титульник – приветствие

Здравствуйте, многоуважаемая комиссия. Вашему вниманию предоставляется дипломный проект студентов группы ИСП-5 Аверина Максима Игоревича и ИСП-7с Костюничева Андрея Игоревича на тему «Интеграция программного модуля «Колл центр с клиентской базой».

**\*клик\*** второй слайд – введение

В современном мире, где бизнес-процессы становятся все более сложными и разветвлёнными, обеспечение высокого уровня обслуживания клиентов становится одним из ключевых факторов успеха для любой компании. Одним из наиболее эффективных инструментов для улучшения качества обслуживания является использование колл-центра, который позволяет оперативно реагировать на запросы клиентов и предоставлять им высококачественную поддержку.

**\*клик\*** третий слайд – цель проекта

Цель данной дипломной работы заключается в разработке программного модуля, который позволит интегрировать колл-центр с клиентской базой данных. Такая интеграция позволяет значительно повысить качество обслуживания клиентов и улучшить уровень удовлетворённости.

**\*клик\*** четвёртый слайд – анализ предметной области

Программный модуль "Колл-центр с клиентской базой" предназначен для интеграции колл-центра компании с её клиентской базой данных. Он позволяет операторам колл-центра быстро получать доступ к информации о клиентах. Это значительно повышает качество обслуживания клиентов и улучшает уровень удовлетворённости. Он обеспечивает автоматическую синхронизацию данных между колл-центром и клиентской базой. Также предоставляет возможность операторам колл-центра регистрировать новых клиентов, обновлять информацию в базе данных и отслеживать запросы клиентов.

В целом, программный модуль "Колл-центр с клиентской базой" является важным инструментом для компаний, которые хотят повысить качество обслуживания клиентов и улучшить свой бизнес. Он позволяет операторам колл-центра быстро получать доступ к информации о клиентах, что позволяет им предоставлять высококачественное обслуживание и повышать уровень удовлетворённости клиентов.

**\*клик\*** пятый слайд – тех задание

С тем, что такое колл-центр с клиентской базой, мы разобрались. Теперь необходимо поставить себе конкретные задачи для реализации системы. Для этого существует понятие технического задания.

Техническое задание – документ, устанавливающий основные назначения, предъявляемые к разрабатываемому изделию.

В разрабатываемой CRM системе колл-центр с клиентской базой необходимо реализовать функциональный блок клиентской части, который должен в себя включать:

* возможность просмотра, добавления, редактирования и удаления информации о клиентах в базе данных;
* возможность поиска клиентов по ФИО;
* возможность просмотра заметок, которые оставил сотрудник о клиенте, а также просмотр, редактирование и удаление существующих заметок;
* возможность просмотра всех сотрудников;
* возможность просмотра всех ролей, и кто из сотрудников к какой роли относится;
* возможность просмотра, редактирования и удаления своего профиля.

И так же функциональный блок администраторской части приложения, который должен в себя включать:

* возможность добавления, редактирования и удаления информации о клиентах в базе данных;
* возможность поиска клиентов по ФИО;
* возможность просмотра заметок, которые оставил сотрудник о клиенте, а также просмотр, редактирование и удаление существующих заметок;
* возможность добавления, редактирования и удаления информации о сотрудниках;
* возможность добавление новых сотрудников, а также редактирование информации о сотрудниках и их удаление;
* возможность просмотра всех ролей, и кто из сотрудников к какой роли относится;
* возможность просмотра, редактирования и удаления своего профиля.

И так же необходимо реализовать нефункциональные требования:

* интерфейс программы должен быть интуитивно понятным и удобным для использования;
* программа должна быть масштабируемой и легко поддерживаемой;
* безопасность данных должна быть обеспечена путем использования аутентификации и авторизации пользователей, а также шифрованием данных в базе данных;
* программа должна быть разработана в соответствии с принципами объектно-ориентированного программирования и архитектурой MVC;
* программа должна быть разработана с использованием современных технологий и инструментов разработки;
* разработка приложения должна вестись в среде разработки Visual Studio 2022 или более поздней версии.

**\*клик\***

К требованиям технических характеристик CRM системы можно отнести:

операционная система: Windows 7/8/10

процессор: Intel Core i3 @ 2.4 и выше

оперативная память: 2 Gb и больше

манипуляторы: Клавиатура и Мышь

**\*клик\*** шестой слайд – схема прецедентов

Несмотря на то, что цели уже поставлены, перед нами всё равно пустой проект. Для начала разберёмся с действующими лицами системы. В этом нам поможет схема прецедентов.

Схема прецедентов отражает отношения между пользователями и прецедентами.

Прецедент – это часть функциональности системы. Часть, благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат.

**\*клик\***

В разрабатываемой crm системе существует один субъект «Пользователь», имеющий следующие возможности:

* регистрация нового клиента - этот прецедент позволяет оператору колл-центра зарегистрировать нового клиента в базе данных CRM системы. Оператор должен заполнить форму с персональными данными клиента, контактной информацией и другими необходимыми полями;
* поиск клиента - этот прецедент позволяет оператору колл-центра найти клиента в базе данных CRM системы по различным параметрам, таким как имя, фамилия, номер телефона и т.д.;
* обновление информации о клиенте - этот прецедент позволяет оператору колл-центра обновить информацию о клиенте, например, изменить его адрес или номер телефона;
* создание и обработка заметки о клиенте - этот прецедент позволяет оператору колл-центра создать замету на обслуживание. Оператор должен заполнить форму с описанием проблемы или запросом клиента;

**\*клик\*** седьмой слайд – функ схема

Далее необходимо продумать, каким образом будут связаны отдельные функциональные части системы. Это реализовано с помощью функциональной схемы, которая и разъясняет процессы системы.

**\*клик\***

Входными данными являются данные о клиентах, которые хранятся в клиентской базе. Колл-центр получает звонки от клиентов и проводит анализ проблемы, которую нужно решить. Затем колл-центр выбирает подходящего специалиста из списка доступных и передает ему информацию о проблеме клиента.

Специалист проводит дополнительный анализ проблемы и предоставляет решение. После этого колл-центр обратно связывается с клиентом и предоставляет ему решение проблемы. Если проблема не может быть решена немедленно, то сотрудник оставляет заметку о клиенте для дальнейшего рассмотрения.

**\*клик\*** восьмой слайд – послед

С субъектами системы определились. Но как этот субъект будет существовать в приложении? Что он будет делать? Ответить на эти вопросы нам поможет диаграмма последовательности.

Диаграмма последовательности – это UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта, а также взаимодействие актёров системы в рамках прецедента.

UML-диаграмма – это диаграмма, наглядно представляющая совокупность элементов модели системы в виде графа на унифицированном графическом языке объектного моделирования (Unified Modeling Language).

**\*клик\***

Представленная на диаграмме последовательности CRM система колл-центра с клиентской базой предназначена для пользователя.

Диаграмма начинается с актора, который открывает приложение. Затем контроллер возвращает актору View с главным меню.

После этого актор выбирает категорию из меню и выполняет некоторую операцию, дальше контроллер запрашивает данные у базы данных, помещает их в модель и возвращает View с результатом операции.

**\*клик\*** девятый слайд – классы

Класс – это расширяемый шаблон программного кода для создания объектов, предоставляющий начальные значения состояния и реализации поведения. Классы – неотъемлемая часть объектно-ориентированного программирования.

Диаграмма классов – это UML-диаграмма, демонстрирующая общую структуру иерархии классов системы.

**\*клик\***

При создании CRM системы колл-центр, были реализованы следующие классы:

* UsersController - это класс контроллера, который обрабатывает запросы, связанные с пользователем. Он содержит методы действий для отображения, создания, редактирования и удаления пользователей. Класс принимает экземпляр 'CRMContext' в конструкторе, чтобы иметь доступ к базе данных и выполнить соответствующие операции;
* RolesController - это класс контроллера, который обрабатывает запросы, связанные с ролями пользователей. Он содержит методы действий для отображения и получения информации о ролях пользователей. Как и в предыдущем случае, класс также принимает экземпляр 'CRMContext' в конструкторе для доступа к данным ролей;
* NotesController - это класс контроллера, который обрабатывает запросы, связанные с заметками. Он содержит методы действий для отображения, создания, редактирования и удаления заметок. Класс также использует экземпляр 'CRMContext' для доступа к данным заметок, а также для получения информации о контактах и пользователях;
* HomeController – это класс контроллера, в котором определены методы действий (action methods) для обработки запросов на главную страницу приложения (Index) и отображения страницы ошибки (Error). Он также содержит поле \_context типа CRMContext, который используется для доступа к базе данных. В нем определены методы действий для обработки запросов, связанных с контактами. Это включает отображение списка контактов (ContactsControllerIndex), отображение деталей конкретного контакта (Details), создание нового контакта (Create), редактирование существующего контакта (Edit) и удаление контакта (Delete). Также используется поле \_context для доступа к базе данных;
* в AccountsController – это класс контроллера, в котором определены методы действий для обработки запросов, связанных с учетными записями пользователей. Включает в себя методы для отображения деталей учетной записи (Details), аутентификации пользователя (Login), регистрации нового пользователя (Register), редактирования учетной записи (Edit) и удаления учетной записи (Delete). Также используется поле \_context для доступа к базе данных;

В каждом классе используется конструктор, принимающий параметр типа CRMContext. Это означает, что контекст базы данных CRMContext инжектируется в контроллер при создании экземпляра класса. Это позволяет контроллеру получать доступ к данным из базы данных и выполнять операции CRUD (создание, чтение, обновление, удаление) в контексте CRM.

Классы контроллеров используют атрибут [Authorize], который указывает на то, что методы действий требуют аутентификации пользователя. Это означает, что пользователи должны быть аутентифицированы, чтобы иметь доступ к этим методам.

* Contact – этот класс представляет собой модель контакта, которая используется в CRM системе для хранения информации о клиентах. Он содержит свойства для идентификации контакта (Id), ФИО (FIO), номера телефона (Phone), области (Region) и города (City), а также флаг IsDeleted, который указывает на то, был ли контакт удален из системы. Класс также содержит атрибуты данных, которые определяют правила валидации для каждого свойства, например, минимальную и максимальную длину строки, регулярное выражение для проверки формата номера телефона и т.д.
* LoginModel – этот класс представляет собой модель для входа в систему, которая используется для аутентификации пользователей. Он содержит свойства для ввода имени пользователя (UserName) и пароля (Password), которые используются для проверки подлинности пользователя при входе в систему. Класс также содержит атрибуты данных, которые определяют правила валидации для каждого свойства, например, обязательность заполнения поля и формат пароля. Эти правила помогают гарантировать правильность и безопасность данных, вводимых пользователем при входе в систему.
* Note – этот класс представляет собой модель для заметки, которая содержит свойства для идентификационного номера (Id), содержания заметки (Content), идентификационного номера контакта (ContactId), идентификатора пользователя (UserId) и флага удаления (IsDeleted). Свойство Content имеет атрибуты данных, которые определяют ограничения на длину текста и обязательность заполнения поля. Свойства ContactId и UserId также имеют атрибуты данных, которые определяют обязательность заполнения поля. Флаг IsDeleted используется для пометки заметки как удаленной
* Role - этот класс представляет собой модель для роли, которая содержит свойства для идентификационного номера (Id) и названия роли (Name). Свойство Name имеет атрибуты данных, которые определяют ограничения на длину текста и обязательность заполнения поля. Этот класс используется для хранения информации о ролях пользователей в системе, где требуется определение уровня доступа пользователя к различным функциям системы.
* User - этот класс представляет собой модель для пользователя, который содержит свойства для идентификационного номера (Id), имени (Name), фамилии (Surname), даты рождения (DateOfBirth), логина (Login), пароля (Password), идентификатора роли (RoleId) и флага удаления (IsDeleted). Свойства Name, Surname, Login и Password имеют атрибуты данных, которые определяют ограничения на длину текста и обязательность заполнения поля. Свойство DateOfBirth имеет атрибут данных, который определяет тип данных и формат даты. Свойство RoleId имеет атрибут данных, который определяет допустимый диапазон значений и обязательность заполнения поля.

**\*клик\*** десятый слайд – заключение

В ходе выполнения дипломного проекта разработана CRM система с клиентской базой.

В результате выполнения данного дипломного проекта был разработан программный модуль, который успешно интегрирует колл-центр с клиентской базой данных. Был проведён анализ существующих решений, что позволило выбрать наиболее подходящие технологии и методы интеграции. Разработанный модуль обеспечивает автоматическую синхронизацию данных между колл-центром и клиентской базой, что значительно повышает эффективность работы и уменьшает вероятность ошибок.

Результаты данной работы могут быть использованы в различных сферах бизнеса, где требуется эффективное взаимодействие с клиентами через колл-центр. Разработанный модуль может быть адаптирован под конкретные потребности компании и интегрирован в её существующую систему управления клиентской базой данных.

Таким образом, выполнение данной дипломной работы позволило разработать программный модуль, который упрощает и оптимизирует процесс работы колл-центра, повышает качество обслуживания клиентов и способствует увеличению прибыли компании.

**\*Видео\***

* Выберите в меню нужный вам раздел.
* Вы можете зайти в раздел с пользователями, там находятся все сотрудники колл-центра.
* Вы можете зайти в раздел с заметками, тут вы можно посмотреть все заметки, оставленные о клиентах, а также самим добавлять/редактировать/удалять заметки для этого выберите соответствующую кнопку в заметке.
* Вы можете зайти в раздел с контактами, тут вы можно посмотреть все контакты, добавленные сотрудниками колл-центра, а также самим добавлять/редактировать/удалять/находить по ФИО контакты для этого выберите соответствующую кнопку.
* Вы можете зайти в раздел с ролями, тут вы можно посмотреть все роли, а также посмотреть какой сотрудник к какой роли относится.
* Ну и последнее, что вы можете сделать, так это зайти в свой профиль, там вы можете отредактировать или удалить свой аккаунт.

