



Федеральное государственное бюджетное образовательное учреждение высшего образования
«ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ ДВАЖДЫ ГЕРОЯ
СОВЕТСКОГО СОЮЗА, ЛЕТЧИКА-КОСМОНАВТА А.А. ЛЕОНОВА»

Колледж космического машиностроения и технологий

«Допущен к защите»

председатель цикловой комиссии

Д.Н. Никонова

«14» июня 2023 г.

Дипломный проект
РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ
РЕЗЕРВНОГО КОПИРОВАНИЯ ДАННЫХ

Пояснительная записка

ДП.09.02.03.23.номер по приказуПЗ

Обучающийся группы ПЗ-20 Корниецкий Ф.А.

Руководитель дипломного проекта Лихторенко О.С.

Результат защиты _____

СОДЕРЖАНИЕ

| | |
|---|----|
| ВВЕДЕНИЕ..... | 3 |
| 1. ТЕОРИТИЧЕСКАЯ ЧАСТЬ..... | 5 |
| 1.1. Описание предметной области..... | 5 |
| 1.2. Постановка задачи | 9 |
| 1.3. Анализ существующих программных решений..... | 12 |
| 2. ПРОЕКТНАЯ ЧАСТЬ..... | 17 |
| 2.1. Описание алгоритмов решения | 17 |
| 2.2. Обоснование выбора средств разработки..... | 21 |
| 2.3. Описание ER-модели..... | 26 |

| | | | | | | | | | | | | | |
|-------------|--|----------------|---------|------|---|--|--|--|----------------|------|--------|----|--|
| | | | | | ДП.09.02.03.23.1ПЗ | | | | | | | | |
| Изм. | | № докум. | Подпись | Дата | | | | | | | | | |
| Разраб. | | Корницкий Ф.А. | | | РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ РЕЗЕРВНОГО КОПИРОВАНИЯ ДАННЫХ | | | | Лит. | Лист | Листов | | |
| Провер. | | Фамилия И.О. | | | | | | | | | 3 | 59 | |
| Конс. эк.ч. | | Фамилия И.О. | | | | | | | ККМТ гр. ПЗ-20 | | | | |
| Н. Контр. | | Фамилия И.О. | | | | | | | | | | | |
| Утверд. | | Никонова Д.Н. | | | | | | | | | | | |

ВВЕДЕНИЕ

В современном мире информация становится одним из самых ценных ресурсов. В связи с этим возрастает необходимость в надежных методах защиты данных от потерь и повреждений. Разработка эффективных систем резервного копирования и восстановления данных приобретает особое значение в условиях постоянно увеличивающегося объема информации и растущих требований к ее безопасности.

Целью данного дипломного проекта является разработка программы для резервной архивации данных, которая позволит пользователям автоматизировать процесс создания резервных копий, обеспечивая сохранность и доступность важной информации. Проект включает исследование существующих методов и инструментов для резервного копирования, анализ их преимуществ и недостатков, а также разработку и тестирование собственного программного решения, соответствующего современным требованиям к надежности и удобству использования.

Введение в тему резервного копирования данных начинается с рассмотрения актуальности данной проблемы в условиях цифровой трансформации бизнеса и повседневной жизни. Мы также рассмотрим основные понятия, связанные с процессом резервного копирования, такие как типы резервных копий, стратегии их хранения и методы восстановления данных. В ходе проекта будет разработана программа, ориентированная на обеспечение высокой скорости и надежности копирования, простоты настройки и использования, а также интеграции с различными системами хранения данных.

Таким образом, данный проект направлен на создание комплексного решения для резервного копирования данных, которое будет соответствовать высоким требованиям современных пользователей и обеспечивать надежную защиту информации от возможных потерь.

В процессе разработки программного обеспечения, она получила название “Backup”.

В первой главе пояснительной записки будет описана предметная область, постановка задачи на дипломный проект и анализ аналогов-конкурентов.

Во второй главе будет описание алгоритмов решения, обоснование средств разработки, описание диаграммы базы данных и т.д.

В третьей главе будет описано управление проектом разработки, расчет затрат на разработку и внедрение программы, а также руководство пользователя.

1. ТЕОРИТИЧЕСКАЯ ЧАСТЬ

1.1. Описание предметной области

В XXI веке информация и данные стали одними из важнейших ресурсов, с которыми работают пользователи компьютеров. В повседневной жизни и профессиональной деятельности пользователи взаимодействуют с различными типами файлов — от текстовых документов и электронных таблиц до фотографий и видеофайлов. Важность этих данных делает их защиту критически важной. Потеря данных может привести к серьезным последствиям, будь то утрата личных воспоминаний или критически важной деловой информации. Именно поэтому существует необходимость в надежных методах резервного копирования данных.

Резервное копирование данных является ключевым элементом в стратегии защиты информации. Существует множество причин, по которым данные могут быть утеряны или повреждены, включая аппаратные сбои, вирусные атаки, ошибки пользователя и даже природные катастрофы. В условиях растущих объемов данных и увеличивающихся требований к их безопасности, необходимость в эффективных и надежных решениях для резервного копирования становится все более актуальной.

История программ для резервного копирования данных начинается в 90-х годах, когда персональные компьютеры и цифровые данные начали широко использоваться. Первые программы для резервного копирования были разработаны с акцентом на функциональность, но зачастую им не хватало удобства использования и интуитивного интерфейса. В те времена основное внимание уделялось надежности и функциональности, что привело к созданию сложных и не всегда удобных для пользователя решений.

С тех пор технологии значительно эволюционировали. Современные программы для резервного копирования должны соответствовать новым требованиям пользователей, включая простоту в использовании, высокую скорость выполнения задач и интуитивно понятный интерфейс. Однако многие из существующих решений, разработанных несколько десятилетий

назад, не отвечают этим требованиям, что делает их неудобными для современных пользователей.

Современные пользователи ожидают от программ для резервного копирования нескольких ключевых характеристик:

1. Интуитивно понятный интерфейс. Программы должны быть простыми и понятными для пользователей всех уровней, от новичков до опытных пользователей. Это включает в себя доступность основных функций и возможность легко настроить параметры резервного копирования.
2. Высокая скорость работы. В условиях увеличивающихся объемов данных программы должны обеспечивать быструю и эффективную обработку данных, минимизируя время, необходимое для создания резервных копий и их восстановления.
3. Надежность. Программы должны гарантировать целостность и сохранность данных, обеспечивая надежную защиту от потерь и повреждений.
4. Гибкость настроек. Пользователи должны иметь возможность настроить программы под свои специфические нужды, включая выбор методов и частоты резервного копирования, а также управление местами хранения данных.
5. Совместимость с различными системами хранения. Программы должны поддерживать работу с различными типами носителей, включая портативные диски хранения информации.

Существует несколько методов резервного копирования данных, каждый из которых имеет свои преимущества и недостатки:

1. Полное резервное копирование. При полном резервном копировании создается точная копия всех данных. Это самый надежный метод, но он требует значительного времени и места для хранения.

2. Инкрементное резервное копирование. Этот метод создает копии только тех данных, которые были изменены с момента последнего резервного копирования. Это значительно уменьшает время и объем данных, необходимых для копирования, но может усложнить процесс восстановления.
3. Дифференциальное резервное копирование. Этот метод похож на инкрементное копирование, но копирует все изменения, произошедшие с момента последнего полного резервного копирования. Он предлагает компромисс между скоростью копирования и сложностью восстановления данных.
4. Зеркальное резервное копирование. Этот метод создает точную копию данных в реальном времени. Это обеспечивает максимальную актуальность данных, но требует значительных ресурсов и может быть уязвим для тех же угроз, что и оригинальные данные.

Многие из существующих программ для резервного копирования, разработанных в 90-х годах, обладают рядом недостатков:

1. Сложный интерфейс. Пользователи часто сталкиваются с трудностями при настройке и использовании таких программ, что может привести к ошибкам и потере данных.
2. Низкая скорость работы. Старые программы могут не справляться с современными объемами данных, что замедляет процесс резервного копирования и восстановления.
3. Ограниченная функциональность. Многие старые программы не поддерживают современные методы и системы хранения данных, такие как облачные сервисы.
4. Отсутствие обновлений и поддержки. Многие из этих программ больше не обновляются, что делает их уязвимыми к новым

угрозам и несовместимыми с современными операционными системами и аппаратными платформами.

С учетом вышеуказанных проблем, существует явная необходимость в разработке новых программ для резервного копирования данных, которые бы отвечали современным требованиям пользователей. Такие программы должны быть:

1. Простыми в использовании. Интуитивно понятный интерфейс и легкость настройки должны быть приоритетами.
2. Быстрыми и эффективными. Программы должны обеспечивать высокую скорость выполнения задач, минимизируя время, необходимое для резервного копирования и восстановления данных.
3. Совместимыми с различными системами хранения. Поддержка работы с различными носителями, типом файловой структуры носителя.
4. Надежными. Гарантия целостности и сохранности данных должна быть ключевой характеристикой.
5. Обновляемыми и поддерживаемыми. Постоянные обновления и поддержка со стороны разработчиков должны обеспечивать защиту от новых угроз и совместимость с современными системами.

1.2. Постановка задачи

Современные информационные технологии предъявляют высокие требования к сохранности данных. Пользователи, работающие с компьютерными средствами, ежедневно создают и обрабатывают различные файлы, будь то рабочие документы, личные фотографии или важные бизнес-документы. Потеря этих данных может иметь серьезные последствия, что делает процессы их защиты и восстановления особенно актуальными. Цель данного дипломного проекта заключается в разработке программного решения для резервного архивирования данных, которое будет удовлетворять современным требованиям пользователей и обеспечивать надежную защиту данных.

Резервное копирование данных — критически важный процесс для сохранения целостности и доступности информации. Современные пользователи сталкиваются с множеством угроз, таких как аппаратные сбои, вирусные атаки, ошибки пользователей и даже природные катастрофы. Несмотря на наличие большого количества программ для резервного копирования, многие из них устарели и не соответствуют современным стандартам удобства и эффективности. Поэтому существует острая необходимость в создании нового, более современного и удобного программного обеспечения.

Цель дипломного проекта — разработка программы для резервного копирования данных, которая обеспечит:

1. Интуитивно понятный интерфейс. Программа должна быть простой в использовании, понятной для пользователей всех уровней, и обеспечивать легкость настройки.
2. Высокая скорость работы. Программа должна эффективно справляться с современными объемами данных, обеспечивая быстрое создание и восстановление резервных копий.
3. Надежность. Программа должна гарантировать целостность и сохранность данных, защищая их от потерь и повреждений.

4. Гибкость настроек. Пользователи должны иметь возможность настраивать параметры резервного копирования в соответствии с их специфическими потребностями.
5. Совместимость с различными системами хранения данных. Программа должна поддерживать работу с различными видами локальных хранилищ данных.
6. Регулярные обновления и поддержка. Программа должна постоянно обновляться, обеспечивая защиту от новых угроз и совместимость с современными системами.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Исследование существующих решений. Провести анализ современных программ для резервного копирования данных, выявить их сильные и слабые стороны, чтобы учесть это при разработке нового продукта.
2. Определение требований к новой программе. Сформулировать конкретные требования к функциональности, интерфейсу, скорости работы и надежности программы на основе анализа существующих решений и потребностей пользователей.
3. Разработка архитектуры программы. Создать архитектуру программы, которая обеспечит гибкость, масштабируемость и надежность, соответствуя современным стандартам программного обеспечения.
4. Реализация программы. Написать код программы, внедряя все необходимые функции, обеспечивая высокую скорость и надежность работы.
5. Тестирование и отладка. Провести всестороннее тестирование программы для выявления и устранения возможных ошибок и недостатков.

6. Документирование. Подготовить полную документацию по программе, включая руководство пользователя и техническую документацию, чтобы облегчить её использование и поддержку.
7. Оценка производительности и надежности. Провести сравнительный анализ производительности и надежности программы в сравнении с существующими решениями для подтверждения её преимуществ.

1.3. Анализ существующих программных решений

В ходе проектирования дипломного проекта был произведен анализ имеющихся программных решений для более подробного представления способов реализации данной задачи. Ниже будут представлены некоторые из них.

WinRAR — это мощный инструмент для управления архивами, который поддерживает большинство популярных форматов архивов, включая RAR и ZIP. Он позволяет создавать, распаковывать и управлять архивами, а также предлагает функции для восстановления поврежденных архивов и защиты их паролем. Основное окно программы продемонстрировано на рисунке 1.1.

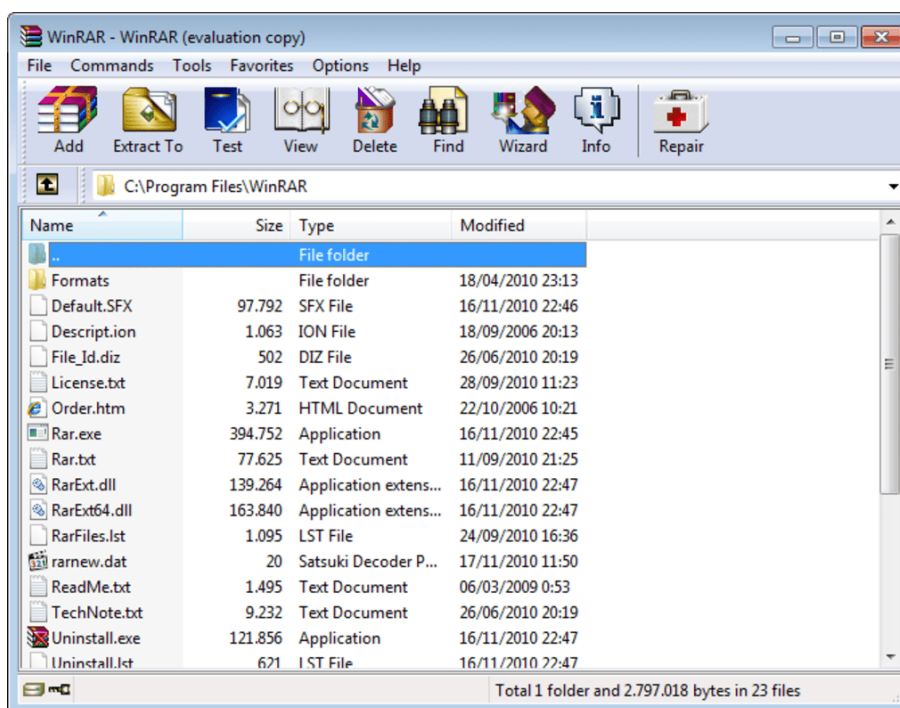


Рисунок 1.1. WinRAR.

Сильные стороны:

- Поддержка множества форматов архивов.
- Высокая степень сжатия данных.
- Функции для восстановления поврежденных архивов.
- Возможность защиты архивов паролем.

Слабые стороны:

- Платная лицензия после пробного периода.

- Менее интуитивный интерфейс для начинающих пользователей.
- Низкая скорость работы программы.

7-Zip — это бесплатное программное обеспечение с открытым исходным кодом для создания и управления архивами. Он поддерживает множество форматов архивов и известен своей высокой степенью сжатия формата 7z. Основное окно программы продемонстрировано на рисунке 1.2.

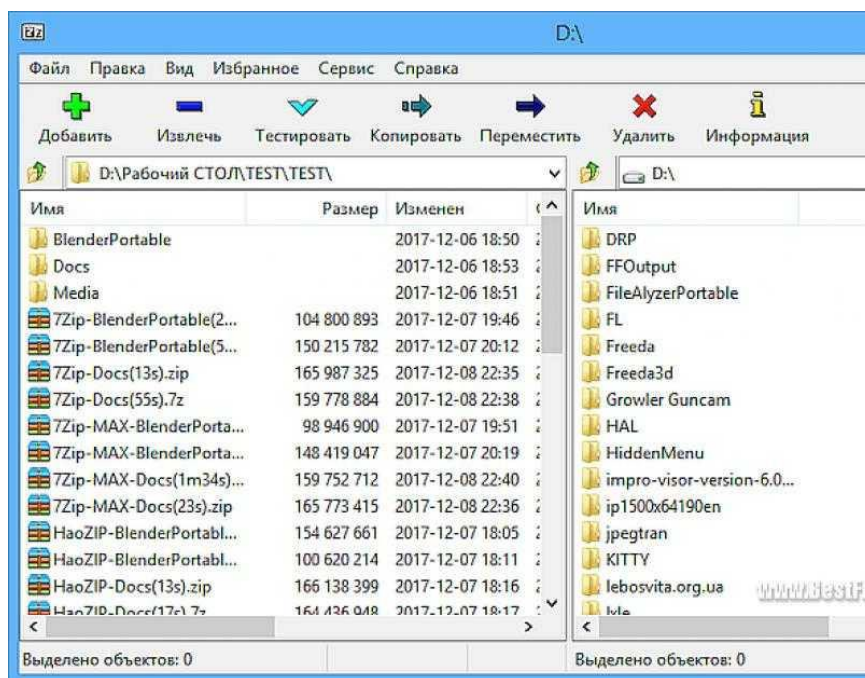


Рисунок 1.2. 7Zip.

Сильные стороны:

- Бесплатный и с открытым исходным кодом.
- Высокая степень сжатия, особенно для формата 7z.
- Поддержка большого количества форматов архивов.
- Простота в использовании.
- Высокая скорость работы программы.

Слабые стороны:

- Менее привлекательный интерфейс.
- Отсутствие некоторых продвинутых функций, присутствующих в платных аналогах.

- Плохая работа с файлами, имеющим длинное имя (более 256 ASCII символов)

PeaZip — это бесплатная утилита для управления архивами, поддерживающая более 200 форматов архивов. Она предлагает интуитивно понятный интерфейс и множество дополнительных функций, таких как шифрование и разбивка архивов на части. Основное окно программы продемонстрировано на рисунке 1.3.

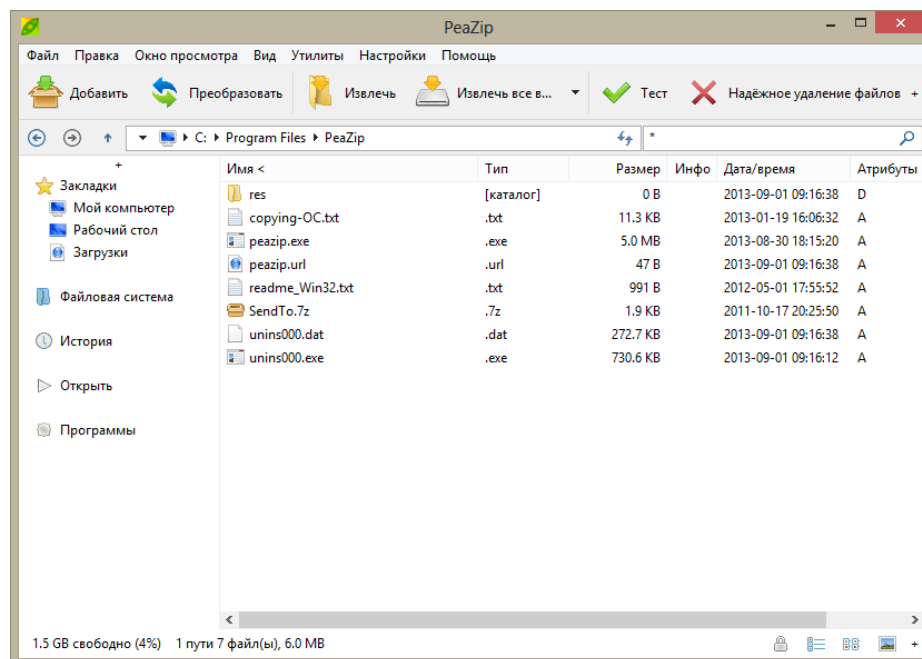


Рисунок 1.3. PeaZip.

Сильные стороны:

- Бесплатное использование.
- Поддержка большого количества форматов архивов.
- Интуитивно понятный интерфейс.
- Дополнительные функции, такие как шифрование и разбивка на части.

Слабые стороны:

- Скорость работы может быть ниже по сравнению с конкурентами.
- Некоторые функции могут быть сложными для новичков.
- Отсутствие возможности обезопасить архив паролем

WinZip — это коммерческое программное обеспечение для создания и управления архивами, широко известное благодаря поддержке ZIP-формата. Оно предлагает множество дополнительных функций, включая интеграцию с облачными сервисами и шифрование данных. Основное окно программы продемонстрировано на рисунке 1.4.

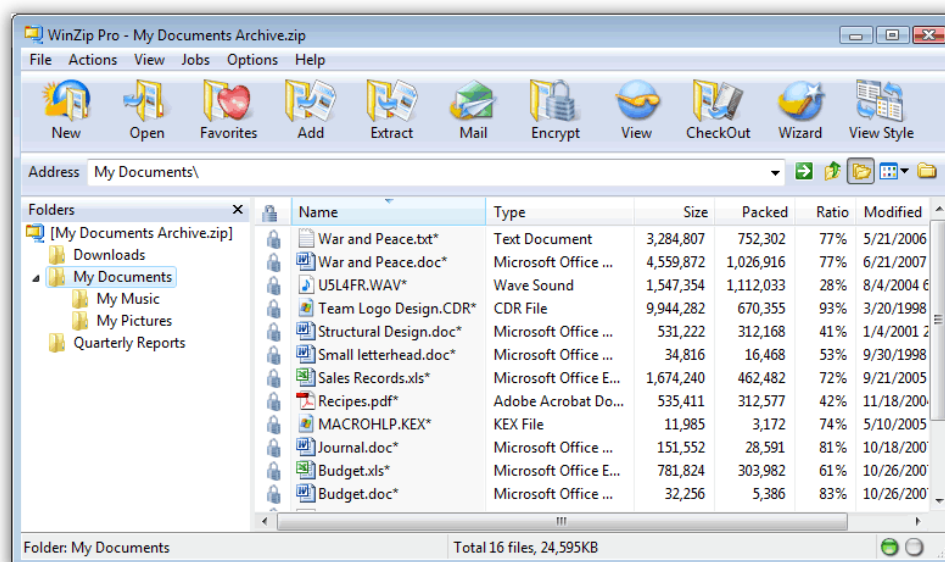


Рисунок 1.4. WinZip.

Сильные стороны:

- Поддержка облачных сервисов.
- Удобный и интуитивно понятный интерфейс.
- Высокая степень сжатия и шифрование данных.
- Расширенные функции для работы с архивами.

Слабые стороны:

- Платная лицензия.
- Ограниченная поддержка форматов по сравнению с другими программами.
- Низкая скорость архивации данных

BandiZip — это бесплатный и легкий инструмент для создания и управления архивами. Он поддерживает множество форматов архивов и

известен своей высокой скоростью работы и простым интерфейсом. Основное окно программы продемонстрировано на рисунке 1.5.

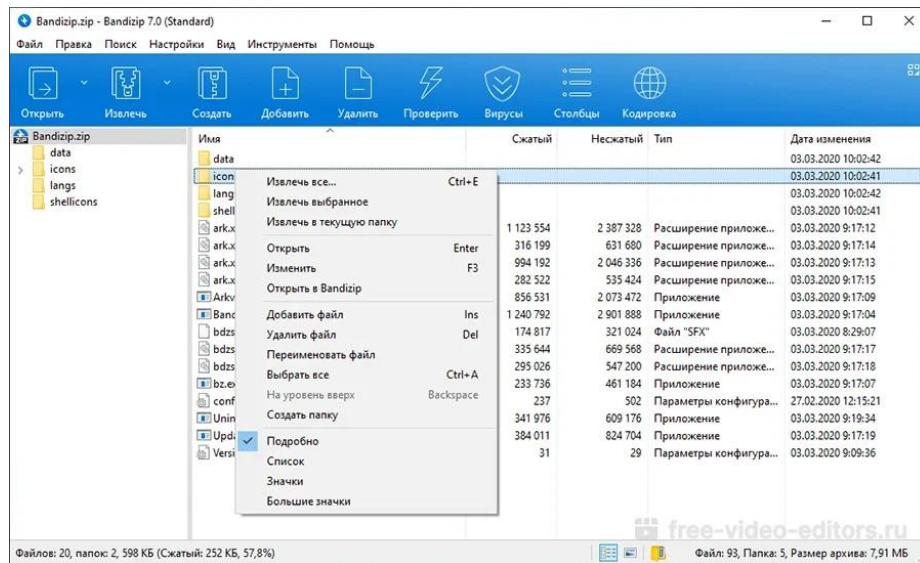


Рисунок 1.5. BandiZip.

Сильные стороны:

- Бесплатное использование.
- Высокая скорость работы.
- Поддержка множества форматов архивов.
- Интуитивно понятный интерфейс.

Слабые стороны:

- Ограниченные функции по сравнению с более продвинутыми программами.
- Периодические обновления могут содержать рекламу.

2. ПРОЕКТНАЯ ЧАСТЬ

2.1. Описание алгоритмов решения

При постановке задачи на дипломный проект была поставлена совокупность точно заданных правил решения минимального набора задач.

Для создания программы резервного копирования данных необходимо было реализовать следующие основные алгоритмы:

- Авторизация
- Регистрация
- Изменение темы оформления
- Настройка архивации данных
- Создание профиля архивации
- Добавление данных в профиль архивации
- Сброс настроек пользователя
- Смена пароля пользователя

Ниже, на рисунке 2.1, изображена use-case диаграмма, наглядно дающая понимание о взаимодействии пользователя и администратора с программой «Васкур».

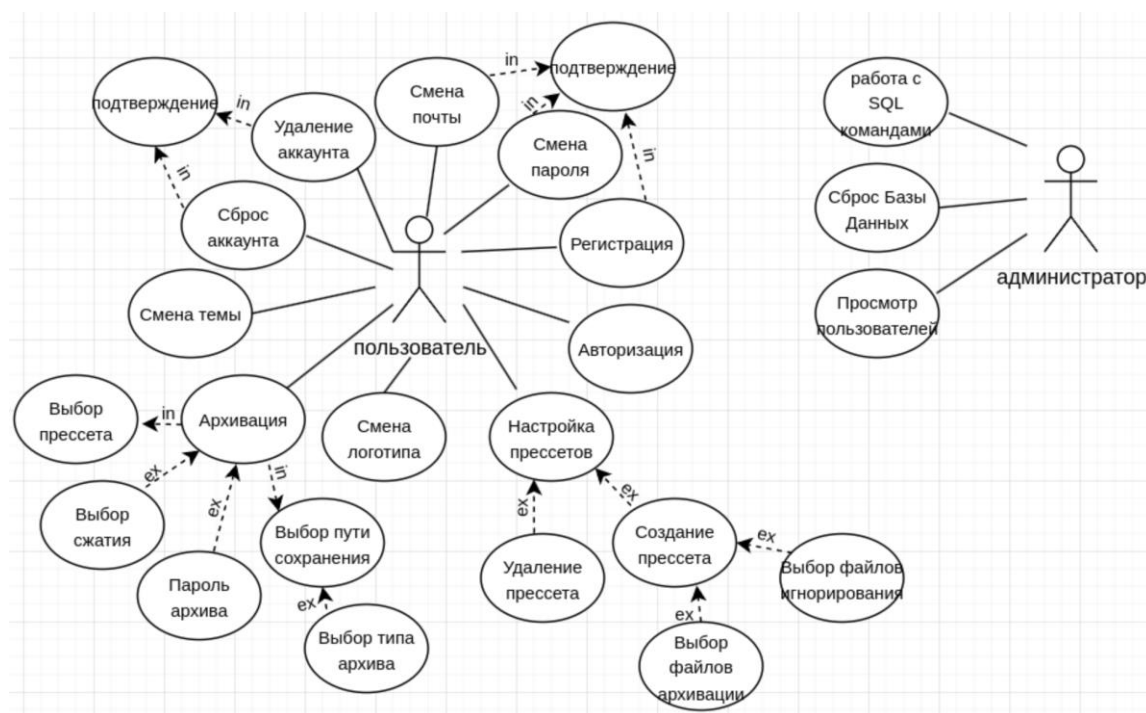


Рисунок 2.1. Use-case диаграмма.

Алгоритм Регистрации:

1. Начало
2. Открытие окно Регистрации
3. Ввод данных (Почта, логин)
4. Проверка данных на корректность
5. Отправка кода подтверждения на почту
6. Подтверждения почты клиенты
7. Ввода данных (Пароль)
8. Проверка сложности пароля
9. Регистрация клиента (Создание записи в базе данных)
- 10.Конец

Алгоритм Авторизации:

1. Начало
2. Открытие окно Авторизации
3. Ввод данных (Логин, пароль)
4. Проверка данных на корректность
5. Авторизация пользователя (Создание сессии)
6. Конец

Алгоритм Изменения темы оформления:

1. Начало
2. Авторизация пользователя
3. Переход в настройки приложения
4. Загрузка всех тем из базы данных
5. Выбор подходящей темы
6. Обновление интерфейса
7. Конец

Алгоритм Настройки архивации данных:

1. Начало
2. Авторизация пользователя
3. Создание профиля архивации

4. Выбор профиля архивации
5. Выбор пути сохранения
6. Выбор типа архива
7. Выбор сжатия архива
8. Выбор пароля к архиву (нет/есть)
9. Конец

Алгоритм Создания профиля архивации:

1. Начало
2. Авторизация пользователя
3. Запуск программы для редактирования профилей архивации
4. Создание нового профиля
5. Ввод названия профиля
6. Сохранение изменений
7. Конец

Алгоритм Добавление данных в профиль архивации:

1. Начало
2. Авторизация пользователя
3. Запуск программы для редактирования профилей архивации
4. Переход в созданный ранее профиль
5. Добавление локальных путей для архивации
6. Добавление локальных путей для игнорирования
7. Сохранение изменений
8. Конец

Алгоритм Сброса настроек пользователя:

1. Начало
2. Авторизация пользователя
3. Переход в меню управления аккаунтом пользователя
4. Нажатие на кнопку «Сброс аккаунта»
5. Подтверждение перед началом сброса
6. Подтверждение паролем от аккаунта

7. Сброс аккаунта
8. Применение изменений (Изменение данных в базе данных)
9. Конец

Алгоритм Смены пароля пользователя:

1. Начало
2. Авторизация пользователя
3. Переход в меню управления аккаунтом пользователя
4. Нажатие на кнопку «Смена пароля»
5. Отправка кода подтверждения на почту клиента
6. Подтверждение аккаунта с помощью кода-подтверждения
7. Ввод нового пароля
8. Подтверждение изменений
9. Сохранение
10. Конец

2.2. Обоснование выбора средств разработки

Для успешной реализации дипломного проекта по разработке программы для резервного архивирования данных важно выбрать подходящий инструментарий. В данном проекте использовались следующие инструменты: база данных MySQL, язык программирования Python, интегрированная среда разработки (IDE) JetBrains PyCharm, редактор кода Visual Studio Code и библиотека для создания графического интерфейса Flet. В этой главе мы рассмотрим плюсы и необходимые свойства каждого из этих инструментов, обосновывая их выбор.

MySQL - свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB.

В качестве базы данных для дипломного проекта была выбрана СУБД MySQL, за счет своих сильных сторон, описанных ниже:

- **Высокая производительность:** MySQL обеспечивает быструю обработку запросов, что важно для эффективного управления данными.
- **Надежность:** MySQL зарекомендовала себя как стабильная и надежная система управления базами данных, что критически важно для обеспечения целостности данных.
- **Широкая поддержка:** MySQL поддерживается большинством языков программирования и платформ, что делает её универсальным выбором.
- **Сообщество и документация:** Сильное сообщество пользователей и обширная документация упрощают процесс разработки и устранения ошибок.
- **Масштабируемость:** MySQL позволяет легко масштабировать приложения, что полезно при увеличении объема данных и нагрузки.

- **Безопасность:** Возможность настройки прав доступа и использования шифрования для защиты данных.
- **Совместимость с Python:** Отличная интеграция с Python через библиотеки, такие как MySQL Connector и SQLAlchemy.

Python - высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

Основной язык программирования Python был выбран, так как моя специализация проходит именно по нему, а также за счет его сильных сторон в данном проекте, описанных ниже:

- **Простота и читабельность кода:** Python известен своей лаконичностью и понятностью, что ускоряет процесс разработки и упрощает поддержку кода.
- **Богатая экосистема библиотек:** Большое количество доступных библиотек и фреймворков для различных задач, включая обработку данных, сетевые запросы и создание GUI.
- **Широкое сообщество:** Активное сообщество разработчиков, большое количество документации и обучающих материалов.
- **Кроссплатформенность:** Возможность разработки и выполнения программ на различных операционных системах без значительных изменений в коде.
- **Совместимость с базами данных:** Поддержка интеграции с различными системами управления базами данных.
- **Модульность и расширяемость:** Возможность легко добавлять новые функции и модули.
- **Поддержка современных стандартов:** Соответствие последним версиям стандартов и возможность использования современных библиотек и инструментов.

JetBrains PyCharm - кроссплатформенная интегрированная среда разработки для языка программирования Python, разработанная компанией JetBrains на основе IntelliJ IDEA. Предоставляет пользователю комплекс средств для написания кода и визуальный отладчик.

Интегрированная среда разработки JetBrains PyCharm, была выбрана, так как это среда, которая лучше всего интегрируется с языком программирования Python:

- Мощные инструменты отладки: PyCharm предоставляет инструменты для отладки, что помогает быстро выявлять и устранять ошибки.
- Умный редактор кода: Поддержка автодополнения, рефакторинга кода и анализа синтаксиса улучшает производительность и качество кода.
- Интеграция с системами контроля версий: Поддержка Git, SVN и других систем контроля версий облегчает управление проектом.
- Поддержка веб-разработки: Встроенные инструменты для разработки веб-приложений и работы с популярными фреймворками.
- Интеграция с Python: Полная поддержка Python, включая виртуальные среды и пакеты.
- Многоплатформенность: Возможность работы на различных операционных системах.
- Поддержка работы с базами данных: Встроенные инструменты для работы с MySQL и другими базами данных.

Visual Studio Code - текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга.

Дополнительный редактор кода Visual Studio Code, был выбран для незначительной разработки web-страниц на языках HTML, CSS и JavaScript, за счет своей универсальности и простоты работы с данным редактором:

- Бесплатный и кроссплатформенный: Visual Studio Code доступен бесплатно и поддерживает работу на разных операционных системах.
- Многочисленные расширения: Большое количество доступных расширений для поддержки различных языков программирования и инструментов.
- Интеграция с Git: Встроенная поддержка Git для управления версиями кода.
- Легкость и скорость: Высокая производительность и небольшие системные требования.
- Настраиваемость: Возможность настройки и персонализации интерфейса и функционала под потребности разработчика.

Библиотека для разработки GUI интерфейса на языке программирования Python - Flet , был выбран за счет:

- Простота использования: Flet предоставляет простой API для создания пользовательских интерфейсов, что ускоряет процесс разработки.
- Кроссплатформенность: Возможность создания приложений, работающих на различных операционных системах.
- Интеграция с Python: Полная совместимость с Python, что упрощает разработку и интеграцию с другими модулями и библиотеками.
- Гибкость и расширяемость: Возможность легко настраивать и расширять интерфейсы.
- Современные компоненты: Поддержка современных UI-компонентов и возможностей.

- Поддержка адаптивного дизайна: Возможность создания интерфейсов, которые корректно отображаются на различных устройствах и экранах.
- Высокая скорость работы: Данная библиотека была написана на языке программирования C++, который славится оптимизацией и скоростью работы.

2.3. Описание ER-модели