



ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ
«КОЛЛЕДЖ СВЯЗИ №54»
ИМЕНИ П.М. ВОСТРУХИНА

полное название образовательного учреждения

ДОПУСКАЮ К ЗАЩИТЕ

Заместитель директора по УПР *О.В.Корешков*

(дата)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Разработка информационной системы складского учета медицинский аптеки

Выпускная квалификационная работа должна быть выполнена в виде:
дипломной работы и демонстрационного экзамена

студентом группы ЗИСП11-1

Дениса Владимировича Хлутцева

(И.О.Фамилия)

(подпись, дата)

Основная профессиональная образовательная программа по специальности
09.02.07 Информационные системы и программирование

Форма обучения очная

Руководитель преподаватель

Кристина Александровна Бастрыкина

Руководитель мастер производственного обучения

Кристина Александровна Бастрыкина

(ученая степень, должность, И.О.Фамилия)

(подпись, дата)

Председатель предметной (междисциплинарной, модульной) комиссии

Виктор Александрович Плаксо

(И.О.Фамилия)

(подпись, дата)

Москва
2020



ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ
«КОЛЛЕДЖ СВЯЗИ №54»
ИМЕНИ П.М. ВОСТРУХИНА

полное название образовательного учреждения

УТВЕРЖДАЮ

Заместитель директора по УПР *О.В.Корешков*

(дата)

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Студенту

Хлутцеву Денису Владимировичу

(фамилия, имя, отчество полностью)

I. Тема выпускной

*Разработка информационной системы складского
учета медицинской аптеки*

квалификационной работы

II. Срок сдачи студентом законченной работы

9 апреля 2020 г.

III. Исходные данные

Microsoft SQL Server

Visual Studio 2016

IV. Перечень подлежащих разработке вопросов

1. *Анализ и классификация информационных систем*
2. *Обоснование проблемы*
3. *Описание объекта*
4. *Разработка информационной системы складского учета медицинской аптеки (практическая часть)*
5. _____

V. Перечень графического /иллюстрационного материала

1. *Скриншоты экрана (практическая часть)*
2. *Презентация*
3. _____
4. _____
5. _____
6. _____

VI. Дата выдачи задания «9» апреля 2020 г.

Руководитель

Бастрыкина К.А

(подпись)

Задание принял к исполнению

Хлутцев Д.В

(подпись)

« 9 » апреля 2020 г.



ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ
«КОЛЛЕДЖ СВЯЗИ №54»
ИМЕНИ П.М. ВОСТРУХИНА

полное название образовательного учреждения

КАЛЕНДАРНЫЙ ПЛАН ВКР

(с указанием сроков выполнения отдельных этапов)

для специальности 09.02.07 Информационные системы и программирование

(группа 3-ИСП11-1, 3-ИСП11-2, 3-ИСП11-3, 3-ИСП11-4вб)

Сроки преддипломной практики		4 недели с 20.04.2020 по 17.05.2020
	Выбор темы, руководителя, оформление заявления	С 03.02.2020 по 07.02.2020
	Утверждение темы ВКР	10.02.2020
	Выполнение задания по теме ВКР	с 18.05.2020 по 14.06.2020
	Предоставление отчета по практике руководителю	с 11.05.2020 по 15.05.2020
	Аттестация по практике	16.05.2020
Подготовка ВКР		4 недели с 18.05.2020 по 14.06.2020
	Утверждение задания на ВКР	1 неделя
	Подбор и анализ исходной информации	с 18.05.2020 по 24.05.2020
	Подготовка и утверждение плана (оглавления) ВКР	
	Работа над разделами (главами) и устранение замечаний руководителя ВКР	1 неделя с 25.05.2020 по 31.05.2020
	Согласование содержания ВКР, устранение замечаний	1 неделя с 01.06.2020 по 07.06.2020
	Оформление и представление руководителю полного текста работы. Получение отзыва руководителя ВКР.	1 неделя 08.06.2020
	Предоставление студентом готовой ВКР рецензенту	09.06.2020
	Сдача ДЭ	По отдельному графику
	Предзащита ВКР	10.06-13.06.2020

Руководитель _____ (подпись)

Руководитель _____ (подпись)

План принял к исполнению «___» _____ 2020 г.

_____ (подпись студента)

ОГЛАВЛЕНИЕ

Введение.....	5
Глава I. Анализ информационных систем.....	6
1.1 Информационные системы. Этапы разработки информационных систем.....	6
1.2 Методы разработки информационных систем.....	9
1.3 Выбор средств разработки.....	15
Выводы.....	15
Глава II. Разработка информационной системы складского учета медицинской аптеки.....	17
2.1 Определение функций и требований программы.....	17
2.2 Разработка базы данных для информационной системы.....	17
2.3 Разработка информационной системы.....	19
Выводы.....	23
Заключение.....	24
Список литературы.....	25

					09.02.07 – ЗИСП11-1		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.		Хлутцев Д.В.			Разработка информационной системы складского учета медицинской аптеки	Лит.	Лист
Провер.		Бастрыкина К.А					2
Рецензент						ГБПОУ КС№54 им. П.М. Вострухина	
Н. Контр.							
Утверд.							
						Листов	25

ВВЕДЕНИЕ

Стремительное развитие информационных систем, применение новейших технологий и коммуникаций в коммерческой деятельности и повседневной жизни обусловило возникновение новых экономических явлений, таких как электронная коммерция.

Электронная коммерция – это широкая сфера экономики, которая включает в себя такие виды деятельности, как создание информационных систем, мобильных приложений, различных приложений для автоматизации некоторых процессов систем и другие виды приложений. Это могут быть как примитивные программы, так и серьезные корпоративные многофункциональные приложения. При этом каждая фирма должна иметь своё приложение. Это обусловлено требованиями современного рынка и уже является объективной необходимостью. Также, перечень задач, которые могут быть реализованы приложением зависят от финансового благополучия фирмы. Интернет предоставляет новые каналы сбыта продукции, открывает широкие возможности для рекламы и маркетинговых исследований.

Для осуществления электронной коммерции необходимо современное программное приложение. На сегодняшний день существует большое количество технологий, которые позволяют делать эффективные информационные системы, удобные в использовании, как со стороны администратора, так и со стороны пользователей.

Именно эти факторы обуславливают актуальность данной выпускной квалификационной работы (ВКР) на тему «Разработка информационной системы складского учёта медицинской аптеки».

Целью ВКР является разработка информационной системы складского учёта медицинской аптеки.

Для достижения поставленной цели необходимо выполнить следующие задачи:

					09.02.07 – ЗИСП11-1	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

- ✓ проанализировать и расставить цели создания информационной системы складского учёта медицинской аптеки;
- ✓ ознакомиться с существующими системами складского учёта медицинских аптек;
- ✓ изучить устройства и компоненты современных складских систем;
- ✓ проанализировать теоретический материал, касающийся методов разработки информационной системы;
- ✓ создать информационную систему складского учёта медицинской аптеки. В рамках решения данной задачи:
 - дать характеристику информационной системы;
 - провести сравнение существующих информационных систем складского учёта и создать более инновационную информационную систему;
- ✓ произвести экономической расчет затрат, необходимых на создание информационной системы складского учёта медицинской аптеки.

Объектом исследования в данной ВКР выступает информационная система складского учёта, предметом – медицинская аптека.

В работе будут использованы следующие методы исследования:

- 1) метод познания – описание, анализ, классификация;
- 2) общенаучные методы – анализ публикационного массива по теме;
- 3) эксперимент, заключающийся в модернизации информационной системы складского учёта медицинской аптеки.

Данная выпускная квалификационная работа выполнена на 30 страница машинопечатного текста, состоит из введения, двух глав, заключения и списка использованной литературы.

ГЛАВА 1. АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ

1.1. Информационные системы. Этапы разработки информационных систем

Под информационной системой (ИС) понимается прикладная подсистема в программной реализации, которая ориентирована на сбор, систематизацию, хранение, поиск и обработку информации требуемого типа.

Информационные системы предназначены для автоматизации процессов, подчиняются конкретной логике обработки информации, предполагают ее хранение, а также представление в виде отчетов. При помощи разработки ИС реально производить интеграцию информационных ресурсов предприятий, применять их для более эффективного управления.

Основная цель информационных систем – эффективное внедрение всех ресурсов, увеличение значения управленческих принимаемых решений, ориентированных на повышении доходов. Процесс разработки ИС представлен на рисунке 1.1.

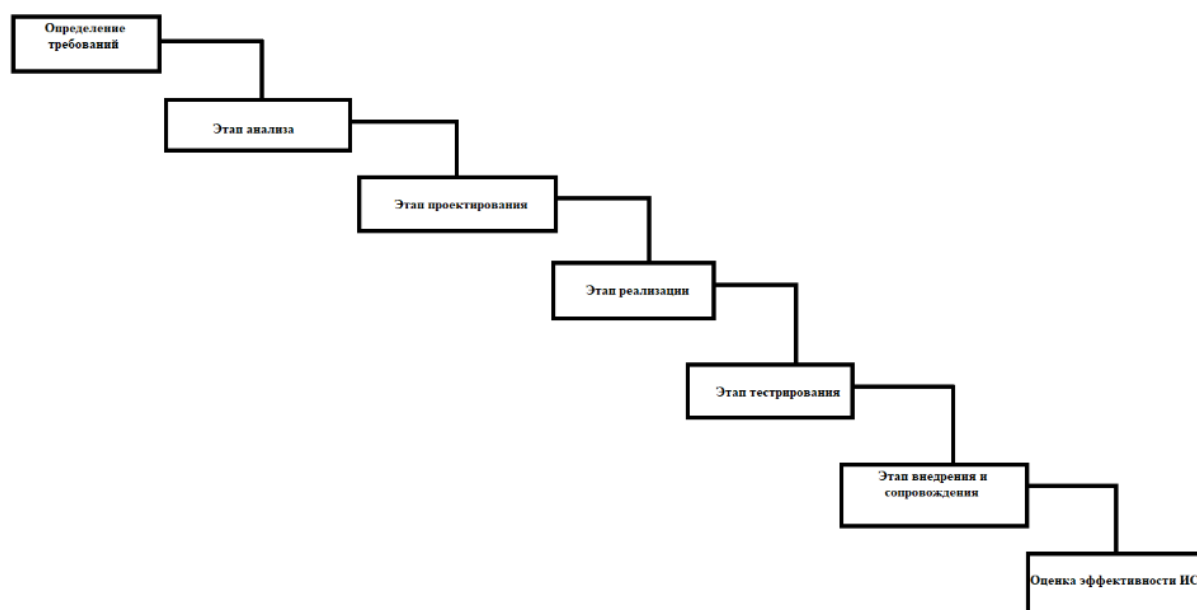


Рисунок 1.1. Этапы разработки ИС

Определение требований. Разработка любой системы начинается с постановки задачи. ИС, как правило, создается для большого количества пользователей. Каждый из них предъявляет собственные требования к системе. На этом этапе необходимо выявить всех потенциальных пользователей ИС, и для каждого из них составить список требований к ней. Так будут сформулированы основные функциональные требования к системе.

Этап анализа. Аналитическая модель структурирует функциональные требования к системе. Она описывает уже внутренний вид системы, используя язык разработчиков. Она представляет собой анализ каждого варианта использования и определяет его дальнейшую реализацию.

Этап проектирования. Это самый трудоемкий этап разработки информационной системы. На данном этапе необходимо разработать проекционную модель всей системы в целом и каждого из ее блоков. Для каждой задачи, которая будет реализована в рамках системы, необходимо описать возможные методы ее решения. Эти методы следует сравнить между собой по критериям, значимым с точки зрения системы, на основании чего выбрать лучший из них. Именно этот метод должен быть реализован впоследствии в программе. Также на этом этапе происходит проектирование базы данных. Сложные информационные системы, как правило, структурированы, т.е. представляют собой совокупность нескольких функциональных блоков. На этапе проектирования должна быть строго описана функциональность каждого из блоков. Здесь же обосновывается выбор методов интеграции блоков в единый информационный комплекс.

Этап реализации. На этапе реализации происходит непосредственно написание программы на выбранном языке программирования. В техническом задании должен быть обоснован выбор именно этого языка, а также выбор СУБД и иных программных средств.

Этап тестирования. На этапе тестирования необходимо проверить корректность функционирования системы в нормальных условиях функционирования (когда в систему вводятся корректные исходные данные),

в граничных условиях (когда на вход подаются допустимые, но редко используемые параметры или граничные параметры) и в экстремальных условиях (когда на вход системы подаются некорректные данные). Модель тестирования должна описывать результаты, которые были получены при обработке всех этих данных.

Этап внедрения и сопровождения. На этом этапе происходит обеспечение стабильной работы и снижение рисков возникновения сбоев в работе информационных систем; оперативное исправление технических неполадок в работе систем; предоставление новых версий, обновлений и дополнений, консультации по вопросам эксплуатации и администрирования информационных систем; консультации по установке и настройке новых версий, обновлений, дополнений и т.д.

Оценка эффективности ИС. На этом этапе собираются отзывы у клиента о процессе использования информационной системы и выявляются требования по улучшению ее работы.

1.2. Методы разработки информационных систем

Написание программного кода является неотъемлемой частью создания программы и информационной системы в целом.

Программирование – это процесс создания компьютерных программ, который основывается на использовании языков программирования, на которых записываются исходные тексты программ. Программа, написанная на процедурном языке, представляет собой последовательность команд, определяющих алгоритм решения задачи. Основная команда – присвоение, с помощью которой определяется и меняется память компьютера.

Основные методы программирования:

1) Алгоритмическое программирование.

Алгоритм – это формальное описание способа решения задачи путем разбиения ее на конечную по времени последовательность действий

					09.02.07 – ЗИСП11-1	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

(элементарных операций). Под словом «формальное» подразумевается, что описание должно быть абсолютно полным и учитывать все возможные ситуации, которые могут встретиться по ходу решения. Под элементарной операцией понимается действие, которое по заранее определенным критериям (например, очевидности) не имеет смысла детализировать.

В связи с этим основная идея самого простого варианта программирования алгоритмического – разбиение (декомпозиция) программы на последовательность фрагментов, каждый из которых выполняет одно или несколько действий. Линейность алгоритма и программы состоит в данном случае в характерном требовании к фрагменту – чтобы его выполнение всегда начиналось с первой команды и всегда заканчивалось на самой последней, то есть, чтобы нельзя было попасть на команды фрагмента извне и передать управление из фрагмента на другие команды в обход заключительной.

Алгоритм программы на выбранном языке записывается с помощью команд описания данных, вычисления значений и управления последовательностью выполнения программы. При этом предполагается, что текст программы представляет собой линейную последовательность операторов присваивания, цикла и условных операторов.

Для алгоритмического описания способа решения задачи в виде конечной последовательности действий характерны следующие свойства алгоритма:

- дискретность;
- определенность;
- результативность;
- массовость.

Более подробно эти свойства описаны на рисунке 1.2.

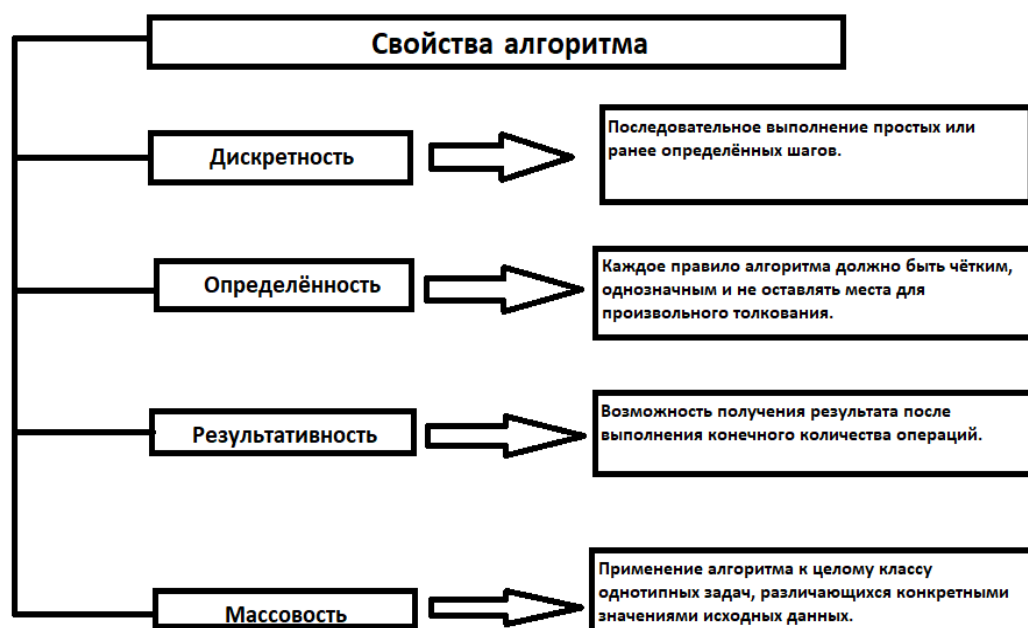


Рисунок 1.2. Свойства алгоритма.

При этом алгоритм может быть задан следующими способами: словесно, графически и сразу в виде программы.

Для представления алгоритма в графической форме, как известно, используются специальные блочные символы, с помощью которых логическая структура решения любой задачи может быть представлена комбинацией трех основных структур:

- последовательной структурой;
- условной структурой;
- циклической структурой.

Методом линейного программирования можно решать не очень сложные задачи и составлять программы, содержащие несколько сот строк кода. После этого понятность исходного текста резко падает из-за того, что общая структура алгоритма теряется за конкретными операторами языка, выполняющими слишком детальные, элементарные действия. Возникают многочисленные вложенные условные операторы и операторы циклов, логика становится совсем запутанной, при попытке исправить один ошибочный оператор вносится несколько новых ошибок, связанных с особенностями работы этого оператора, результаты выполнения которого нередко

учитываются в самых разных местах программы. Поэтому набрать и отладить длинную линейную последовательность операторов становится практически невозможно.

2) Методика структурного программирования.

При создании средних по размеру приложений чаще используется структурное программирование, идея которого заключается в том, что структура программы должна отражать структуру решаемой задачи, чтобы алгоритм решения был ясно виден из исходного текста. Для этого надо иметь средства для создания программы не только с помощью трех простых комбинаций операторов, но и с помощью средств, более точно отражающих конкретную структуру алгоритма.

С этой целью в программирование введено понятие подпрограммы — набора операторов, выполняющих нужное действие и не зависящих от других частей исходного кода. Программа разбивается на множество мелких подпрограмм, каждая из которых выполняет одно из действий, предусмотренных исходным заданием. Комбинируя эти подпрограммы, удастся формировать итоговый алгоритм уже не из простых операторов, а из законченных блоков кода, имеющих определенную смысловую нагрузку, причем обращаться к таким блокам можно по названиям. Получается, что подпрограммы — это новые операторы или операции языка, определяемые программистом.

Наличие подпрограмм позволяет вести проектирование и разработку приложения сверху вниз — такой подход называется нисходящим проектированием. Сначала выделяется несколько подпрограмм, решающих самые глобальные задачи (например, инициализация данных, главная часть и завершение), потом каждый из этих модулей детализируется на более низком уровне, разбиваясь в свою очередь на небольшое число других подпрограмм, и так происходит до тех пор, пока вся задача не окажется реализованной.

Такой подход удобен тем, что позволяет человеку постоянно мыслить на предметном уровне, не опускаясь до конкретных операторов и переменных.

Кроме того, появляется возможность некоторые подпрограммы не реализовывать сразу, а временно откладывать, пока не будут закончены другие части. Например, если имеется необходимость вычисления сложной математической функции, то выделяется отдельная подпрограмма такого вычисления, но реализуется она временно одним оператором, который просто присваивает заранее выбранное значение. Когда все приложение будет написано и отлажено, тогда можно приступить к реализации этой функции.

Немаловажно, что небольшие подпрограммы значительно проще отлаживать, что существенно повышает общую надежность всей программы.

Очень важная характеристика подпрограмм – это возможность их повторного использования. С интегрированными системами программирования поставляются большие библиотеки стандартных подпрограмм, которые позволяют значительно повысить производительность труда за счет использования чужой работы по созданию часто применяемых подпрограмм.

3) Процедуры и функции.

Возможность применения подпрограмм относит язык программирования, в котором имеется возможность их формирования, к классу процедурных языков. В таких языках подпрограммы бывают двух видов – процедуры и функции. Отличаются они тем, что процедура просто выполняет группу операторов, а функция вдобавок вычисляет некоторое значение и передает его обратно в главную программу (возвращает значение). Это значение имеет определенный тип.

Процедура может содержать такие же разделы описаний, что и программа, а именно: разделы описания модулей, меток, констант, типов, переменных, процедур и функций.

В языке программирования С# понятия «процедура» нет – там имеются только функции, а если никакого значения функция не вычисляет, то считается, что она возвращает значение типа «никакое» (void).

Чтобы работа подпрограммы имела смысл, ей надо получить данные из внешней программы, которая эту подпрограмму вызывает. Данные передаются подпрограмме в виде параметров или аргументов, которые обычно описываются в ее заголовке так же, как переменные.

Подпрограммы вызываются, как правило, путем простой записи их названия с нужными параметрами. В языке программирования Visual Basic есть оператор CALL для явного указания того, что происходит вызов подпрограммы.

Подпрограммы активизируются только в момент их вызова. Операторы, находящиеся внутри подпрограммы, выполняются, только если эта подпрограмма явно вызвана. Пока выполнение подпрограммы полностью не закончится, оператор главной программы, следующий за командой вызова подпрограммы, выполняться не будет.

Подпрограмма состоит из нескольких частей: заголовка с параметрами, тела подпрограммы (операторов, которые будут выполняться при ее вызове) и завершения подпрограммы.

4) Объектно-ориентированное программирование.

В объектно-ориентированной декомпозиции мир представлен совокупностью автономных действующих сущностей, которые взаимодействуют друг с другом, чтобы обеспечить поведение системы, соответствующее более высокому уровню. При этом, каждый объект системы обладает своим собственным поведением, и каждый из них моделирует некоторый объект реального мира. Объект является вполне осязаемой вещью, которая демонстрирует определенное поведение.

Некое множество объектов, имеющих общую структуру и общее поведение, объединяется в этом случае в классы. Понятия класса и объекта тесно связаны, однако существует важное различие. В то время как объект обозначает конкретную сущность, определенную во времени и в пространстве, класс определяет лишь абстракцию существенного в объекте. Можно говорить о классе «Млекопитающие», который включает характеристики, общие для

всех млекопитающих. Для указания конкретного представителя млекопитающих – говорят: «это – млекопитающее» или «то – млекопитающее».

Объединяя понятия структуры классов и структуры объектов с пятью признаками сложных систем, мы приходим к тому, что фактически все сложные системы можно представить одной и той же формой, которая в общем виде может быть представлена.

1.3. Выбор средств разработки

Интегрированная среда разработки – это комплекс программных средств, используемый программистами для разработки программного обеспечения (ПО).

Существуют несколько видов сред разработки, для выявления лучшей проведём сравнительный анализ в таблице 1.

Таблица 1. Сравнительный анализ.

Название	Необходимые характеристики для ПК	Достоинства	Недостатки
Visual Studio	Процессор с тактовой частотой не ниже 1,8 ГГц, 2 ГБ ОЗУ, до 130 ГБ свободного места на жёстком диске, Видеоадаптер с минимальным разрешением 720р.	Многофункциональность, бесплатность (версия Community edition), множество плагинов, облачные хранилища, корпоративность, поддержка платформ.	Баги при переходах с триал-версии, сложность.
Project Rider	Процессор с тактовой частотой не ниже 2,0 ГГц, 2 ГБ ОЗУ, до 130 ГБ свободного места на жёстком диске,	Поддержка полного цикла, функциональность, кроссплатформенность, контроль версий,	Стоимость, часть функциональности ещё в разработке.

	минимальное разрешение 1024x768.	поддержка нескольких запущенных программ.	
Eclipse	Процессор с тактовой частотой не ниже 1,5 ГГц, 1 ГБ ОЗУ, до 100 ГБ свободного места на жёстком диске.	Множество плагинов, активное сообщество, кастомизация, бесплатность, функциональность.	Сложность, нет гарантий надёжности.

Таким образом, можно сказать, что Visual Studio наиболее подходящая интегрированная среда разработки для создания информационной системы складского учёта медицинской аптеки.

Выводы

Были рассмотрены возможные виды информационных систем. Проведя сравнительный анализ IDE, можно сказать, что наиболее удачным вариантом среды разработки будет считаться Visual Studio.

ГЛАВА II. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ СКЛАДСКОГО УЧЕТА МЕДИЦИНСКОЙ АПТЕКИ

В I главе ВКР упоминалось, что для эффективного ведения бизнеса предприятию, в том числе аптеке, необходима информационная система.

Целью II главы выпускной квалификационной работы является разработка информационной системы, для автоматизации и оптимизации работы склада медицинской аптеки.

2.1. Определение функций и требований программы

Целью второй главы дипломной работы является создание информационной системы складского учёта медицинской аптеки. Программа должна иметь следующие основные возможности:

- 1) авторизация пользователей;
- 2) распределение прав пользователей в зависимости от роли (оператор, администратор);
- 3) создание, хранение, вывод и изменение данных о всех сущностях системы.

Для работы в системе пользователь должен иметь компьютер со следующими техническими характеристиками:

- процессор с тактовой частотой не менее 2,0 ГГц, с двумя и более ядер;
- оперативная память от 2 Гб;
- не менее 1ГБ свободного места на жестком диске;
- наличие доступа в глобальную сеть Интернет.

Далее, разработаем базу данных для информационной системы «Складской учёт медицинской аптеки».

2.2. Проектирование и разработка базы данных для ИС

Выделим основные сущности информационной системы:

					09.02.07 – ЗИСП11-1	Лист
						17
Изм.	Лист	№ докум.	Подпись	Дата		

- пользователи системы;
- роли пользователей;
- склад препаратов.

Приведенные сущности должны быть отражены в базе данных.

У каждого пользователя системы должны быть имя, фамилия, электронная почта, логин, пароль и роль. Создадим таблицу Пользователи с соответствующими полями (рисунок 2.1.).

DESKTOP-BLCLROD...dbo.Пользователи			
	Column Name	Data Type	Allow Nulls
PK	id	int	<input type="checkbox"/>
	Имя	nvarchar(50)	<input type="checkbox"/>
	Фамилия	nvarchar(50)	<input type="checkbox"/>
	[Электронная почта]	nvarchar(50)	<input type="checkbox"/>
	Логин	nvarchar(50)	<input type="checkbox"/>
	Пароль	nvarchar(50)	<input type="checkbox"/>
	Роль	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Рисунок 2.1. Сущность Пользователи

Далее, таблица ролей. У каждой роли есть id и название. Всего существует две роли пользователей – операторы и администратор. Создадим и сразу заполним данную таблицу как показано на рисунке 2.2.

DESKTOP-BLCLROD\...yYchet - dbo.Роль		
	id	Роль
▶	1	admin
	2	operator

Рисунок 2.2. Сущность Роль

Сущность Препараты содержит данные о названии препарата, номере на складе, количестве и поставщике. Проект таблицы можно увидеть на рисунке 2.3.

DESKTOP-BLCLROD\...- dbo.Препараты			
	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	[Название препарата]	nvarchar(50)	<input type="checkbox"/>
	[Номер на складе]	nvarchar(50)	<input type="checkbox"/>
	Количество	nvarchar(50)	<input type="checkbox"/>
	Поставщик	nvarchar(50)	<input type="checkbox"/>

Рисунок 2.3. Сущность Препараты

Далее, займемся разработкой самого приложения на языке программирования C# в среде разработки Visual Studio.

2.3. Проектирование и разработка информационной системы.

В первую очередь необходимо предоставить пользователю возможность авторизоваться в системе. Приложение производит поиск в базе данных и, если находит пользователя, открывает главное окно приложения. Также, для всего приложения необходимо выбрать единый стиль и набор цветов. Форма авторизации представлена на рисунке 2.4.

Рисунок 2.4. Форма Авторизации

При авторизации будем проверять, что пользователь существует в базе данных и переходить на соответствующую главную страницу.

Код представлен на рисунке 2.5. Полный листинг кода расположен в приложении к дипломной работе.

```

namespace Складской_учёт
{
    ссылка:3
    public partial class Authorization : Form
    {
        ссылка:1
        public Authorization()
        {
            InitializeComponent();
            textBox2.PasswordChar = '*';
        }

        ссылка:1
        private void button2_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        ссылка:1
        private void button1_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-BLCLROD\SQL EXPRESS;Initial Catalog=СкладскойYchet;Integrated Security=True");
            SqlDataAdapter sda = new SqlDataAdapter("Select Count (*) From Пользователи where Логин = '"+ textBox1.Text +"' and Пароль = '"+textBox2.Text +"'", con);
            DataTable dt = new DataTable();
            sda.Fill(dt);
            if (dt.Rows[0][0].ToString() == "1")
            {
                this.Hide();
                Sklad s = new Sklad();
                s.Show();
            }
            else
            {
                MessageBox.Show("Логин или Пароль введён не верно, пожалуйста укажите корректные данные", "Неверные данные", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}

```

Рисунок 2.5. Код страницы авторизации

Далее, создадим главную страницу для сотрудников (рисунок 2.6.), а также другие страницы.

Рисунок 2.6. Главная страница сотрудников

Функционал на главной странице сотрудников. Слева будет отображаться иконка аптеки и кнопка информации. Справа расположены кнопка «Выход», «Внести изменения», «Открыть», «Найти», «Добавить» и

«Очистить поля». По центру разместим два текстовых поля для внесения данных и две кнопки «Администрирование БД» и «Добавить оператора БД».

При нажатии кнопки информация запускается информационное табло(рисунок 2.7.).

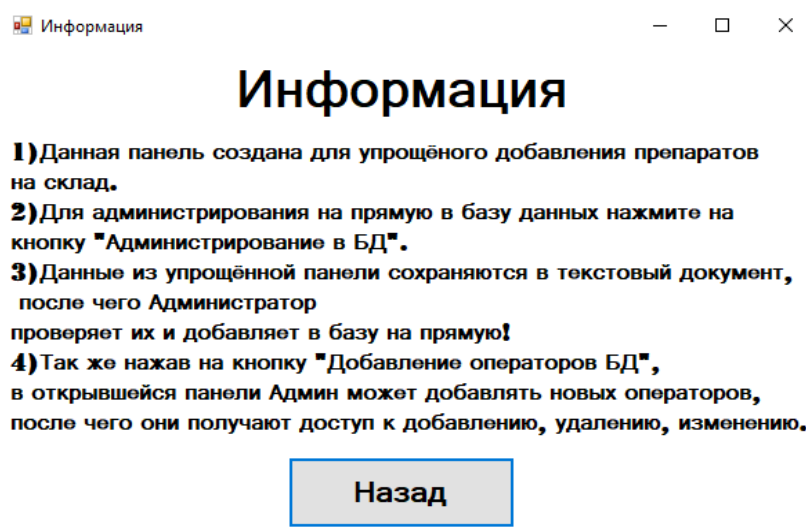


Рисунок 2.7. Информационное табло

После ознакомления с информацией что бы вернуться на главную страницу сотрудника необходимо нажать кнопку «Назад».

Кнопки на главной странице сотрудников «Выход», «Внести изменения», «Открыть», «Найти», «Добавить» и «Очистить поля» предназначены для упрощённого добавления препаратов на склад, после ввода данных и нажатия кнопки «Добавить», данные добавляются и сохраняются в текстовый документ их можно поменять с помощью кнопки «Внести изменения». После добавления данных Администратор может открыть текстовый докумен, проверить данные и внести их в базу на странице Администрирование БД. Что бы перейти на эту страницу нужно нажать кнопку «Администрирование БД» после чего будет осуществлён переход (рисунок 2.8.).

Администрирование БД						
Файл						
Обновить						
	id	Название препарата	Номер на складе	Количество	Поставщик	Command
▶	1	Тавегил	10001	100	Медторг	Delete
	2	Супрастин	10002	250	Медторг	Delete
	3	Парацетомол	10003	150	Главфарм	Delete
	4	Мезим	10004	300	Биоритм	Delete
	6	Мукалтин	10005	400	МедиаМед	Delete
*						

Рисунок 2.8. Страница администрирования БД

На данной странице представлена таблица «Препараты» из ранее подключённой нами базы данных. Так же на данной странице реализованы функции, удалить, добавить, изменить данные с помощью столбца «Command», если поменять данные в каком-либо столбце команда «Delete» (Удалить) примет значение «Update» (Обновить) и после нажатия на неё данные в таблице обновятся и сохранятся в базу данных. Если заполнить новые строки данными, пустое значение в столбце Command примет значение «Insert» (Добавить) и после нажатия на неё новые данные будут добавлены в таблицу и сохранены в базе данных. Команда «Delete» (Удалить) предназначена для удаления данных из таблицы и соответственно из базы данных. Для того что бы покинуть страницу администрирования базе данных и вернуться на главную страницу пользователей нужно нажать на вкладку Файл и в выпадающем списке нажать Выход.

Так же на главной странице пользователей есть кнопка Добавление операторов БД при нажатии на которую мы попадаем на страницу Добавление операторов БД (рисунок 2.9.).

Добавление оператора БД								
Файл								
Обновить								
	id	Имя	Фамилия	Электронная почта	Логин	Пароль	Роль	Command
▶	1	Петр	Петров	petrov22@mail.ru	admin	admin	admin	Delete
	2	Иван	Иванов	ivanov33@mail.ru	operator	oper	operator	Delete
*								

Рисунок 2.9. Страница добавление операторов БД

Работа на данной странице осуществляется так же, как на странице администрирования БД, но только администратором. Администратор может удалять и изменять данные операторов, а также добавлять новых. После того как Администратор задаёт логин и пароль нового оператора и сохраняет их в базе, оператор может осуществить вход в систему на панели авторизации и получить доступ к полному функционалу программы.

На этом разработка информационной системы складского учёта медицинской аптеки окончена.

Выводы

В результате разработки и реализации проекта складского учета медицинской аптеки стало наиболее эффективно управлять поставками на складе и пользователями приложения. Система стала более удобной в управлении, более автоматизированной благодаря функциям удаления, добавления и обновления данных.

Само приложение состоит из базы данных, а также непосредственно программы, написанной в среде разработки Visual Studio на языке программирования C#.

ЗАКЛЮЧЕНИЕ

Склады играют значительную роль в производстве, необходимо наличие автоматизированных систем складского учета товара, чтобы увеличить производительность и минимизировать затраты.

Стоит отметить, что склад – это здания (сооружения) и разнообразные устройства, которые предназначены для приемки, размещения и хранения товаров. Кроме этого, склад обладает большим количеством функций, которые позволяют производителям и потребителям эффективно взаимодействовать друг с другом, зачастую использование приложений существенно облегчает весь логистический процесс.

Безусловно, для любых складов характерно существование множества операций. Они реализуются в процессе его работы. И, конечно же, все эти операции должны быть контролируемы, и отвечали единым стандартам и требованиям. От того, как проводятся складские операции, зависит как сохранность груза, так и его целостность.

Одним из способов для автоматизации работы на складе, является внедрение информационных технологий, перевод всей информации о товарах в цифровой вид, а также написание приложения для представления данной информации в удобном виде. Именно это и было достигнуто в ходе данной выпускной квалификационной работы.

СПИСОК ЛИТЕРАТУРЫ

- 1) Абрамян, Михаил Visual C# на примерах / Михаил Абрамян. - М.: БХВ-Петербург, 2015. - 572 с.
- 2) Албахари, Джозеф C# 6.0. Справочник. Полное описание языка / Джозеф Албахари , Бен Албахари. - М.: Вильямс, 2018. - 406 с.
- 3) Вагнер, Билл. Эффективное программирование на C# / Билл Вагнер. – М.: ЛОРИ, 2017. – 320 с.
- 4) Ватсон, Б. C# 4.0 на примерах (C# 4.0. How-To) / Б. Ватсон. - М.: БХВ-Петербург, 2017. - 608 с.
- 5) Голдштейн, Саша Оптимизация приложений на платформе .Net / Саша Голдштейн , Дима Зурбалева , Идо Флатов. - М.: ДМК Пресс, 2017. - 524 с
- 6) Гэри, Маклин Холл Адаптивный код на C#. Проектирование классов и интерфейсов, шаблоны и принципы SOLID / Гэри Маклин Холл. - М.: Вильямс, 2015. - 432 с.
- 7) Культин, Н.Б. Microsoft Visual C# в задачах и примерах (+ CD-ROM) / Н.Б. Культин. - М.: БХВ-Петербург, 2017. - 308 с.
- 8) Культин, Никита Борисович Основы программирования в Microsoft Visual C# 2010 (+ CD-ROM) / Культин Никита Борисович. - М.: БХВ-Петербург, 2016. - 201 с.
- 9) Магда, Ю. С. NI Measurement Studio. Практика разработки систем измерения и управления на C# / Ю.С. Магда. - М.: ДМК Пресс, 2017. - 190 с.
- 10) Джеффри Рихтер: CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е издание/Д. Рихтер. - М.: Питер, 2019. - 436 с.
- 11) Фримен, А. LINQ. Язык интегрированных запросов в C# 2010 для профессионалов / А. Фримен. - М.: Диалектика / Вильямс, 2015. - 961 с.

Электронные ресурсы

					09.02.07 – ЗИСП11-1	Лист
Изм.	Лист	№ докум.	Подпись	Дата		25

- 1) <https://metanit.com/sharp/tutorial/1.1.php> - программирование на C#
- 2) <https://habr.com/ru/hub/csharp/> - основы программирования на C#
- 3) <http://progopedia.ru/language/csharp/> - самоучитель C#
- 4) <https://professorweb.ru/> - объектно-ориентированное программирование C#
- 5) <http://www.sql-tutorial.ru/> - базы данных MSSQL
- 6) <https://proglib.io/p/practice-ms-sql> - практика по MSSQL



ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ
«КОЛЛЕДЖ СВЯЗИ №54»
ИМЕНИ П.М. ВОСТРУХИНА

ОТЗЫВ РУКОВОДИТЕЛЯ

на выпускную квалификационную работу

Разработка информационной системы складского учета медицинской аптеки

студента

Хлутцева Дениса Владимировича

(фамилия, имя, отчество)

1. Актуальность работы:

Четко прослеживается актуальность данной работы, обусловленная необходимостью выполнения работ, связанных с разработкой информационных систем.

2. Отличительные положительные стороны работы:

В первой главе ВКР подробно изучен теоретический материал, касающийся информационных

систем: свойства, виды и этапы разработки.

Вторая глава является логическим продолжением первой и довершает целостность ощущений

от прочтения ВКР. Автор наглядно демонстрирует процесс разработки информационной системы с помощью интегрированной среды разработки Visual Studio 2016

3. Практическое значение

Продемонстрирован процесс разработки информационной системы

4. Недостатки и замечания

Работа выполнена в соответствии с требованиями, предъявляемыми к содержанию и оформлению ВКР, и заданием, выданным руководителем 20.03. 2020 г.

5. Оценка образовательных достижений студента

При выполнении ВКР Хлутцев Д.В проявил инициативу и самостоятельность в проведении исследований. Показал себя как вдумчивый, опытный и инициативный студент, который способен решать различные сложные задачи в области научных исследований как теоретического, так и экспериментального характера.

Выпускная квалификационная работа может быть допущена к защите и заслуживает оценки **«отлично»**

Руководитель Бастрыкина Кристина Александровна

(фамилия, имя, отчество)

подпись

преподаватель ГБПОУ Колледжа связи №54.

им. П.М Вострухина

ученая степень, ученое звание, должность, место работы

« 20 » мая 2020 г.