Содержание

|  |  |
| --- | --- |
| Введение | 3 |
| 1 Назначение и цели разработки | 4 |
| 2 Разработка технического проекта на основе анализа требований | 5 |
| 2.1 Определение спецификаций программного обеспечения | 5 |
| 2.2 Проектирование модели данных и диаграммы классов | 10 |
| 2.3 Проектирование диаграммы развертывания | 18 |
| 3 Реализация | 19 |
| 3.1 Обоснование выбора средств разработки | 19 |
| 3.2 Разработка базы данных в среде СУБД | 20 |
| 3.3 Описание программных модулей | 22 |
| 4 Тестирование программных модулей | 26 |
| 4.1 Модульное тестирование | 26 |
| 4.2 Интеграционное тестирование | 27 |
| 5 Эксплуатационная документация | 28 |
| 5.1 Руководство пользователя | 28 |
| Заключение | 29 |
| Список использованных источников | 30 |
| Приложение А Техническое задание. Требования к программным модулям | 32 |
| Приложение Б Программный код | 36 |
| Приложение В Формы выходных документов | 38 |
| Приложение Г Скрипт базы данных | 39 |
| Приложение Д Тестирование | 41 |

Введение

В современных условиях эффективное управление представляет собой ценный ресурс организации, наряду с финансовыми, материальными, человеческими и другими ресурсами. Следовательно, повышение эффективности управленческой деятельности становится одним из направлений совершенствования деятельности предприятия в целом.

Наиболее очевидным способом повышения эффективности трудового процесса является его автоматизация. Бурное развитие информационных компьютерных технологий, совершенствование технической платформы и появление новых классов программных продуктов привело в наши дни к изменению подходов к автоматизации управления предприятиями монтажа специализированного оборудования.

Компании, производящие специализированное оборудование и его установку, становятся все более популярны в настоящее время. Многие начинающие предприниматели открывают свой бизнес именно в сфере услуг. Поэтому возрос спрос на информационные системы, автоматизирующие процессы и задачи, характерные для данного вида деятельности.

Обрабатывать большие объемы данных без использования электронно-вычислительных машин (ЭВМ) довольно сложно. Наиболее очевидным способом повышения эффективности трудового процесса является его автоматизация.

Учитывая все перечисленное, перед нами встает задача разработки информационной системы в целях ускорения, повышения эффективности и удобства процесса учета произведенных работ на объектах.

Дипломный проект включает в себя проектирование и разработку базы данных и серверного компонента в рамках информационной системы, предназначенной для автоматизации фиксирования произведенных работ.

1 Назначение и цели разработки

Заданием предусмотрена разработка серверного компонента информационной системы фотофиксации «FixPhoto». Разработанный серверный компонент предназначен для автоматизации рабочих мест менеджера и специалиста по монтажу в компании ООО «Зодиак-Электро», а также для сбора, обработки и хранения информации о сотрудниках организации, объектах, произведенных работ.

Автоматизация позволит получить следующие преимущества по сравнению с нынешней реализацией данных задач:

- быстрое формирование отчетов по выполненным работам за выбранный менеджером период;

- быстрая фиксация выполненных работ специалистом;

- ускорение процесса обновления информации об объектах компании.

Данная система позволит максимально сократить присутствие человеческого фактора в ряде важных задач, таких как добавление нового сотрудника, обмен фотографиями выполненных работ, формирование отчетности, учет сотрудников организации.

2 Разработка технического проекта на основе анализа требований

2.1 Определение спецификаций программного обеспечения

Рассмотрим определение вариантов использования (прецедентов).

Система «FixPhoto» требуется прежде всего следующим заинтересованным лицам:

- менеджер компании;

- специалист по монтажу.

На начальном этапе проектирования системы можно ограничиться двумя важными ролями действующих лиц: менеджер и специалист по монтажу. Соответственно основные прецеденты (варианты использования) для разрабатываемой системы следующие.

Прецеденты для менеджера:

- П1 – создать задачу;

- П2 – зарегистрировать пользователя;

Прецеденты для специалиста по монтажу:

- П3 – выполнить задачу.

Диаграмма вариантов использования, созданная средствами MS Visio, для проектируемой системы представлена на рисунке 1.

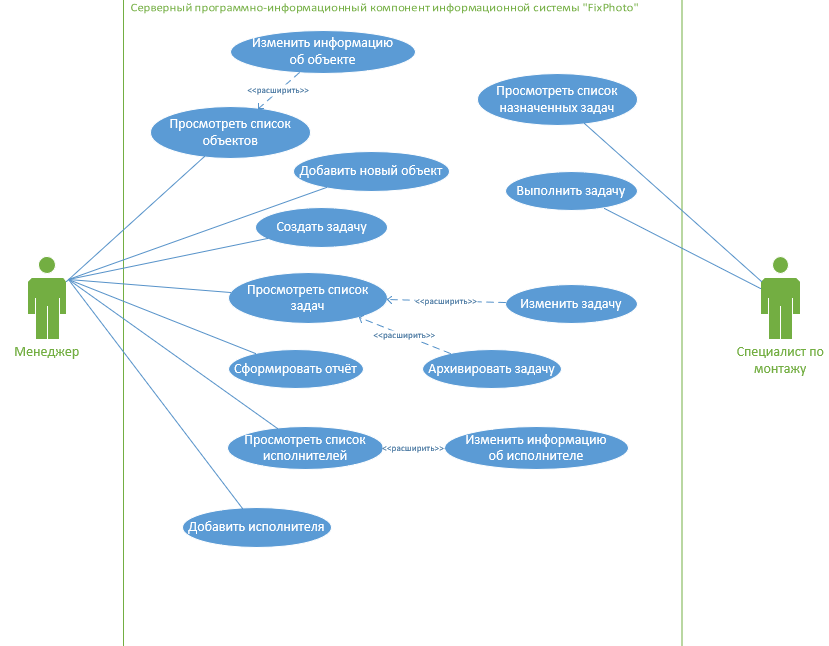


Рисунок 1 – Диаграмма вариантов использования серверного компонента информационной системы “FixPhoto”

В таблице 1 представлено описание главного раздела сценария прецедента (варианта использования) «Создать задачу» [10].

Таблица 1 - Главный раздел сценария «Создать задачу»

|  |  |
| --- | --- |
| Вариант использования | Создать задачу |
| Актеры | Менеджер |
| Краткое описание | Менеджер создает задачу на заданный объект компании |
| Цель | Успешно создать задачу |
| Тип | Базовый |
| Ссылки на другие варианты использования | Включает в себя создание объекта при его отсутствии в базе данных |

В таблице 2 описана последовательность действий менеджера, приводящая к успешному выполнению прецедента (варианта использования) «Создать задачу».

Таблица 2 - Сценарий успешного выполнения варианта использования «Создать задачу»

|  |  |
| --- | --- |
| Действия актеров | Отклик системы |
| 1.Менеджер создает задачу  Исключение 1. В списке нет необходимого объекта | 2. Система возвращает список задач  3. Система возвращает список объектов |
| 4. Менеджер сохраняет задачу | 5. Система сохраняет задачу |

В таблице 3 представлен сценарий обработки исключительных ситуаций для варианта использования «Создать задачу».

Таблица 3 - Обработка исключительных ситуаций для варианта использования «Создать задачу»

|  |  |
| --- | --- |
| Действия актеров | Отклик системы |
| Исключение 1. В списке нет необходимого объекта | |
| 4. Менеджер создает новый объект | 5. Система добавляет новый объект в базу данных |
| 6. Менеджер сохраняет задачу | 7. Система сохраняет задачу |

С помощью диаграммы деятельности, представленной на рисунке 2, описан алгоритм реализации описанных сценариев для варианта использования «Создать задачу».

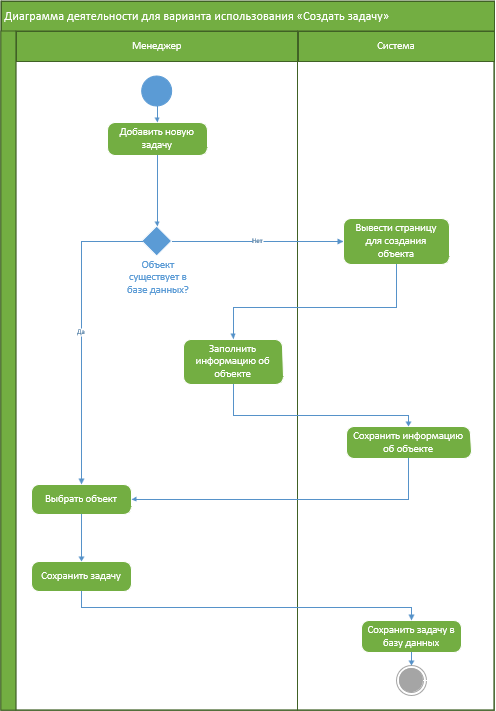


Рисунок 2 – Диаграмма деятельности для варианта использования «Создать задачу»

На диаграмме последовательностей, представленной на рисунке 3, описаны требования реализации сценария «Создать задачу» с учетом динамики во времени на уровне сообщений [9].

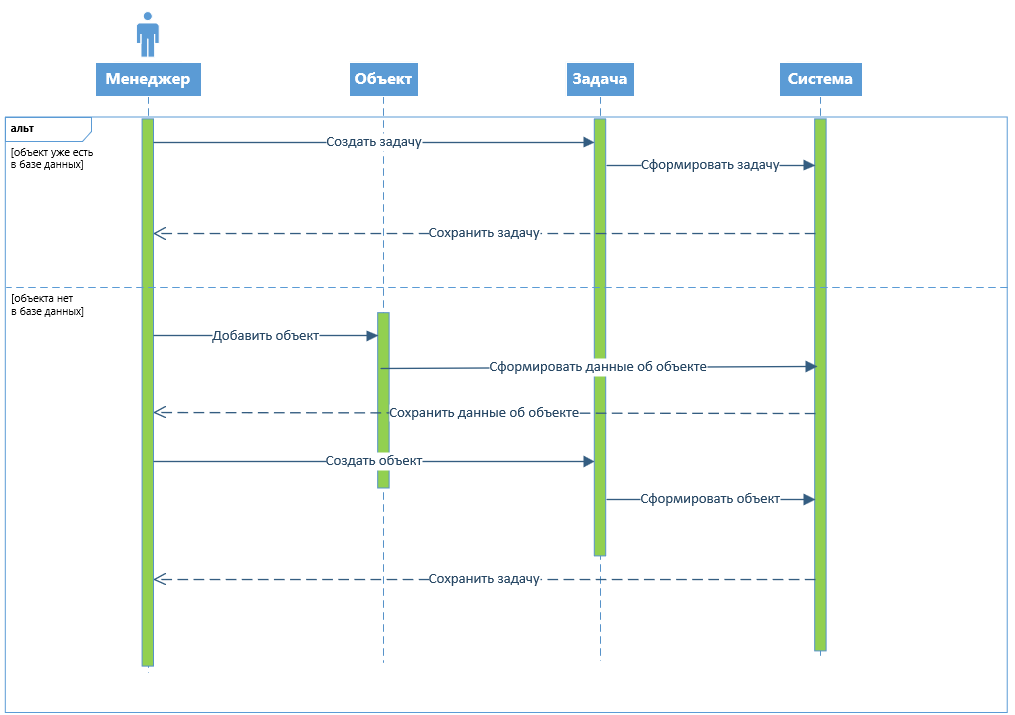


Рисунок 3 - Диаграмма последовательности для варианта использования «Создать задачу»

Разработанные спецификации программного обеспечения на языке UML средствами MS Visio показывают основные действия пользователей автоматизируемой системы и алгоритмы их выполнения.

2.2 Проектирование модели данных и диаграммы классов

На основе анализа требований заказчика к функциям программной системы, требованиям к организации входных и выходных данных, с учетом спроектированных требований к реализации функций, описанных в диаграммах вариантов использования, деятельности и последовательности, была разработана модель данных системы и описана в виде ER-диаграммы, позволяющая четко описать требования к представлению логической структуры данных, на основе которой в последующем будет разработана физическая структура данных для хранения во внешней памяти и программной обработки.

На рисунке 4 представлена ER – диаграмма для информационной системы «FixPhoto».

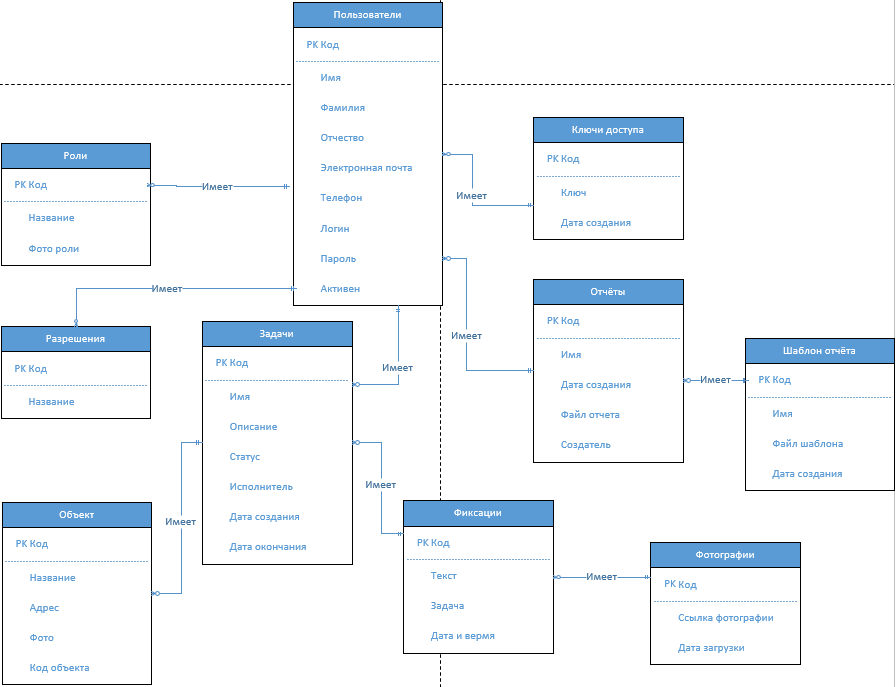


Рисунок 4 – ER-диаграмма для информационной системы «FixPhoto»

В таблице 4 представлены сведения о сущностях и связях между сущностями ER-диаграммы из рисунка 4 [7].

Таблица 4 – Характеристики сущностей и связей между сущностями   
ER-диаграммы для информационной системы “FixPhoto”

|  |  |  |  |
| --- | --- | --- | --- |
| Сущность | Назначение сущности | Ключ | Характеристики связей |
| Пользователи | Содержит данные о пользователях:  - имя;  - фамилия;  - отчество;  - электронная почта;  - телефон;  - логин;  - пароль;  - активен. | Код пользователя (PK) | Пользователь – Роль (1..1)  Пользователь – Ключ доступа (1..1)  Пользователь – Отчёты (1..n)  Пользователь – Фиксации (1..n)  Пользователь – Разрешения (1..n)  Пользователь – Задачи (1..n) |
| Роли | Содержит данные о ролях:  - название;  - фото роли. | Код роли (PK) | Пользователь – Роль (1..1) |
| Ключи доступа | Содержит данные о ключах доступа  - ключ;  - дата создания. | Код ключа доступа (PK) | Ключ пользователя – Пользователь (1..1) |
| Разрешения | Содержит данные о разрешениях:  - название. | Код разрешения (PK) | Разрешение – Пользователь (n..1) |
| Задачи | Содержит данные о задачах:  - имя;  - описание;  - статус; | Код задачи (PK) | Задача – Пользователь (n..1)  Задача – Объект (n..1)  Задача – Фиксации (1..n) |

Продолжение таблицы 4

|  |  |  |  |
| --- | --- | --- | --- |
| Сущность | Назначение сущности | Ключ | Характеристики связей |
|  | - исполнитель;  - дата создания;  - дата окончания. |  |  |
| Объекты | Содержит данные об объектах:  - название;  - адрес;  - фото;  - код объекта. | Код объекта (PK) | Объект – Задача (1..n) |
| Фиксации | Содержит данные о фиксациях:  - текст;  - задача;  - дата и время;  - дата регистрации;  - электронная почта;  - телефон. | Код фиксации (PK) | Фиксация – Задача (n..1)  Фиксация – Фотография (1..n) |
| Фотографии | Содержит данные о фотографиях:  - ссылка на фотографию;  - дата создания. | Код фотографии (PK) | Фотография – Фиксация (n..1) |
| Отчёт | Содержит данные об отчётах:  - имя;  - дата создания;  - файл отчёта;  - создатель. | Код отчёта (PK) | Отчёт – Пользователь (n..1)  Отчёт – Шаблон отчёта (n..1) |
| Шаблон отчёта | Содержит данные о шаблонах отчётов:  - имя; | Код шаблона отчёта (PK) | Шаблон отчёта – Отчёт (1..n) |

Продолжение таблицы 4

|  |  |  |  |
| --- | --- | --- | --- |
| Сущность | Назначение сущности | Ключ | Характеристики связей |
|  | - файл шаблона;  - дата создания. |  | Шаблон отчёта – Отчёт (1..n) |

На следующем этапе была спроектирована диаграмма классов, которая описывает требования к внутренней организации проектируемого программного обеспечения на уровне интерфейсов и классов, их кооперации, атрибутов и методов классов, правил их взаимодействия.

На рисунке 5 представлена диаграмма классов для информационной системы “FixPhoto”

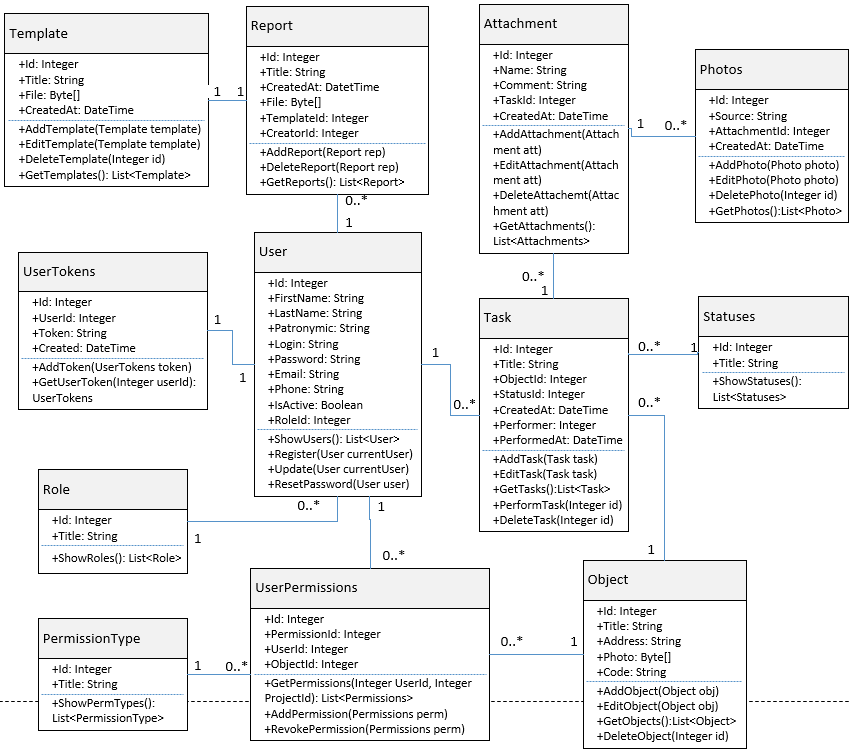


Рисунок 5 – Диаграмма классов для информационной системы “FixPhoto”

Данная диаграмма классов описывает типы объектов информационной системы “FixPhoto” и статические отношения, которые существуют между ними, а также отображаются операции классов.

В таблице 5 представлены сведения о структурных сущностях, свойствах, а также их методах [5].

Таблица 5 – Характеристика диаграммы классов для информационной системы “FixPhoto”

|  |  |  |
| --- | --- | --- |
| Имя класса | Атрибуты класса | Методы класса |
| User | +Id – содержит код пользователя;  +FirstName – содержит имя пользователя;  +LastName – содержит фамилию пользователя;  +Patronymic – содержит отчество пользователя;  +Login – содержит логин пользователя;  +Password – содержит пароль пользователя;  +Email – содержит электронную почту пользователя;  +Phone – содержит мобильный телефон пользователя;  +IsActive – содержит информацию о статусе пользователя;  +RoleId – содержит код роли пользователя. | +ShowUsers() – отображает список пользователей в системе;  +ResetPassword() – изменить пароль пользователя;  +Register() – регистрация пользователя;  +Update() – обновить информацию о пользователе. |
| Role | +Id – содержит код роли;  +Title – содержит название роли. | +ShowRoles() – отображает список ролей в системе. |

Продолжение таблицы 5

|  |  |  |
| --- | --- | --- |
| Имя класса | Атрибуты класса | Методы класса |
| UserTokens | +Id – содержит код токена пользователя;  +UserId – содержит код пользователя;  +Token - содержит ключ доступа пользователя;  +Created – содержит дату и время создания ключа доступа | +AddToken() – добавить токен пользователя;  +GetUserToken() – получить токен пользователя. |
| Template | +Id – содержит код шаблона отчёта;  +Title – содержит название шаблона;  +File – содержит файл шаблона отчёта;  +CreatedAt – содержит дату и время создания шаблона. | +AddTemplate() – добавить шаблон;  +EditTemplate() – изменить шаблон;  +DeleteTemplate() – удалить шаблон;  +GetTemplates() – получить список шаблонов. |
| Report | +Id – содержит код отчёта;  +Title – содержит название отчёта;  +CreatedAt - содержит дату и время создания отчёта;  +File – содержит файл отчёта;  +TemplateId – содержит код шаблона;  +CreatorId – содержит код создателя отчёта. | +AddReport() – добавить отчёт;  +DeleteReport() – удалить отчёт;  +GetReports() – получить список отчётов. |
| Attachment | +Id – содержит код фиксации;  +Name – содержит имя фиксации; | +AddAttachment() – создать фиксацию;  +EditAttachment() – изменить фиксацию; |

Продолжение таблицы 5

|  |  |  |
| --- | --- | --- |
| Имя класса | Атрибуты класса | Методы класса |
|  | +Comment – содержит комментарий фиксации;  +TaskId – содержит код задачи;  +CreatedAt – содержит дату и время создания фиксации. | +DeleteAttachment – удалить фиксацию;  +GetAttachments() – получить список фиксаций. |
| Task | +Id – содержит код задачи;  +Title – содержит название задачи;  +ObjectId – содержит код объекта;  +StatusId – содержит код статуса;  +CreatedAt – содержит дату и время создания;  +Performer – содержит код исполнителя задачи;  +PerformedAt – содержит дату и время выполнения задачи. | +AddTask() – добавить задачу;  +EditTask() – изменить задачу;  +GetTasks() – получить список задач;  +PerformTask() – выполнить задачу;  +DeleteTask() – удалить задачу. |
| Photos | +Id – содержит код фотографии;  +Source – содержит путь фотографии;  +AttachmentId – содержит код фиксации;  +CreatedAt – содержит дату и время создания. | +AddPhoto() – добавить фотографию;  +EditPhoto() – изменить фотографию;  +DeletePhoto() – удалить фотографию;  +GetPhotos() – получить список фотографий. |
| Statuses | +Id – содержит код статуса;  +Title – содержит название статуса. | +ShowStatuses() – получить список статусов. |

Продолжение таблицы 5

|  |  |  |
| --- | --- | --- |
| Имя класса | Атрибуты класса | Методы класса |
| Object | +Id – содержит код объекта;  +Title – содержит название объекта;  +Address – содержит адрес объекта;  +Photo – содержит фотографию объекта;  +Code – содержит системный код объекта. | +AddObject() – добавить объект;  +EditObject() – изменить объект;  +GetObjects() – получить список объектов;  +DeleteObject() – удалить объект. |
| UserPermissions | +Id – содержит код разрешений пользователя на объект;  +PermissionId – содержи код типа разрешения;  +UserId - cодержит код пользователя;  +ObjectId – содержит код объекта. | +GetPermissions() – получить список разрешений;  +AddPermission() – добавить разрешение;  +RevokePermission – отобрать разрешение. |
| PermissionType | +Id – содержит код типа разрешения;  +Title – содержит название типа разрешения; | +ShowPermTypes() – получить список типов разрешений |

Данная таблица описывает типы объектов информационной системы “FixPhoto” и отображает операции классов.

2.3 Проектирование диаграммы развертывания

Диаграмма развертывания, созданная средствами MS Visio, для проектируемой системы представлена на рисунке 6.

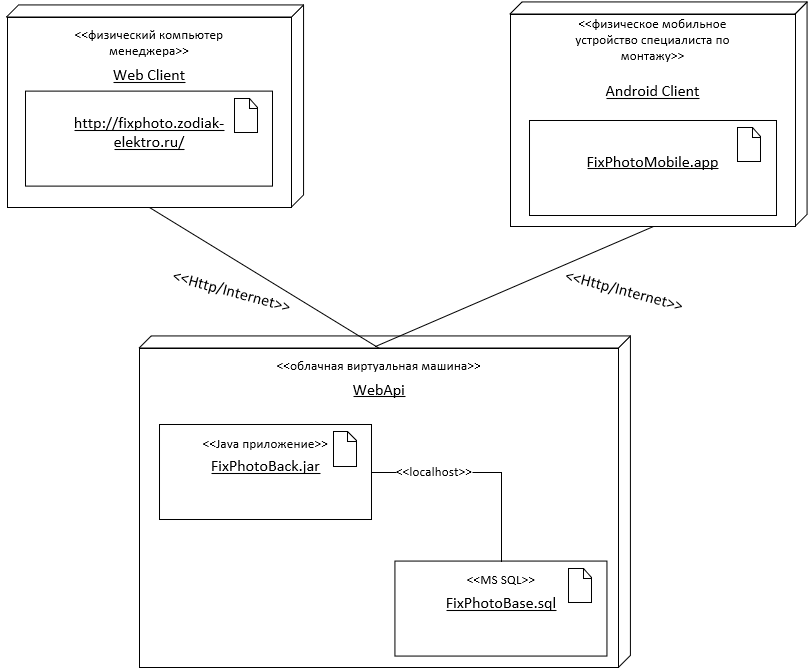


Рисунок 6 – Диаграмма развертывания для информационной системы “FixPhoto”

На выше представленной диаграмме отображен узел под названием “WebApi”, который и разрабатывается в рамках данного дипломного проекта.

Узел “Web Client” является рабочим местом менеджера, а узел “Android Client” в свою очередь является рабочим устройством специалиста по монтажу. Данные узлы передают данные по протоколу соединения Http на узел “WebApi”, являющимся сервером данных.

3 Реализация

3.1 Обоснование выбора средств разработки

В таблице 6 представлены некоторые возможные средства разработки для серверного компонента информационной системы “FixPhoto”.

Таблица 6 – Возможные средства разработки.

|  |  |  |
| --- | --- | --- |
| Язык программирования и технология | Плюсы | Минусы |
| Java, Spring | Автоконфигурация, встроенные серверы, работа в автономном режиме, возможность распределить нагрузку, используя микросервисы | Spring boot создает множество зависимостей автоматически, что приводит к большему размеру файла развертывания |
| C#, ASP .NET Core | Запросы к БД пишутся, используя типизированный Compile-time LINQ, централизированный репозиторий библиотек, легкая модель асинхронного программирования | ASP .NET Core является более молодой технологией, по сравнению с Spring. Имеет менее обширную документацию и сообщество, чем Spring. |

Java и фреймворк Spring позволяет в короткие сроки создать серверное приложение с удобной контейнеризацией и работой с данными. Также имеет хорошие инструменты для генерирования кода, что упрощает разработку.

ASP .NET Core имеет сложную структуру файлов проекта, ошибки в работе среды разработки Visual Studio.

Так как в техническом задании (см. Приложение А) указано, что система должна работать под операционной системой Windows, то выбор останавливается на языке программирования Java с использованием фреймворка Spring.

3.2 Разработка базы данных в среде СУБД

На основе ранее спроектированной ER-диаграммы в среде Microsoft SQL Server Management Studio 18 была разработана база данных FixPhoto. На рисунке 10 представлена диаграмма разработанной базы данных [4].

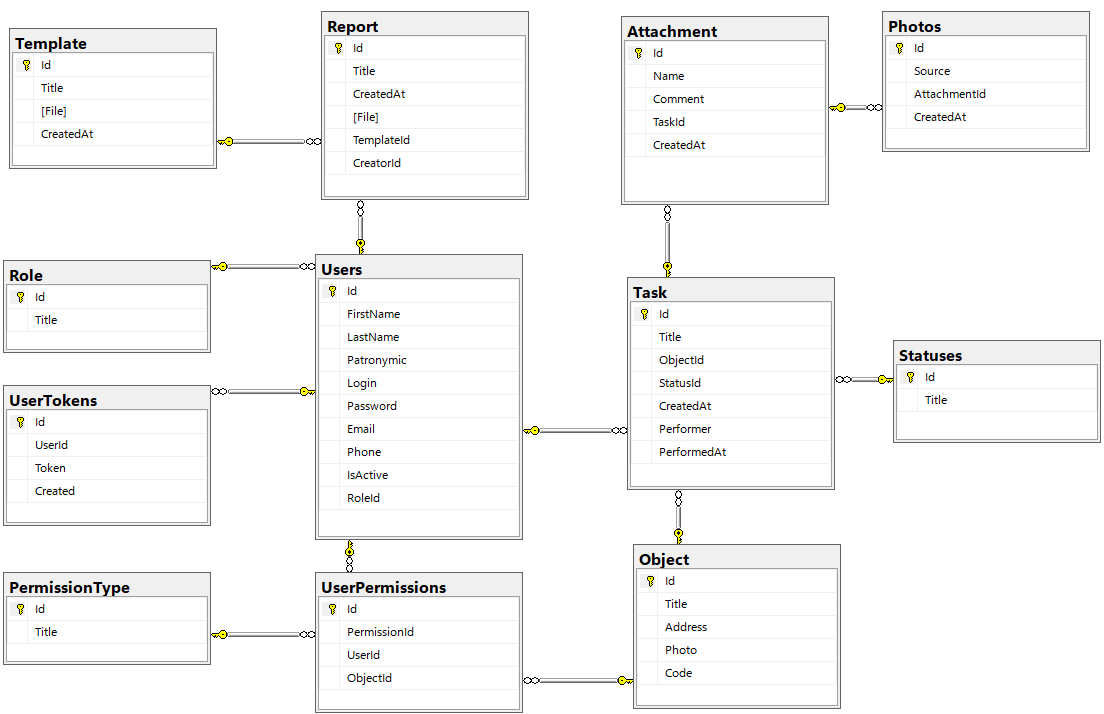


Рисунок 7 – Диаграмма базы данных FixPhoto

В таблице 7 представлено описание таблиц базы данных FixPhoto.

Таблица 7 – Описание таблиц базы данных FixPhoto

|  |  |
| --- | --- |
| Название таблицы | Краткое описание |
| Template | Предназначена для хранения информации о шаблонах отчётов |
| Report | Предназначена для хранения информации об отчётах |
| Role | Предназначена для хранения информации о ролях |
| User | Предназначена для хранения информации о пользователях |
| UserTokens | Предназначена для хранения информации о ключах доступа пользователей |
| PermissionTypes | Предназначена для хранения информации о типах разрешений |
| UserPermissions | Предназначена для хранения информации о разрешениях пользователей |
| Attachment | Предназначена для хранения информации о фиксациях специалистов по монтажу |
| Task | Предназначена для хранения информации о задачах |
| Object | Предназначена для хранения информации об объектах |
| Statuses | Предназначена для хранения информации о статусах задач |

Скрипт разработанной базы со всеми данными находится в Приложении Г.

3.3 Описание программных модулей

На рисунке 11 представлена информация о структуре проекта приложения

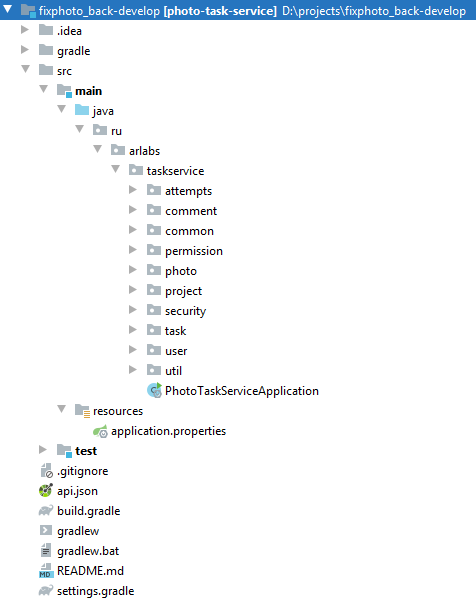
****

Рисунок 8 – Структура проекта

В таблице 8 описаны назначения директорий из проекта приложения.

Таблица 8 – Назначение директорий

|  |  |
| --- | --- |
| Название модуля | Назначение |
| Attempts | Хранит сервис загрузки фотографий в облако |
| Comment | Хранит сервис фиксаций |
| Common | Хранит общие для сервисов ошибки |
| Photo | Хранит сервис для обработки фотографий |
| Project | Хранит сервис для работы с объектами |
| Security | Хранит сервис безопасности сервера |
| Task | Хранит сервис работы с задачами |
| User | Хранит сервис для работы с пользователями |
| Util | Хранит вспомогающие классы |

На рисунке 9 представлен запрос регистрации пользователя:

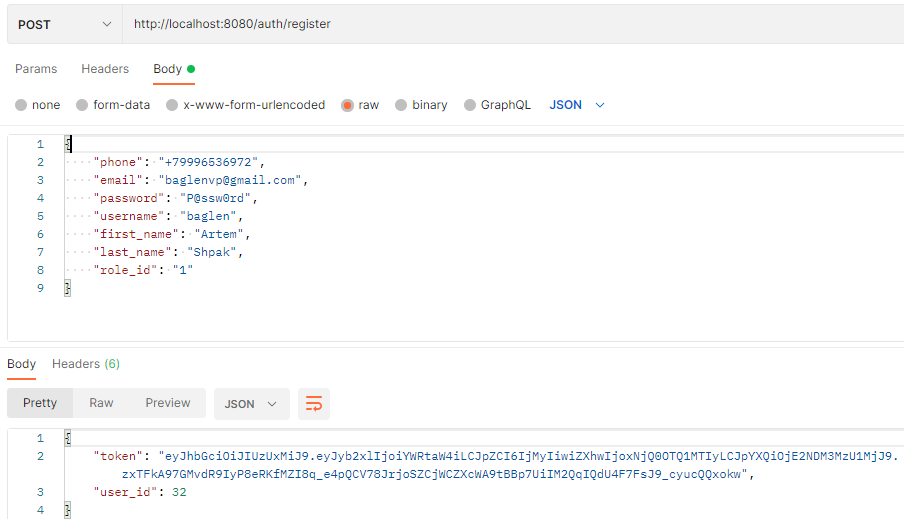
****

Рисунок 9 – Запрос регистрации пользователя

Код обработки запроса регистрации пользователя, представленного на рисунке 9 отображен на рисунке Б.1.

Код обработки запроса авторизации пользователя представлен на рисунке Б.2.

Результат работы обработки запроса авторизации пользователя представлен на рисунке 10.

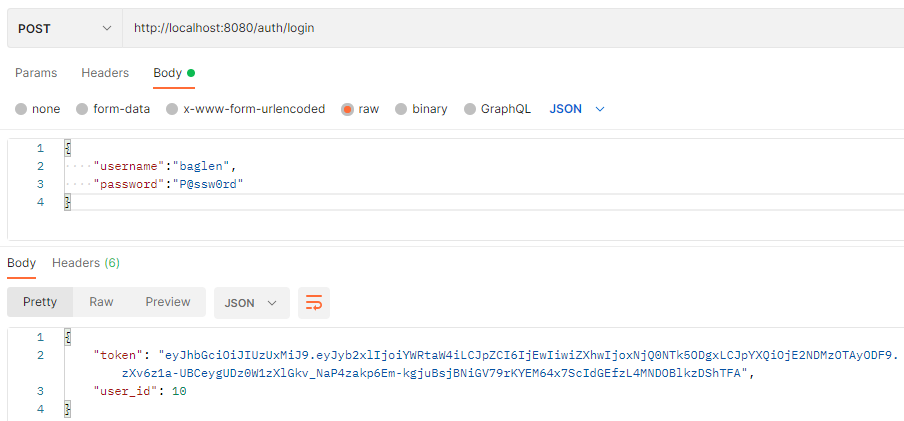


Рисунок 10 – Результат работы поиска

Запрос для формирования отчета представлено на рисунке 11.

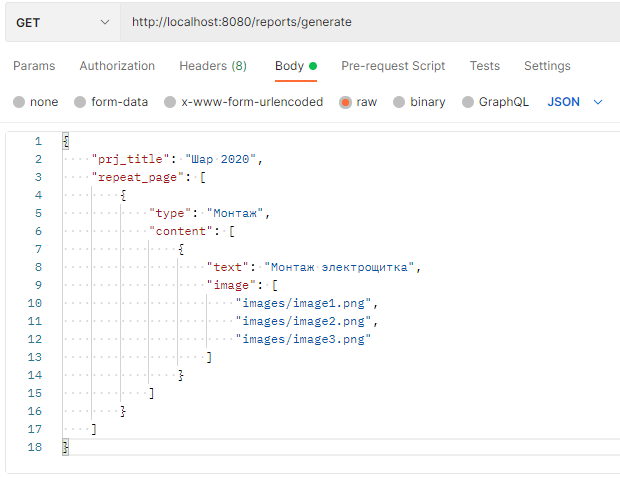


Рисунок 11 – Запрос для формирования отчета

Код обработки запроса для формирования отчёта, представленного на рисунке 11 представлен на рисунках Б.3 и Б.4.

Образец выходного документа представлен на рисунке 12.



Рисунок 12 – Образец выходного документа

Другие образцы выходных документов находятся на рисунках В.1 и В.2.

4 Тестирование программных модулей

4.1 Модульное тестирование

Задачей тестирования является проверка метода matches(), который проверяет введенный пароль пользователя и зашифрованный на сходство[3]. Тест кейсы представлены в Приложении Д.

На рисунке 13 представлен результат позитивного теста.

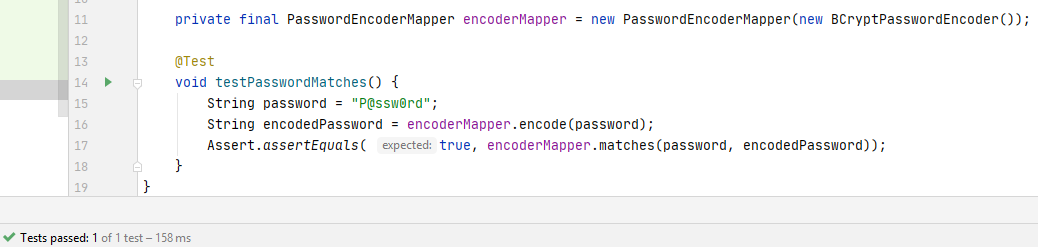


Рисунок 13 – Результат позитивного теста

На рисунке 14 представлен результат негативного теста.

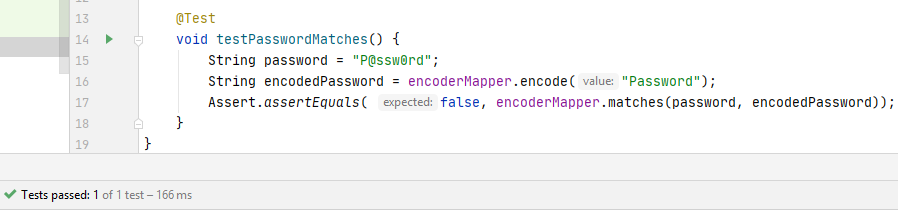


Рисунок 14 – Результат негативного теста

Негативный тест заключался в том, чтобы проверить, как отработает метод matches (), если не вводить символ «@» и 0[8].

4.2 Интеграционное тестирование

Интеграционный тест #1:

Цель теста: задачей тестирования является проверка интеграционной связи между репозиторием задач и выбранной задачей.

Описание теста: для удаления была выбрана задача из базы данных с первичным ключом «10».

Ожидаемый результат: «True».

Фактический результат: «True».

На рисунке 15 представлен результат позитивного теста.

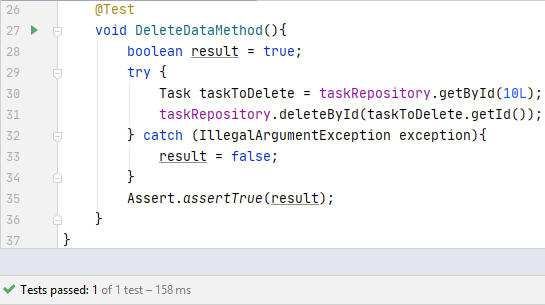


Рисунок 15 – Результат интеграционного теста

5 Эксплуатационная документация

5.1 Руководство разработчика

Заключение

В результате выполнения дипломной проектной работы был разработан серверный программно-информационный компонент информационной системы “FixPhoto” с помощью средств разработки языка программирования Java и фреймворка Spring, а также с использованием системой управления базами данных SQL Server Management Studio.

Cерверный программно-информационный компонент соответствует требованиям и пожеланиям заказчика и обладает следующими функциональными возможностями:

- ведение учета сотрудников и их персональных данных в электронном виде;

- ведение учета работ, произведенных компанией, в электронном виде;

- сбор, хранение и обработка информации об объектах;

- формирование отчетов за выбранный период и объект времени по произведенным работам;

- произведение фиксации произведенной работы по задачам.

Разработанный серверный программно-информационный компонент можно рассматривать как удобное средство для автоматизации рабочих мест менеджера и специалиста по монтажу в компаниях по монтажу оборудования.

Список использованных источников

1. ГОСТ 19.201-78. Единая система программной документации (ЕСПД). Техническое задание. Требования к содержанию и оформлению (с Изменением N 1) = Unified system for program documentation. Technical specifications for development. Requirements to contents and form of presentation: межгосударственный стандарт: издание официальное: утвержден и введен в действие Постановлением Государственного комитета СССР по стандартам от 18 декабря 1978 г. № 3351: введен впервые: дата введения 1980-01-01. – Москва: Стандартинформ, 2010. – 4 с. – Текст непосредственный.

2. РД 50-34.698-90. Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов: руководящий документ по стандартизации: издание официальное: утверждены и введены в действие Постановлением Государственного комитета СССР по управлению качеством продукции и стандартами от 27 декабря 1990 г. № 3380: дата введения 1992-01-01 / Разработан Министерством электротехнической промышленности и приборостроения СССР. – Москва.: ИПК Издательство стандартов, 2002 г. – 27 с. –Текст непосредственный.

3. Семакин, И.Г. Основы алгоритмизации и программирования: учебник для студ. учреждений среднего профессионального образования / И.Г. Семакин, А.П. Шестаков. – 2-е изд., стер. – М.: Издательский центр «Академия», 2018. – 304 с. – (Профессиональное образование). – ISBN 978-5-4468-6228-3. – Текст: непосредственный.

4. Федорова, Г.Н. Разработка, администрирование и защита баз данных: учебник для студ. учреждений среднего профессионального образования / Г.Н. Федорова. – 2-е изд., стер. – М.: Издательский центр «Академия», 2018. – 288 с. – (Профессиональное образование). – ISBN 978-5-4468-6994-7. – Текст: непосредственный.

5. Онлайн-школа IT профессий и сообщество программистов. [сайт] – Текст: электронный. – URL: https://itproger.com/ (дата обращения 15.05.2021).

6. Портал для помощи программистам. Текст: электронный. – URL: https://codernet.ru/ (дата обращения 27.03.2021).

7. Сайт о программировании. Полное руководство по языку программирования С# 9.0 и платформе .NET 5: [сайт] – Текст: электронный. – URL: https://metanit.com/sharp/tutorial (дата обращения 20.10.2020).

8. IT Курсы программирования онлайн – обучение программированию, видео уроки. [сайт] - Текст: электронный. Видеоматериалы. – URL: https://itvdn.com/ru (дата обращения 21.04.2021).

9. UML-диаграммы классов. Программирование. [сайт] – Текст: электронный. – URL: https://prog-cpp.ru/uml-classes (дата обращения 17.04.2021).

10. Программирование на C#. [сайт] – Текст: электронный. – URL: https://stepik.org/course/4143/promo (дата обращения 11.05.2021).

Приложение А

(обязательное)

Техническое задание. Требования к программным модулям

1 Введение

Настоящее техническое задание распространяется на разработку серверного программно-информационного компонента информационной системы "FixPhoto". Данный компонент предназначен для осуществления безопасного сбора, хранения, обработки фотографий для компании ООО "Зодиак-Электро".

Автоматизированная система фиксирования хода выполнения задач "FixPhoto" позволяет быстро и удобно сотрудникам внутри компании отправлять фотографии выполненных задач, формировать отчеты по выбранным объектам и временному периоду.

Подобная автоматизация выполнения монтажных работ повышает эффективность управленческой деятельности и позволяет создать единую систему, работающую как слаженный механизм, своевременно анализировать процессы работы организации и повысить качество и эффективность выполнения работ.

2 Основания для разработки

Основанием для разработки является заказ системы FixPhoto для автоматизации работы сбора фотографий, их обработка и формирование отчетов для компании ООО «Зодиак-Электро» и согласовано с директором Блиновым Е.А.

Наименование темы разработки – «Разработка серверного программно-информационного компонента информационной системы "FixPhoto" ООО "Зодиак-Электро"».

3 Назначение разработки

Система призвана автоматизировать работу сотрудников компании:

- вести учет сотрудников компании;

- вести учет объектов;

- вести учет хода выполнения монтажных работ;

- вести учет поставленных задач.

Пользователями системы выступают специалисты по монтажу и менеджер.

4 Требования к программе или программным модулям

4.1 Требования к функциональным характеристикам

После запуска программы пользователю отображается окно авторизации с формой для ввода логина и пароля.

В системе существует всего 3 типа пользователей – менеджер, бухгалтер и администратор. В соответствии с типом пользователя программа открывает соответствующий интерфейс.

Для менеджера программа предоставляет следующие возможности:

- просмотр и изменение списка записей организации;

- просмотр и изменение списка клиентов организации;

- просмотр списка предоставляемых услуг организации.

Для бухгалтера программа предоставляет следующие возможности:

- просмотр и изменение списка предоставляемых организацией услуг;

- просмотр и изменение списка сотрудников организации;

- формирование отчетов по записям за выбранный период.

Для администратора программа предоставляет следующие возможности:

- просмотр и изменение списка пользователей системы;

- просмотр истории авторизации;

- сброс пароля пользователя.

Для всех типов пользователей программа предоставляет возможность изменения своих персональных данных.

4.2 Требования к надежности и безопасности

Серверный компонент должен соответствовать современному уровню требований к надежности программного обеспечения:

- предусматривать контроль вводимой информации и блокировку некорректных действий пользователя при работе с системой;

- обеспечивать целостность информации, хранящейся в базе данных;

- обрабатывать все возможные исключение и сообщать о них пользователю.

4.3 Требования к составу и параметрам технических средств

Требования к компьютеру сотрудника представлены в таблице 1.

Таблица А.1 – Требования к компьютеру, выполняющего роль сервера

|  |  |
| --- | --- |
| Компонент | Требование |
| Жесткий диск | Минимум 200 ГБ свободного места |
| Монитор | Требуется монитор с разрешением 1280х720 или более высоким |
| Оперативная память | Не менее 32 ГБ с последующим увеличением по мере роста размера базы данных |
| Быстродействие процессора | Минимум: процессор х64 с тактовой частотой 1,4 ГГц |
| Операционная система | Windows 10 |

4.4 Требования к информационной и программной совместимости

На серверах должны быть установлены ОС Windows Server 2019, сервер БД Microsoft SQL Server Express 2019, SQL Server Management Studio 18.8.

Приложение Б

(обязательное)

Программный код

Программный код представлен в контексте текстового редактора IntelliJ Idea Ultimate Edition версии 2020.1.

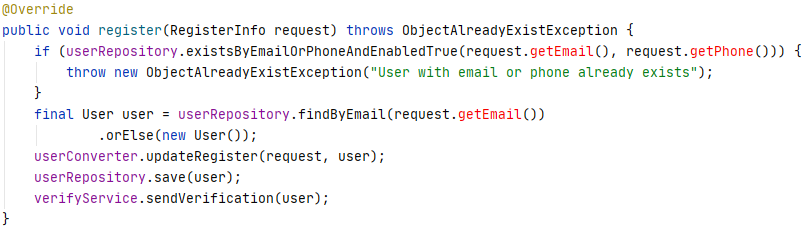
**

Рисунок Б.1 – Программный код обработки запроса регистрации пользователя



Рисунок Б.2 – Программный код обработки запроса авторизации пользователя

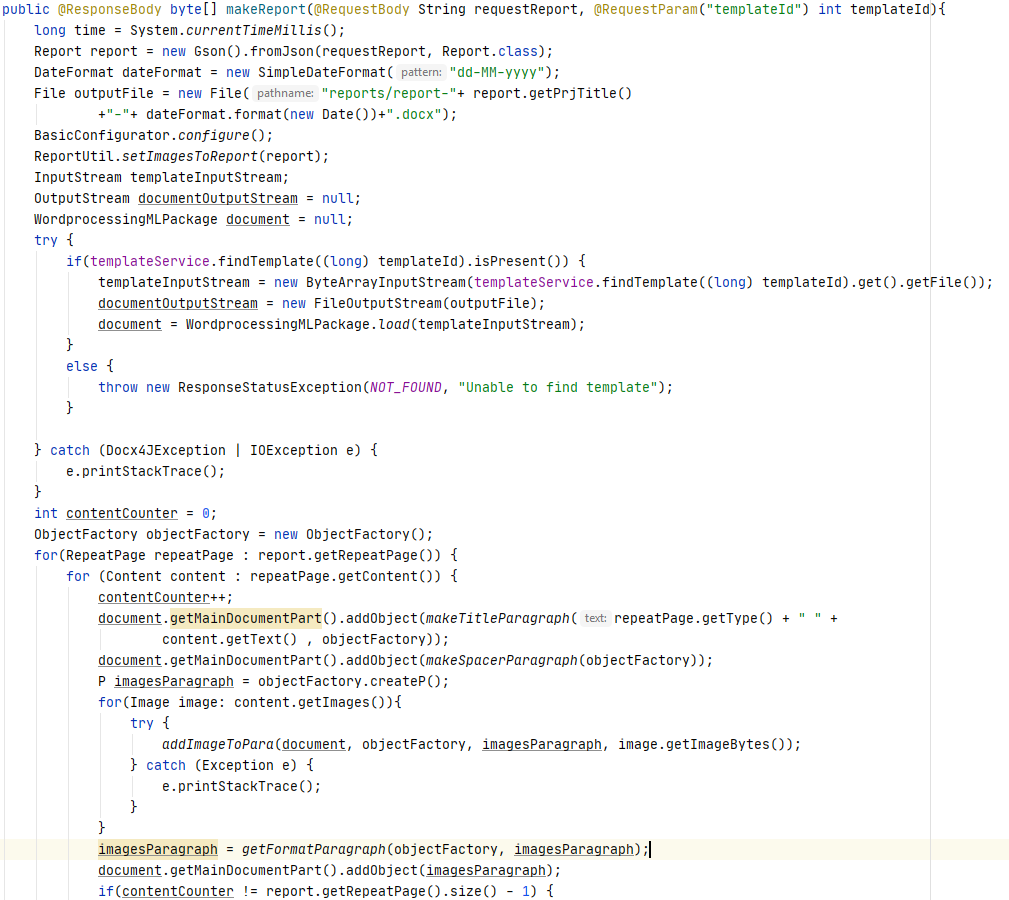


Рисунок Б.3 – Программный код обработки запроса формирования отчёта. Блок 1

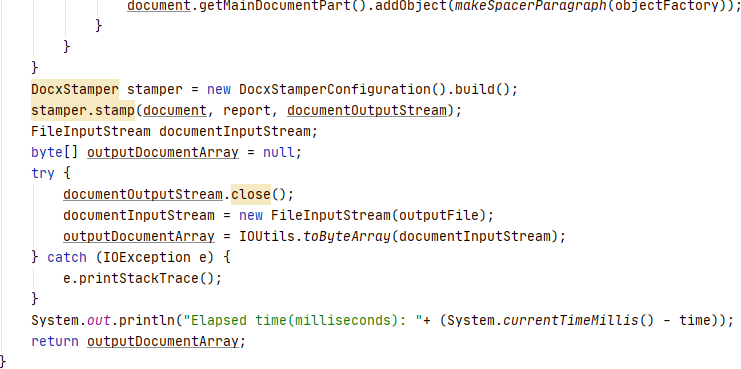


Рисунок Б.4 – Программный код обработки запроса формирования отчёта. Блок 2

Приложение В

(обязательное)

Формы выходных документов

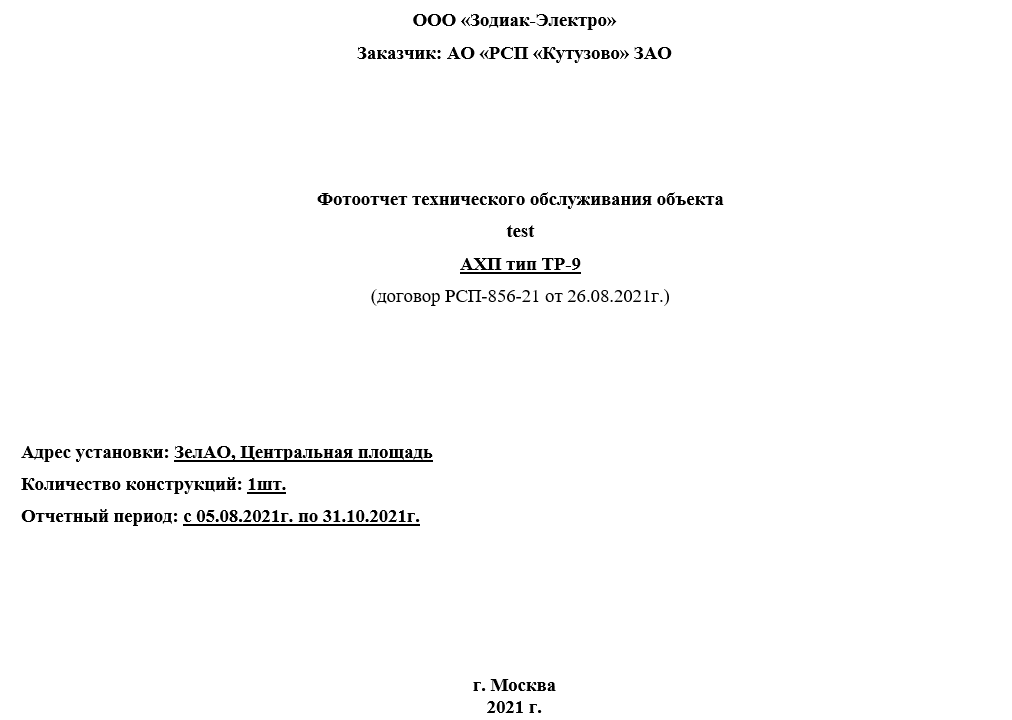


Рисунок В.1 – Первый лист отчёта в формате .docx

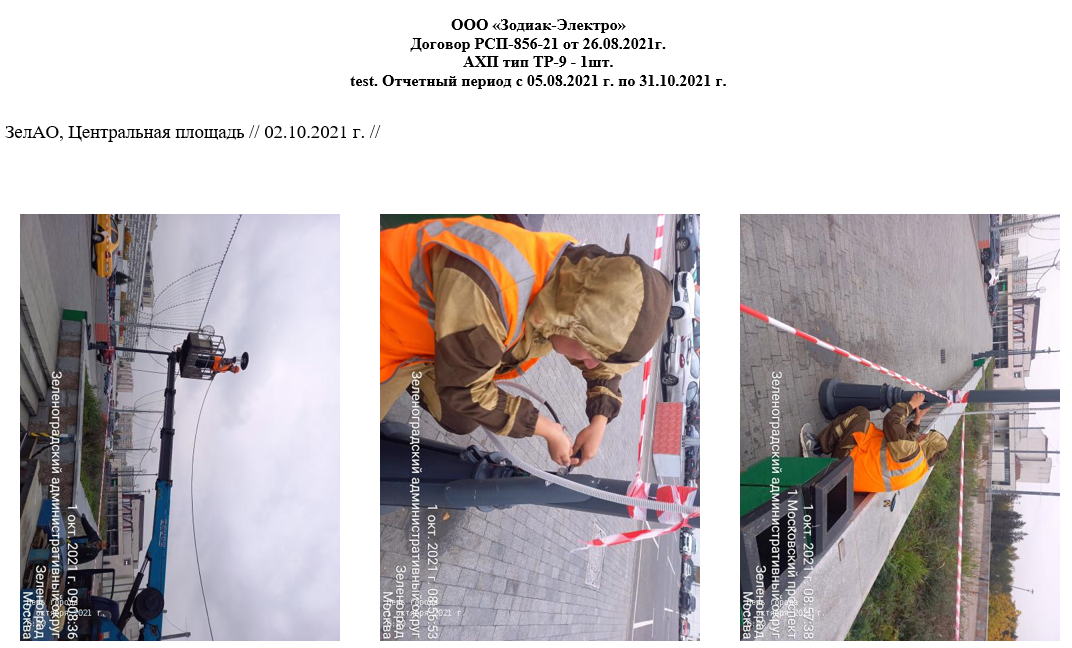


Рисунок В.2 – Второй лист отчёта в формате .docx

Приложение Г

(обязательное)

Скрипт базы данных



Рисунок Г.1 – Скрипт базы данных. Блок 1

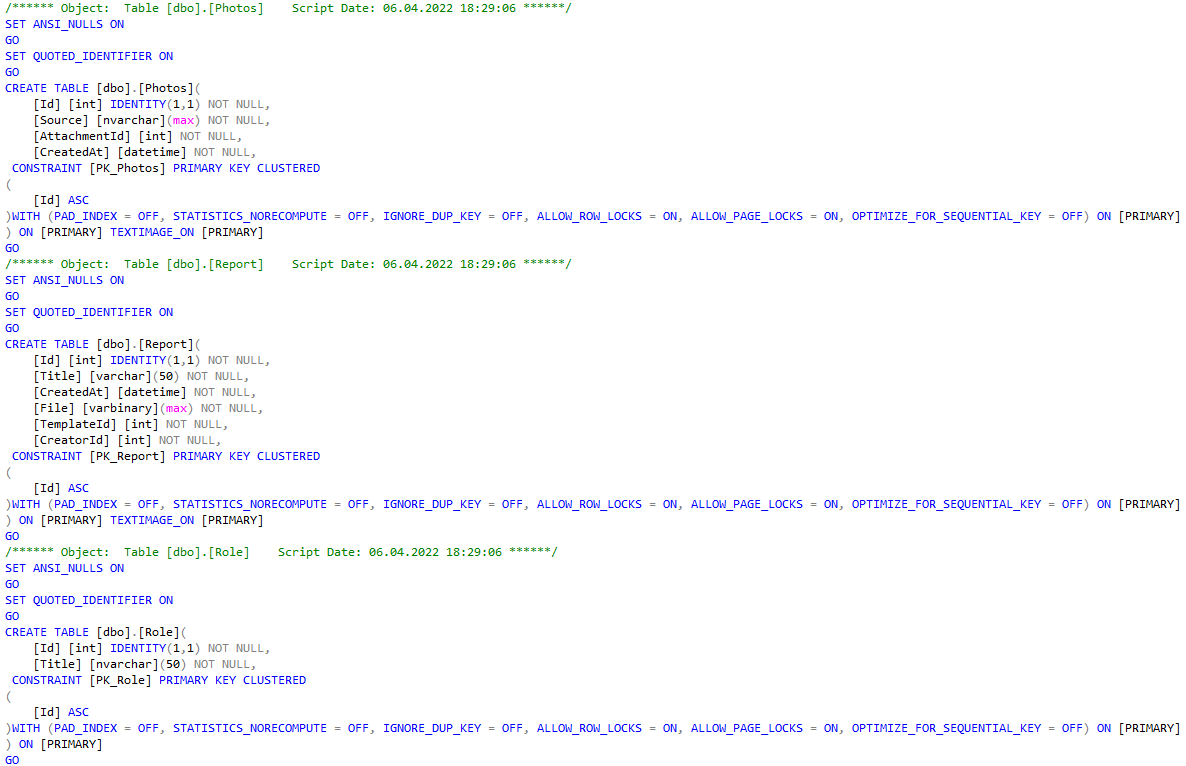


Рисунок Г.2 – Скрипт базы данных. Блок 2



Рисунок Г.3 – Скрипт базы данных. Блок 3

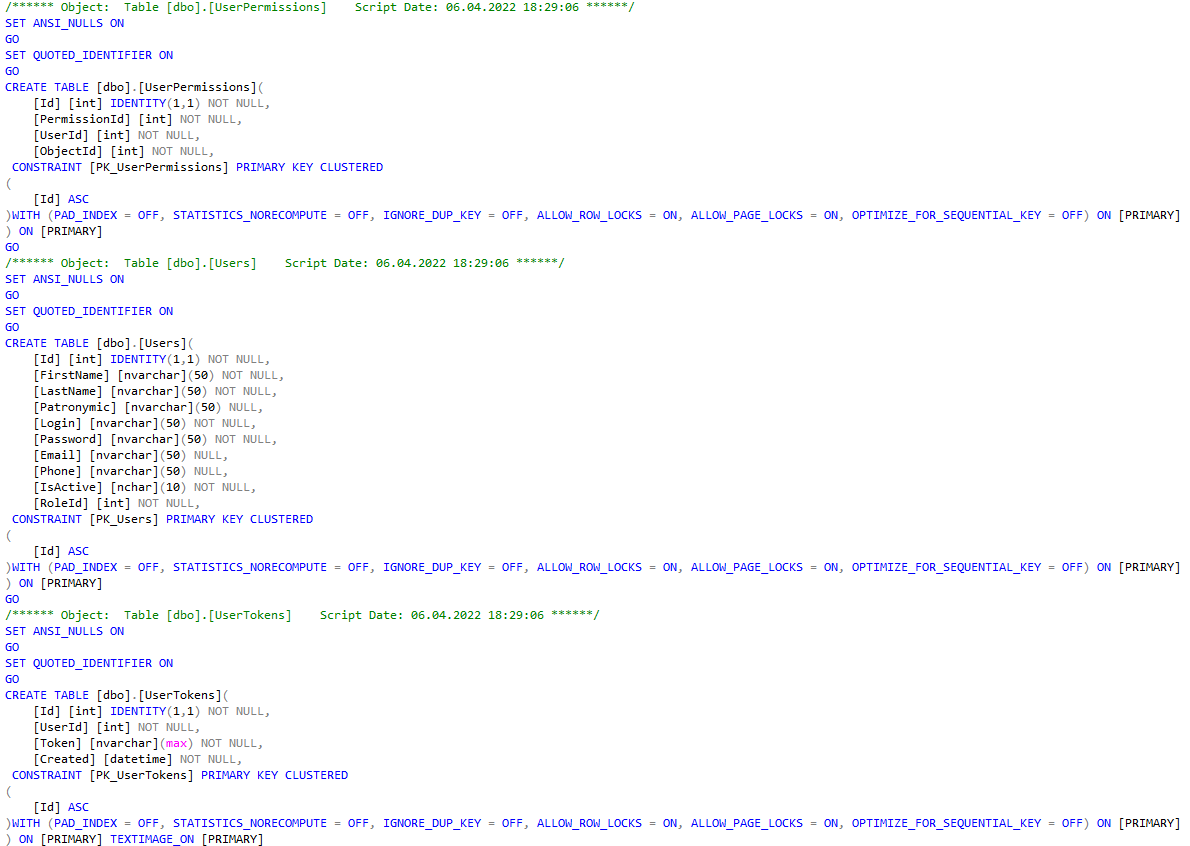


Рисунок Г.4 – Скрипт базы данных. Блок 4

Приложение Д

(обязательное)

Тестирование

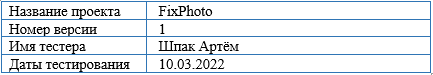


Рисунок Д.1 – Общая информация о тестировании

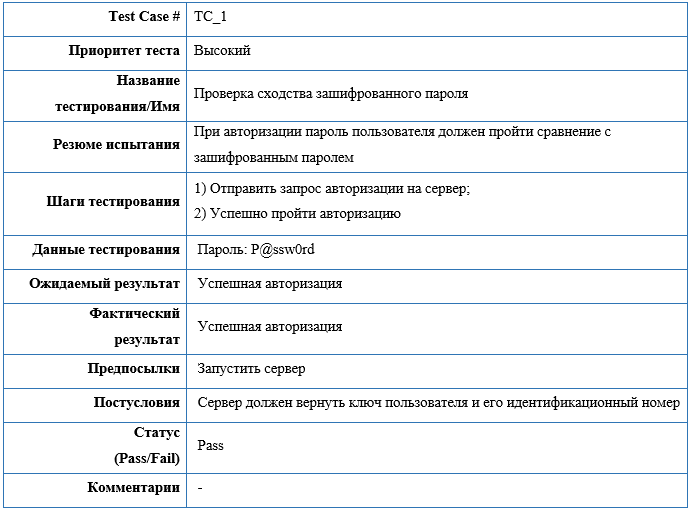


Рисунок Д.2 – Проверка сходства зашифрованного пароля. Блок 1

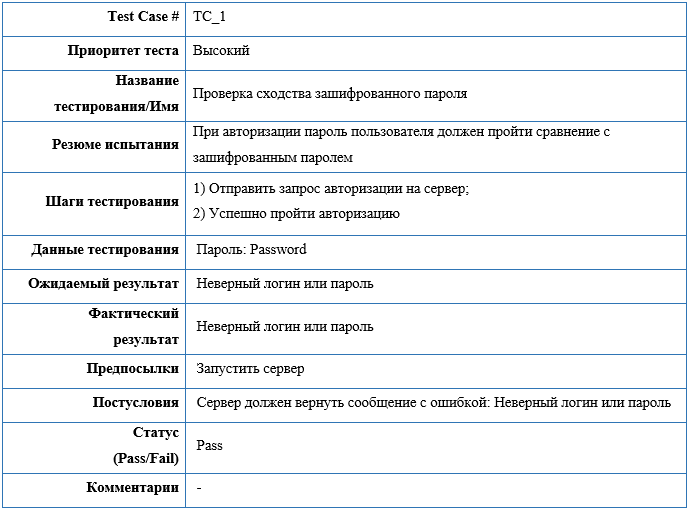


Рисунок Д.3 – Проверка сходства зашифрованного пароля. Блок 2