ВВЕДЕНИЕ

В современном информационном обществе компьютерные клубы остаются важными учреждениями, предоставляющими услуги доступа к компьютерам и интернету. Администраторы компьютерных клубов играют ключевую роль в управлении и контроле за работой клуба, обеспечивая безопасность, эффективность и удобство для пользователей.

Приложение "Компьютерный клуб" будет предоставлять широкий набор инструментов и возможностей для управления компьютерами, пользовательскими аккаунтами, расписанием, а также мониторинга и контроля за активностью пользователей. Благодаря использованию технологии WPF, можно создать современный и привлекательный пользовательский интерфейс, соответствующий последним трендам и стандартам.

Это приложение будет способствовать эффективному функционированию компьютерного клуба, улучшению общего опыта пользователей и обеспечению безопасности и конфиденциальности данных. Оно будет предоставлять администратору все необходимые инструменты для управления и контроля работы компьютерного клуба, а также обеспечит легкость использования и интуитивно понятный интерфейс.

В итоге, разработка приложения "Компьютерный клуб" для администратора на платформе WPF с использованием языка программирования C# позволит упростить и автоматизировать ряд задач, связанных с администрированием компьютерного клуба, повысив его эффективность и качество обслуживания.

1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Описание предметной области

Компьютерный клуб представляет собой учреждение, где пользователи могут получить доступ к компьютерам, высокоскоростному интернету, программному обеспечению и другим ресурсам. Это место популярно среди студентов, профессионалов, геймеров и людей, которые временно не имеют доступа к собственным компьютерам или нуждаются в специфическом программном обеспечении.

Администратор компьютерного клуба играет важную роль в эффективном функционировании и управлении клубом. Он отвечает за регистрацию пользователей, назначение времени использования компьютеров, обеспечение безопасности и соблюдение правил поведения, а также обеспечение качественного обслуживания и решение возникающих проблем.

Анализ предметной области позволяет определить основные функциональные и нефункциональные требования, которые приложение для администратора компьютерного клуба должно удовлетворять. Это включает в себя разработку удобного и интуитивно понятного пользовательского интерфейса, обеспечение безопасности данных и конфиденциальности, а также эффективное использование ресурсов клуба для обеспечения лучшего опыта пользователей и удовлетворения их потребностей.

1.2 Обоснование выбора модели хранения данных

В качестве модели хранения данных была использована реляционная база данных.

Реляционная база данных — это совокупность взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа. Строка таблицы содержит данные об одном объекте (например, товаре, клиенте), а столбцы таблицы описывают различные характеристики этих объектов — атрибутов (например, наименование, код товара, сведения о клиенте). Записи имеют одинаковую структуру — они состоят из полей, хранящих атрибуты объекта. Каждое поле описывает только одну характеристику объекта и имеет строго определенный тип данных. Все записи имеют одни и те же поля, только в них отображаются различные информационные свойства объекта.

В реляционной базе данных каждая таблица должна иметь первичный ключ – поле или комбинацию полей, которые единственным образом идентифицируют каждую строку таблицы. Ключ должен быть уникальным и однозначно определять запись. По значению ключа можно отыскать единственную запись. Ключи служат также для упорядочивания информации в БД.

Таблицы реляционной БД должны отвечать требованиям нормализации отношений. Нормализация отношений – это формальный аппарат ограничений на формирование таблиц, который позволяет устранить дублирование, обеспечивает непротиворечивость хранимых в базе данных. Имеется три нормальные формы отношений.

Первая нормальная форма. Реляционная таблица приведена к первой нормальной форме тогда и только тогда, когда ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

Вторая нормальная форма. Реляционная таблица задана во второй нормальной форме, если она удовлетворяет требованиям первой нормальной формы и все ее поля, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом. Чтобы привести таблицу ко второй нормальной форме, необходимо определить функциональную зависимость полей. Функциональная зависимость полей — это зависимость, при которой в экземпляре информационного объекта определенному значению ключевого реквизита соответствует только одно значение описательного реквизита.

Третья нормальная форма. Таблица находится в третьей нормальной форме, если она удовлетворяет требованиям второй нормальной формы, ни одно из ее не ключевых полей не зависит функционально от любого другого не ключевого поля. Реляционные таблицы могут быть связаны друг с другом, следовательно, данные могут извлекаться одновременно из нескольких таблиц. Таблицы связываются между собой для того, чтобы в конечном счете уменьшить объем БД. Связь каждой пары таблиц обеспечивается при наличии в них одинаковых столбцов. Существуют следующие типы информационных связей:

1) один-к-одному – предполагает, что одному атрибуту первой таблицы соответствует только один атрибут второй таблицы и наоборот;

2) один-ко-многим – предполагает, что одному атрибуту первой таблицы соответствует несколько атрибутов второй таблицы;

3) многие-ко-многим – предполагает, что одному атрибуту первой таблицы соответствует несколько атрибутов второй таблицы и наоборот.

1.3 Обоснование выбора языка программирования

В качестве языка программирования было принято решение использовать объектно-ориентированный язык, то есть язык, относящийся к объектно-ориентированному программированию. Объектно-ориентированное программирование (ООП) – методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.

Одним из наиболее известных языков ООП является С# – язык, относящийся к классу языков С. Он имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML. Основные его плюсы заключаются в том, что он способен иметь четкую структуру, логичен и в то же время понятен. Он имеет достаточно большое количество так называемого "синтаксического сахара" - готовых конструкторов, которые позволяют писать гораздо меньшее количество строк, тем самым оптимизируя работу программного продукта. Еще стоит упомянуть, что все это работает на базе платформы .NET Framework. Помимо этого, приятным плюсом является наличие большого количества библиотек и шаблонов, позволяющих не тратить время на написание большого количества строк кода.

1.4 Обоснование выбора СУБД

Одним из ключевых факторов, которые позволили принять решение о выборе MS SQL Server в качестве модели хранения данных для приложения "Компьютерный клуб", является его высокая надежность и стабильность. MS SQL Server прошел множество тестов и доказал свою способность обрабатывать большие объемы данных без сбоев и потерь информации. Это обеспечивает надежную и стабильную основу для хранения и управления данными приложения компьютерного клуба.

Кроме того, MS SQL Server обладает расширенными возможностями, которые позволяют эффективно управлять данными. Система масштабирования и репликации MS SQL Server позволяет распределить нагрузку и обеспечить высокую доступность системы. Это особенно важно для компьютерного клуба, где могут быть сотни и тысячи пользователей, использующих приложение одновременно.

Резервное копирование и восстановление данных в MS SQL Server обеспечивают защиту от потери данных и позволяют быстро восстановить работоспособность системы в случае сбоя.

Другим преимуществом выбора MS SQL Server является его совместимость со стандартами SQL и языком запросов. Это обеспечивает гибкость и возможность использовать различные инструменты и системы для работы с данными. Кроме того, MS SQL Server имеет широкое сообщество разработчиков и пользователей, что облегчает доступ к документации, учебным материалам и форумам поддержки. Это позволяет разработчикам находить решения и получать советы от опытных специалистов.

Наконец, интеграция MS SQL Server с другими продуктами Microsoft является значимым фактором выбора. Если в компьютерном клубе уже используются продукты Microsoft, такие как .NET Framework, Visual Studio или Azure, то использование MS SQL Server обеспечит более гармоничную и совместимую работу в экосистеме Microsoft. Разработчики могут использовать знакомые инструменты и средства разработки, а также взаимодействовать с другими сервисами и продуктами Microsoft, что облегчит разработку, развертывание и поддержку приложения "Компьютерный клуб".

Все эти факторы, в сочетании с репутацией и широким применением MS SQL Server в индустрии, обосновывают выбор этой модели хранения данных для приложения "Компьютерный клуб".

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

2.1 Диаграмма Use-case

Диаграмма Use-case (диаграмма прецедентов) представляет собой графическое представление отношений между акторами и прецедентами, иллюстрирующее варианты использования системы. Она помогает визуализировать, как различные пользователи или внешние системы взаимодействуют с системой и какие функции или действия они могут выполнить.

На рисунке 2.1 представлены основные действия и взаимодействия, которые может выполнять администратор приложения "Компьютерный клуб". Он может управлять пользователями, управлять расписанием использования компьютеров, управлять компьютерами и мониторить активность пользователей. Каждое действие имеет свои подзадачи и возможности, которые могут быть выполнены администратором для эффективного управления компьютерным клубом.

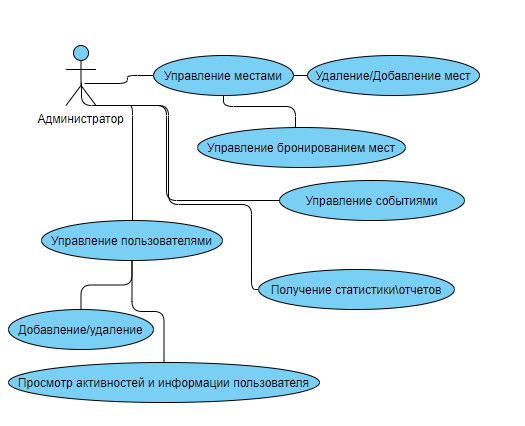


Рисунок 2.1 Диаграмма Use case