1. Общие сведения

1.1. Наименование системы

Автоматизированная система управления (АСУ) для заведений общественного питания (или иной предметной области).

1.2. Заказчик системы

«Омский государственный университет им. Ф.М. Достоевского» Институт среднего профессионального образования и довузовской подготовки

1.3. Разработчик системы

«Омский государственный университет им. Ф.М. Достоевского» Институт среднего профессионального образования и довузовской подготовки (ФИО)

1.4. Сроки разработки

Начало: 17.06.2025

Окончание: 17.06.2025

1.5. Стадии и этапы разработки

Техническое проектирование

Реализация

Тестирование

Ввод в эксплуатацию

2. Цели и назначение создания автоматизированной системы

2.1. Цель разработки

Автоматизация процессов управления в [предметной области], повышение эффективности работы сотрудников, минимизация ошибок ручного учета.

2.2. Назначение системы

Система предназначена для:

Управления учетными процессами (заказы, клиенты, сотрудники и т. д.).

Контроля выполнения задач сотрудниками.

Формирования отчетности.

3. Характеристика объектов автоматизации

3.1. Описание предметной области

АС предназначена для автоматизации работы [организации/предприятия], включая:

Управление персоналом.

Учет заказов/транзакций.

Контроль выполнения задач.

3.2. Пользователи системы

Таблица 1. Пользователи системы

Роль

Функции

Администратор

Управление пользователями, настройка системы, контроль данных

[Роль 2]

[Функции]

[Роль 3]

[Функции]

4. Требования к автоматизированной системе

4.1. Функциональные требования

Регистрация и аутентификация пользователей.

Управление данными (добавление, редактирование, удаление).

Формирование отчетов.

Контроль статусов операций.

4.2. Требования к интерфейсу

Интуитивно понятный веб-интерфейс.

Разделение прав доступа.

Адаптивность под разные устройства.

4.3. Требования к безопасности

Разграничение прав доступа.

Ограничение неверных попыток входа.

Резервное копирование.

4.4. Технические требования

Серверная часть:

Процессор - Intel(R) Core(TM) i5-10400 CPU @ 2.90GHz 2.90 GHz

ОЗУ - 8,00 ГБ

Хранилище данных - SSD 512 Гб

Клиентская часть: Windows 10, .NET.

СУБД: MySQL Workbench

5. Состав и содержание работ по созданию АС

Анализ требований.

Проектирование архитектуры.

Разработка интерфейсов.

Программная реализация.

Тестирование.

Ввод в эксплуатацию.

6. Порядок разработки автоматизированной системы

Использование гибких методологий (Agile/Scrum).

Этапное тестирование.

Взаимодействие с заказчиком на каждом этапе.

7. Порядок контроля и приемки АС

Проведение приемо-сдаточных испытаний.

Подписание акта ввода в эксплуатацию.

Гарантийное сопровождение.

8. Требования к подготовке объекта автоматизации

Обучение персонала.

Настройка оборудования.

Миграция данных (при необходимости).

9. Требования к документированию

Руководство пользователя.

Техническая документация.

API-документация (если требуется).

10. Источники разработки

ГОСТ 34.

Внутренние стандарты компании.

Аналогичные решения на рынке.

Руководство пользователя на информационную систему

Нащзвание\_Системы

1. Начало работы

Для запуска приложения необходимо найти соответствующий ярлык на рабочем столе или в меню "Пуск" и выполнить двойной клик левой кнопкой мыши. После этого откроется окно авторизации, где потребуется ввести учетные данные - логин и пароль, выданные системным администратором. В случае первого входа рекомендуется сразу сменить временный пароль на постоянный, используя соответствующую функцию в личном кабинете.

2. Основной интерфейс входные и вых данные

Главное окно приложения содержит основное меню, расположенное в верхней части экрана, и рабочую область, занимающую большую часть интерфейса. В левой части может находиться панель навигации с доступными разделами системы. В правом нижнем углу отображается информация о текущем пользователе и кнопка выхода из системы. Интерфейс выполнен в классическом стиле оконного приложения с возможностью изменения размеров окон и их перетаскивания.

3. Работа с данными

Для просмотра списка записей используется соответствующая вкладка в основном меню. Данные отображаются в виде таблицы с возможностью сортировки по любому из столбцов. Чтобы открыть детальную информацию о конкретной записи, необходимо сделать двойной клик по соответствующей строке таблицы. Добавление новых данных осуществляется через кнопку "Создать" в верхней панели инструментов. При этом открывается форма ввода, где обязательные для заполнения поля помечены красной звездочкой.

4. Поиск и фильтрация

В верхней части каждого списка данных находится строка поиска, позволяющая осуществлять быстрый поиск по ключевым словам. Для более точного поиска можно использовать расширенный фильтр, доступный по нажатию соответствующей кнопки рядом с полем поиска. В появившемся окне можно задать несколько критериев отбора одновременно. Результаты фильтрации автоматически обновляются в основном окне списка.

5. Завершение работы

Перед закрытием приложения рекомендуется сохранить все внесенные изменения. Для корректного завершения работы следует использовать пункт меню "Выход" или соответствующую кнопку в правом нижнем углу интерфейса. Не рекомендуется закрывать приложение через крестик в углу окна, так как это может привести к потере несохраненных данных.

…-01 13 01

**1. ОБЩИЕ СВЕДЕНИЯ**

1.1. Обозначение и наименование программы

Наименование программы – …..

Обозначение программы – …….

**2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ**

ПО  выполняет следующие функции:

1) контроль своевременного выполнения задач по связи дежурным расчетом/аппаратной;

2) отображение состояния сеансов на визуальных формах;

3) отображение состояния технических средств на визуальных формах;

4) ввод и редактирование данных о составе  средств;

5) ввод и редактирование управляемых параметров технических средств;

6) идентификации и аутентификации пользователей путем авторизации

………..

**3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ**

3.2.1. Основные классы

Необходимые для программы классы и их исходные файлы представлены в таблице 1.

…-01 13 01

Таблица 1

|  |  |  |
| --- | --- | --- |
| Наименование класса | Файл объявления(\*.h) | Файл описания(\*.cpp) |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

3.2.2. Описание классов

Для запуска приложения используется функция main, которая принимает аргументы приложения. В качестве первого аргумента передается IP-адрес сервера, на котором находится база данных ПРЦ (далее БД). Все остальные настройки программы берутся из базы данных.

Функция main прежде всего выполняет соединение с базой данных MySQL. Если соединение с базой данных по каким-то причинам не произошло (не тот IP-адрес, либо база не установлена, либо не работает СУБД MySQL), функция main выдает сообщение об этом и завершает свою работу.

Иначе, функция далее выполняет следующие действия:

* создает объекты данных для дальнейшего управления;
* запускает главное окно приложения Название.

Далее приводится краткое описание основных классов программы.

Класс SqlDatabase обеспечивает соединение с базой данных MySQL и предоставляет функции для работы с БД.

Класс ManageApp на основе содержимого БД создает список объектов данных, которые являются основными элементами программы и модели данных

…-01 13 01

включающие в себя разные наборы объектов данных.

Класс MainWindow обеспечивает работу главного окна приложения, которое включает в себя обработку команд меню, визуальное отображение состояния технических средств комплекса, вызов диалоговых форм управления техническими средствами.

Класс ObjectData имеет несколько конструкторов и может создавать объекты с типом, номером и данными, структурированными в XML-формат.

Класс ObjectManage является потомком ObjectData и является основным классом программы, так как выполняет важнейшие функции:

* содержит статический указатель на класс SqlDatabase, который используется во всех файлах программы;
* инициализирует объект, загружая его данные из БД (функция loadObjFromDB);
* при необходимости создает диалоговую форму для объекта данных, используя указатель на класс BaseClassDlg;
* обновляет данные объекта при их изменении в процессе работы, в том числе и на диалоговой форме.

Класс BaseClassDlg создает базовую диалоговую форму для управления ТС, в которой определяется основной стиль внешнего вида и поведения окна при его вызове и в процессе выполнения команд.

Классы QAK, QBFM, QDUKK, QPK, QMRPU, QUFKS, QUFSOCH, QSEV, QMRPU являются потомками BaseClassDlg. Они создают диалоговые формы управления техническими средствами ПРЦ.

Класс DlgMainWin – базовый класс для создания диалоговых форм с наполнением в зависимости от заданного делегата. В данном случае реализовано отображение графики и отображение моделей данных в виде элементов – потомков класса QWidget.

Класс BaseClassKmd обеспечивает процесс отправки и выполнения

…-01 13 01

команд, отображения результатов выполнения на соответствующих диалогах управления средствами.

Класс BaseThread – базовый класс создания потоков в программе, а класс DemonThreads осуществляет выполнение потоков из очереди потоков.

Класс Magazine – шаблонный класс очереди, который обеспечивает добавление и выборку элементов очереди.

Класс Socket – класс на основе QUdpSocket осуществляет прием и передачу данных через заданный порт.

По командам меню Данные вызываются классы: QEditAdrLIS, QEditLM, QEditOper, QEditPRA, QEditRDM, QEditRPDU, QEditWP, которые отвечают за редактирование определенных таблиц базы данных.

**6. ВЫЗОВ И ЗАГРУЗКА**

Запуск программы осуществляется с помощью ярлыка «АРМ ….», расположенного на рабочем столе.

…-01 13 01

После запуска приложения появится диалоговое окно регистрации пользователя. После ввода в данном окне логина и пароля выполняется процедура аутентификации пользователя и, в случае успешного ее выполнения, загружается главное окно программы…...