МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

РАЗРАБОТКА АСОИ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ C#, ASP.NET, SQL, Bootstrap5 ДЛЯ МАГАЗИНА ПО ПРОДАЖЕ ГИТАР «AmDm.by»

Курсовое проектирование

по дисциплине «Проектирование автоматизированных систем»

КП.1-53 01 02.10030282

Исполнитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Казымов Н.А., АСОИ-191

(подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Крутолевич С.К.

(подпись)

Дата допуска к защите \_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_

Могилёв 2022

Содержание

[Введение 4](#_Toc122090293)

[1 Анализ бизнес-процессов 5](#_Toc122090294)

[1.1 Обоснование начала разработки АСОИ 5](#_Toc122090295)

[1.2 Функциональные требования к АСОИ 5](#_Toc122090296)

[1.3 Прочие требования к АСОИ 6](#_Toc122090297)

[2 Проектирование структуры базы данных 7](#_Toc122090298)

[3 Проектирование архитектуры проекта 10](#_Toc122090299)

[3.1 Разработка диаграммы взаимодействия 10](#_Toc122090300)

[3.2 Структура классов АСОИ 10](#_Toc122090301)

[3.3 Диаграмма состояний 11](#_Toc122090302)

[4 Управление процессом разработки программного обеспечения 13](#_Toc122090303)

[4.1 Определение трудоёмкости разработки 13](#_Toc122090304)

[4.2 Отчёт о разработке программных компонентов 13](#_Toc122090305)

[5 Тестирование разработанного ПО 15](#_Toc122090306)

[Заключение 16](#_Toc122090307)

[Список использованных источников 17](#_Toc122090308)

Введение

Темой курсового проектирования является разработка АСОИ для оптимизации рабочих процессов магазина по продаже гитар «AmDm.by».

В результате внедрения системы улучшиться скорость и качество обслуживания клиентов, уменьшится время на оформление и комплектацию заказа, что, в свою очередь, способствует экономическому росту организации.

Пояснительная записка к курсовому проектированию содержит 6 разделов:

* анализ бизнес-процессов – приводится структура формируемых документов в виде таблиц;
* проектирование структуры базы данных – приводится структура БД в виде таблиц;
* проектирование архитектуры проекта – приводится структура интерфейсов, запросов и процедур в виде таблиц;
* управление процессом разработки программного обеспечения – таблицы трудоёмкости и календарный график;
* тестирование разработанного ПО – раздел включает в себя информацию по тестированию ПО;
* руководство пользователя – приводится описание процесса формирования документов с формами;
* заключение – содержит обобщение выполненной работы.

1. Анализ бизнес-процессов
   1. Обоснование начала разработки АСОИ

По требованию заказчика в качестве объекта автоматизации был выбран магазин по продаже гитар и аксессуаров к ним.

Система создаётся на основании следующих документов:

* отчёт по продажам за месяц;
* чек заказа.

Создание интернет-магазина – один из наиболее выгодных и перспективных инструментов онлайн-бизнеса, позволяющий при меньших затратах охватить большую аудиторию. Интернет-торговля на собственной платформе связана с существенно меньшим числом издержек, чем торговля в магазине. Такой вид бизнеса стал особенно актуален в условиях опасной эпидемиологической обстановки, когда обычные магазины сталкивались с ограничениями деятельности вплоть до закрытия

Проектируемую систему планируется использовать на рабочих местах сотрудников магазина музыкального оборудования.

В музыкальном магазине работают люди со средним и высшим образованием.

* 1. Функциональные требования к АСОИ

Назначение проекта – проектирование многопользовательской системы. Систему предполагается создать для улучшения качества обслуживания покупателей, учёта товарно-материальных ценностей и ускорения работы персонала магазина. Так как система позволяет увеличить скорость обслуживания, то возрастает число обслуживаемых покупателей.

Критерии оценки достижений целей системы:

* увеличение количества клиентов засчёт уменьшения времени обслуживания клиентов;
* улучшение качества обслуживания клиентов;
* увеличение скорости обработки информации о движении тмц для создания отчётов.

Система должна быть адаптивной к изменениям и простой в использовании, для большего ее распространения.

Система должна удовлетворять следующим требованиям:

* надёжность хранения данных;
* безопасность хранения данных;
* доступность системы с любого компьютера корпоративной сети;
* защищённости информации, хранящейся в системе, от внешних воздействий, хакерских атак и других аварийных ситуаций;
* квалификация персонала (персонал должен быть обучен правилам работы с ИС).

В качестве функциональных требований выступают требования по формированию документов «Чек заказа» и «Отчёт по продажам за месяц».

1. Поля документа «Чек заказа»

| Поле в документе | Обозначение |
| --- | --- |
| Номер заказа | [Id] |
| Фамилия заказчика | [Surname] |
| Имя заказчика | [Name] |
| Отчество заказчика | [Patronymic] |
| Дата заказа | [OrderDate] |
| Наименование товара | [ProductName] |
| Количество позиции товара | [ProductCount] |
| Стоимость заказа | [TotalCost] |

1. Поля документа «Отчёт по продажам за месяц»

| Поле в документе | Обозначение |
| --- | --- |
| Дата запроса | [RequestDate] |
| Артикул товара | [ProductNumber] |
| Название товара | [ProductName] |
| Цена товара | [Price] |
| Количество проданных экземпляров | [SalesCount] |
| Общая прибыль | [TotalGain] |

* 1. Прочие требования к АСОИ

Минимальные системные требования:

* процессор: Intel Core i3-4160 или соответствующий ему аналог от AMD;
* оперативная память: 2 ГБ;
* жёсткий диск: 2 ГБ;
* видеокарта: GeForce 8800 (с 512 МБ видеопамяти) или Radeon HD3850 (с 512 МБ видеопамяти);
* жёсткий диск: 500 ГБ;

Рекомендуемые системные требования:

* процессор: Intel Core i5 9700KF 3.6GHz;
* оперативная память: 8 ГБ (для Windows 10);
* видеокарта: GeForce GTX 1050ti (с 4 ГБ видеопамяти);
* жесткий диск: 1 ТБ свободного места.

1. Проектирование структуры базы данных

На основании всей предоставленной документации была разработана структура базы данных. Все таблицы приведены к третьей нормальной форме. Ниже приведена структура каждой из таблиц базы данных.

1. Структура таблицы Users

| Key | Name | Type | Not Null | Unique | Len |
| --- | --- | --- | --- | --- | --- |
| PK | u\_id | int | + | + | - |
|  | u\_login | varchar | + | + | 30 |
|  | u\_password | varchar | + | - | 18 |

1. Структура таблицы ProductTypes

| Key | Name | Type | Not Null | Unique | Len |
| --- | --- | --- | --- | --- | --- |
| PK | pt\_id | int | + | + | - |
|  | pt\_name | nvarchar | + | + | 100 |

1. Структура таблицы Products

| Key | Name | Type | Not Null | Unique | Len |
| --- | --- | --- | --- | --- | --- |
| PK | p\_number | int | + | + | - |
| FK | p\_type | int | + | - | - |
|  | p\_name | nvarchar | + | + | 200 |
|  | p\_producer | Nvarchar | + | - | 200 |
|  | p\_price | Money | + | - | - |
|  | p\_description | nvarchar | - | - | 300 |
|  | p\_amount | int | + | - | - |
|  | p\_is\_deleted | bit | + | - | - |

1. Структура таблицы Customers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Name | Type | Not Null | Unique | Len |
| PK | c\_id | int | + | + | - |
|  | c\_surname | nvarchar | + | - | 50 |
|  | c\_name | nvarchar | + | - | 50 |
|  | c\_patronymic | nvarchar | + | - | 50 |
|  | c\_phone | varchar | + | - | 13 |

1. Структура таблицы OrderStatuses

| Key | Name | Type | Not Null | Unique | Len |
| --- | --- | --- | --- | --- | --- |
| PK | os\_id | int | + | + | - |
|  | os\_name | nvarchar | + | + | - |

1. Структура таблицы ShoppingCarts

| Key | Name | Type | Not Null | Unique | Len |
| --- | --- | --- | --- | --- | --- |
| PFK | sc\_order | int | + | + | - |
| PFK | sc\_product | int | + | + | - |
|  | sc\_count | int | + | - | - |

1. Структура таблицы Staff

| Key | Name | Type | Not Null | Unique | Len |
| --- | --- | --- | --- | --- | --- |
| PK | s\_personnel\_number | int | + | + | - |
| FK | s\_user | int | - |  | - |
|  | s\_surname | nvarchar | + | - | 50 |
|  | s\_name | nvarchar | + | - | 50 |
|  | s\_patronymic | nvarchar | + | - | 50 |
|  | s\_post | nvarchar | + | - | 50 |
|  | s\_is\_fired | bit | + | - | - |

1. Структура таблицы PaymentTypes

| Key | Name | Type | Not Null | Unique | Len |
| --- | --- | --- | --- | --- | --- |
| PK | pt\_id | int | + | + | - |
|  | pt\_name | nvarchar | + | + | 75 |

1. Структура таблицы Orders

| Key | Name | Type | Not Null | Unique | Len |
| --- | --- | --- | --- | --- | --- |
| PK | o\_id | int | + | + | - |
| FK | o\_customer | int | - | - | - |
| FK | o\_employee | int | + | - | - |
| FK | o\_status | int | + | - | - |
| FK | o\_payment\_type | int | + | - | - |
|  | o\_total\_cost | money | + | - | - |
|  | o\_order\_date | datetime | + | - | - |
|  | o\_payment\_date | datetime | - | - | - |

Для отображения информационной модели рассматриваемого процесса используются следующие сущности:

* «Customer» – хранение информации о покупателе: id покупателя, фамилия, имя, отчество, контактный телефон;
* «Employee» –хранение информации о сотруднике магазина: табельный номер, id пользователя в системе, фамилия, имя, отчество, должность, флаг, уволен ли сотрудник;
* «Order» – хранение информации о заказе: id заказа, id покупателя, id сотрудника, принявшего заказ, id статуса заказа, id типа оплаты, общая стоимость заказа,

дата заказа, дата оплаты заказа.

* «OrderStatus» – хранение информации о статусах заказов: id статуса, наименование статуса заказа;
* «PaymentType» – хранение информации о типе оплаты: id типа оплаты, наименование типа оплаты;
* «Product» – хранение информации о товаре: артикул, id типа товара, наименование, производитель, цена, описание, количество на складе, флаг, удалён ли товар;
* «ProductType» – хранение информации о типе товара: id типа товара, наименование типа товара;
* «User» – хранение информации о пользователе системы: id пользователя, логин, пароль.

1. Проектирование архитектуры проекта
   1. Разработка диаграммы взаимодействия

Для описания процессов, происходящих в клиентской части web-приложения, была разработана диаграмма взаимодействия. Она описывает все на данный момент реализованные варианты использования приложения, которые доступны пользователю. Диаграмма взаимодействия представлена в графической части.

Архитектура АСОИ представлена паттерном Model-View-Controller, где:

* модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя своё состояние;
* представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели;
* контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

Для связи приложения с базой данных используется библиотека Entity Framework Core.

В таблице 3.1 перечислены классы, реализованные в ходе разработки приложения.

1. Классы АСОИ

|  |  |
| --- | --- |
| Имя класса | Стереотип класса |
| Customer | «entity» |
| Employee | «entity» |
| Order | «entity» |
| OrderStatus | «entity» |
| PaymentType | «entity» |
| Product | «entity» |
| ProductType | «entity» |
| ShoppingCart | «entity» |
| User | «entity» |

* 1. Структура классов АСОИ

Диаграмма классов – это диаграмма, которая демонстрирует общую структуру классов, их атрибутов и взаимосвязей между ними.

Атрибуты класса определяют состав и структуру данных, которые хранятся в объектах этого класса. Каждый атрибут имеет имя и тип, определяющий, какие данные он представляет.

1. Интерфейс класса

|  |  |
| --- | --- |
| Класс | |
| Поля | |
| Имя | Тип данных |
|  |  |
| Метод | |
| Имя | Тип данных |
|  |  |

1. Интерфейс класса

|  |  |
| --- | --- |
| Класс | |
| Поля | |
| Имя | Тип данных |
|  |  |
| Метод | |
| Имя | Тип данных |
|  |  |

1. Интерфейс класса

|  |  |
| --- | --- |
| Класс | |
| Поля | |
| Имя | Тип данных |
|  |  |
| Метод | |
| Имя | Тип данных |
|  |  |

1. Интерфейс класса

|  |  |
| --- | --- |
| Класс | |
| Поля | |
| Имя | Тип данных |
|  |  |
| Метод | |
| Имя | Тип данных |
|  |  |

Перечисленные выше классы, которые представляют собой контроллеры, содержат всю основную бизнес логику данного приложения. Они позволяют просматривать, изменять, добавлять и удалять информацию.

* 1. Диаграмма состояний

Диаграмма состояний используется для описания взаимодействия между программой и пользователем. Диаграмма моделирует переходы между диалоговыми формами и список внутренних действий в форме. На переходах между формами отображается имя кнопки, вызвавшей событие перехода.

Диаграмма состояний похожа на диаграмму деятельности, но деятельность в случае диаграммы состояний заменена состоянием, переходы символизируют действия.

Состояние содержит имя или имя и список внутренних действий. Список внутренних действий содержит перечень действий или деятельностей, которые выполняются во время нахождения объекта в данном состоянии. Данный список фиксированный.

Список основных действий включает в себя:

1. Управление процессом разработки программного обеспечения
   1. Определение трудоёмкости разработки

В таблице 4.1 указаны примерные трудозатраты разработки элементов АСОИ в виде количества часов, затраченного на создание того или иного элемента АСОИ.

1. Трудоёмкость разработки программного обеспечения

| Элемент | Число полей | Число диалоговых элементов | Число вычисляемых процедур | Число страниц печатной формы | Трудоёмкость разработки, ч |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. Календарный план разработки

| Компонент | Даты |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

* 1. Отчёт о разработке программных компонентов

В разработке применялся следующий стек технологий:

* платформа ASP.NET для кроссплатформенной разработки веб-приложений;
* язык C# как основной язык платформы ASP.NET;
* библиотека FastReport.Core для генерации отчётов;
* ORM-технология Entity Framework Core от компании Microsoft для доступа к данным;
* Microsoft SQL Server 2019 как СУБД
* HTML5/CSS3.

В разработке приложения использовались ORM Entity Framework Core и подход Code-First, которые позволяю непосредственно в коде C# создавать сущности базы данных, а также и контекст этой базы данных. Фрагмент кода, в котором происходит создание сущностей базы данных и контекста базы данных, приведён ниже:

1. Тестирование разработанного ПО

Заключение

Автоматизация и информатизация необходима для интернет-магазинов, так как она позволяет обрабатывать заказы в реальном времени, что в конечном счёте приводит к оптимизации торгового процесса.

Во время выполнения курсового проектирования была разработана многопользовательская информационная система магазина по продаже гитар и аксессуаров к ним.

При создании АСОИ использовался следующий стек технологий:

* язык C# и платформа ASP.NET;
* ORM-технология Entity Framework Core 7;
* библиотека FastReport.Core;
* CASE-средство Sparx System Enterprise Architect 15.2 Build 1560 05-Nov-2021;
* СУБД Microsoft SQL Server 2019;
* HTML5/CSS3.

Были разработаны такие элементы проектирования систем, как: диаграмма вариантов использования, диаграмма бизнес-процессов, диаграмма классов АСОИ, диаграмма классов БД, диаграмма последовательности, диаграмма состояний.

Проведено тестирование реализованных систем. Проведенное тестирование показало целостность и правильность составленного кода взаимодействия с созданной базой данных.

Все пункты технического задания курсового проектирования были выполнены. Таким образом, система выполняет поставленную перед ней задачу.

Список использованных источников

1. Буч, Г. Язык UML. Руководство пользователя: пер. с англ. / Г. Буч, Д. Рамбо, И. Якобсон; – 2–е изд.–М.: ДМК Пресс, 2007. – 496с.
2. Куликов, С. C. Реляционные базы данных в примерах : практическое пособие для программистов и тестировщиков / С. С. Куликов. ­ Минск: Четыре четверти, 2020. — 424 с.
3. Чамберс Джеймс, Пэкетт Дэвид, Тиммс Саймон ASP.NET Core. Разработка приложений. — СПб.: Питер, 2018. — 464 с.: ил. — (Серия «Для профессионалов»).
4. Сильвио Морето Bootstrap в примерах. / Пер. с англ. Рагимов Р.Н. / Науч.ред. Киселев А.Н. – М.: ДМК Пресс, 2017 – 314 с.: ил.
5. Смит Дж. П. Entity Framework Core в действии: пер. с англ. / Д.А. Беликова. – М.: ДМК Пресс, 2022. – 690 с.