**3. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ МАГАЗИНА АВТОЗАПЧАСТЕЙ**

**3.1 Описание средств разработки приложения**

Данный дипломный проект будет реализован в виде приложения на языке C# с помощью среды разработки Visual Studio 2013. Данная среда представляет собой интегрированный набор средств обеспечения производительности разработчика, облачных служб и расширений, которые позволяют создать уникальные приложения. [8]

В данном проекте была использована технология ADO.NET, которая предоставляет доступ к данным для приложений, основанных на Microsoft .Net и является не развитием более ранней технологии ADO, а самостоятельной технологией. В отличии от классической ADO, которая была в основном предназначена для тесно связанных клиент-серверных систем, ADO.NET больше нацелена на автономную работу с помощью объектов DataSet. Эти типы представляют локальные копии любого количества взаимосвязанных таблиц данных, каждая из которых содержит набор строк и столбцов. Однако самое фундаментальное различие между классической ADO и ADO.NET состоит в том, что ADO.NET является управляемой кодовой библиотекой, и, значит, подчиняется тем же правилам, что и любая управляемая библиотека. [8]

Также в разрабатываемой информационной системе используется реляционная структура базы данных, в ней все данные представлены в виде простых таблиц, разбитых на строки и столбцы, на пересечении которых расположены данные. Главное достоинство таблиц заключается в их понятности. С табличной информацией человек сталкивается практически каждый день. В реляционной БД строка таблицы называется записью, а столбец – полем. В общем виде реляционная БД имеет вид, представленный на рисунке 3.1.1.

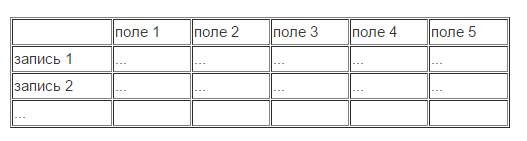


Рисунок 3.1.1. – Реляционная БД

Каждое поле таблицы имеет имя. Одна запись содержит информацию об одном объекте той реальной системы, модель которой представлена в таблице. Поля являются различными характеристиками объекта. Значения полей в одной строке относятся к одному объекту. В реляционной БД не должно быть совпадающих записей. [8]

Таблицы в реляционных базах данных обладают рядом свойств. Основными являются следующие:

* Столбцы располагаются в определенном порядке, который создается при создании таблицы. В таблице может не быть ни одной строки, но обязательно должен быть хотя бы один столбец.
* У каждого столбца есть уникальное имя (в пределах таблицы), и все значения в одном столбце имеют один тип (число, текст, дата).

В данном дипломном проекте есть две подсистемы:

– СУБД, содержащая базу данных;

– клиентская часть.

Эти подсистемы связаны между собой и связь эта работает следующим образом:

1) Пользователь выбирает нужную ему функцию;

2) Программными средствами отправляется запрос на соответствующую функции операцию над данными, содержащимися в БД;

3) СУБД пытается выполнить данную операцию и возвращает результат клиентской части. [8]

Клиентская часть состоит из 2-х подсистем:

– подсистема пользователя;

– подсистема администратора.

Функциональная часть подсистемы пользователя подразумевает, что обычному пользователю не будут доступны такие функции работы с базой данных, как добавление, редактирование и удаление информации. Пользователю доступны только две функции: поиск и просмотр информации о товарах и сотрудниках.

В данном дипломном проекте бы использована система WPF (Windows Presentation Foundation). Эта система предназначена для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая подсистема в составе .NET Framework использующая язык XAML. С помощью WPF можно создавать широкий спектр как автономных, так и запускаемых в браузере приложений. [XAML](https://ru.wikipedia.org/wiki/XAML" \o "XAML) представляет собой язык декларативного описания интерфейса, основанный на [XML](https://ru.wikipedia.org/wiki/XML). [8]

WPF предоставляет широкий спектр возможностей по созданию интерактивных настольных приложений:

- Привязка данных. Это гибкий механизм, который позволяет через расширения разметки XAML связывать различные данные (от значений свойств элементов управления до общедоступных свойств, реализующих поля базы данных через [Entity Framework](https://ru.wikipedia.org/wiki/Entity_Framework)).

- Стили**.** Позволяют создавать стилевое оформление элементов и, как правило, используются только в разметке.

- Графика**.** WPF представляет обширный, масштабируемый и гибкий набор графических возможностей. Графика, не зависит от разрешения и устройства. Основной единицей измерения в графической системе WPF является аппаратно-независимый пиксель, который составляет 1/96 часть дюйма независимо от фактического разрешения экрана. Также графическая система WPF использует преимущества графического оборудования, чтобы уменьшить использование ЦП. [8]

Каждое окно приложения состоит из двух файлов (рисунок 3.1.2):

1. С расширением .cs (Там код.. функции разные и переменные)
2. С Расширением .xaml (Там чисто разметка окна)

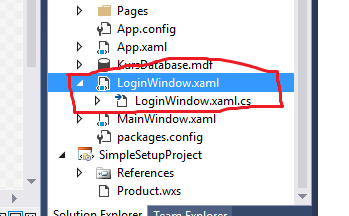


Рисунок 3.1.2 – Расширения файлов

В данном проекте была использована библиотека EntityFramework, которая помогает нам работать с базой данных. EntityFramework может создать базу данных из классов, благодаря чему нам не нужно будет думать об SQL, у нас будет код, а из него мы будем получать базу данных. Это называется методом Code-First. [8]

Также в дипломном проекте был использован шаблон для приложений ModernUI, который делает кнопки, таблицы и текстовые поля стилистически красиво оформленными. [8]

**3.2.Описание процесса информационного взаимодействия пользователя с программой**

При начале работы с нашей программой мы можем войти как администратор и как пользователь (отличие между ними только в предложенном функционале)

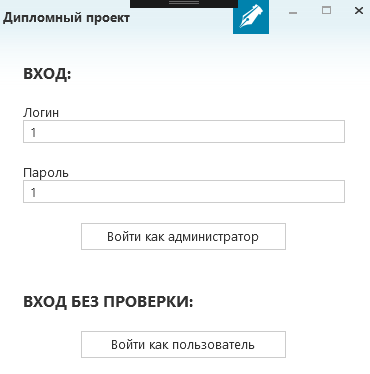


Рисунок 3.2.1 – Вход в приложение

В клиентской части дипломного проекта предусмотрено меню с основными компонентами: Персонал, Товары, Покупки.

В разделе Персонал имеется две вкладки: Должности и Персонал

При переходе по вкладке должность мы переходим в окно, в котором можем добавлять новые должности для персонала нашего магазина.

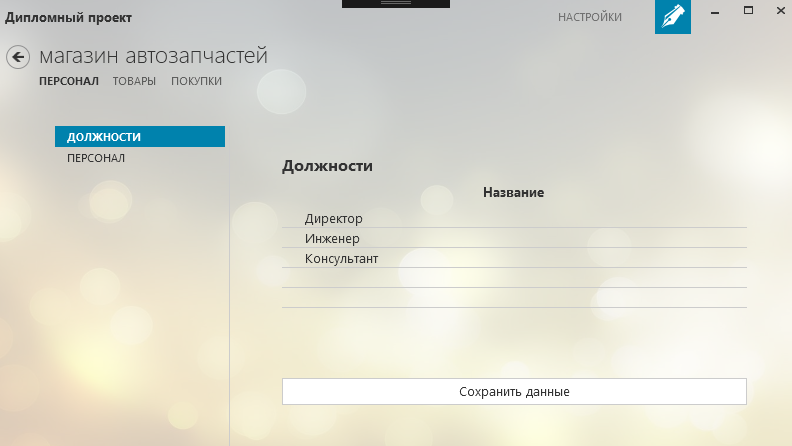


Рисунок 3.2.2 – Вкладка должности

После того как мы добавим должности для нашего персонала мы можем перейти во вкладку персонал и добавить сотрудников нашего магазина, присвоив им соответствующие должности

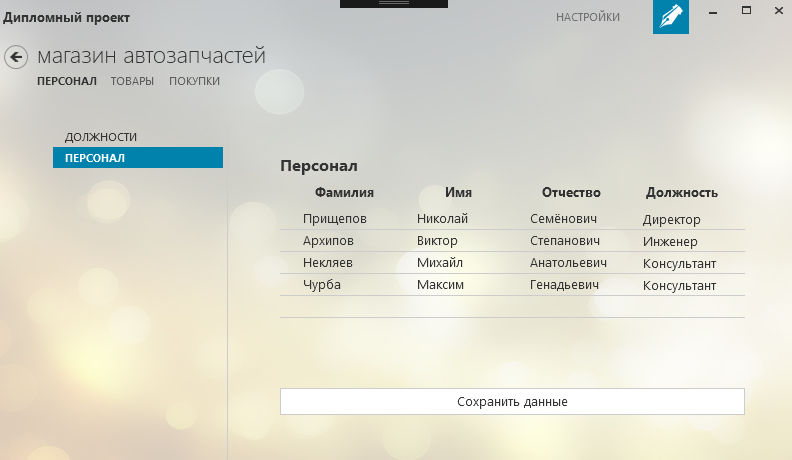


Рисунок 3.2.3 – Вкладка персонал

Далее, мы можем каждому сотруднику настроить график работы, в котором будут указаны дни недели, когда он работает, с указанием начала и конца смены.

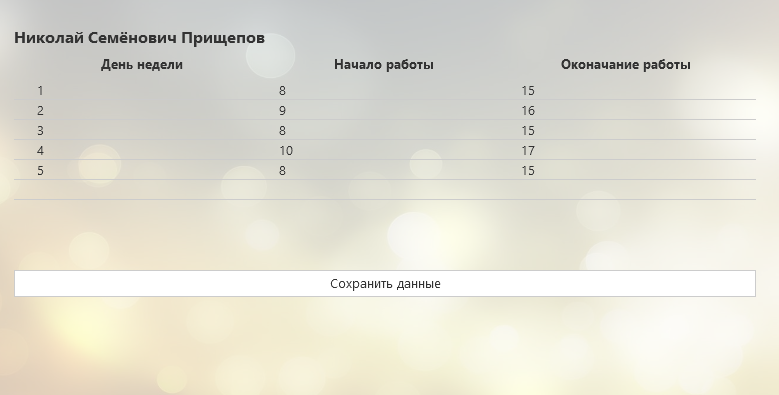


Рисунок 3.2.4 – Графики работы

В разделе Товары имеется три вкладки: Категории товаров, Брэнды автомобилей и Список товаров. Перейдя по вкладке категории товаров, мы сможем добавить необходимые нам категории товаров

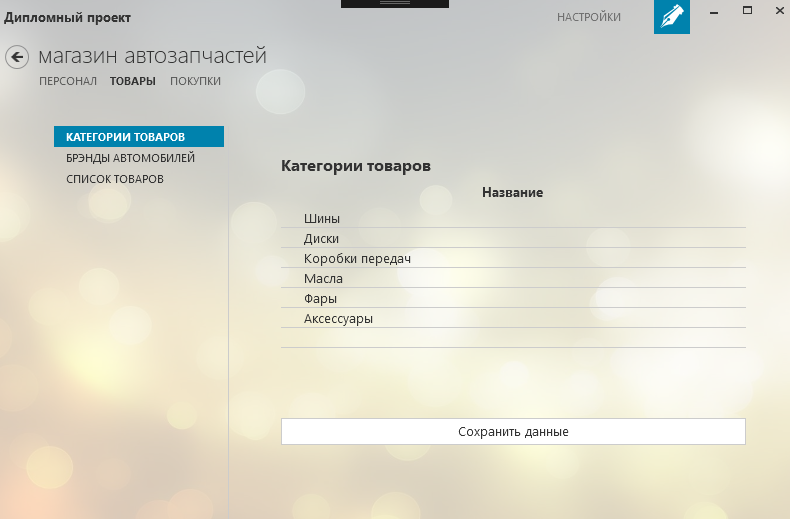


Рисунок 3.2.5 – Категории товаров

Во вкладке брэнды автомобилей мы также добавляем необходимые нам брэнды авто

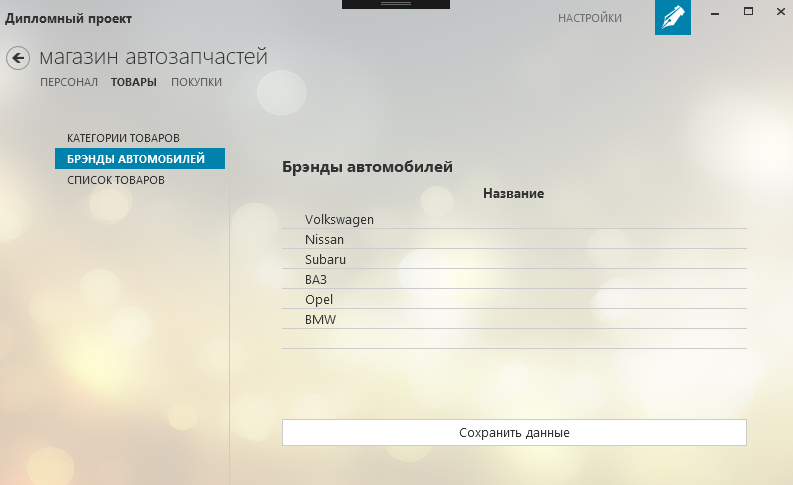


Рисунок 3.2.6 – Брэнды авто

Во вкладке список товаров храниться информация о, имеющихся в нашем магазине товарах, Администратор присваивает каждому товару определённый брэнд, категорию, название, также он устанавливает цену этого товара, пишет его описание и указывает его количество на складе. Далее пользователь сможет просматривать эти данные.



Рисунок 3.2.7 – Список товаров

После того как пользователь определился с заказом, он обращается к администратору, который в свою очередь формирует заказ и распечатывает чек.

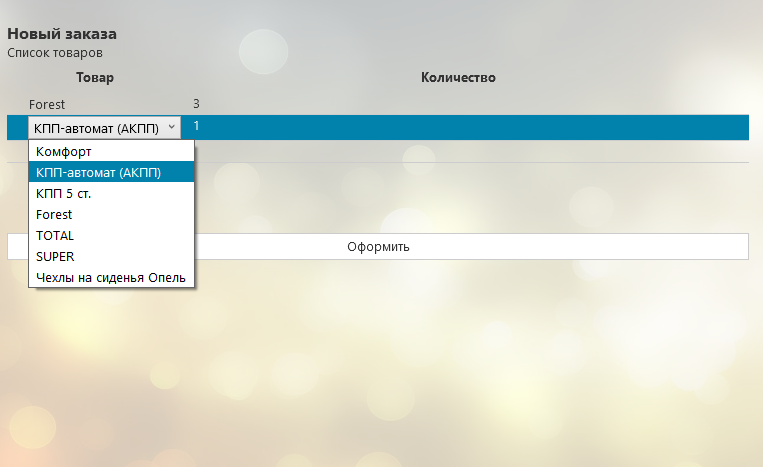


Рисунок 3.2.8 – Оформление чека

Чек формируется в формате Word, в нём указывается вся информация о заказе и рассчитывается его общая сумма, после чего чек распечатывается.

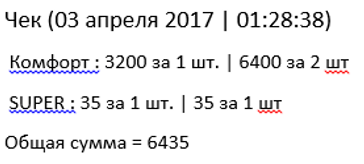


Рисунок 3.2.9 – Чек

Администрация магазина в любой момент может отслеживать все заказы, которые были осуществлены в магазине и, при необходимости, сможет подтвердить покупку того или иного товара. С указанием полученной прибыли.

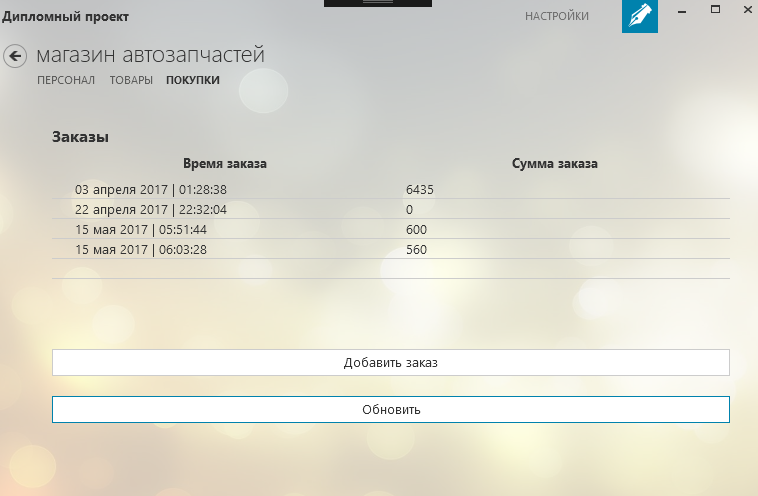


Рисунок 3.2.10 – Список заказов

**3.3.Тестирование программного модуля.**

Для тестирования программного модуля был проведён Acceptance Test (AT): полное тестирование системы или ее части как на корректных, так и на некорректных данных/сценариях. Данный вид тестирования направлен на подтверждение того, что приложение может использоваться по назначению при любых условиях. [12]

Серьезность (**Severity**) - это атрибут, характеризующий влияние дефекта на работоспособность приложения.

S1 Блокирующая (Blocker) Блокирующая ошибка, приводящая приложение в нерабочее состояние, в результате которого дальнейшая работа с тестируемой системой или ее ключевыми функциями становится невозможна. Решение проблемы необходимо для дальнейшего функционирования системы. [12]

S2 Критическая (Critical) Критическая ошибка, неправильно работающая ключевая бизнес логика, дыра в системе безопасности, проблема, приведшая к временному падению сервера или приводящая в нерабочее состояние некоторую часть системы, без возможности решения проблемы, используя другие входные точки. Решение проблемы необходимо для дальнейшей работы с ключевыми функциями тестируемой системой. [12]

S3 Значительная (Major) Значительная ошибка, часть основной бизнес логики работает некорректно. Ошибка не критична или есть возможность для работы с тестируемой функцией, используя другие входные точки. [12]

S4 Незначительная (Minor) Незначительная ошибка, не нарушающая бизнес логику тестируемой части приложения, очевидная проблема пользовательского интерфейса. [12]

S5 Тривиальная (Trivial) Тривиальная ошибка, не касающаяся бизнес логики приложения, плохо воспроизводимая проблема, малозаметная посредствам пользовательского интерфейса, проблема сторонних библиотек или сервисов, проблема, не оказывающая никакого влияния на общее качество продукта. [12]

Результаты проведения Acceptance Test представлены в виде таблицы 3.3.1, а подробное описание всех возникающих проблемных ситуаций приложения в таблице 3.3.2

Таблица 3.3.1. – Результаты Acceptance Test

|  |  |  |  |
| --- | --- | --- | --- |
| Тип теста | | AT | |
| Дата теста | | 22.04.2017 | |
| Тестировщик | | Беловский Максим Сергеевич | |
| Операционная система | | Windows 8 | |
| Результаты тестирования | | | |
| Статус | Количество | | Процент (%) |

Продолжение таблицы 3.3.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ОК | | | 45 | 75 | |
| Протестировано частично | | | 0 | 0 | |
| Trivial | | | 0 | 0 | |
| Minor | | | 7 | 11,66666667 | |
| Major | | | 0 | 0 | |
| Critical | | | 7 | 11,66666667 | |
| Blocked | | | 1 | 1,666666667 | |
| Недоступно | | | 0 | 0 | |
| Не реализовано | | | 0 | 0 | |
| Не протестировано | | | 0 | 0 | |
| Итог проверок | | | 60 | 100 | |
| Модуль | Тип теста | Функциональность | Результат | Дефекты | Комментарии |
| Окно входа в приложение | АТ | Фун: Ввод данных в ячейку логин | ОК |  |  |
| Фун: Ввод данных в ячейку пароль | ОК |  |  |
| Фун: Вход от имени пользователя | ОК |  |  |
| Фун: Вход от имени администратора | OK |  |  |
| Фун: Закрытие приложения | OK |  |  |
| Фун: Вход от имени администратора с неправильным паролем | Blocked | Выводится сообщение об ошибке | Появляется окно, в котором указано что введён неверный пароль |
| Вкладка товары | АТ | Фун: Переход во вкладку категории товаров | ОК |  |  |
| Фун: Добавление новой категории товаров вводя корректные данные | ОК |  |  |

Продолжение таблицы 3.3.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Фун: Добавление новой категории товаров вводя некорректные данные | Minor | Данные вводятся | Данные должны проходить проверку на корректность |
| Фун: Сохранение данных | OK |  |  |
| Фун: Переход во вкладку бренды автомобилей | ОК |  |  |
| Фун: Добавление нового бренда авто вводя корректные данные | OK |  |  |
| Фун: Добавление нового бренда авто вводя некорректные данные | Minor | Данные вводятся | Данные должны проходить проверку на корректность |
| Фун: Переход во вкладку список товаров | OK |  |  |
| Фун: Присваивание товару категории | OK |  |  |
| Фун: Присваивание товару бренда | ОК |  |  |
| Фун: Ввод названия товара используя корректные данные | OK |  |  |
| Фун: Ввод названия товара используя некорректные данные | Minor | Данные вводятся | Данные должны проходить проверку на корректность |
| Фун: Ввод цены товара используя корректные данные | OK |  |  |

Продолжение таблицы 3.3.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Фун: Ввод цены товара используя некорректные данные | Critical | Поле подсвечивается красным цветом | Невозможно продолжить работу программы с некорректными данными |
| Вкладка персонал | АТ | Фун: Переход во вкладку должности | ОК |  |  |
| Фун: Добавление новой должности используя корректные данные | ОК |  |  |
| Фун: Добавление новой должности используя некорректные данные | Minor | Данные вводятся | Данные должны проходить проверку на корректность |
| Фун: Переход во вкладку персонал | ОК |  |  |
| Фун: Добавление фамилии работника вводя корректные данные | ОК |  |  |
| Фун: Добавление фамилии работника вводя некорректные данные | Minor | Данные вводятся | Данные должны проходить проверку на корректность |
| Фун: Добавление имени работника вводя корректные данные | OK |  |  |
| Фун: Добавление имени работника вводя некорректные данные | Minor | Данные вводятся | Данные должны проходить проверку на корректность |
| Фун: Добавление отчества работника вводя корректные данные | OK |  |  |

Продолжение таблицы 3.3.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Фун: Добавление отчества работника вводя некорректные данные | Minor | Данные вводятся | Данные должны проходить проверку на корректность |
| Фун: Присваивание работнику должности | OK |  |  |
| Фун: Просмотр рассписания работы работника | OK |  |  |
| Фун: Ввод корректных данных дня недели | OK |  |  |
| Фун: Ввод некорректных данных дня недели | Critical | Поле подсвечивается красным цветом | Невозможно продолжить работу программы с некорректными данными |
| Фун: Ввод корректных данных начала работы | OK |  |  |
| Фун: Ввод некорректных данных начала работы | Critical | Поле подсвечивается красным цветом | Невозможно продолжить работу программы с некорректными данными |
| Фун: Ввод корректных данных окончания работы | OK |  |  |
| Фун: Ввод некорректных данных окончания работы | Critical | Поле подсвечивается красным цветом | Невозможно продолжить работу программы с некорректными данными |

Продолжение таблицы 3.3.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вкладка покупки | АТ | Фун: Добавление заказа | ОК |  |  |
| Фун: Выбор товара | ОК |  |  |
| Фун: Ввод количества товара используя корректные данные | ОК |  |  |
| Фун: Ввод количества товара используя некорректные данные | Critical | Поле подсвечивается красным цветом | Невозможно продолжить работу программы с некорректными данными |
| Фун: Оформление заказа | ОК |  |  |
| Фун: Обновление списка заказов | ОК |  |  |
| Фун: Проссмотр списка покупов в заказе | ОК |  |  |
| Фун: Отмена заказа | ОК |  |  |
| Вкладка настройки | АТ | Фун: Переход во вкладку цветовая гамма | ОК |  |  |
| Фун: Выбор цветовой гаммы | ОК |  |  |
| Фун: Выбор темы оформления приложения | ОК |  |  |
| Фун: Выбор размера шрифта | ОК |  |  |
| Фун: Переход во вкладку резервные копии | ОК |  |  |
| Фун: Создание новой резервной копии | ОК |  |  |

Продолжение таблицы 3.3.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Фун: Восстановление выбранной резервной копии | ОК |  |  |
| Фун: Переход во вкладку администраторы | ОК |  |  |
| Фун: Ввод логина администратора используя корректные данные | ОК |  |  |
| Фун: Ввод логина администратора используя некорректные данные | Critical | Поле подсвечивается красным цветом | Невозможно продолжить работу программы с некорректными данными |
| Фун: Ввод пароля администратора используя корректные данные | ОК |  |  |
| Фун: Ввод пароля администратора используя некорректные данные | Critical | Поле подсвечивается красным цветом | Невозможно продолжить работу программы с некорректными данными |
| Фун: Сворачивание приложения | ОК |  |  |
| Фун: Максимизация и минимализация приложения | ОК |  |  |

Таблица 3.3.2. – Описание всех проблемных ситуаций

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | Headline | | Severity | Description | Result | Expected result |
| N1 | Фун: Вход от имени администратора с неправильным паролем | | Blocked | Пытаемся войти от имени администратора введя неверные данные | Выводится сообщение об ошибке | Осуществление входа в приложение от имени администратора |
| N2 | Фун: Добавление новой категории товаров вводя некорректные данные | Minor | | Вводим некорректные данные в графу категории товаров | Данные вводятся | Данные должны проходить проверку на корректность |
| N3 | Фун: Добавление нового бренда авто вводя некорректные данные | Minor | | Вводим некорректные данные в графу бренды автомобилей | Данные вводятся | Данные должны проходить проверку на корректность |
| N4 | Фун: Ввод названия товара используя некорректные данные | Minor | | Вводим некорректные данные в графу названия товаров | Данные вводятся | Данные должны проходить проверку на корректность |
| N5 | Фун: Ввод цены товара используя некорректные данные | Critical | | Ввод букв в графу цена товара | Поле подсвечивается красным цветом | Данные должны ввестись |
| N6 | Фун: Добавление новой должности используя некорректные данные | Minor | | Вводим некорректные данные в графу должности | Данные вводятся | Данные должны проходить проверку на корректность |
| N7 | Фун: Добавление фамилии работника вводя некорректные данные | Minor | | Вводим некорректные данные в графу фамилии | Данные вводятся | Данные должны проходить проверку на корректность |

Продолжение таблицы 3.3.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N8 | Фун: Добавление имени работника вводя некорректные данные | Minor | Вводим некорректные данные в графу имени | Данные вводятся | Данные должны проходить проверку на корректность |
| N9 | Фун: Добавление отчества работника вводя некорректные данные | Minor | Вводим некорректные данные в графу отчества | Данные вводятся | Данные должны проходить проверку на корректность |
| N10 | Фун: Ввод некорректных данных дня недели | Critical | Ввод букв в графу дня недели | Поле подсвечивается красным цветом | Данные должны ввестись |
| N11 | Фун: Ввод некорректных данных начала работы | Critical | Ввод букв в графу начала работы | Поле подсвечивается красным цветом | Данные должны ввестись |
| N12 | Фун: Ввод некорректных данных окончания работы | Critical | Ввод букв в графу окончания работы | Поле подсвечивается красным цветом | Данные должны ввестись |
| N13 | Фун: Ввод количества товара используя некорректные данные | Critical | Ввод букв в графу количества товаров | Поле подсвечивается красным цветом | Данные должны ввестись |
| N14 | Фун: Ввод логина администратора используя некорректные данные | Critical | Вводим некорректные данные в графу логин администратора | Поле подсвечивается красным цветом | Данные должны ввестись |

Продолжение таблицы 2.3.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N15 | Фун: Ввод пароля администратора используя некорректные данные | Critical | Вводим некорректные данные в графу пароль администратора | Поле подсвечивается красным цветом | Данные должны ввестись |

Было произведено 60 проверок и получены данные о количестве ошибок возникающих в приложении как на корректных так и на некорректных сценариях (таблица 3.3.3.). Также были выведены два графика отражающие соотношение количества ошибок друг с другом по всему приложению (рисунок 3.3.4.) и соотношение ошибок в каждой вкладке приложения (рисунок 3.3.5.)

Таблица 3.3.3. – Результаты проверки приложения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Окно входа в приложение | Вкладка товары | Вкладка персонал | Вкладка покупки | Вкладка настройки |
| ОК | 5 | 10 | 11 | 7 | 12 |
| Протестировано частично | 0 | 0 | 0 | 0 | 0 |
| Trivial | 0 | 0 | 0 | 0 | 0 |
| Minor | 0 | 3 | 4 | 0 | 0 |
| Major | 0 | 0 | 0 | 0 | 0 |
| Critical | 0 | 1 | 3 | 1 | 2 |
| Blocked | 1 | 0 | 0 | 0 | 0 |
| Недоступно | 0 | 0 | 0 | 0 | 0 |
| Не реализовано | 0 | 0 | 0 | 0 | 0 |
| Не протестировано | 0 | 0 | 0 | 0 | 0 |

Рисунок 3.3.4. – Соотношение количества ошибок по всему приложению

Рисунок 3.3.4 представляет собой круговую диаграмму которая в полной мере отображает соотношение количества ошибок различных уровней по всему приложению.

Рисунок 3.3.5. – Соотношение ошибок в каждой вкладке приложения

Рисунок 3.3.5 представляет собой столбиковую диаграмму которая в полной мере отражает количество ошибок различных уровней в каждом отдельном окне приложения

Выводы по 3 главе

Ознакомившись с различными средствами разработки приложений были выбраны наиболее подходящие для создания информационной системы магазина автозапчастей.

После выбора необходимых средств разработки и написания самой программы был составлен сценарий информационного взаимодействия с пользователем.

Затем было произведено полное тестирование программы на корректных и некорректных сценариях которое помогло выявить и устранить все дефекты разрабатываемой информационной системы.

**4. ТЕХНИКО-ЭКОНОМЧЕСКОЕ ОБОСНОВАНИЕ ЭФФЕКТИВНОСТИ, РАЗРАБОТКИ И ИСПОЛЬЗОВАНИЯ ИНФОРМАЦОННОЙ СИСТЕМЫ МАГАЗИНА АВТОЗАПЧАСТЕЙ**

**4.1 Расчет стоимостной оценки затрат**

Данное программное обеспечение представляет собой информационную систему магазина автозапчастей. Основные функции, выполняемые данным приложением следующие: возможность добавления, выборки и модификации данных в БД, реализация двух режимов работы с приложением: Режим администратора и режим пользователя, возможность оформления заказа с последующей распечаткой чека.

Разрабатываемое ПО решает задачи по хранению и получению объективной информации о наличии тех или иных товаров в магазине, а также об обслуживающем его персонале. Используется данный программный продукт непосредственно в магазине автозапчастей с целью отслеживания и внесения в базу данных товаров, а также для формирования заказов от покупателей.

ПО разрабатывается по индивидуальному заказу предприятия. Потенциальными пользователями данного приложения выступают сотрудники магазина автозапчастей.

Расчет затрат на разработку ПО производится в разрезе следующих статей:

– затраты на основную заработную плату разработчиков;

– затраты на дополнительную заработную плату разработчиков;

– отчисления на социальные нужды;

– прочие затраты (амортизация оборудования, расходы на

электроэнергию, командировочные расходы, накладные расходы и т.п.). [9]

Затраты на основную заработную плату команды разработчиков определяются исходя из состава и численности команды, размеров месячной заработной платы каждого из участников команды, а также общей трудоемкости разработки программного обеспечения.

Расчет величины основной заработной платы участников команды осуществляется по формуле 4.1.1:

Зо =чi · ti, (4.1.1)

где n – количество исполнителей, занятых разработкой конкретного ПО;

– часовая заработная плата i-го исполнителя (руб.);

– трудоемкость работ, выполняемых i-м исполнителем (ч). [9]

Часовая заработная плата определяется путем деления месячной заработной платы на количество рабочих часов в месяце (можно принять равным 168 часам). При этом месячная заработная плата определяется по фактическим данным предприятия. Трудоемкость определяется исходя из сложности разработки программного продукта. Также в затраты на основную заработную плату команды разработчиков входит и премиальный фонд, создаваемый для материального стимулирования участников команды. Его размер может браться условно в диапазоне 50-100% от размера основной заработной платы. [9]

Расчет затрат на основную заработную плату осуществляется в форме таблицы 4.1.1

Таблица 4.1.1 – Расчет затрат на основную заработную плату команды разработчиков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| №№ | Участник команды | Выполняемые работы | Месячная  заработная  плата, р. | Часовая  заработная  плата, р. | Трудоемкость работ, часов | Основная  заработная  плата, р. |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 11 | Бизнес-аналитик | Создание функциональных требований к разрабатываемому ПО, работа с документацией | 1100 | 6,55 | 240 | 1572 |
| 22 | Ведущий программист | Написание программного кода в соответствии с предъявляемыми требованиями | 1200 | 7,14 | 250 | 1785 |
| 33 | Тестировщик | Тестирование программного продукта на различных стадиях разработки | 700 | 4,16 | 180 | 748,8 |
| ПРЕМИЯ 75% | | | | | | 3079,35 |
| Итого затраты на основную заработную плату разработчиков | | | | | | 7185,15 |

Затраты на дополнительную заработную плату команды разработчиковвключает выплаты, предусмотренные законодательством о труде (оплата отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью исполнителей), и определяется по формуле 4.1.2:

Зд = , (4.1.2)

где – затраты на основную заработную плату с учетом премии (руб.);

– норматив дополнительной заработной платы, рекомендуется брать в пределах 10-20% (или по согласованию с консультантом по экономическому разделу). [9]

Отчисления на социальные нужды (в фонд социальной защиты населения и на обязательное страхование) определяются в соответствии с действующими законодательными актами по формуле 4.1.3:

Зсоц = , (4.1.3)

где – норматив отчислений на социальные нужды (согласно действующему законодательству). [9]

Расчет прочих затрат осуществляется в процентах от затрат на основную заработную плату команды разработчиков с учетом премии (табл.4.1.1) по формуле 4.1.4:

Зпз = , (4.1.4)

где – норматив прочих затрат, рекомендуется брать в пределах 100-150%. [9]

9340,695

Полная сумма затрат на разработку программного обеспечения находится путем суммирования всех рассчитанных статей затрат (см. таблицу 4.2.2).

Таблица 4.1.2 – Затраты на разработку программного обеспечения

|  |  |
| --- | --- |
| Статья затрат | Сумма, р. |
| Основная заработная плата команды разработчиков | 7185,15 |
| Дополнительная заработная плата команды разработчиков | 25 |
| Отчисления на социальные нужды | 2875 |
| Прочие затраты | 9340,695 |
| Общая сумма затрат на разработку | 20495,640375 |

**4.2 Расчет стоимостной оценки результата**

Любое программное обеспечение разрабатывается для удовлетворения какой-либо потребности, получения эффекта. В общем виде эффект может быть 2 видов:

1. Экономический эффект. Использование ПО напрямую влияет на экономические показатели деятельности пользователя

2. Неэкономический эффект. Это эффект, напрямую не связанный с экономическими результатами деятельности компании: социальный, экологический, политический, технический. Данный эффект зачастую невозможно оценить в стоимостном выражении, либо процесс оценки является сложным, трудоемким и неточным. [9]

Экономический эффект при разработке программного обеспечения по индивидуальному заказу.

Экономический эффект рассчитывается для организации-разработчика. Цена любого продукта состоит из трех частей: себестоимости (суммы всех затрат на производство и реализацию продукта), прибыли и косвенных налогов (т.е. налогов которые включаются в цену товаров). Поскольку в Республике Беларусь в настоящее время взимается один косвенный налог – налог на добавленную стоимость, то расчет прибыли осуществляется по формуле:

П = Ц - НДС - Зр , (4.2.1)

где, Ц – цена реализации ПО заказчику (руб.);

– сумма расходов на разработку ПО (руб.);

П – прибыль, получаемая организацией-разработчиком от реализации данного ПО (руб.);

НДС – сумма налога на добавленную стоимость (руб.). [9]

Сумма налога на добавленную стоимость в таком случае рассчитывается по формуле:

, (4.2.2)

где НДС – ставка налога на добавленную стоимость, (20%).

Так как наша организация-разработчик ПО не освобождена от уплаты налога на прибыль, нам следует найти чистую прибыль:

, (4.2.3)

где – ставка налога на прибыль, %. [9]

##### **4.3 Расчет показателей эффективности использования программного продукта.**

В нашем случае использование программного обеспечение ведет к получению экономического эффекта, это значит, что необходимо сравнить размер инвестиций в разработку программного продукта (табл. 4.1.2.), и получаемый годовой экономический эффект. [9]

Так как сумма инвестиций меньше, чем сумма годового эффекта, это означает, что инвестиции окупятся менее чем за год. Поэтому следует остановиться на расчете одного показателя – рентабельности инвестиций:

, (4.3.3)

Разработка данного ПО целесообразна, так как рентабельность инвестиций составляет более 100% и, следовательно, необходимо инвестировать средства в разработку ПО вместо того чтобы положить их в банк. [9]