Техническое задание на учебную практику

по МДК.01.01 Разработка программных модулей

группа 403 студента Пендрагона Константина Александровича

Требуется разработать информационную систему (ИС) ООО «Playerok» – видеоигры. Товаром в данной ИС является видеоигра. Для этого требуется разработать основные программные модули ИС для следующих пользователей:

* неавторизованный клиент только может просматривать каталог видеоигр, авторизованный клиент дополнительно может просматривать свои заказы на видеоигры;
* администратор магазина может добавлять/редактировать/удалять видеоигры.

**Модуль №1. Исследование предметной области**

1. Проанализировать техническое задание, составить краткую спецификацию разрабатываемого программного решения (описание предметной области): цель ИС, актуализация, область применения, входные и выходные данные.
2. Создать диаграмму прецедентов (Use Case) пользователей ИС. ✅

**Модуль №2. Анализ и проектирование бизнес-процессов**

1. Создать контекстную диаграмму IDEF0. ✅
2. Определить основные составляющие действия в работе ИС. Представить в словесном описании и формате общей блок-схемы основной алгоритм приобретения видеоигры. Основной алгоритм представить в виде блок-схемы выполнить по правилам, установленным ГОСТ 19.701.
3. Детализировать в виде алгоритма функцию определения количества игр интересующего игрового жанра. Детальный алгоритм представить в виде блок-схемы выполнить по правилам, установленным ГОСТ 19.701.
4. Разработайте диаграмму деятельности для прецедента «Добавление новой видеоигры».

**Модуль №3. Проектирование реляционного хранилища данных**

1. Спроектировать ER-диаграмму для ИС. Обязательна 3 нормальная форма с обеспечением ссылочной целостности. При разработке диаграммы обратите внимание на согласованную осмысленную схему именования, создайте необходимые первичные и внешние ключи, определите ограничения внешних ключей, отражающие характер предметной области (типами данных на данном этапе можно пренебречь).
2. Создать словарь данных (Data Dictionary) – набор информации, описывающий, какой тип данных хранится в БД, их формат, структуру и способы использования данных. Обратите внимание на соответствие вашей диаграммы и словаря данных. Используйте подходящие типы данных, ограничения и форматы. Отразите в документе ограничения, определенные в таблицах, включая первичные ключи, отношения внешнего ключа с другими таблицами и ненулевые ограничения. Не забудьте сделать необходимые пояснения и комментарии к неоднозначным полям.
3. В проектируемой базе данных требуется хранить:

* информацию о видеоиграх: название видеоигры, игровой жанр, цена и скидка на игру, разработчик игры, дата выхода видеоигры, изображение игрового кадра, описание игрового сюжета;
* информация о клиентах: ФИО, контактные данные, паспортные данные;
* информация о пользователях ИС магазина: ФИО, аккаунт, роль в системе (администратор видеоигр, зарегистрированный клиент);
* заказы на видеоигры: информация о видеоигре, информация о клиенте, дата оформления заказ, статус заказа (отложен, в кредит, оплачен частично, оплачен полностью, возвращен), комментарий к заказу.

**Модуль №4. Разработка баз данных, объектов баз данных и импорт**

1. Создать БД ООО «Playerok» на сервере баз данных. Создайте таблицы основных сущностей, атрибуты, отношения и необходимые ограничения. Выполните названия таблиц и полей в едином стиле, согласно отраслевой документации.
2. Подготовьте данные для импорта и загрузите в разработанную базу данных.
3. Реализовать востребованные SQL-запросы к разработанной БД и оформить результаты запросов в виде отчетов. Запросы оформить в виде хранимых процедур.

**Модуль №5. Разработка desktop-приложения**

1. **Руководство по стилю**

Разработайте основные требования к оформлению окон приложения. Обеспечьте соблюдение требований всех компонентов в следующих областях:

* цветовая схема✅
* размещение логотипа✅
* использование шрифтов✅
* установка иконки приложения✅

1. **Требования и рекомендации к разработке приложения**
2. Используйте соответствующие названия для ваших приложений и файлов. Так, например, наименование настольного приложения должно обязательно включать название компании-заказчика.
3. Файловая структура проекта должна отражать логику, заложенную в приложение. Например, все окна содержатся в одной директории, классы сущностей – в другой.
4. Каждая сущность должна быть представлена в программе как минимум одним отдельным классом. Классы должны быть небольшими, понятными и выполнять одну единственную функцию (Single responsibility principle).
5. Визуальные компоненты должны соответствовать руководству по стилю.
6. Макет и технические характеристики. Все компоненты системы должны иметь единый согласованный внешний вид, соответствующий руководству по стилю, а также следующим требованиям:

* разметка и дизайн (предпочтение отдается масштабируемой компоновке);
* должно присутствовать ограничение на минимальный размер окна;
* должна присутствовать возможность изменения размеров окна, где это необходимо;
* увеличение размеров окна должно увеличивать размер контентной части, например, таблицы с данными из БД);
* группировка элементов (в логические категории);
* использование соответствующих элементов управления (например, выпадающих списков для отображения подстановочных значений из БД);
* расположение и выравнивание элементов (метки, поля для ввода и т.д.);
* последовательный переход фокуса по элементам интерфейса (по нажатию клавиши TAB);
* последовательный пользовательский интерфейс, позволяющий перемещаться между существующими окнами в приложении (в том числе обратно, например, с помощью кнопки «Назад»); ✅
* соответствующий заголовок на каждом окне приложения (не должно быть значений по умолчанию типа MainWindow, Form1 и т.п.). ✅

1. Обратная связь с пользователем. Уведомляйте пользователя о совершаемых им ошибках или о запрещенных в рамках задания действиях, запрашивайте подтверждение перед удалением, предупреждайте о неотвратимых операциях, информируйте об отсутствии результатов поиска и т.п. Окна сообщений соответствующих типов (например, ошибка, предупреждение, информация) должны отображаться с соответствующим заголовком и пиктограммой. Текст сообщения должен быть полезным и информативным, содержать полную информацию о совершенных ошибках пользователя и порядок действий для их исправления. Также можно использовать визуальные подсказки для пользователя при вводе данных.
2. Обработка ошибок:

* не позволяйте пользователю вводить некорректные значения в текстовые поля сущностей. Например, в случае несоответствия типа данных или размера поля введенному значению, оповестите пользователя о совершенной им ошибке;
* при возникновении непредвиденной ошибки приложение не должно аварийно завершать работу.

1. Оформление кода:

* идентификаторы переменных, методов и классов должны отражать суть и/или цель их использования, в том числе и наименования элементов управления (например, не должно быть значений по умолчанию типа Form1, button3);
* идентификаторы должны соответствовать соглашению об именовании (Code Convention) и стилю CamelCase (для C# и Java) и snake\_case (для Python);
* допустимо использование не более одной команды в строке.

1. Комментарии:

* используйте комментарии для пояснения неочевидных фрагментов кода. Запрещено комментирование кода;
* не используйте комментарии для пояснения очевидных действий. Комментарии должны присутствовать только в местах, которые требуют дополнительного пояснения;
* используйте тип комментариев, который в дальнейшем позволит сгенерировать XML-документацию, с соответствующими тегами (например, param, return(s), summary и др.).

1. **Требования к интерфейсу**

Разработать интерфейс программного модуля по составленному алгоритму в среде разработки в соответствии технического задания. Реализовать последовательности алгоритма по этапам (выходные данные должны соответствовать алгоритму, обрабатывающему входные данные). Реализовать алгоритм с использованием всех необходимых данных. В качестве источников данных для реализации алгоритмов используйте БД. Для работы с разными сущностями используйте разные формы, где это уместно.

1. **Функционал приложения**
2. *Авторизация*

При запуске приложения окно входа – первое, что видит пользователь. На ней пользователю предлагается ввести свой логин и пароль. Есть возможность перейти на экран просмотра каталога видеоигр в роли гостя. Только после удачной авторизации пользователь получает доступ к остальным модулям системы.

После входа в любую учетную запись должна быть реализована возможность выхода на главный экран – окно входа. При переходе в любую учетную запись в интерфейсе (правый верхний угол) должны отображаться ФИО пользователя.

При вводе пароля используется маска. Предоставлена возможность просмотра введенного пароля.

После первой попытки неуспешной авторизации система выдает сообщение о неуспешной авторизации, а затем помимо ввода логина и пароля просит ввести captcha. CAPTCHA должна содержать минимум 4 символа (буква или цифра), которые выведены не в одной линии. Символы должны быть либо перечеркнуты, либо наложены друг на друга.

Блокировка системы. После попытки неудачной авторизации с вводом captcha, система блокирует возможность входа на 10 секунд.

1. *Список каталога видеоигр*

Очень важно постоянно получать актуальную информацию о видеоиграх, которые хранятся в базе данных, согласно предоставленному макету. При отсутствии изображения игрового кадра необходимо вывести картинку-заглушку. Необходимо реализовать вывод видеоигр каталога, которые хранятся в БД, согласно предоставленному макету.

Вывод должен осуществляться списком.

**Описание видеоигры**

Название видеоигры:

Игровой жанр:

Разработчик:

~~Цена:~~

Скидка:

Цена со скидкой:

Фото игрового кадра

Пользователь должен иметь возможность искать игру по названию, сортировать видеоигры (по возрастанию и убыванию) по их стоимости.

Кроме этого, пользователь должен иметь возможность отфильтровать данные по игровым жанрам. Выбор категории должен быть из выпадающего списка. Первым элементом в выпадающем списке должен быть “Все жанры”, при выборе которого настройки фильтра сбрасываются.

В верхней части окна необходимо показывать количество выведенных данных и общее количество записей в базе.

Если у позиции товара в каталоге снижена стоимость (ненулевая скидка), то цена отображается перечеркнутым шрифтом.

1. *Добавление/редактирование видеоигры*

Необходимо реализовать возможность добавления новой позиции в каталог видеоигр и редактирования имеющейся позиции. Редактировать данные может только администратор системы.

Редактирование и добавление видеоигры выполняется в отдельном окне. Переходы на данное окно должны быть реализованы из окна каталога видеоигр. Добавление новой видеоигры осуществляется при нажатии кнопки “Добавить игру”, для редактирования – при нажатии на конкретный элемент в списке игр.

В окне должны быть предусмотрены следующие поля: название игры, игровой жанр, разработчик, описание (с возможностью многостраничного ввода), стоимость, скидка, изображение.

Пользователь может добавить/заменить изображение игрового кадра.

Проводить валидацию данных. Стоимость товара может включать сотые части, а также не может быть отрицательной. Скидка не должна превышать 30%.

После редактирования/добавления игр, данные в окне каталога видеоигр должны быть обновлены.

1. *Формирование заказа*

Реализуйте возможность формирования заказа на приобретение видеоигр из окна просмотра каталога видеоигр.

При просмотре каталога пользователь может выделить игру, по нажатию правой клавиши мыши вызвать контекстное меню с командой «Выбрать игру». При нажатии на команду меню происходит добавление выбранной игры в формируемый заказ.

При добавлении игры в заказ, в интерфейсе появляется кнопка для просмотра заказа. Просмотр заказа должен быть реализован в отдельном модальном окне. При просмотре заказа отображается вся информация о выбранной игре, в том числе изображение игровых кадров. При формировании заказа:

* ведется подсчет суммы заказа с учетом стоимости игры и величины скидки;
* автоматически формируется код заказа, который состоит из 3 цифр, генерирующихся случайным образом при создании заказа;
* информация о заказе хранится в БД.

Формируется чек к заказу с возможностью сохранения в pdf и его дальнейшей печати. Чек содержит описание заказа на товары.

**Талон заказа №Код заказа**

Дата оформления заказа:

Описание видеоигры:

Название:

Игровой жанр:

Разработчик:

Итого:

Сумма заказа без скидки:

Величина скидки:

Сумма заказа со скидкой:

**Модуль №6. Отладка модуля**

Выполнить отладку разрабатываемого программного обеспечения с использованием инструментальных средств. Сохранить и представить результаты в скриншотах. Для реализации процесса отладки использовать реализованный программный код любого окна. Продемонстрировать умения пользоваться отладчиком для следующих действий:

* установка точек останова в программном коде;
* выполнять работу приложения в режиме Отладки в пошаговом режиме, отслеживать последовательность выполнения команд;
* просматривать текущие значения переменных в пошаговом режиме с помощью наведения курсора мыши;
* отслеживать изменения значений локальных величин в окне Отладчика для просмотра локальных переменных.

**Модуль №7. Руководство пользователя**

Необходимо разработать руководство пользователя для разработанного настольного приложения, которое описывает последовательность действий для выполнения всех функций ИС.

При подготовке документации старайтесь использовать живые примеры и скриншоты вашей системы для более наглядного пояснения шагов работы с различным функционалом.