ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ «КОЛЛЕДЖ АВТОМАТИЗАЦИИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ № 20»

**ОТЧЕТ   
О ПРОХОЖДЕНИИ ПРАКТИКИ**

учебная практика

(наименование вида практики)

Котовой Екатерины Евгеньевны

(Ф. И. О. обучающегося)

ПМ.02 Осуществление интеграции программных модулей

курс 2

группа ИСП212

формы обучения очная

специальность 09.02.07 Информационные системы и программирование

Место прохождения практики

ГБПОУ КАИТ № 20, г. Москва, ул. 1-я Мясниковская, д. 16

Срок прохождения практики с «21» апреля 2025 г. по «26» апреля 2025 г.

В объеме 36 часа

Отчёт принят

/ Лисавина А.В. / преподаватель

(дата, должность, Ф.И.О руководителя практики)

г. Москва, 2025

**СОДЕРЖАНИЕ**

[ВВОДНАЯ ЧАСТЬ 4](#_Toc196051336)

[КРАТКОЕ ОПИСАНИЕ ПРЕДПРИЯТИЯ И ЕГО ДЕЯТЕЛЬНОСТИ 5](#_Toc196051337)

[ОПИСАНИЕ ВЫПОЛНЕННОГО ИНДИВИДУАЛЬНОГО ЗАДАНИЯ 6](#_Toc196051338)

[ВЫВОД 7](#_Toc196051339)

# ВВОДНАЯ ЧАСТЬ

Я, Котова Екатерина Евгеньевна, проходила учебную практику на базе колледжа ГБПОУ КАИТ №20.

Моим индивидуальным заданием было создание use-case и ER-диаграммы, а также разработка базы данных для учета расходов и доходов. Кроме того, мне необходимо было создать приложение, которое позволяет добавлять, редактировать и удалять финансовые транзакции, а также просматривать отчеты о финансовой деятельности.

Для разработки РЭ необходимо проанализировать все его функции, а затем создать структуру документа, включающую разделы и перечень требований.

В рамках тестирования сценариев выполняется анализ приложения, выявляются критические сценарии использования, и разрабатывается план тестирования, который включает в себя последовательные шаги и ожидаемые результаты.

# КРАТКОЕ ОПИСАНИЕ ПРЕДПРИЯТИЯ И ЕГО ДЕЯТЕЛЬНОСТИ

# [Государственное бюджетное профессиональное образовательное учреждение города Москвы «Колледж автоматизации и информационных технологий № 20»](https://kait20.mskobr.ru/)

# КАИТ № 20 наладил прочные связи с ведущими организациями Москвы, ежегодно подготавливая более 600 конкурентоспособных специалистов по множеству направлений для экономики столицы. В колледже активно развивается Студенческий Спортивный клуб «Драйв», Волонтерский отряд «Команда доброты», Военно-патриотический клуб и поисковый отряд «Дозор», музей «Страницы Великой Истории», колледжный театр «АйТиАрт», Студенческий медиацентр. Создан проект «ПРОдвижение», направленный на профориентацию школьников в каникулярный период.

В колледже функционирует **6 отделений**, каждое из которых предлагает программы обучения, соответствующие современным требованиям рынка труда. Реализуются **программы по 4 направлениям подготовки:**

* ИТ-специальности;
* Транспортные системы и технологии;
* Производство и инженерные технологии;
* Креативная индустрия.

В КАИТ № 20 проводится**бесплатное очное обучение с элементами дистанционных образовательных технологий для лиц с инвалидностью и ОВЗ.** На базе колледжа работает **автошкола**с собственной площадкой, эстакадой и кабинетом автосимулятора. Для студентов КАИТ № 20 действуют скидки на обучение.

# ОПИСАНИЕ ВЫПОЛНЕННОГО ИНДИВИДУАЛЬНОГО ЗАДАНИЯ

В рамках проекта была разработана Use-Case диаграмма и модель ER-диаграммы базы данных. Эти инструменты помогли визуализировать бизнес-процессы и связи между сущностями, что стало основой для структурирования будущей базы данных. (Рис.1, 2)

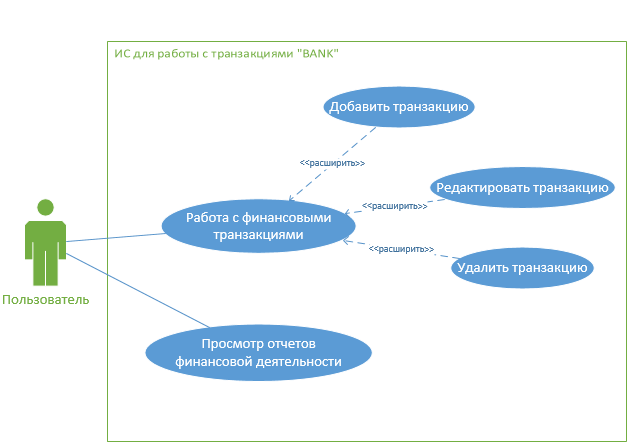


Рисунок 1 – use-case диаграмма

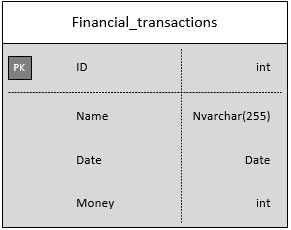


Рисунок 2 – Er-диаграмма

Была создана база данных с именем BANK, в которой была разработана таблица Financial\_transactions. Таблица включает в себя различные столбцы, отражающие ключевые параметры финансовых операций, что обеспечило эффективное хранение и управление данными. (Рис. 3)

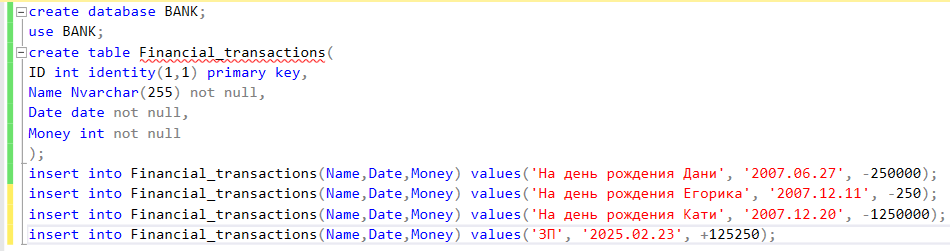


Рисунок 3 – Создание таблицы Financial\_ transactions

В рамках разработки была установлена связь между кодом приложения и самой базой данных. Для этого был добавлен объект класса BANKEntities, который представляет контекст базы данных. Для управления этим контекстом был реализован метод GetContext(). Данный метод проверяет наличие существующего экземпляра \_context: если он уже создан, метод возвращает его; в противном случае, происходит пересоздание контекста. Это обеспечивает эффективное управление подключением к базе данных и оптимизирует работу приложения. (Рис. 4)

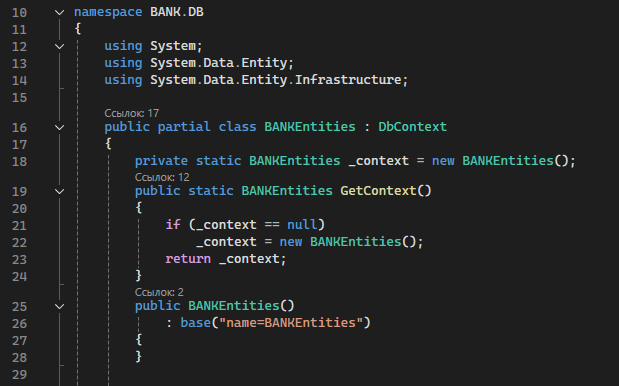


Рисунок 4 – Контекст базы данных

В приложении было разработано главное меню, содержащее две кнопки: «Транзакции» и «Фин. отчёт». Эти элементы интерфейса обеспечивают удобный переход между формами, облегчая навигацию для пользователей. (Рис. 5, 6)

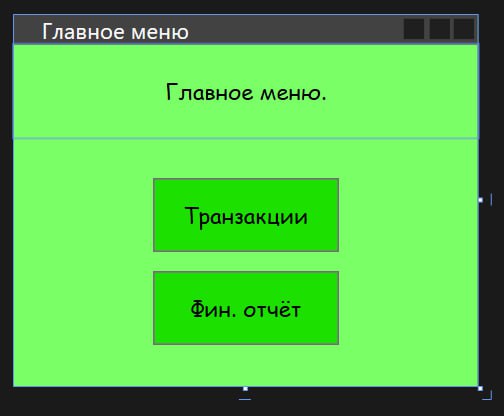


Рисунок 5 – Главное меню

В коде главного меню были разработаны два метода обработки кликов для кнопок, отвечающих за переход к формам «Транзакции» и «Фин. отчет». В каждом из этих методов создается объект соответствующего класса формы. После этого происходит отключение видимости главного меню, что позволяет сосредоточить внимание пользователя на выбранной задаче. Затем открывается диалоговое окно с необходимой формой, обеспечивая пользователю доступ к функционалу. По завершении работы с формой, видимость главного меню восстанавливается, что позволяет пользователю легко вернуться к основному интерфейсу приложения. (Рис. 6)

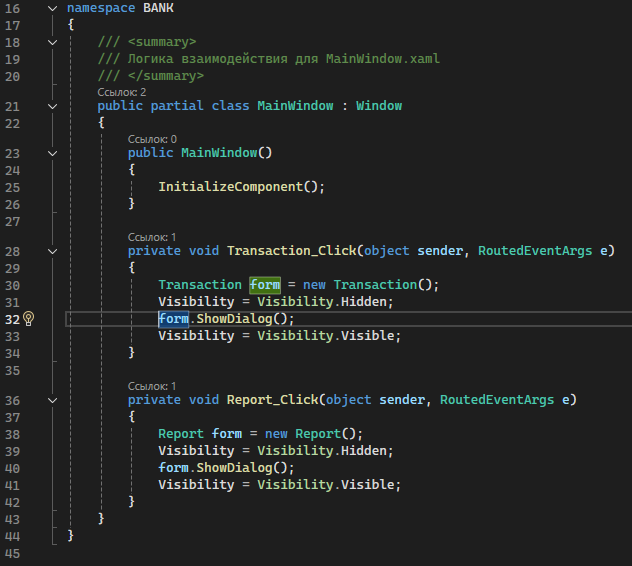


Рисунок 6 – Код для перехода на другие формы по кнопкам

*Транзакции*

На странице транзакций внедрено отображение информации из базы данных, что позволяет пользователям удобно просматривать и работать с транзакциями. Также были добавлены формы для ввода новых данных: «Описание» (для ввода события полученных или потраченных денег), «Сумма денег» (для ввода числа потраченных или полученных денег), календарь (для выбора даты события) и кнопки для отправки в базу данных: «Добавить транзакцию», «Редактировать транзакцию» и «Удалить транзакцию». Это делает процесс управления финансовыми операциями интуитивно понятным и доступным. (Рис. 7)

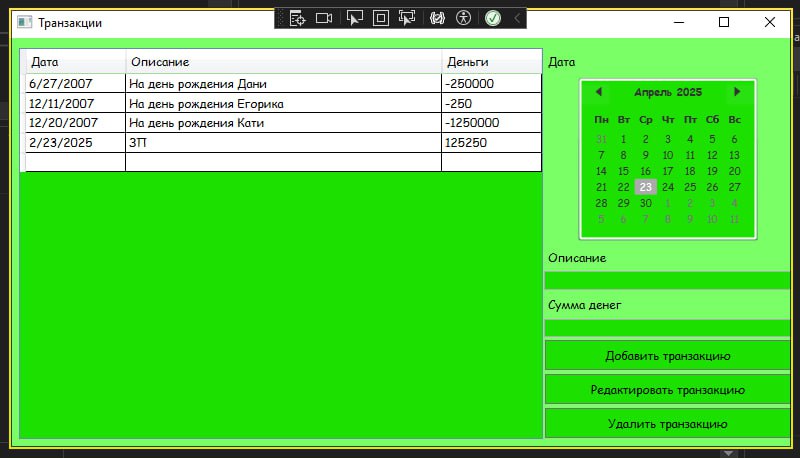
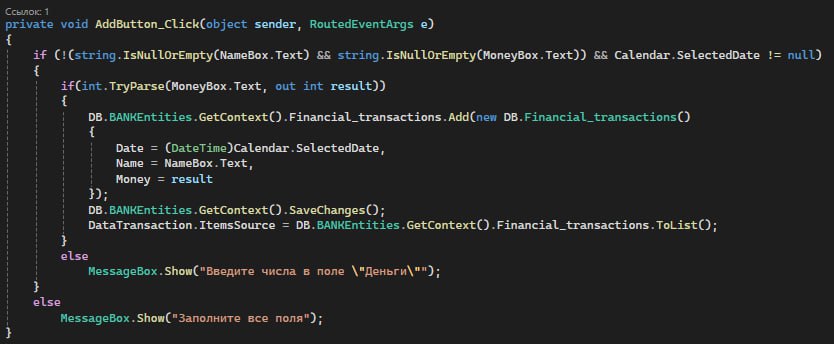


Рисунок 7 – Окно транзакций

В рамках реализации функционала добавления транзакции был разработан метод, который выполняет несколько ключевых шагов. Сначала происходит проверка заполненности всех обязательных полей формы. Это гарантирует, что пользователь предоставил всю необходимую информацию для успешного добавления транзакции. Затем осуществляется валидация данных, в частности, проверка возможности преобразования значения поля "Сумма денег" из строкового формата в числовой. Если какое-либо из условий не выполняется, метод генерирует соответствующее сообщение об ошибке, информируя пользователя о необходимости исправления введенных данных. При соблюдении всех условий метод обращается к таблице транзакций в базе данных и добавляет новый элемент, представляющий собой класс транзакций. После успешного добавления данных происходит сохранение изменений в базе и обновление отображаемого в DataGrid списка транзакций, что позволяет пользователю видеть актуальную информацию. (Рис. 8)

Рисунок 8 – Добавление транзакций в базу данных

В процессе для редактирования транзакции реализован метод, который выполняет несколько последовательных шагов для обеспечения корректности данных. Сначала осуществляется проверка заполненности всех обязательных полей формы, а также подтверждение выбора элемента из базы данных. Это гарантирует, что пользователь выбрал транзакцию, которую необходимо редактировать. Затем выполняется валидация данных, включая проверку возможности преобразования значения поля "Сумма денег" из строкового формата в числовой. В случае, если какое-либо из условий не выполняется, метод генерирует сообщение об ошибке, информируя пользователя о необходимости исправления введенных данных. Если все условия соблюдены, создается переменная item, значение которой представляет собой выбранный элемент, приведенный к классу транзакций. После этого происходит обновление данных, включая дату, описание и сумму в объекте item. В завершение, все изменения сохраняются в базе данных, и происходит обновление отображаемого списка транзакций в DataGrid, что позволяет пользователю видеть актуальную информацию. (Рис. 9)

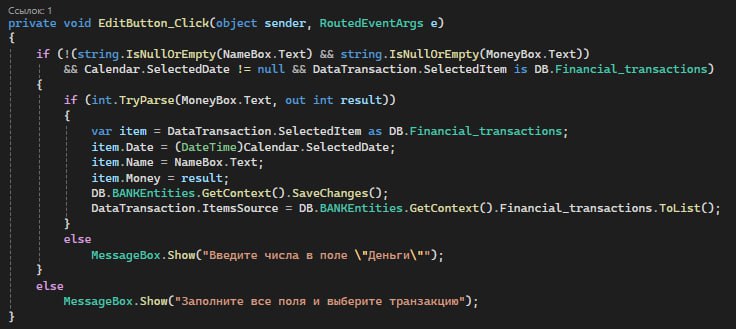


Рисунок 9 – Редактирование транзакций

В рамках реализации функционала удаления транзакции был разработан метод, который выполняет несколько ключевых шагов для обеспечения корректности операции. Сначала осуществляется проверка, выбран ли элемент в DataGrid. Это гарантирует, что пользователь действительно намерен удалить конкретную транзакцию. После подтверждения выбора, метод обращается к таблице транзакций в базе данных и удаляет соответствующий элемент, приведенный к классу транзакций. Затем изменения сохраняются в базе данных, что обеспечивает актуальность информации. В завершение, происходит обновление DataGrid, что позволяет пользователю видеть обновленный список транзакций без удаленного элемента. (Рис. 10)

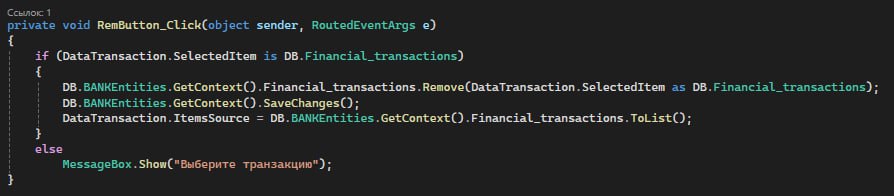


Рисунок 10 – Удаление транзакций

В процессе работы с транзакциями реализован метод выбора элемента в DataGrid, который выполняет несколько ключевых шагов для корректного отображения информации. Сначала осуществляется проверка возможности преобразования выбранного элемента в класс транзакций. Это гарантирует, что данные, с которыми мы собираемся работать, соответствуют необходимому формату. После успешной проверки создается переменная item, значение которой представляет собой преобразованный элемент, приведенный к классу транзакций. Затем происходит заполнение полей формы: значения для полей "Сумма денег" и "Описание" извлекаются из объекта item и вводятся в соответствующие поля ввода. В завершение, устанавливается отображение даты в календаре на нужную дату, что позволяет пользователю легко видеть и редактировать информацию о выбранной транзакции. (Рис. 11)

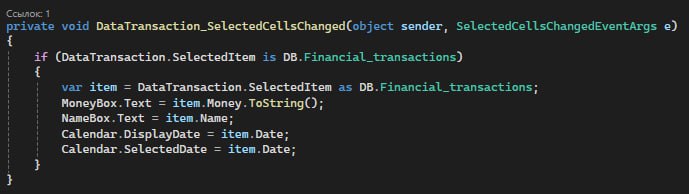


Рисунок 11 – Выбор транзакций в DataGrid

*Фин. отчёт*

На странице с финансовыми отчетами пользователям предоставляется возможность выбора категорий «Траты» и «Прибыль». Это позволяет более детально ознакомиться с произведёнными расходами и полученными доходами, что способствует лучшему анализу финансовой деятельности. (Рис. 12)

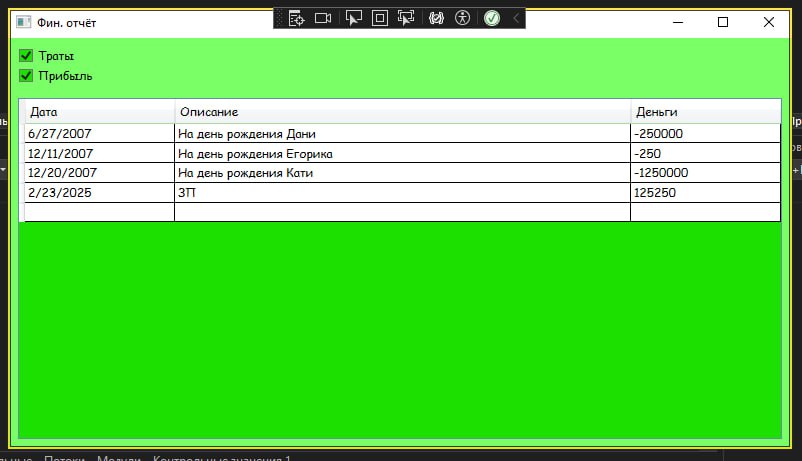


Рисунок 12 – Окно отчета транзакций

В процессе формирования отчета реализована функция обновления DataGrid, а также управление состоянием чекбоксов для категорий "Траты" и "Прибыль". При установке или снятии галочек в соответствующих чекбоксах происходит вызов метода UpdateDB(), который отвечает за обновление данных в DataGrid. В этом методе создается пустой список транзакций, который будет заполняться в зависимости от состояния чекбоксов. Если чекбокс "Траты" активен, в список добавляются транзакции с положительными значениями. В противном случае этот блок пропускается. Аналогичная проверка выполняется для чекбокса "Прибыль": если галочка установлена, в список добавляются соответствующие транзакции. Если же чекбокс отсутствует, то возвращается значение, указывающее на то, что галочка считается установленной. Важно отметить, что при проверке состояния чекбокса "Прибыль" учитывается возможность его отсутствия. Если чекбокс не существует, метод возвращает значение, что галочка присутствует. В противном случае, состояние чекбокса приводится к булевому значению. После завершения всех проверок и добавления элементов в список, данные переносятся в DataGrid, что позволяет пользователю видеть актуальную информацию о транзакциях. (Рис. 13)

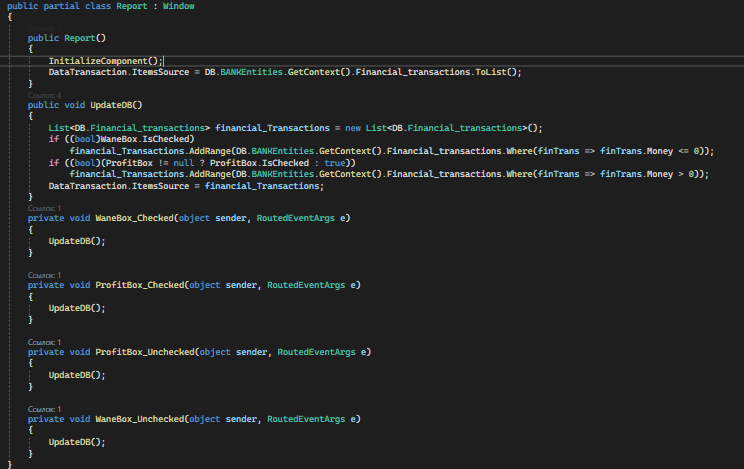


Рисунок 13 – Код вывода таблицы отчета с выбором «Траты» или «Прибыль»

Было разработано и проведено пять тестовых сценариев, в рамках которых тестировалась функциональность программы. (Рис. 14)



Рисунок 14 – Тестовый сценарий 1

По завершении проекта было составлено руководство по эксплуатации, предназначенное для пользователей. Этот документ содержит необходимые инструкции по использованию приложения, что обеспечивает легкость в освоении всех функций. (Рис. 15)

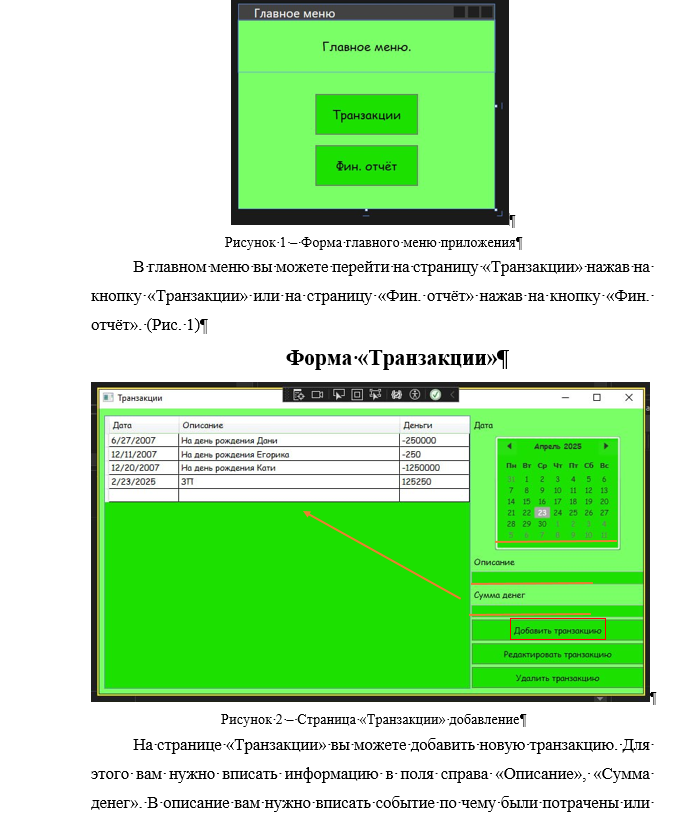


Рисунок 15 – Страница документа руководства по эксплуатации

# ВЫВОД

Создана база данных на платформе MS SQL Server, включающая таблицу для транзакций. Приложение было разработано на языке C# с использованием WPF и обладает функционалом для просмотра отчетов, добавления, редактирования и удаления транзакций.

В результате выполнения проекта была успешно реализована база данных и приложение, позволяющее удобно управлять транзакциями. Этот опыт стал ценным шагом в освоении процесса разработки баз данных и создания приложений для их обработки.

В завершение работы были подготовлены 2 документа: руководство пользователя и пять тестовых сценариев.