Excel» является самой популярной программой для работы с электронными таблицами, созданная корпорацией «Microsoft» для операционных систем «Microsoft Windows», «Windows NT», «Mac OS», «Android», «iOS» и «Windows Phone».

Самая первая версия программы «Microsoft Excel» была выпущена в 1985 году и предназначалась для операционной системы «Mac». На тот период времени «Excel» была инновационной, по сравнению с аналогичными продуктами, программа позволяла менять параметры отображения табличных данных.

До 2003 версии «Microsoft Excel» использовал свой собственный бинарный формат данных «BIFF». Это означало, что для работы со всеми файлами с расширением «.xls», обязательно требовалась установка программы «Microsoft Excel». Эта программа предоставляла «API» для работы с табличным файлом.

Начиная с 2007 версии начинается использование нового формата «Microsoft Office Open XML». Файлы с таким форматом имеют расширение «.xlsx». Этот формат подразумевает представление табличного файла в виде отдельных файлов с расширением «.xml», которые можно редактировать и без самой программы «Excel». Такой формат позволяет работать с табличными файлами без предустановки необходимых компонентов, главное ознакомиться с документацией по работе с «Microsoft Office Open XML».

Языки программирования как средства для автоматизации отчетности являются более гибкими решениями, позволяя реализовать весь процесс формирования отчета на свое усмотрение. Главным недостатком является сложность разработки и поддержка итогового программного продукта. В большинстве случаев все средства разработки программ, работающих с табличными файлами «Excel», поддерживают только новый формат «Microsoft Office Open XML».

Сегодня на рынке программного обеспечения для работы с файлами формата «Excel» имеются самые различные решения. Каждое решение имеет свои преимущества и недостатки.

Наиболее популярными компонентами разработки для работы с табличными файлами по критерию стабильности являются:

* библиотека «EPPlus;
* модуль «Xlsxwriter»;
* библиотека «NPOI»;
* язык программирования «VBA».
  1. Обзор библиотеки «EPPlus»

В современном мире разработки приложений нередко встает необходимость работы с «Excel» документами. Чаще всего это разного рода отчеты, но иногда «xls/x» файлы используются в качестве хранилища данных. Например, если пользователь должен иметь возможность загрузить данные в приложение или выгрузить, в человеко-читаемом виде, «Excel» де-факто является стандартом. Относительно дружелюбный интерфейс, прозрачная структура, в купе с его распространенностью... трудно навскидку назвать решение лучше [4].

«EPPlus» отличается производительностью, из-за хорошей оптимизации работа с данными в табличном файле происходит быстрее, по сравнению с другими библиотеками.

На «github», и не только, можно найти ряд библиотек, бесплатных и не только. Пожалуй, самой популярной является «EPPlus» [4]. Исходный код библиотеки «EPPlus» можно использовать бесплатно только до четвертой версии библиотеки. Начиная с пятой версии и выше данное решение перешло на платную лицензию.

На официальном сайте проекта приводится следующее определение:

«EPPlus — это библиотека .NET Framework/.NET Core для управления электронными таблицами Office Open XML, распространяемая через Nuget. Версия 5 поддерживает .NET Framework с версии 3.5 и .NET Core с версии 2.0. EPPlus не имеет зависимостей от любой другой библиотеки, такой как Microsoft Excel» [5]. По определению видно, что «EPPlus» является самостоятельной библиотекой, которая работает напрямую с расширением «XML».

«Epplus» предлагает возможности работы со следующими компонентами:

* стилизация ячеек;
* форматирование данных;
* стилизация текста;
* настройка границ;
* вывод графика;
* настройка выравнивания.

Основным недостатком данной библиотеки является её неспособность работать с устаревшим форматом «XLS». Этот недостаток исходит из того что «EPPlus» не работает с «API» приложения «Excel», а напрямую взаимодействует с внутренним форматом «XML».

* 1. Обзор модуля для «Python» «Xlsxwriter»

Для тех, кто разрабатывает программы для работы с «Excel» файлами на языке программирования «Python» есть замечательный модуль «Xlsxwriter». Модуль имеет открытый исходный код, поэтому является бесплатным для пользования.

Его основными преимуществами являются:

* удобная работа с формулами, как видно из рисунка 1.3, формулы составляются точно так же, как и в «Excel»;
* упрощенная работа с форматами ячеек, на рисунке 1.4;
* постройка сложных графиков, на рисунке 1.5.

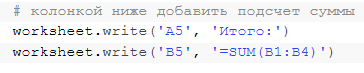


Рисунок 1.3 – Использование формул в модуле «Xlsxwriter»



Рисунок 1.4 – Изменение формата текста передачей дополнительного параметра в метод

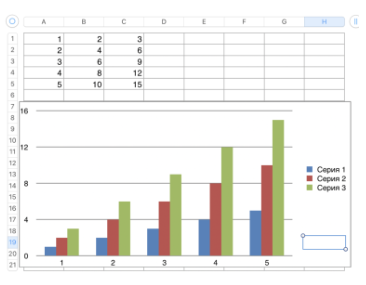


Рисунок 1.5 – Построенный график с помощью модуля «Xlsxwriter»

* 1. Обзор библиотеки «NPOI»

Для работы с файлами «Microsoft Office» существует хорошая библиотека для языка программирования «Java» под названием «Apache POI», была создана версия этой библиотеки на языке программирования «C#». Её назвали «NPOI», функционал полностью дублирует оригинальную библиотеку, разработанную для «Java».

Основным преимущество данной библиотеки является стабильность и производительность по сравнении с «EPPlus». На рисунке 1.6 приведено сравнение двух библиотек. Как видно из рисунка 1.6, библиотека «NPOI» выигрывает по поддержке кода, что отражается в общей стабильности работы её модулей [6].

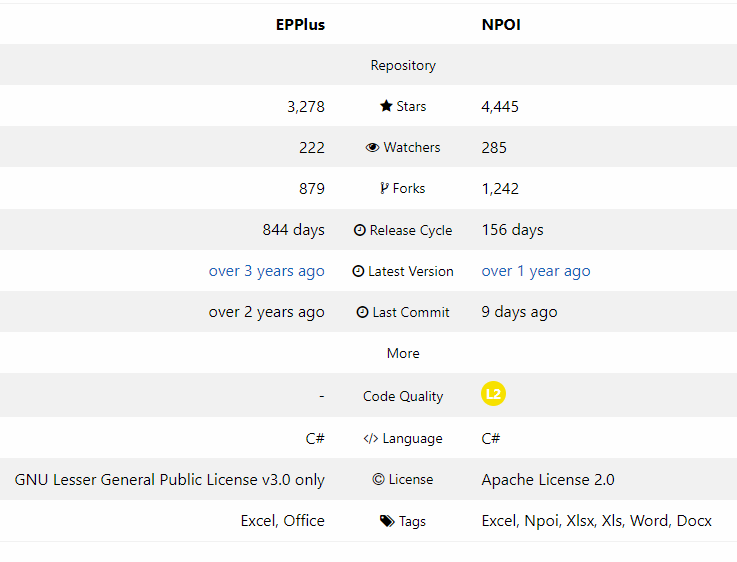


Рисунок 1.6 – Сравнение библиотек «NPOI» и «EPPlus»

* 1. Обзор языка программирования «VBA»

Язык «VBA» освоить не трудно, это достаточно легкий язык программирования, и с его помощью можно быстро получить ощутимые результаты — профессионально сконструированные приложения, решающие практически все задачи в операционной среде «Windows». «Visual Basic for Applications» относится к числу объектно-ориентированных языков, то есть при разработке проектов на «VBA» все данные и действия над ними представляются в виде объектов.

«VBA» основан на технологии визуального программирования, предлагающей конструирование рабочей поверхности приложения и элементов его управления непосредственно на экране, а также запись всей программы или ее частей при помощи макрорекордера.

С помощью макросов «VBA» можно автоматизировать ежедневную рутинную работу в программе «Microsoft Excel» — ведь действия, доступные пользователю на ее рабочем листе, составляют 10% от всех возможностей приложения.

Кроме того, VBA объединяет (интегрирует) приложения «Microsoft Office», позволяет, не выходя из «Excel», управлять работой других приложений и внедрять в «Excel» объекты из других приложений пакета [7].

Основными преимуществами «VBA» являются:

* легкость в освоении;
* выполнение скрипта в среде офисных приложений;
* среда выполнения может быть встроена в приложения;
* высокая распространенность среди офисных приложений.

К недостаткам можно отнести:

* излишняя открытость кода;
* проблемы с обратной совместимостью.
  1. Изучение существующего бизнес-процесса, подлежащего автоматизации

На данный момент формирование сводного отчета происходит согласно диаграмме изображенном на рисунке 1.7. Пользователю каждый раз приходится открывать файл-отчет, считывать и идентифицировать данные записанные в разных ячейках. И только после успешной расшифровки данные заносятся в сводный отчет. Количество итераций равно количеству файлов-отчетов.

При таком подходе появляются ряд проблем, связанных с человеческим фактором:

* проблема пропуска данных;
* проблема возникновения ошибок при чтении данных;
* проблема ввода ошибочных данных;
* проблема дублирования информации.

И более того каждая итерация открытия и чтения файла-отчета, а в дальнейшем запись данных в итоговый файл, отнимает много времени у пользователя.

Однотипные повторяющиеся действия отнимают не только время, но и способствуют быстрой усталости и потере концентрации и внимания на рабочем месте.

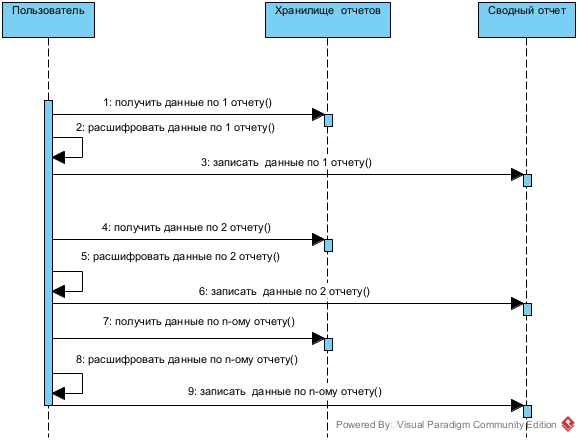


Рисунок 1.7 – Диаграмма последовательности процесса формирования сводного отчета сотрудником

* 1. Описание бизнес-процесса с использованием средств автоматизации

Для уменьшения возможного количества ошибок, появляющихся при составлении сводного отчета, предлагается диаграмма последовательности, представленная на рисунке 1.8.

При таком подходе, пользователю нужно будет всего лишь отправить сигнал программному модулю о составлении сводного отчета. После остается только проверка правильности составления сводного отчета программой.

Вся работа по поиску, идентификации, чтении и записи информации в многочисленных файлах отводится программному модулю.

Пользователь все равно должен будет поработать с данными, но уже в этом случае ему надо только проконтролировать правильность составления сводного отчета, вместо самостоятельно ручного ввода. При необходимости у пользователя есть возможность в любой момент подправить данные по тому или иному файлу-отчету.

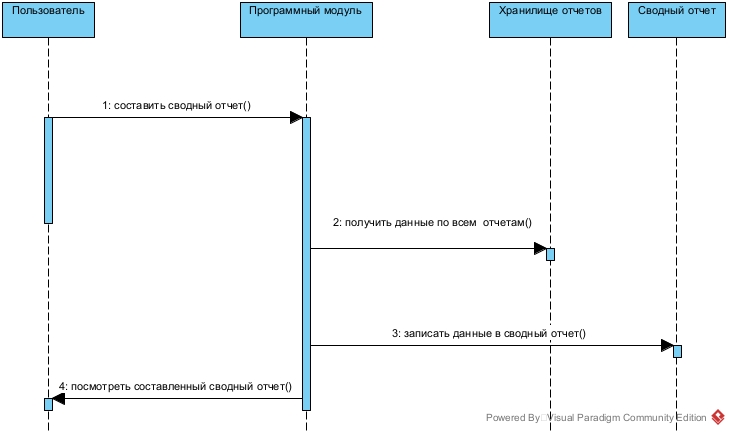


Рисунок 1.8 – Диаграмма последовательности процесса формирования сводного отчета программным модулем

* 1. Вывод по первому разделу

В первой части выпускной квалификационный работы был сделан аналитический обзор предметной области, изучены основные понятия.

Были приведены примеры генераторов отчетов и их области применения. Выяснено что каждое решение по автоматизации генерации отчетов является специфичным и может быть не применимо в другой области генерации отчетов.

Были показаны примеры средств разработки, при помощи которых можно создать генератор отчетов только для работы с табличными файлами. Выяснено что каждое средство имеет свои преимущества и недостатки. Общим для всех недостатком является отсутствие поддержки старого формата табличного файла «XLS».

Это обусловлено с тем что большинство решений работают напрямую с внутренним форматом «XML», игнорируя внешний «XLSX, формат «XLS» не имеет внутреннего формата, поэтому реализация поддержки этого формата крайне затруднительна.

Для более детального понимания бизнес-процесса, подлежащего автоматизации, было осуществлено его моделирование в нотации UML. А именно был описан существующий бизнес-процесс предметной области, для сравнения с предлагаемым бизнес-процессом после внедрения автоматизации.

1. Разработка проекта по автоматизации формирования отчета
   1. Постановка задачи

В процессе работы с документацией постоянно возникают задачи автоматизации того или иного процесса составления отчетов. Приложения, помогающие при работе с документами, позволяют существенно оптимизировать бизнес – процессы в организации. Составление сводного отчета в отделе трудоустройства управления занятости населения представляет собой монотонный процесс считывания и копирования данных из одного документа в итоговый.

Необходимо автоматизировать процесс формирования сводного отчета о предоставлении государственной услуги сопровождения в ГКУ ЦЗН субъекта.

Тип отчетности ежемесячная. Каждый месяц в ЦЗН субъекта с каждого районного или городского ЦЗН отправляются отчеты в виде файлов формата «XLSX». Сводный отчет состоит из данных расположенных в определенных ячейках исходного табличного файла. Формат сводного отчета, такой же, как и у присылаемых, то есть табличный файл «Excel».

Для сводного отчета нужны данные не только с присылаемых отчетов, но и данные со сводного отчета за прошлый месяц, для сравнения изменений и контроля правильности заполнения документа. Среди генераторов отчета представленных на рынке, не имеются генераторов считывающих данные с файла формата «XLSX» и записывающий в такой же формат.

Необходимо создать новый генератор отчетов, так как для решения проблемы в данной предметной области, нужна детальная реализация всех этапов составления сводного отчета.

В предметной области специфичными являются идентификация данных в отчетах, а также преобразование текстовой информации в численную.

* 1. Моделирование бизнес-процессов разрабатываемой системы

Для полного понимания всех аспектов разрабатываемого программного продукта, необходимо смоделировать бизнес-процессы разрабатываемой системы.

В системе имеются три актора с вариантами использования, представленные на рисунке 2.1.

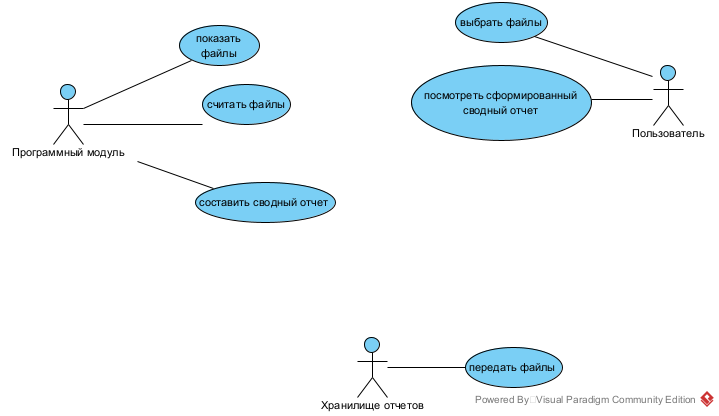


Рисунок 2.1 – Диаграмма вариантов использования предметной области

Для определения очередности взаимодействия сущностей при процедуре составления сводного отчета была создана диаграмма последовательности, изображенная на рисунке 2.2.

Программный модуль может быть в самых разных состояниях от простоя до интенсивной работы, по сути, программный модуль является основным работающим компонентом, поэтому для него была определена диаграмма состояний, которая изображена на рисунке 2.3. На этой диаграмме начальное состояние программы «ожидание», из этого режима вывод начало процедуры создания сводного отчета. В остальных трех состояниях программа непосредственно создает отчет.

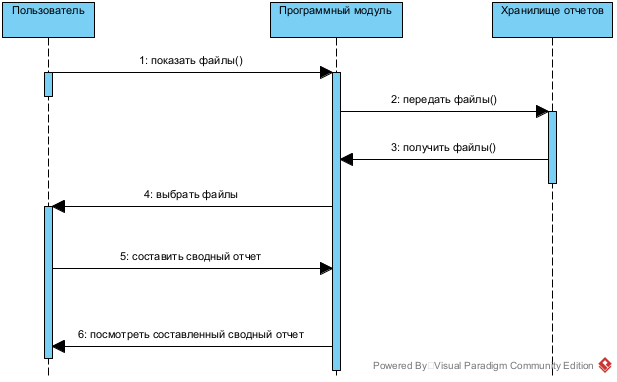


Рисунок 2.2 – Диаграмма последовательности процедуры создания отчета

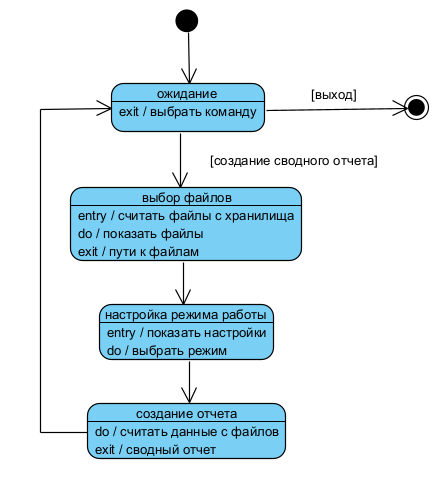


Рисунок 2.3 – Диаграмма состояний программного модуля

Диаграмма обзора взаимодействия формирования отчета представлена на рисунке 2.4. Диаграммы последовательности выбора файла, настройки режима и составления отчета отображены на рисунке 2.5.

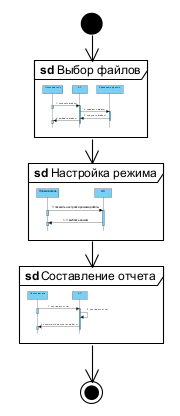


Рисунок 2.4 – Диаграмма обзора взаимодействия формирования отчета

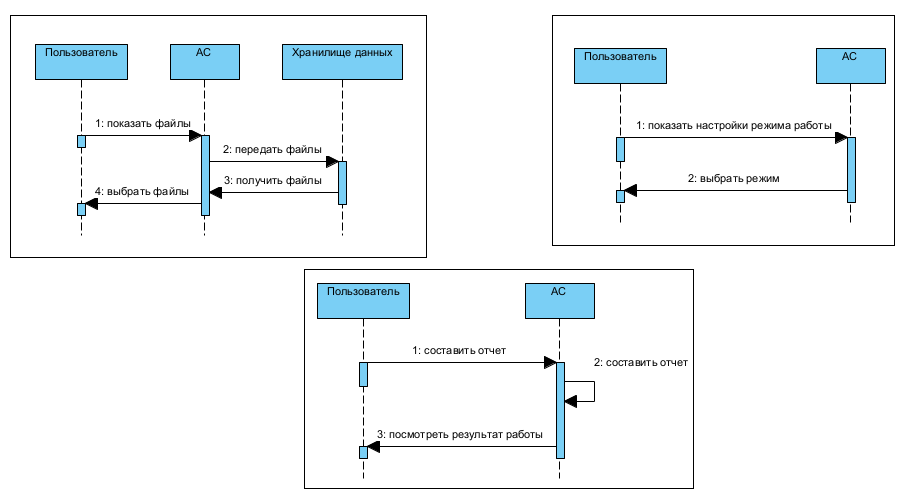


Рисунок 2.5 – Диаграммы последовательности всех элементов диаграммы обзора взаимодействия формирования отчета

* 1. Выбор и обоснование средств разработки

Как уже отмечалось в первом разделе данной работы, генератор отчетов работы с табличными файлами можно реализовать при помощи различных языков программирования. Для нашего случая основными требованиями являются: обеспечение модульности разработки, возможность разработки графического интерфейса, масштабируемость и гибкость программного кода, а также возможность работы с табличными файлами формата «XLSX».

По всем критериям подходят языки программирования «C#» и «Java». Но библиотеки работы с табличными файлами написанные для платформы «.NET» более производительны. Поэтому в качестве языка программирования был выбран «C#».

С учётом выбранного языка программирования и платформы, было принято решение разрабатывать приложение на интегрированной среде разработки «Microsoft Visual Studio».

«Visual Studio» – это лучшая комплексная среда «IDE» для разработчиков «.NET» в «Windows». Полноценный набор инструментов и функций для улучшения и усовершенствования каждого этапа разработки программного обеспечения.

Будет использоваться «Community» версия «Visual Studio», потому что она бесплатная.

Интерфейс данной «IDE» представлен на рисунке 2.6.

Рассмотрим другие среды разработки, которые также можно было бы использовать для разработки.

Самый хороший заменитель «Visual Studio», «SharpDevelop» – свободная среда разработки для «C#», «Visual Basic .NET», «Boo», «IronPython», «IronRuby», «F#», «C++».

Обычно используется как альтернатива «Visual Studio .NET». Это очень хорошая среда разработки для своего времени.

Здесь есть визуальный редактор интерфейса, автозавершение текста, подсветка синтаксиса, шаблоны формирования файлов программных модулей (последние два, сугубо по мнению разработчика, даже лучше, чем в VS).

Также возможности среды можно расширить дополнительным ПО. Однако эту «IDE» перестали обновлять с 2016 года.

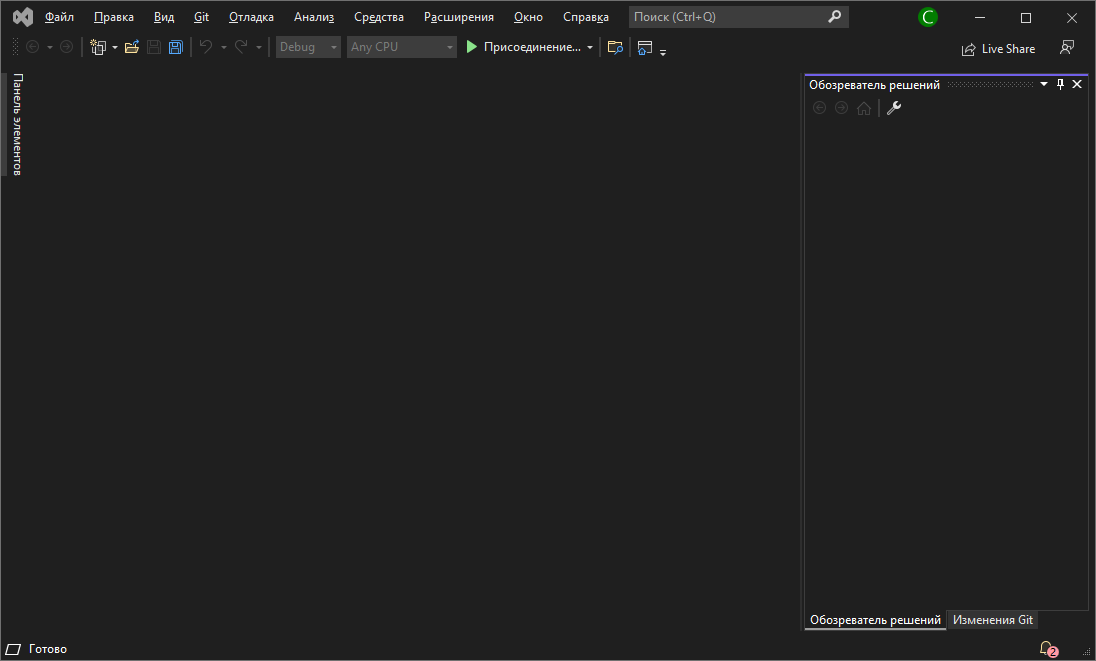


Рисунок 2.6 – Интерфейс «IDE» «Visual Studio Community»

Поэтому изначально на ней, например, нельзя разрабатывать приложения более поздние, чем на «.NET Framework 4.5.1». Для этого нужно вручную устанавливать пакеты платформ. Также у «SharpDevelop» нет инсталлятора тех самых пакетов и прочих дополнений, что есть у «Visual Studio».

Сравнение «Visual Studio» с другими средами разработки, а именно «SharpDevelop», «Geany» и «MonoDevelop», представлена в таблице 2.1.

Из таблицы 2.1, также можно сделать вывод, что «Visual Studio» является наиболее оптимальной «IDE» для разработки программного обеспечения (по крайней мере, в данном случае).

Таблица 2.1 – Сравнение «IDE» для «C#»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IDE | **Visual Studio** | SharpDevelop | Geany | MonoDevelop |
| Windows | **+** | + | + | + |
| Транслятор | **+** | + | - | + |
| Отладчик | **+** | + | - | + |
| Разработка UI | **+** | + | - | + |
| Покрытие кода | **+** | + | - | + |
| Автодополнение | **+** | + | + | + |
| Браузер классов | **+** | + | + | + |
| Подсветка синтаксиса | **+** | + | + | + |
| Поддержка дополнений | **+** | + | + | + |
| Менеджер дополнений | **+** | - | - | - |
| Сворачивание кода | **+** | + | + | + |
| Поддержка разработчиков в настоящее время | **+** | - | + | - |

С учетом выбранного языка программирования, было принято решения об использовании библиотеки «EPPlus» для работы с табличными файлами формата «XLSX».

Аналогами библиотеки «EPPlus» являются пакеты: «ClosedXML» и «NPOI». Все три библиотеки по функциональным возможностям являются почти одинаковыми.

Самым производительным является «EPPlus», а самым надежным «NPOI». Библиотека «ClosedXML» достаточно легка в усвоении.

Выбор библиотеки «EPPlus» связан с его производительностью, а также сравнении динамики популярности, показано на рисунках 2.7 – 2.9.

Как видно из графиков самой популярной библиотекой в среднем является «EPPlus».

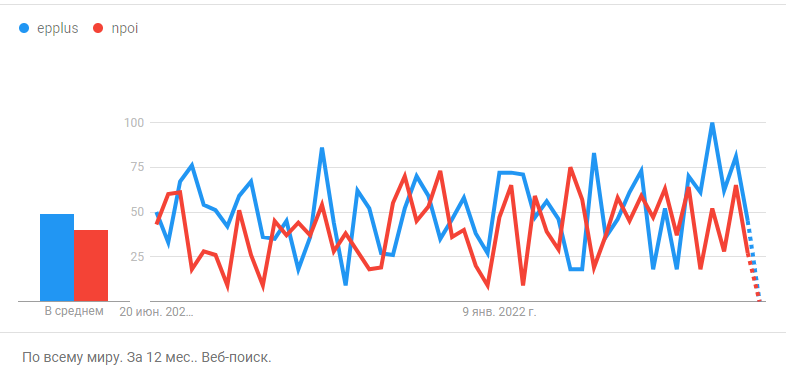


Рисунок 2.7 – Сравнение «EPPlus» и «NPOI»

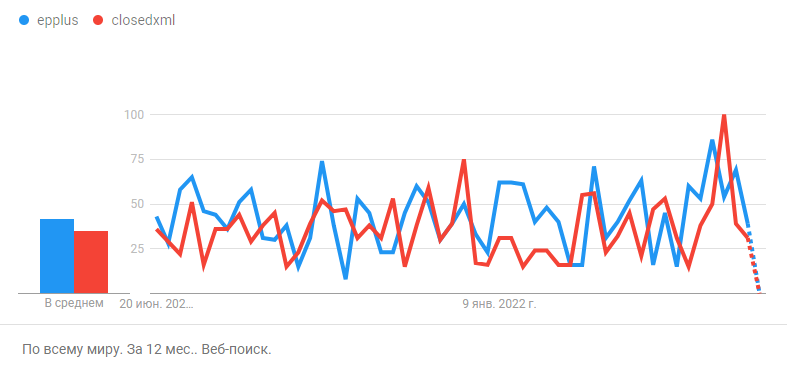


Рисунок 2.8 – Сравнение «EPPlus» и «ClosedXML»

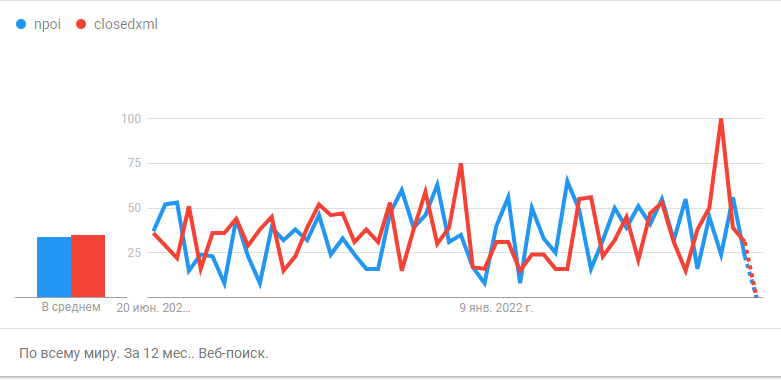


Рисунок 2.9 – Сравнение «NPOI» и «ClosedXML»

На платформе «.NET» имеются различные фреймворки для построения графического интерфейса. Для десктопных решений наиболее используемыми являются фреймворки «WPF» и «WinForms».

«Windows Forms» – графическая система в составе «.Net Framework». Представляет собой обертку вокруг «Win32 API» в управляемом коде. Считается заменой графической системы «MFC», написанной под «C++» и имеет сложную модель для разработки интерфейса программного продукта.

«Windows Presentation Foundation» – это графическая система в составе «.Net Framework 3.0» и более поздних версий. Спроектирована под влиянием технологий «HTML» и «Flash» и использует аппаратное ускорение [8].

Преимущества «WPF»:

* более новый, а, следовательно, больше соответствует современным стандартам;
* более гибкая разработка;
* кроссплатформенность;
* сторонние разработчики библиотек нацелены именно на «WPF»;
* интерфейс описывается при помощи разметки «XAML»;
* наличие механизма привязок.

Преимущества «WinForms»:

* более старый, соответственно более проверенный;
* дизайнер «Visual Studio» изначально разрабатывался именно для этого фреймворка.

В результате сравнения наиболее предпочтительным решением оказался фреймворк «WPF». Еще одним огромным преимуществом этого фреймворка является, то что благодаря механизму привязки данных, можно разрабатывать масштабируемые и гибкие приложения используя архитектуру «MVVM».

«MVVM» – это архитектурный паттерн, разделяющий разрабатываемый программный продукт на три части.

«Model» – часть приложения с бизнес-логикой.

«View» – отображение графических элементов.

«ViewModel» – контекст данных для отображения, служит некой прослойкой между бизнес-логикой и отображением.

Разделение на разные части делает приложение легко поддерживаемым, ускоряет разработку и повышает модульность программного кода.

Взаимодействие частей отображено на рисунке 2.10.

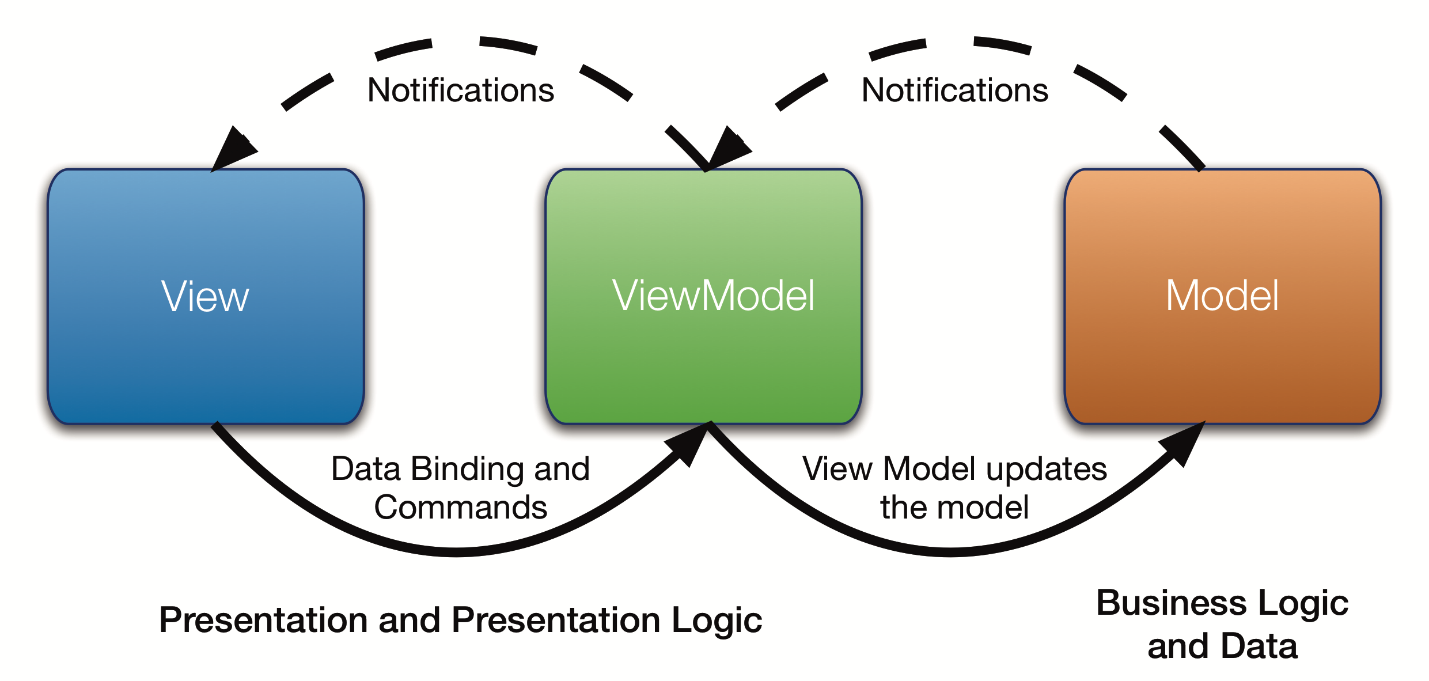


Рисунок 2.10 – Взаимодействие между частями паттерна «MVVM»

Гибкость разработки на «WPF» с использованием паттерна «MVVM» позволяет строить приятные и дружелюбные пользовательские интерфейсы. Поэтому было принято решение использовать элементы управления, предоставляемые корпорацией «Google» в пакете «Material Design in XAML». У этого пакета есть удобное демонстрационное приложение, изображенное на рисунке 2.11.

В приложении достаточно выбрать заинтересовавший элемент управления, и рядом отобразится код создания такого элемента управления, разработчику остается скопировать и подправить этот код.

Пример использования этого приложения показан на рисунке 2.12.

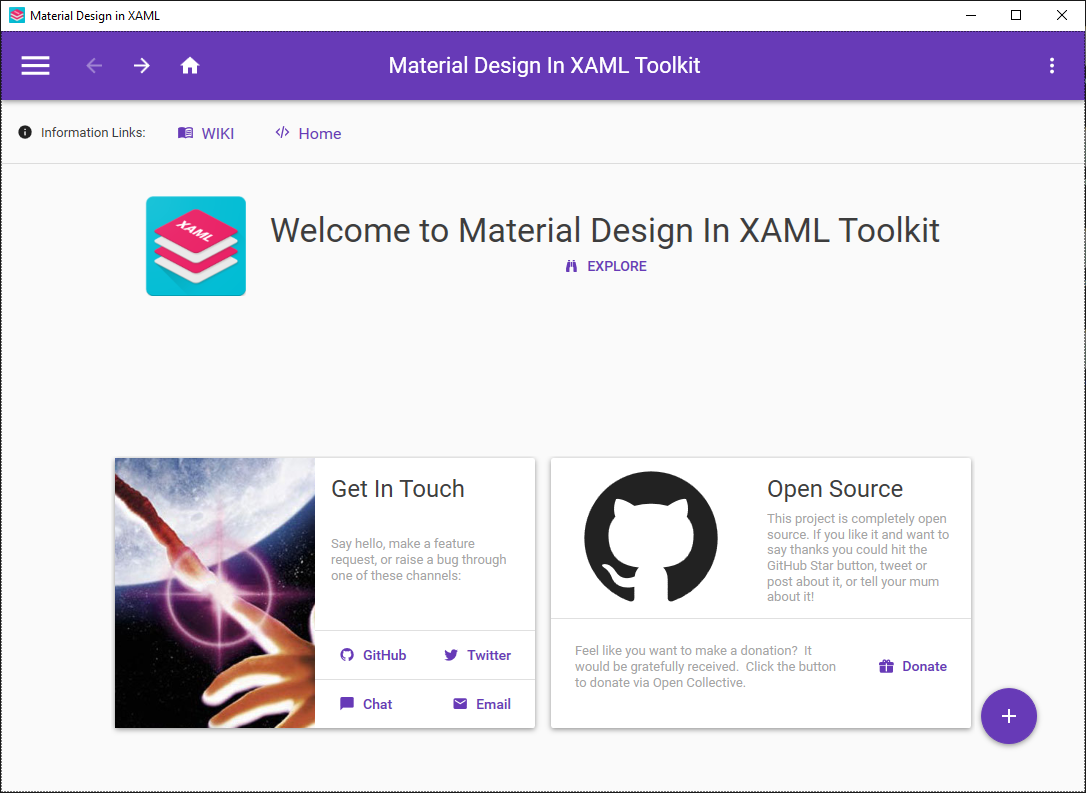


Рисунок 2.11 – Демонстрационное приложение возможностей пакета «Material Design in XAML»

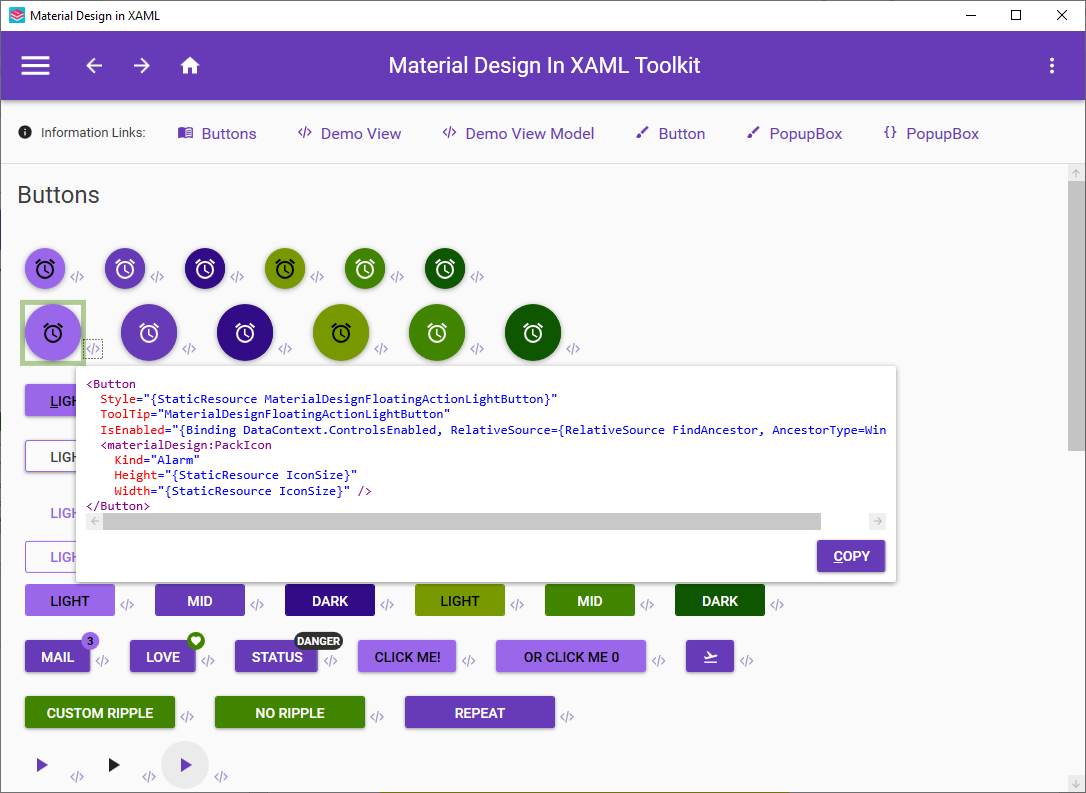


Рисунок 2.12 – Использование демонстрационного приложения

* 1. Чтение и запись данных

Для составления методов и алгоритмов чтения и записи данных в отчетах, необходимо ознакомиться и проанализировать содержимое табличных файлов.

Анализ присылаемых отчетов показал, что:

* необходимые для чтения данные хранятся не в фиксированных ячейках;
* форматы данных в ячейках различаются.

Было определено четыре разных вариантов заполнения. Первый вариант заполнения представлен на рисунке 2.13. В отчетах такого типа необходимо считать данные по шести столбцам и итоговую сумму количества проведенных проверок под таблицей. В разных файлах-отчетах, со схожим вариантом заполнения данные не всегда располагаются в одних и тех же ячейках. Для точного выявления местонахождения данных, необходимо ориентироваться на ячейку со следующим содержимым: «кол-во(месяц)». Под этой ячейкой и будут начинаться ячейки с данными.

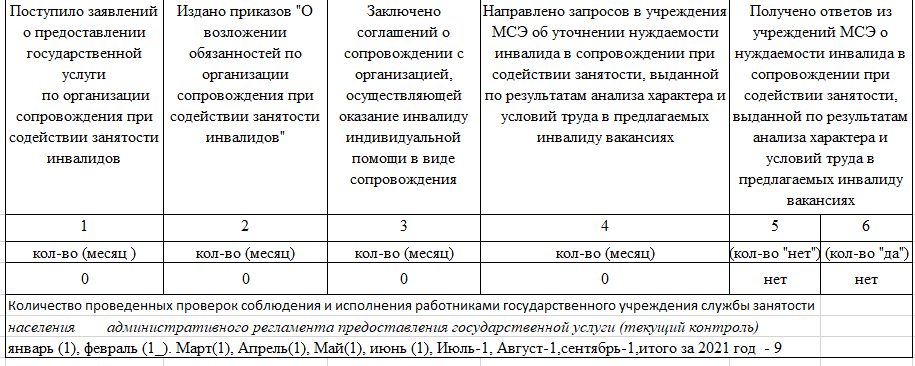


Рисунок 2.13 – Первый вариант заполнения

Для второго варианта заполнения, показанного на рисунке 2.14, характерно наличие ячеек с содержимым «х», поэтому следует ориентироваться на эти ячейки, а данные уже считывать над ними.

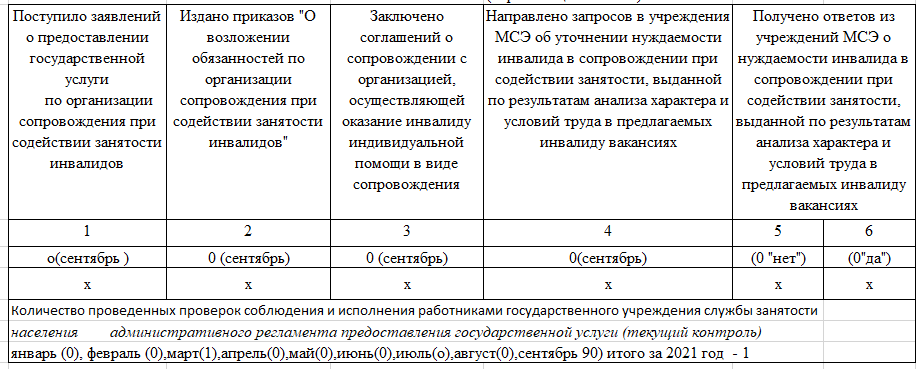


Рисунок 2.14 – Второй вариант заполнения

Для третьего варианта заполнения, показанного на рисунке 2.15, характерно наличие подряд идущих ячеек со значениями «(нет)» и «(да)». Данные располагаются под ориентировочными ячейками.

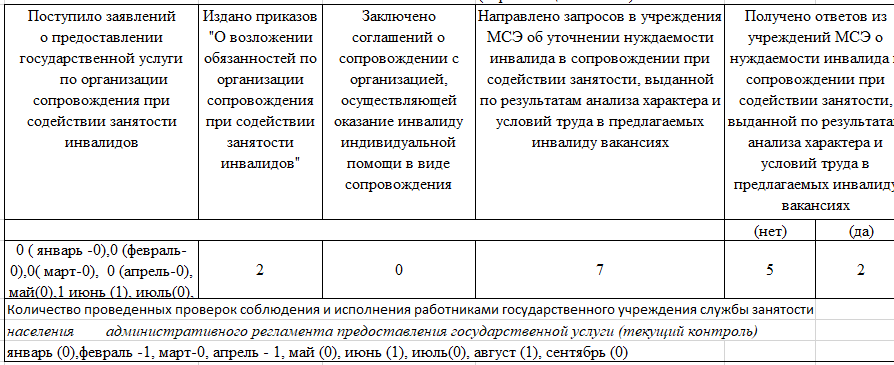


Рисунок 2.15 – Третий вариант заполнения

У последнего, четвертого варианта нет отличительных особенностей, а так как вариантов заполнения всего четыре, то все оставшиеся отчеты попадают в эту категорию.

Пример приведен на рисунке 2.16.

Помимо определения местонахождения данных также следует определить способы чтения этих данных. В сводный отчет должны попасть только числовые данные, большинство отчетов отвечают этим требованиям и в ячейках с данными хранят целое число, но все-таки есть часть ячеек, в которых данные представлены в текстовом формате.

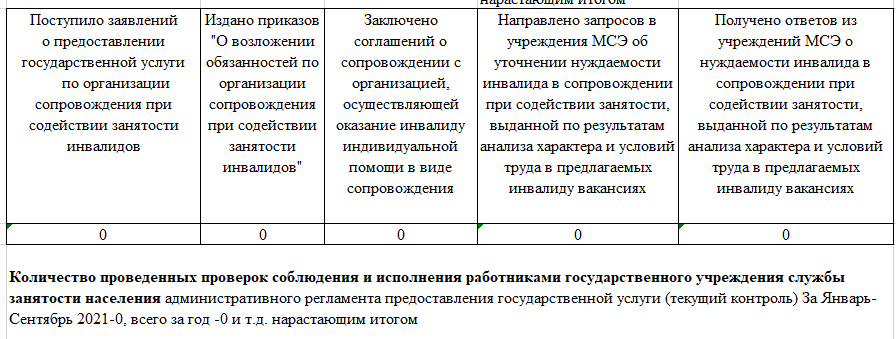


Рисунок 2.16 – Четвертый вариант заполнения

Пример текстовых данных, которые нужно перевести в числовой формат показан на рисунке 2.17.

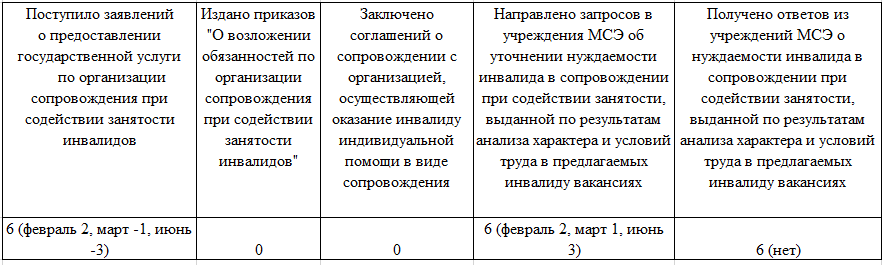


Рисунок 2.17 – Текстовые данные в ячейках

При варианте представления данных в формате как отображено на рисунке 2.17, необходимо убрать содержимое внутри скобок.

При варианте представления данных, показанном на рисунке 2.18, необходимо просуммировать все числа в ячейке, убрав заранее скобки и их содержимое.

В сводном отчете имеются метаданные которые необходимо заполнять сначала в интерфейсе программы, а уже потом записать при составлении сводного отчета.

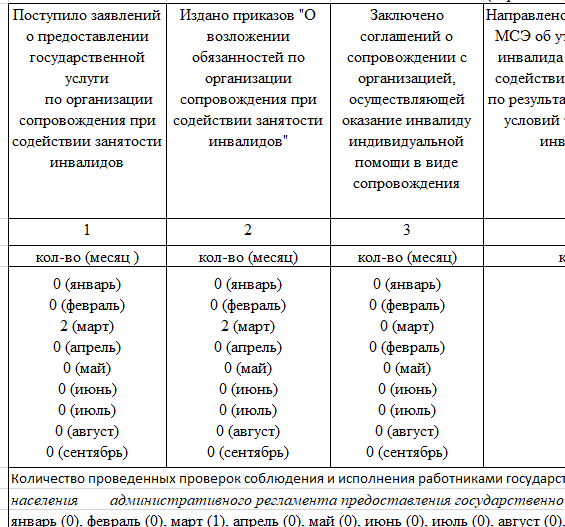


Рисунок 2.18 – Данные в ячейках в неправильном формате

Виды метаданных:

* месяц отчетности;
* дата составления отчета;
* название сводного отчета.

Некоторые метаданные записываются в верхней части сводного отчета, на рисунке 2.19. После метаданных идёт основная часть, состоящая из данных прочитанных с присланных табличных файлов-отчетов имеющие формат «XLSX».

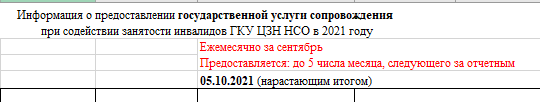


Рисунок 2.19 – Верхняя часть сводного отчета

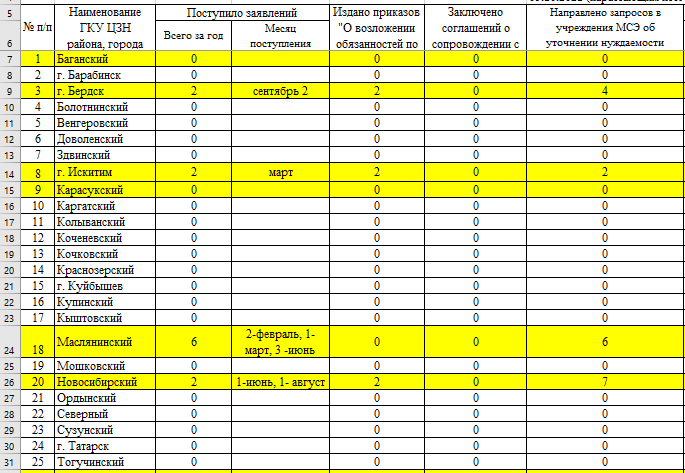


Рисунок 2.20 – Пример основной части сводного отчета

* 1. Вывод по второму разделу

Во втором разделе выпускной квалификационной работы была представлена постановка задачи с подробными условиями и спецификациями проблемы в предметной области.

В постановке задачи, было решение о реализации собственного генератора отчетов. Поэтому, для получения более полного и развернутого понимания и представления о разрабатываемом программном продукте были смоделированы бизнес-процессы разрабатываемой системы. Получены диаграммы:

* диаграмма вариантов использования предметной области;
* диаграмма последовательности процедуры создания отчета;
* диаграмма состояний программного модуля;
* диаграмма обзора взаимодействия формирования отчета;
* диаграммы последовательности всех элементов диаграммы обзора взаимодействия формирования отчета.

Были выбраны средства разработки программного продукта, а именно:

* язык программирования: «C#»;
* среда разработки: «Visual Studio»;
* библиотека для работы с табличными файлами: «EPPlus»;
* фреймворк для разработки графической части: «WPF».

Также было принято решение об использовании в проекте архитектурного паттерна «MVVM», а для более дружелюбного интерфейса использовать пакет «Material Design in XAML».

Были проанализированы все отчеты для определения подробностей чтения и записи данных.

Были выделены основные способы распознавания отчетов:

* ключевое слово «кол-во (месяц)»;
* ключевой символ «х»;
* последовательное расположение ячеек с содержимым «нет» и «да».

Ключевые слова показаны на рисунке 2.21

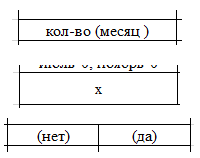


Рисунок 2.21 – Ячейки с ключевыми словами

Выявлено что для составления сводного отчета, перед считыванием данных с присланных отчетов, необходимо заполнить информацию о самом отчете.

Виды метаданных:

* месяц отчетности;
* дата составления отчета;
* название сводного отчета.

1. Разработка приложения
   1. Разработка архитектуры приложения

Любая современная разработка строится на принципах масштабируемости и изменчивости функционала создаваемого программного продукта. Основными преимуществами масштабируемости являются:

* модульная разработка (реализация приложения по частям);
* добавление нового функционала без изменения уже реализованного;
* расширения реализованного функционала без потери работоспособности остальных частей программного обеспечения;
* использование реализованного функционала при необходимости по частям;
* удаление неактуальных элементов функционирования.

Для написания масштабируемого приложения, на начальном этапе разработки необходимо выделить составные части разрабатываемого продукта. Составные части разрабатываемого приложения:

* бизнес – логика;
* пользовательский интерфейс;
* хранилище данных.

Составные части схематично представлены на рисунке 3.1.

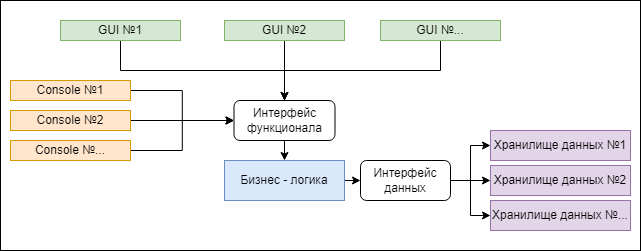


Рисунок 3.1 – Архитектура разрабатываемого ПО

Как видно из рисунка 3.1, при реализованной бизнес – логике можно создавать и настраивать множество различных графических интерфейсов отображения без изменения основного функционала. При этом можно реализовать приложение вовсе и без графической части, а только настроить консоль под «API» бизнес – логики. Интерфейсная прослойка позволяет не думать об ограничениях, накладываемых отдельными частями системы.

Графическая часть не имеет представления о работе бизнес – логики, та в свою очередь не знает откуда и как берутся данные.

Таким образом достигается гибкость итогового программного продукта, в будущем при необходимости можно будет изменить способ хранения данных (например, добавить работу с базой данных или хранить информацию в обычных файлах), графическое отображение или даже саму бизнес – логику без вмешательства в остальные части разрабатываемой системы.

Явной реализацией вышеупомянутой архитектуры в среде «WPF» является паттерн «MVVM». На рисунке 3.2 показана реализация данного паттерна для разрабатываемого проекта.

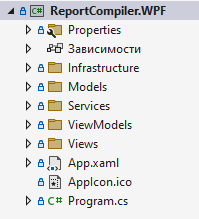


Рисунок 3.2 – Архитектура «MVVM»

* 1. Разработка функционала приложения

Благодаря гибкой архитектуре весь функционал вынесен в отдельные классы. Для работы с файлами был подключен пакет «EPPlus». Установленные в проекте пакеты показаны на рисунке 3.3.

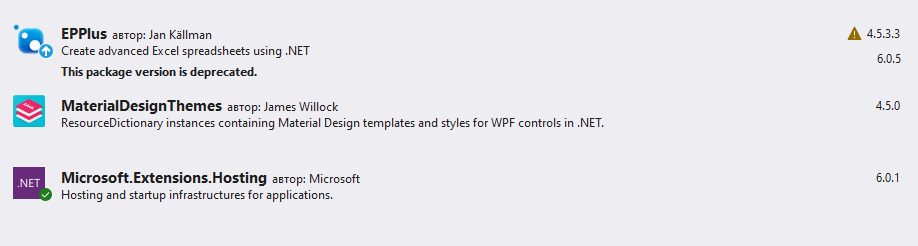


Рисунок 3.3 – Пакеты в проекте

Функционал составления сводного отчета находятся в классах, изображенных на рисунке 3.4.

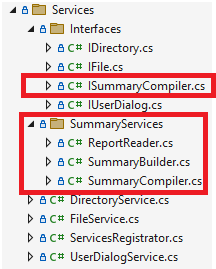


Рисунок 3.4 – Классы с основным функционалом

Интерфейс «ISummaryCompiler» описывает точки взаимодействия для составления сводного отчета, код интерфейса приведен в приложении А.

Интерфейс «ISummaryCompiler» реализует класс «SummaryCompiler». Для составления сводного отчета достаточно передать в качестве аргумента метода экземпляр класса «MetaData».

После получение метаданных класс «SummaryCompiler» составляет сводный отчет используя классы «ReportReader» и «SummaryBuilder».

Код функциональной части приведен в приложении А.

* 1. Разработка графического интерфейса

Приложение не требует разработки особо сложных и адаптируемых дизайнерских решений. Однако даже в таком приложении необходимо построить дружелюбный графический интерфейс для пользователей. Главное окно приложения показано на рисунке 3.5., «XAML» код главного окна приведен в приложении Б.

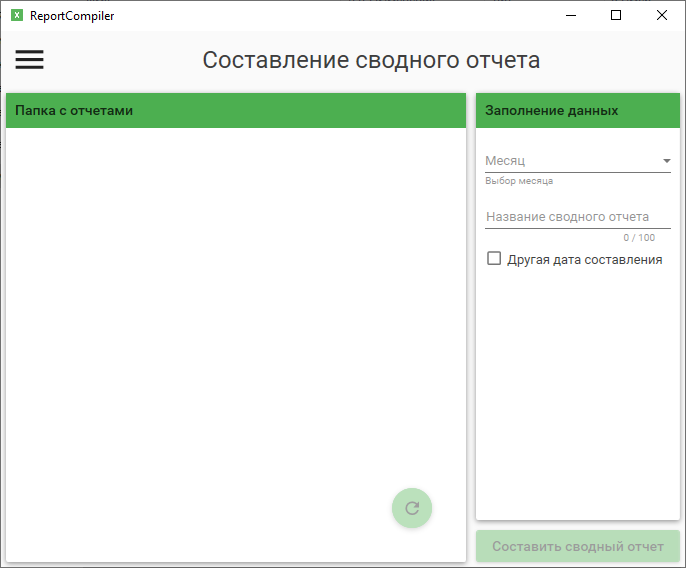


Рисунок 3.5 – Главное окно приложения

Все остальные визуальные элементы показаны на рисунках 3.6 – 3.8.

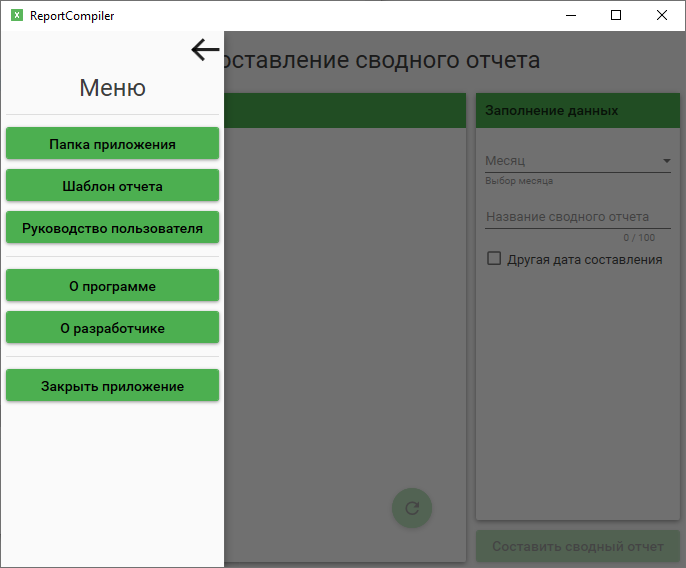


Рисунок 3.6 – Меню приложения

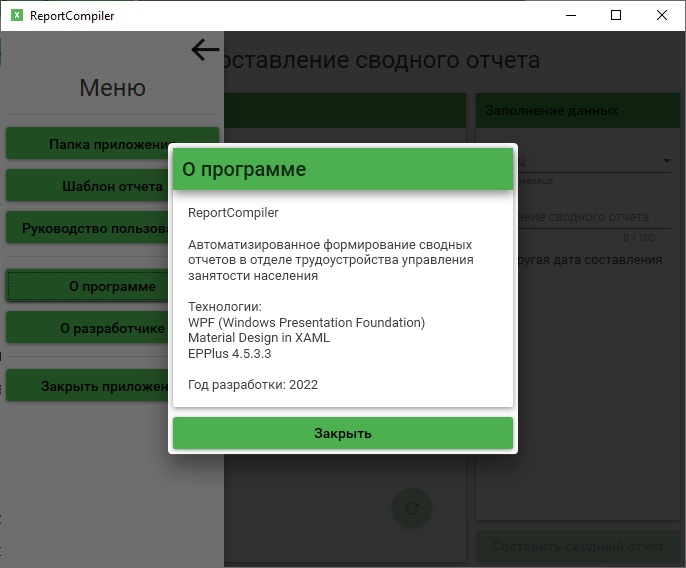


Рисунок 3.7 – Информация о программе

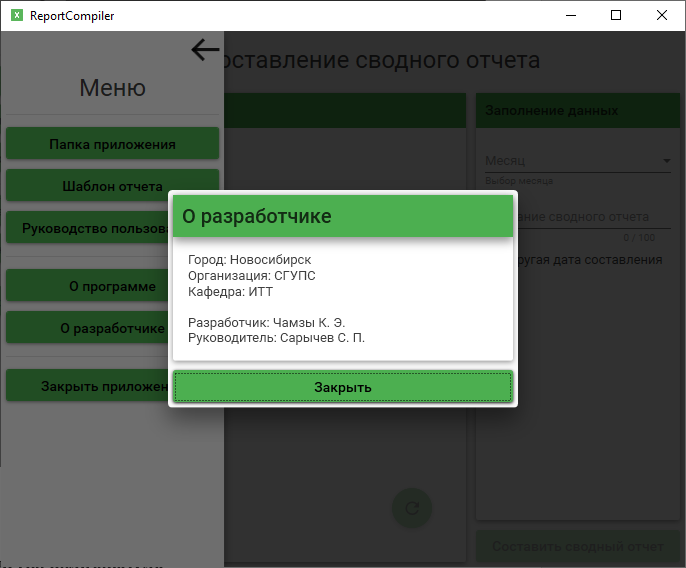


Рисунок 3.8 – Информация о разработчике

* 1. Руководство пользователя

Для старта программы достаточно запустить исполняемый файл, показанный рисунке 3.9.



Рисунок 3.9 – Исполняемый файл программы

После запуска выводится окно на рисунке 3.10, описание к элементам интерфейса представлено на рисунке 3.11.

Пункты меню представлены на рисунке 3.12. Если в директории приложения не будут присутствовать файлы шаблона отчетов или руководства пользователя, то кнопки открывающие соответствующие файлы будут недоступны. Вывод информации о программе и о разработчике не привязаны к файлам, поэтому они доступны всегда, их содержимое показано на рисунках 3.13 и 3.14.

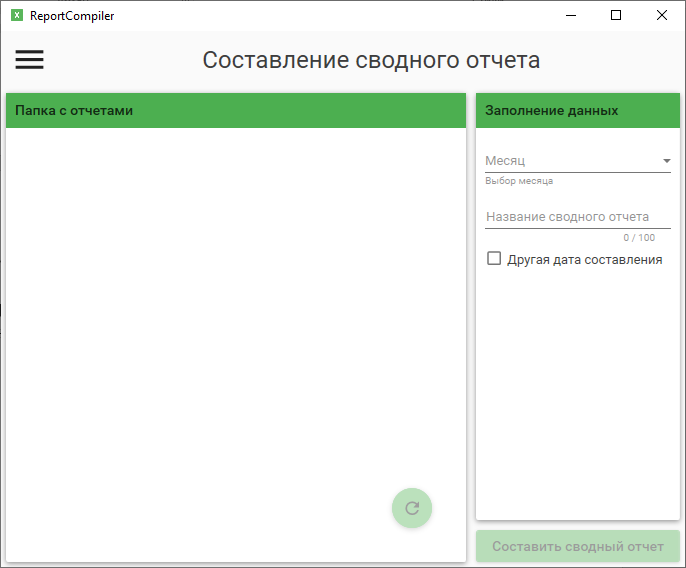


Рисунок 3.10 – Главное окно приложение при старте

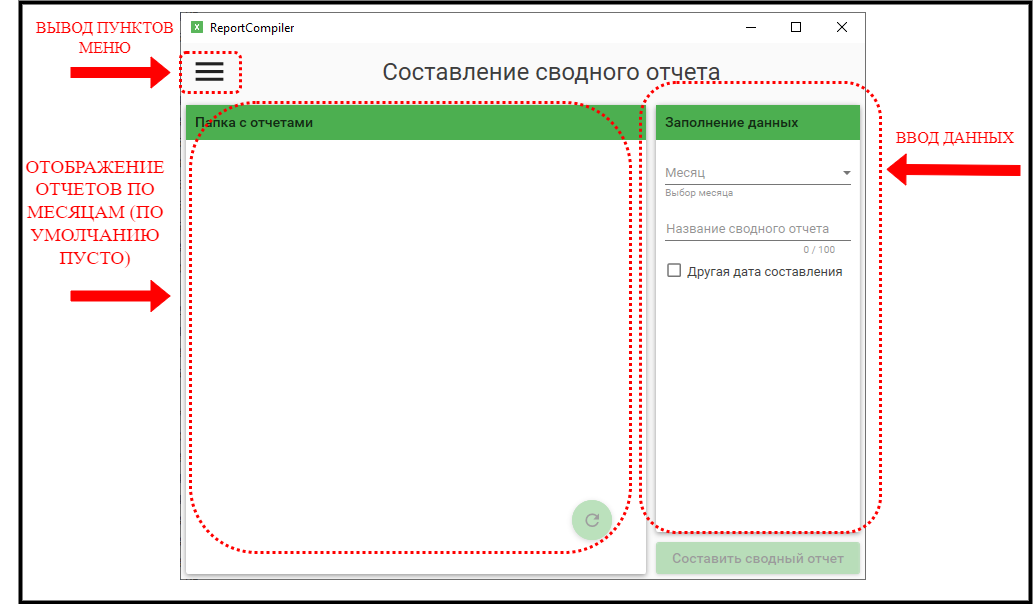


Рисунок 3.11 – Описание интерфейса приложения

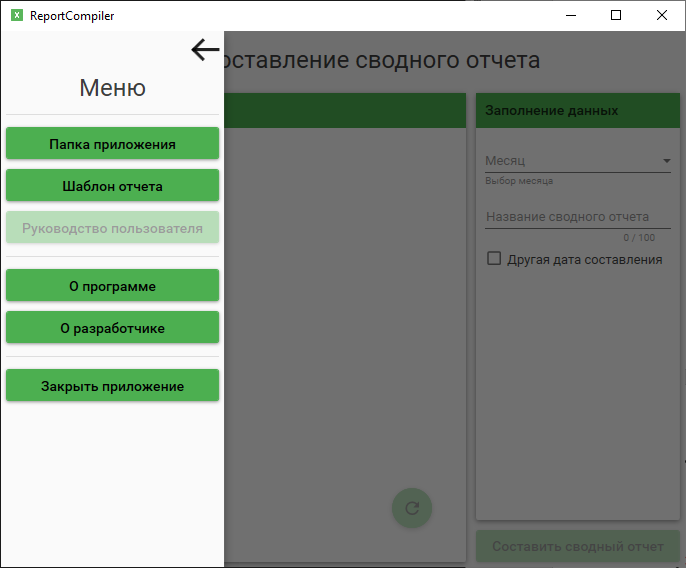


Рисунок 3.12 – Пункты меню

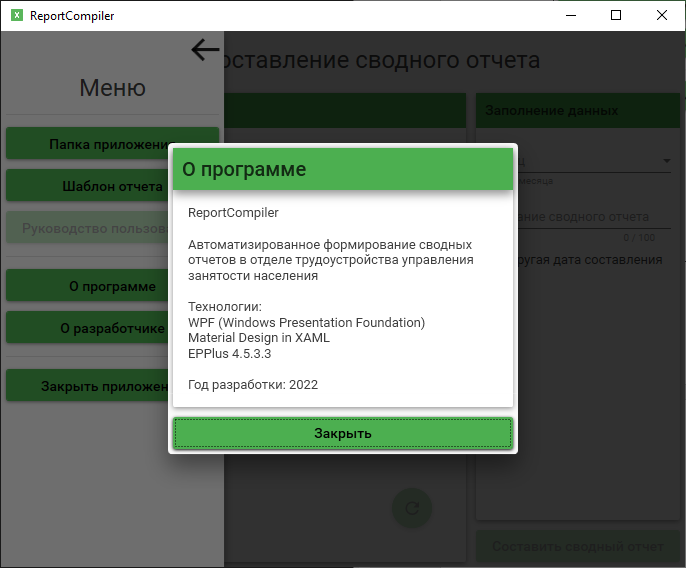


Рисунок 3.13 – Пункт меню «О программе»

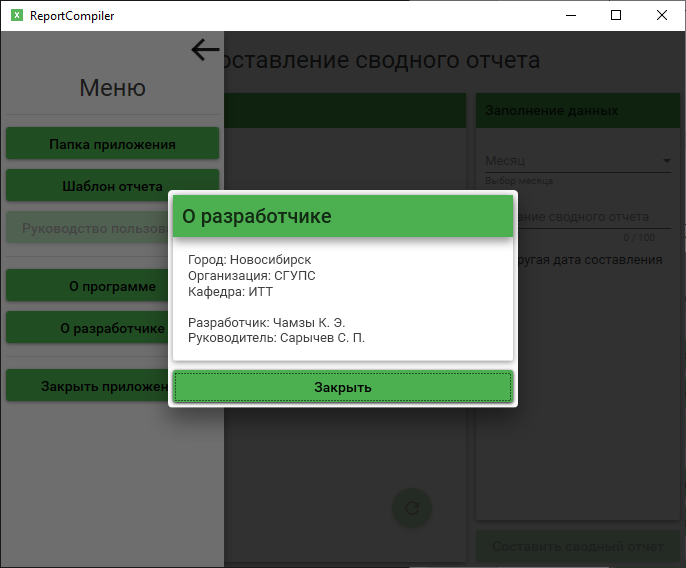


Рисунок 3.14 – Пункт меню «О разработчике»

Для нормальной работы программы требуются файлы, изображенные на рисунке 3.15. Они должны быть расположены строго в директории приложения. Править эти файлы не рекомендуется.

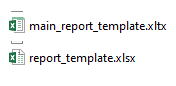


Рисунок 3.15 – Файлы необходимые для работы приложения

Для составления сводного отчета необходимо:

* добавить отчеты в папку с месяцем составления, рисунки 3.16;
* запустить программу;
* выбрать месяц, на рисунке 3.17;
* проверить отчеты по выведенному списку, отображение отчетов в программе показано на левой части рисунка 3.17 (двойным щелчком по определенному отчету можно открыть его);
* ввести название сводного отчета, на рисунке 3.18 (без выбранного месяца и без названия сводного отчета кнопка «Составить сводный отчет» будет неактивна);
* при необходимости, дату которая будет ставится в сводном отчете можно поменять, выбрав другую как показано на рисунке 3.19;
* нажать на кнопку «Составить сводный отчет»;
* после успешного составления сводного отчета, на экране появится сообщение, показанное на рисунке 3.20 (успешно созданный сводный отчет хранится в папке «Сводный отчет» в папке выбранного месяца, это можно увидеть на рисунке 3.21).

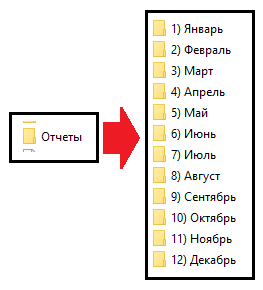


Рисунок 3.16 – Папки для добавления отчетов в приложение (папка «Отчеты» всегда находится в папке с приложением)

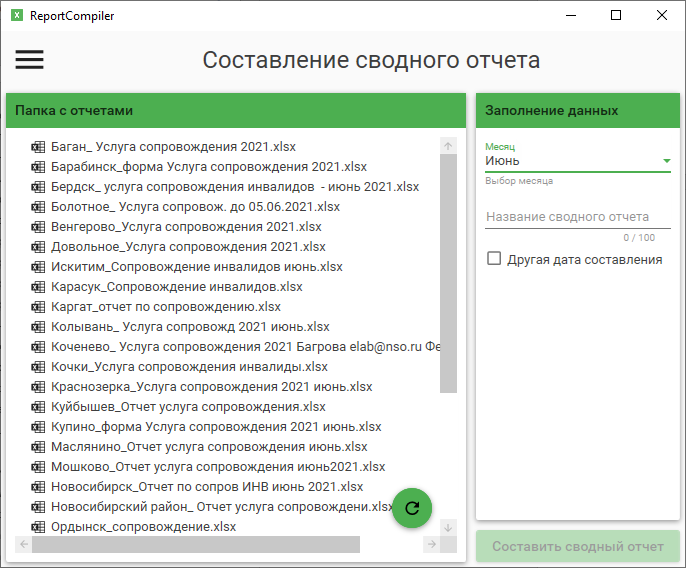


Рисунок 3.17 – Выбор месяца (отображенные отчеты конкретно в данном примере тестовые)

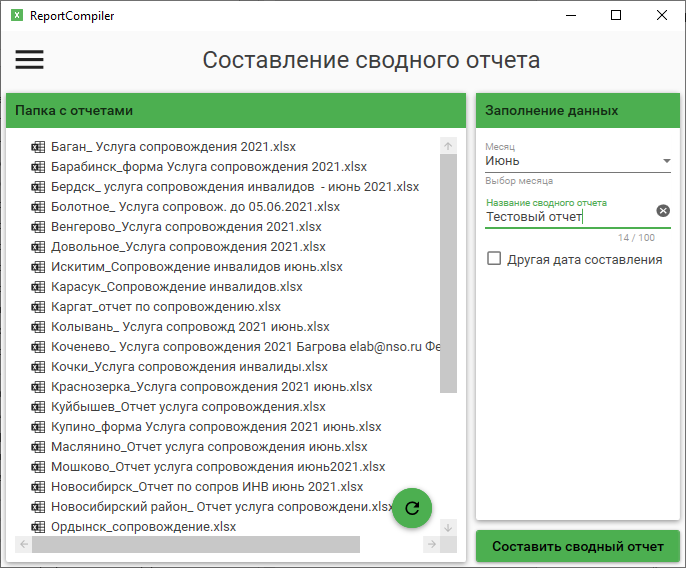


Рисунок 3.18 – Ввод названия сводного отчета

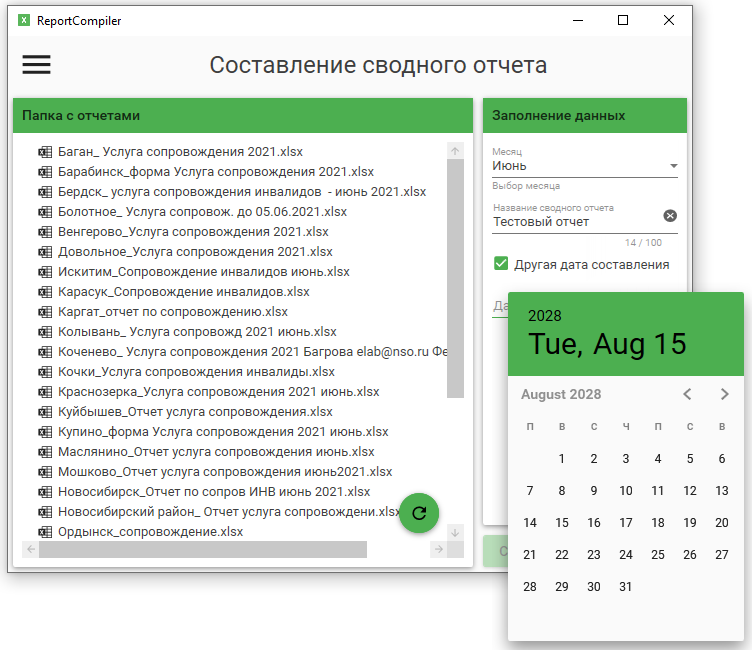


Рисунок 3.19 – Выбор другой даты (необязательное условие составления сводного отчета)

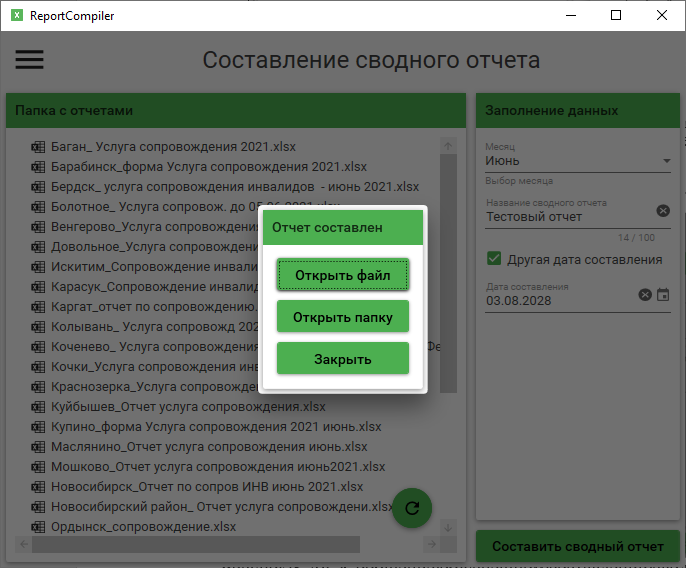


Рисунок 3.20 – Сообщение об успешном составлении сводного отчета

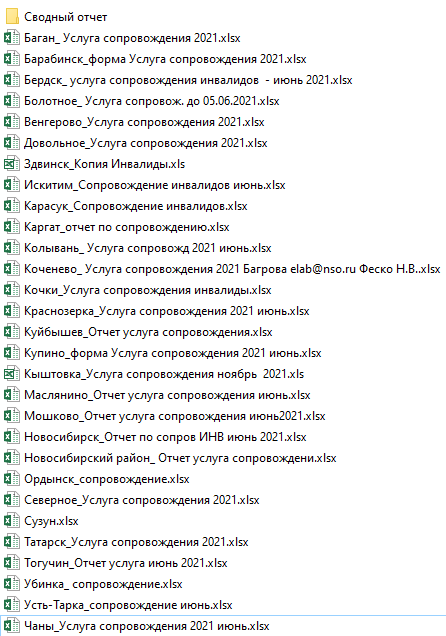


Рисунок 3.21 – В папке каждого месяца есть папка в котором хранятся все сводные отчеты созданные с помощью программы

Для корректной работы программы требуется определенная организованность данных.

Программа может прочитать данные только с определенных отчетов, заполненных по шаблону, файл с шаблонами можно открыть через пункт меню «Шаблон отчета». Если данные не удалось прочитать, то программа выводит название файла, адрес ошибки и список возможных действий (открытие файла, повторная проверка и открытия папки файлов), это показано на рисунке 3.22.

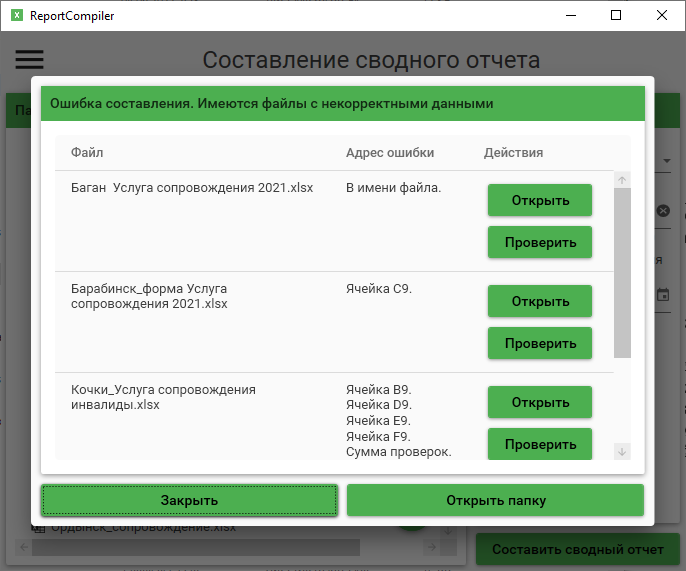


Рисунок 3.22 – Вывод ошибок в отчетах

Чтобы составить сводный отчет, необходимо убрать все ошибки. Для устранения ошибки нужно нажать на кнопку «Открыть» и подправить файл в соответствии с шаблоном. Ошибка в ячейках указывает, что в этих ячейках нету численных данных.

Ошибка «Сумма проверок» сигнализирует что в файле отчета не удалось определить где находится информация о сумме проведенных проверок (либо такая информация отсутствует, либо количество указано по месяцам, но без суммирования).

Ошибка в имени файла связано с определением района (города отчета). Для определения района, необходимо чтобы в имени файла до символа «\_» было записано название административного центра района (или город). Каждое название сопоставляется по правилу, отображенному на рисунке 3.23. Ошибка выводится в том случае если название составлено не по шаблону или его нету в списке соответствий, на рисунке 3.23.

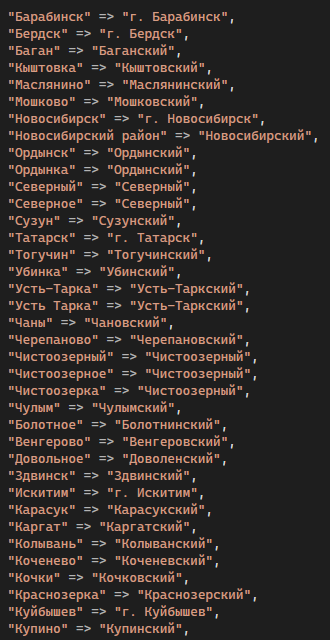


Рисунок 3.23 – Сопоставление названий поселений к названиям районов

После исправления ошибок, необходимо либо заново начать процедуру составления сводного отчета, либо нажать на кнопку «Проверить».

Если файл отчет исчез со списка, это означает что файл был успешно проверен и готов к использованию в программе.

Пример сформированного сводного отчета показан на рисунке 3.24.

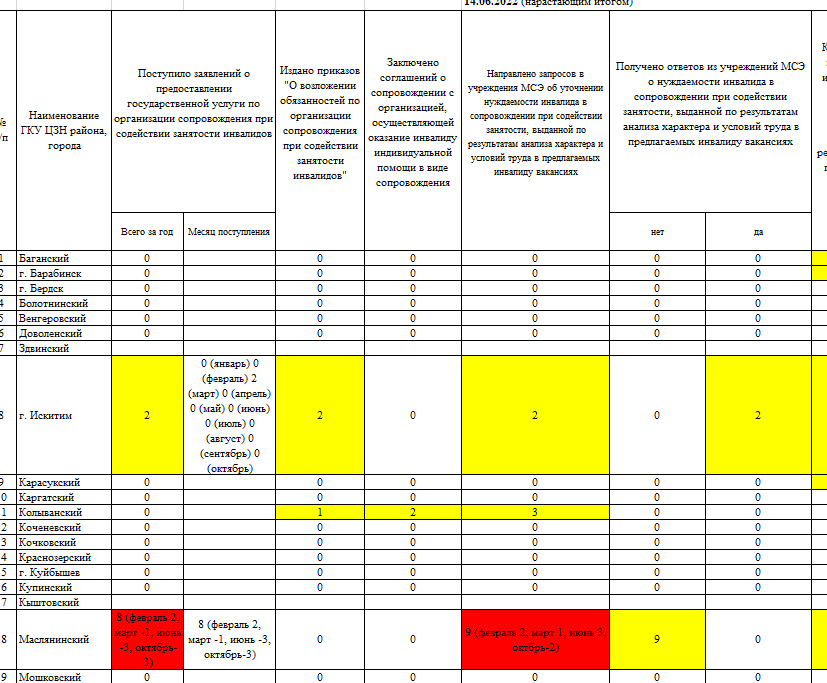


Рисунок 3.24 – Часть сформированного сводного отчета

Все поля со значащими данными отмечаются желтым цветом.

Поля, которые не удалось считать, подсвечиваются красным цветом. Эти поля необходимо заполнить вручную (программное заполнение не рекомендуется из-за большой вероятности внесение ошибочных данных в итоговый сводный отчет).

Если данные по какому-то району (городу) отсутствуют, то это означает что файлов отчетов по этому району (городу) в исходной папке найти не удалось.

Во втором столбце записываются текстовые данные в неизменном виде, для определения месяца поступления, когда проводится суммирование в первом столбце.

* 1. Вывод по третьему разделу

В третьем разделе выпускной квалификационной работы была разработана информационная система автоматизирующая процесс составления сводного отчета в отделе трудоустройства управления занятости населения.

Разработка приложения началась с постройки правильной архитектуры кода, в результате получилась архитектура, изображенная на рисунке 3.25.

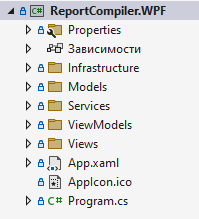


Рисунок 3.25 – Итоговая архитектура разработанного приложения

Разработка с использованием паттерна «MVVM» позволила реализовать основную бизнес-функцию отдельно от других компонентов. Код в итоге поделился на два типа, вспомогательный и основной. Количество основного кода оказалось сильно меньше вспомогательного.

Благодаря использованию пакета «Material Design in XAML» удалось быстро выстроить удобный и легкий для восприятия интерфейс, изображенный на рисунке 3.26.

В руководстве пользователя были описаны все возможности приложения и действия при аварийных ситуациях.

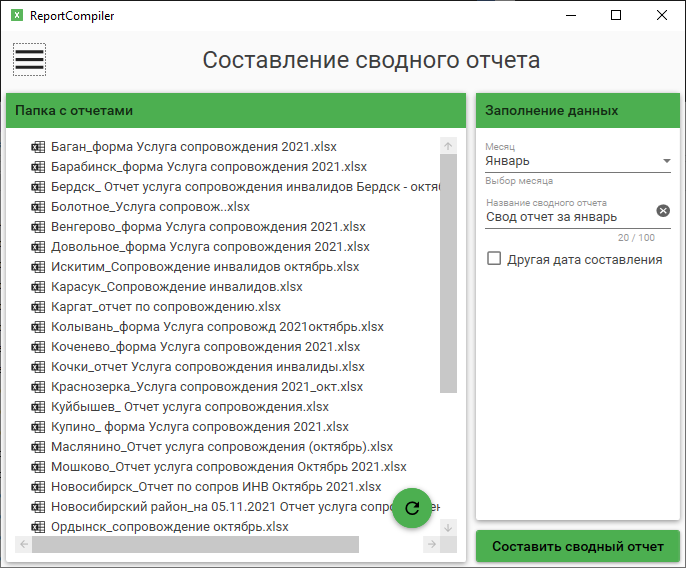


Рисунок 3.26 – Итоговый пользовательский интерфейс

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы были изучены области автоматизации отчетности, генераторы отчетов, а также программные средства способные создать генераторы для специфичных предметных областей.

В результате выполнения выпускной квалификационной работы был автоматизирован процесс составления отчета о предоставлении государственной услуги сопровождения при содействии занятости инвалидов ГКУ ЦЗН.

Для этого были решены следующие задачи:

* проведен аналитический обзор предметной области;
* осуществлен анализ генераторов отчетов, которые уже имеются на рынке в текущий момент;
* сформированы требования к разрабатываемому программному продукту;
* смоделированы бизнес-процессы для разрабатываемой системы;
* были выбраны и обоснованы среды и средства разработки;
* разработана информационная система;
* составлено руководство пользователя;
* проанализированы результаты работы.

В итоге было получено приложение составляющее сводный отчет в отделе трудоустройства управления занятости и исследована область генерации отчетов формата «Excel» программными средствами.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Что такое сводный отчет [Электронный ресурс]. – Режим доступа: – https://car3g.ru/info/chto-takoe-svodnyj-otchet (Дата обращения 14.05.2022).
2. Генератор отчётов [Электронный ресурс]. – Режим доступа: – https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:%D0%93%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80\_%D0%BE%D1%82%D1%87%D1%91%D1%82%D0%BE%D0%B2 (Дата обращения 14.05.2022).
3. Библиотека отчетов JasperReports [Электронный ресурс]. – Режим доступа: – https://java-online.ru/jasperreports.xhtml (Дата обращения 14.05.2022).
4. Старт работы с Excel на C# / Хабр [Электронный ресурс]. – Режим доступа: – https://habr.com/ru/post/525492 (Дата обращения 14.05.2022).
5. Features - EPPlus Software [Электронный ресурс]. – Режим доступа: – https://epplussoftware.com/ru/Developers/Features (Дата обращения 14.05.2022).
6. EPPlus vs NPOI | LibHunt [Электронный ресурс]. – Режим доступа: – https://dotnet.libhunt.com/compare-epplus-vs-npoi (Дата обращения 14.05.2022).
7. Программирование на VBA в Excel 2016. Самоучитель / Н. В. Комолова, Е. С. Яковлева. — СПб. : БХВ-Петербург, 2019. — 432 с. : ил. — (Самоучитель).
8. МакДональд, Мэтью. WPF 4.5: Windows Presentation Foundation в .Net 7 с примерами на C# 2019 для профессионалов. : Пер. с англ. – М. : ООО «И.Д. Вильямс», 2019. – 1024 с. : ил. – Парал. тит. англ..

ПРИЛОЖЕНИЕ А

**Код функциональной части программы**

Код функциональной части представлен на рисунке А.1.

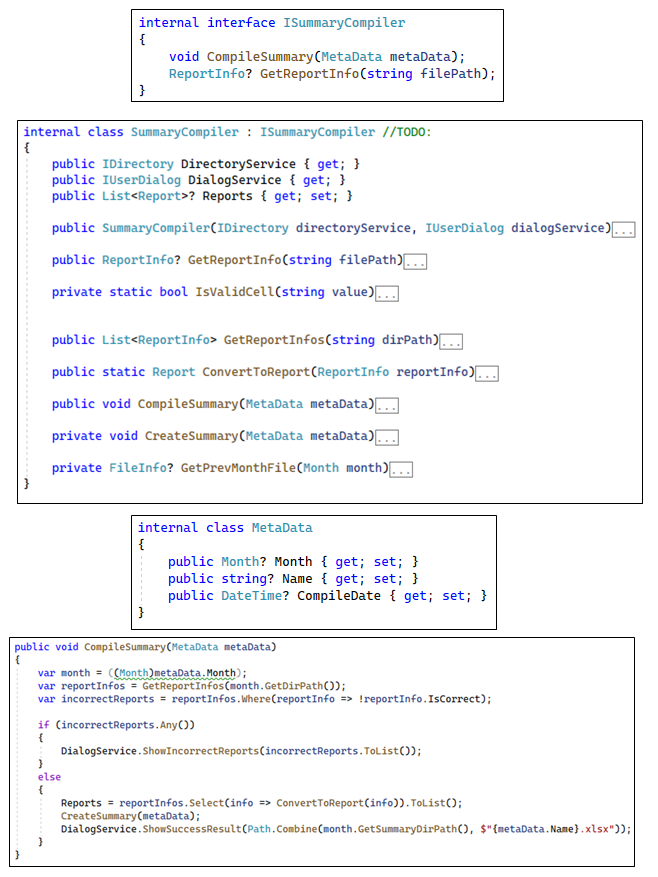


Рисунок А.1 – Код основной бизнес-логики приложения

ПРИЛОЖЕНИЕ Б

**Код графического интерфейса программы**

<Window x:Class="ReportCompiler.WPF.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" mc:Ignorable="d"

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

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

Title="ReportCompiler"

Height="575" Width="700"

MinHeight="500" MinWidth="675"

WindowStartupLocation="CenterScreen"

Style="{StaticResource MaterialDesignWindow}">

<Window.Resources>

<Style

TargetType="Button"

BasedOn="{StaticResource MaterialDesignRaisedButton}">

<Setter

Property="Margin"

Value="5"/>

</Style>

<Style

TargetType="GroupBox"

BasedOn="{StaticResource MaterialDesignCardGroupBox}">

<Setter

Property="Margin"

Value="5"/>

</Style>

</Window.Resources>

<md:DialogHost>

<md:DialogHost.DialogContent>

<ContentControl

Content="{Binding}"/>

</md:DialogHost.DialogContent>

<md:DrawerHost

IsLeftDrawerOpen="{Binding

ElementName=MenuToggleButton,

Path=IsChecked}">

<md:DrawerHost.LeftDrawerContent>

<DockPanel

DataContext="{Binding

Source={StaticResource Locator},

Path=MenuViewModel}">

<ToggleButton

Style="{StaticResource MaterialDesignHamburgerToggleButton}"

HorizontalAlignment="Right"

IsChecked="{Binding

ElementName=MenuToggleButton,

Path=IsChecked,

Mode=TwoWay}"

DockPanel.Dock="Top"/>

<ContentPresenter Content="{Binding}"/>

</DockPanel>

</md:DrawerHost.LeftDrawerContent>

<DockPanel>

<DockPanel DockPanel.Dock="Top">

<ToggleButton

Name="MenuToggleButton"

Style="{StaticResource

MaterialDesignHamburgerToggleButton}"

IsChecked="False"

DockPanel.Dock="Left"

Margin="10"/>

<TextBlock

Text="Составление сводного отчета"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Style="{StaticResource

MaterialDesignHeadline5TextBlock}"/>

</DockPanel>

<DockPanel DockPanel.Dock="Right">

<Button

DockPanel.Dock="Bottom"

Content="Составить сводный отчет"

Command="{StaticResource CreateReportCommand}"

CommandParameter="{Binding

Source={StaticResource Locator},

Path=MetaDataViewModel}"/>

<GroupBox DataContext="{Binding

Source={StaticResource Locator},

Path=MetaDataViewModel}">

<GroupBox.Header>

<TextBlock

Style="{DynamicResource

MaterialDesignSubtitle2TextBlock}"

Text="Заполнение данных"/>

</GroupBox.Header>

<StackPanel MaxWidth="225">

<ContentPresenter Content="{Binding}"/>

</StackPanel>

</GroupBox>

</DockPanel>

<GroupBox DataContext="{Binding

Source={StaticResource Locator},

Path=ReportsViewModel}">

<GroupBox.Header>

<TextBlock

Style="{DynamicResource

MaterialDesignSubtitle2TextBlock}"

Text="Папка с отчетами"/>

</GroupBox.Header>

<ContentPresenter Content="{Binding}"/>

</GroupBox>

</DockPanel>

</md:DrawerHost>

</md:DialogHost>

</Window>