

Министерство образования Республики Беларусь  
УО «Полоцкий государственный университет»

Кафедра технологий программирования

**О Т Ч Ё Т**

о прохождении преддипломной практики  
студента(-ки) 5 курса 16ИТз группы

Журавлёвой Каралины Славомировны

в период с 23.03.2020г. по 19.04.2020г.

Наименование базы практики:	Республиканское унитарное предприятие по оказанию услуг «БелЮрОбеспечение»
-----------------------------	--

Руководитель от предприятия: (учёная степень, звание)	<u>Гончаров О.К.</u>
--	----------------------

Руководитель от кафедры: (учёная степень, звание)	<u>Данченко Е.В.</u>
--	----------------------

Практика защищена  
с оценкой \_\_\_\_\_  
Дата защиты \_\_\_\_\_

Полоцк 2020

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ .....	6
2 СРАВНИТЕЛЬНЫЙ АНАЛИЗ АНАЛОГОВ И ПРОТОТИПОВ.....	8
3 ПОСТАНОВКА ЗАДАЧИ ПРОЕКТИРОВАНИЯ .....	11
4 АНАЛИЗ ТРЕБОВАНИЙ К ПРОЕКТУ .....	12
4.1 Функциональные требования .....	12
4.2 Требования к безопасности.....	13
5 ВЫБОР И ОБОСНОВАНИЕ СРЕДСТВ И МЕТОДОВ РЕШЕНИЯ ЗАДАЧ .....	14
6 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ .....	17
6.1 Функциональная структура приложения.....	17
6.2 Проектирование логической модели данных.....	18
6.3 Разработка диаграммы развертывания системы.....	22
ЗАКЛЮЧЕНИЕ .....	24
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	25
ПРИЛОЖЕНИЕ А Техническое задание .....	26
ПРИЛОЖЕНИЕ Б Диаграмма вариантов использования.....	30
ПРИЛОЖЕНИЕ В Диаграмма «Сущность-связь».....	32
ПРИЛОЖЕНИЕ Г Диаграмма развертывания .....	33

					Отчет по преддипломной практике							
Изм.	Лист	№ докум.	Подпись	Дата								
Разраб.		Журавлёва К.С.						Лит.	Лист	Листов		
Провер.		Данченко Е.В.								3	33	
Реценз.								ПГУ 16ИТз				
Н. Контр.												
Утверд.												

## ВВЕДЕНИЕ

Преддипломная практика является неотъемлемой частью учебного процесса по образовательной программе подготовки дипломированного специалиста по специальности «Программное обеспечение информационных технологий». Целью преддипломной практики является закрепление навыков, полученных в процессе теоретического обучения, приобретение практикантами опыта в решении реальных задач, связанных с проектированием, разработкой и сопровождением программных продуктов, и сбор необходимого материала для написания дипломной работы. В ходе практики приобретаются профессиональные навыки, необходимые для последующего выполнения должностных обязанностей инженера-программиста.

В качестве основного направления во время прохождения практики были выбрана разработка информационной системы «Здравпункт среднего специального или высшего учебного заведения».

Деятельность современного человека, организации или производства невозможно представить без информационных технологий. Существующие на сегодняшний день информационные технологии и программные средства позволяют частично либо полностью автоматизировать труд человека. Автоматизация труда позволяет исключить влияние человеческого фактора. В последнее время данный вопрос стоит особо остро, поскольку в некоторых сферах производства человеческая ошибка может обернуться необратимыми последствиями.

Разработка систем, которые позволяют систематизировать и/или автоматизировать сбор и учет необходимых данных, является перспективным направлением на рынке информационных технологий. Как следствие, в рамках преддипломной практики была поставлена задача изучения возможностей платформ для разработки информационных систем с последующим проектированием структуры и функциональных возможностей программного средства в контексте выбранной предметной области.

Объектом проектирования является процесс функционирования здравпункта в учебных учреждениях. Предметом проектирования стали практические действия, которые направлены на достижение желаемого результата.

В рамках преддипломной практики необходимо решить ряд следующих задач:

- проанализировать предметную область;
- произвести сравнительный анализ аналогов и прототипов;

						Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		

- выделить ключевые функциональные требования;
- выбрать методы и средства проектирования;
- спроектировать систему, выделить основные методы обработки данных;
- сделать вывод о проделанной работе;
- разработать необходимую техническую документацию.

Разработанная информационная система позволит упростить деятельность сотрудников здравпунктов и станет непосредственным помощником при выполнении их функциональных обязанностей.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

## 1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Здравпункт – это лечебно-профилактическое учреждение, которое организуется на предприятиях, в учебных заведениях и др. В здравпункте оказывается доврачебная и первая врачебная помощь при травмах, внезапных заболеваниях, отравлениях, острой сердечно-сосудистой и дыхательной недостаточности. Здравпункт при необходимости организует транспортировку больных и пострадавших в лечебно-профилактические учреждения, проводит мероприятия по борьбе с общей заболеваемостью, по оздоровлению условий обучения и быта.

Персонал здравпункта осуществляет плановые лечебно-оздоровительные, санитарно-гигиенические и противоэпидемические мероприятия: организует медицинский осмотр учащихся; контролирует выполнение администрацией санитарных правил; проводит профилактические прививки; наблюдает за лицами, бывшими в контакте с заразными больными; обучает приемам само- и взаимопомощи при травмах; осуществляет работу с санитарным активом: участвует в разработке и анализе заболеваемости; проводит гигиеническое воспитание и пропаганду здорового образа жизни.

Здравпункты подразделяют на врачебные и фельдшерские. Они, как правило, входят в состав медсанчастей или городских (районных) больниц, поликлиник. При численности учащихся 1200 чел. и более организуется врачебный здравпункт, штат которого составляют врач, средний медработник и санитарка, при контингенте свыше 2000 чел. вводится еще одна должность среднего медработника. Фельдшерский здравпункт создается, как правило, при численности учащихся 500 чел. и более.

К функциям здравпункта в учреждениях образования относятся:

- участие в проведении мероприятий по предупреждению и снижению заболеваемости учащихся, улучшению санитарно-гигиенических условий обучения;
- ведение ежедневного амбулаторного приёма;
- ведение учёта заболеваемости учащихся;
- освобождение обучающихся при необходимости от учебных занятий и практических работ;
- проведение осмотра на амбулаторном приёме при заселении в общежитие;
- распределение на группы здоровья учащихся;
- диспансерное наблюдение отдельных групп обучающихся, имеющих хронические заболевания;

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

- иммунопрофилактика инфекционных болезней учащихся;
- ведение и заполнение необходимой медицинской документации.

Как правило, в здравпункте учёт информации ведёт фельдшер. При заселении учащихся в общежитие фельдшер фиксирует личные сведения учащихся и сведения о его состоянии здоровья. Учащийся должен предоставить фельдшеру справку о том, что он прошел медосмотр. Следующим этапом медосмотра при заселении учащихся в общежитие является осмотр учащихся на кожные заболевания. Результаты осмотра фиксируются. При каком-либо недомогании или для оказания первичной медицинской помощи учащийся обращается к фельдшеру в здравпункт, где фельдшер даёт назначения и при необходимости даёт направление на приём в поликлинику. Если направление не требуется, то фельдшер может выдать учащемуся справку об освобождении от занятий, справка может быть выдана сроком до трёх дней. Сведения об обращении учащихся в медпункт фиксируются в журнале обращений.

Не менее важной обязанностью фельдшера является назначение группы здоровья учащемуся. При наступлении нового учебного года учащийся приносит из поликлиники справку, утвержденную врачом, где указана группа здоровья учащегося. Если требуется освобождение от физических нагрузок, фельдшер подаёт сведения в учебную часть для освобождения учащегося от физической культуры и сельскохозяйственных работ.

В здравпункте ведётся учёт:

- медосмотров учащихся;
- вакцинации учащихся (вакцинация проводится по истечению срока последней вакцинации или при необходимости для предотвращения заражения каким-либо заболеванием);
- заболеваний учащихся (все заболевания учащихся фиксируются у фельдшера);
- амбулаторного приема (учащиеся обращаются при недомогании или для оказания первичной медицинской помощи, всё это фиксируется в журнале обращений);
- личных сведений учащихся (личные сведения учащихся хранятся у фельдшера для контроля за здоровьем и оказания первичной медицинской помощи).

						Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

## 2 СРАВНИТЕЛЬНЫЙ АНАЛИЗ АНАЛОГОВ И ПРОТОТИПОВ

С целью создания более совершенной и уникальной информационной системы требуется проанализировать аналоги и прототипы разрабатываемого программного средства. В ходе анализа интернет-ресурсов было выявлено два аналога системы: 1С-Рарус: Управление медицинской организацией и 1С: Медицина. Клиническая лаборатория.

### *1С-Рарус: Управление медицинской организацией*

Данный программный продукт представляет собой конфигурацию 1С и предназначен для комплексной автоматизации крупных медицинских организаций любой формы собственности, имеющих амбулаторно-поликлинические и стационарные звенья.

Главное окно «1С-Рарус: Управление медицинской организацией» представлено на рисунке 2.1.



Рисунок 2.1 – Главное окно «1С-Рарус: Управление медицинской организацией»

В данной информационной системе ведётся учёт услуг, профосмотров, вакцинаций, диагностики.

Главные достоинства данного программного средства:

- комплексная автоматизация процессов;
- существенное сокращение времени на оформление документации;
- ведение электронного расписания.

## 1С: Медицина. Клиническая лаборатория

Данный программный продукт также представляет собой конфигурацию 1С и предназначен для:

- регистрации пациентов и заказов;
- перевод заказанных услуг в термины технологического процесса лаборатории;
- регистрации результатов исследований;
- анализ результатов на соответствие норме;
- ведение лабораторных журналов;
- архив результатов;
- отчеты лаборатории по исследованиям и услугам.

Главная форма программного средства показана на рисунке 2.2.

Статус	Анализ	Методика	Результат	P..	Норма
<b>Общие свойства мочи</b>					
→	Цвет	ФУМ1	Бурый		Светло-желтый; Желтый
→	Запах мочи	ФУМ1	С запахом аммиака		Не специфический
→	Количество	ФУМ1	2		
→	Прозрачность	ФУМ1	Не полная		Не полная; Полная
→	pH	ФУМ1	2		
<b>Исследование на анализаторе м...</b>					
→	Цвет	Auion Jet	Светло-желтый		Желтый; Светло-желтый
→	Относительная плотность	Auion Jet	2		1,01-1,03
→	Реакция	Auion Jet	2		5-7
→	Белок (качественный)	Auion Jet	++		--
→	Белок (количественный)	Auion Jet	2		0-0,1
→	Глюкоза (качественная)	Auion Jet	++		--
→	Глюкоза (количественная)	Auion Jet	2		1-1
→	Кетоновые тела (качественные)	Auion Jet	++		--
→	Кетоновые тела (количественные)	Auion Jet	2		0-0
→	Реакция на кровь (качественный)	Auion Jet	++		--
→	Реакция на кровь (количественный)	Auion Jet	2		--
→	Билирубин (качественный)	Auion Jet	+		--
→	Билирубин (количественный)	Auion Jet	2		0-0
→	Уробилиноиды (качественный)	Auion Jet	++		--
→	Уробилиноиды (количественный)	Auion Jet	2		0-17
→	Реакция на лейкоциты (качественный)	Auion Jet	++		--
→	Реакция на лейкоциты (количественный)	Auion Jet	2		--
→	Нитриты (качественный)	Auion Jet	++		--
<b>Микроскопия осадка мочи</b>					
→	Лейкоциты (микроскопия)	ММ 1	0 - 2 в поле зрения		Единичные в препарате
→	Эритроциты (микроскопия)	ММ 1	Единичные в препарате		Нет; Единичные в препарате
→	Эпителий плоский	ММ 1	В небольшом количестве		В небольшом количестве
→	Эпителий переходный	ММ 1	В небольшом количестве		Нет; Единичные в препарате
→	Эпителий почечный	ММ 1	В небольшом количестве		Единичные в препарате
→	Цилиндры гиалиновые	ММ 1	5 - 10 в препарате		

Рисунок 2.2 – Окно «Регистрация результатов, анализ на соответствие норме»

Главные достоинства данной конфигурации – комплексная автоматизация процессов и существенное сокращение времени на оформление документации.



В ходе анализа аналогов и прототипов, сформировалось определённая концепция разработки продукта с целью более совершенной реализации программного средства. При проектировании и разработке дипломного проекта необходимо учесть достоинства и упущения аналогов.

Главными недостатками рассматриваемых приложений является сложный интерфейс т.к. большинство функционала не будет использоваться в здравпунктах учреждений образования. Разработка собственной информационной системы позволит решить данные проблемы, разработать нужный функционал и спроектировать интерфейс, наиболее подходящий под изученную предметную область.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		10

### 3 ПОСТАНОВКА ЗАДАЧИ ПРОЕКТИРОВАНИЯ

Целью дипломного проекта является повышение эффективности работы здравпунктов за счет разработки информационной системы, которая решит вопрос автоматизации учета обращений, вакцинации и медосмотров учащихся.

В рамках преддипломной практики поставлены следующие задачи:

- провести анализ предметной области;
- провести сравнительный анализ аналогов и прототипов;
- выделить ключевые функциональные требования к разрабатываемой системе;
- выбрать методы и средства проектирования;
- спроектировать систему, выделить основные методы обработки данных;
- разработать необходимую техническую документацию.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

## 4 АНАЛИЗ ТРЕБОВАНИЙ К ПРОЕКТУ

### 4.1 Функциональные требования

Диаграмма вариантов использования отражает взаимодействие между вариантами использования, представляющими функции системы, и действующими лицами, представляющими людей или системы, получающие или передающие информацию в данную систему.

На диаграмме вариантов использования представлено взаимодействие между вариантами использования и действующими лицами. Она отражает требования к системе с точки зрения пользователя. Таким образом, варианты использования – это функции, выполняемые системой, а действующие лица – это заинтересованные лица по отношению к создаваемой системе.

Диаграмма показывает, какие действующие лица иницируют варианты использования. В сущности, диаграмма вариантов использования иллюстрирует требования к системе.

Диаграмма вариантов использования служит внешним представлением системы. Именно поэтому нет каких-либо взаимозависимостей между вариантами использования и классами внутри системы. Диаграмма вариантов использования приложения представлена в ПРИЛОЖЕНИИ Б.

В связи с тем, что программный продукт подразумевает нескольких пользователей, то на диаграмме вариантов использования изображено два актера – администратор и фельдшер (сотрудник, студент, гость).

Администратор имеет следующие возможности:

- аутентификация;
- смена логина и пароля;
- учет личных сведений учащихся: регистрация нового учащегося, редактирование сведений об учащемся, перевод учащихся на следующий курс и групповое удаление сведений об учащихся группы.

Фельдшер имеет следующие возможности:

- аутентификация;
- смена логина и пароля;
- ведение журнала учета;
- ведение журнала амбулаторного приема;
- учет личных сведений учащихся с возможностью назначения группы здоровья учащемуся;

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

- ведение журнала медосмотров учащихся (плановой медосмотр, медосмотр при заселении в общежитие);
- ведение журнала прививок учащихся;
- получение напоминаний о необходимости проведения прививок;
- экспорт необходимых данных в Microsoft Excel.

## **4.2 Требования к безопасности**

При работе с программным средством различные ее подсистемы взаимодействуют между собой, оперируя большими потоками данных, поэтому необходимо обеспечить устойчивое функционирование приложения при бесперебойной работе персонального компьютера. При возникновении сбоя в работе аппаратуры, восстановление нормального функционирования информационной системы должно производиться при запуске приложения после перезагрузки операционной системы.

Программное средство должно обеспечить контроль корректности вводимых данных, в случае возникновения ошибок при вводе пользователь должен быть проинформирован о них соответствующими сообщениями. Необходимо обеспечить целостность данных на уровне сервера баз данных.

Ограничения по временным характеристикам функционирования системы отсутствуют.

Приложение предназначено для различных групп пользователей. Необходим некоторый набор знаний о предметной области и навыки работы с персональным компьютером.

Более подробно все требования рассмотрены в техническом задании (ПРИЛОЖЕНИЕ А).

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		
						13

## 5 ВЫБОР И ОБОСНОВАНИЕ СРЕДСТВ И МЕТОДОВ РЕШЕНИЯ ЗАДАЧ

### *Выбор среды и языка программирования*

Основными критериями при выборе средства для разработки могут быть следующие: требования заказчика (пользователя), специфические требования для разрабатываемого продукта, опыт и навыки программиста в использовании ранее средства для разработки.

На сегодняшний момент язык программирования C# один из самых мощных, быстро развивающихся и востребованных языков в ИТ-отрасли. На нем пишутся самые различные приложения: от небольших информационных систем до крупных веб-порталов и веб-сервисов, обслуживающих ежедневно миллионы пользователей.

По сравнению с другими языками C# достаточно молодой, но в то же время он уже прошел большой путь. Первая версия языка вышла вместе с релизом Microsoft Visual Studio .NET в феврале 2002 года. C# представили, как язык разработки приложений для платформы Microsoft .NET Framework. Текущей версией языка является версия C# 8.0, которая вышла в сентябре 2019 года вместе с релизом .NET Core 3.

C# является объектно-ориентированным языком. Например, C# поддерживает полиморфизм, наследование, перегрузку операторов, статическую типизацию. Объектно-ориентированный подход позволяет решить задачи по построению крупных, но в тоже время гибких, масштабируемых и расширяемых приложений. И C# продолжает активно развиваться, и с каждой новой версией появляется все больше функциональных возможностей, как, например, лямбды, динамическое связывание, асинхронные методы и т.д.

Фреймворк .NET представляет мощную платформу для создания приложений. Можно выделить следующие ее основные черты:

- поддержка нескольких языков;
- кроссплатформенность;
- мощная библиотека классов;
- разнообразие технологий (например, для работы с базами данных в этом стеке технологий предназначена технология ADO.NET и Entity Framework). [6]

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

Entity Framework представляет специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с данными. Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Отличительной чертой Entity Framework является использование запросов LINQ для выборки данных из БД. С помощью LINQ можно не только извлекать определенные строки из базы данных, но и получать объекты, связанные различными ассоциативными связями. LINQ to Entities обеспечивает поддержку LINQ при запросах к сущностям. Компонент позволяет разработчикам писать запросы к концептуальной модели Entity Framework на языке Visual Basic или Visual C#. Запросы к платформе Entity Framework представляются в виде дерева команд запроса, выполняемого на контексте объектов. Технология LINQ to Entities преобразует запросы Language-Integrated Queries (LINQ) в запросы в виде дерева команд, выполняет эти запросы на платформе Entity Framework и возвращает объекты, которые могут использоваться как платформой Entity Framework, так и технологией LINQ. [4]

Исходя из рассмотренных выше преимуществ при работе с данными и разработке приложений, было принято решение использовать язык программирования C#, организовать работу с базой данных с помощью Entity Framework и языка запросов LINQ. Данный выбор позволит качественно реализовать весь необходимый функционал разрабатываемой информационной системы.

Стоит отметить, что среда программирования Microsoft Visual Studio содержит весь необходимый набор инструментов, чтобы реализовать приложение на основе платформы .NET.

Таким образом в качестве среды программирования была выбрана Microsoft Visual Studio 2019. [8]

### *Выбор архитектуры информационной системы и СУБД*

Основная задача СУБД – дать пользователю базы данных возможность работать с ней, не вникая во все подробности работы на уровне аппаратного обеспечения. То есть, СУБД позволяет обычному пользователю рассматривать базу данных как объект более высокого уровня по сравнению

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

с аппаратным обеспечением, а также предоставляет в его распоряжение набор операций, выражаемых в терминах языка высокого уровня, например, набор операций, которые можно выполнять с помощью языка SQL (Structured Query Language).

В связи с тем, что приложение будет работать на основе платформы .NET и будет разрабатываться в среде программирования Microsoft Visual Studio, ключевой возможностью является встроенная поддержка интерфейсов разработки приложений с помощью СУБД. Microsoft Visual Studio позволяет использовать средства SQL Server Data Tools для проектирования, развертывания и обслуживания баз данных.

Главные особенности MS SQL Server Express:

- тот же механизм баз данных, что и у более сложных версий SQL Server;
- интеграция с Microsoft Visual Studio и Visual Web Developer;
- поддержка хранимых процедур, триггеров и функций;
- поддержка реляционных данных, XML и FILESTREAM;
- службы отчетов, полнотекстовый поиск и поддержка пространственных данных
- интеграция с Microsoft Office;
- поддержка Transact-SQL, ADO.NET и LINQ. [9]

Таким образом в качестве СУБД был выбран MS SQL Server Express, который сможет наиболее гибко взаимодействовать с информационной системой, так как содержит самый полный и функциональный интерфейс разработки приложений для платформы Microsoft .NET Framework.

MS SQL Server Express направлен на то, чтобы сделать управление данными самонастраивающимся, самоорганизующимся и самообслуживающимся механизмом. Это позволит уменьшить до нуля время нахождения сервера в нерабочем состоянии.

## 6 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

### 6.1 Функциональная структура приложения

Метод структурного проектирования предполагает последовательное разложение общей функции обработки данных на простые функциональные элементы («сверху-вниз»).

В результате строится иерархическая схема, отражающая состав и взаимоподчиненность отдельных функций, которая носит название функциональная структура алгоритма приложения.

Последовательность действий по разработке функциональной структуры алгоритма приложения:

- определяются цели автоматизации предметной области и их иерархия (цель-подцель);
- устанавливается состав приложений (задач обработки), обеспечивающих реализацию поставленных целей;
- уточняется характер взаимосвязи приложений и их основные характеристики (информация для решения задач, время и периодичность решения, условия выполнения и др.);
- определяются необходимые для решения задач функции обработки данных;
- выполняется декомпозиция функций обработки до необходимой структурной сложности, реализуемой предполагаемым инструментарием.

Разложение должно носить строго функциональный характер, т.е. отдельный элемент описывает законченную содержательную функцию обработки информации, которая предполагает определенный способ реализации на программном уровне.

Разрабатываемая информационная система предназначена для улучшения работы фельдшера учебного заведения, а именно автоматизация процесса учета обращений учащихся, их медосмотров, вакцинации и заболеваний.

Исходя из анализа требований к системе была разработана функциональная структура данного приложения, которая представлена на рисунке 6.1.

						Лист
						17
Изм.	Лист	№ докум.	Подпись	Дата		



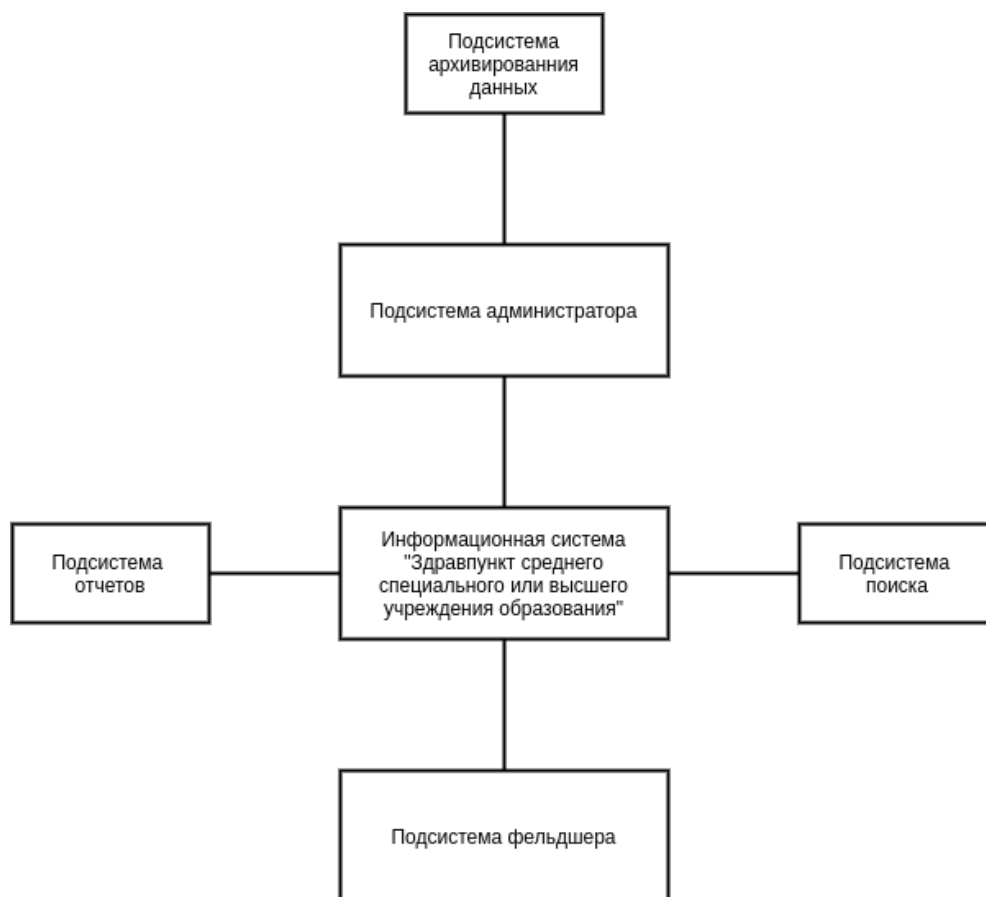


Рисунок 6.1 – Функциональная структура приложения

## 6.2 Проектирование логической модели данных

Логическая модель данных – описание объектов предметной области, их атрибутов и взаимосвязей между ними в том объеме, в котором они подлежат непосредственному хранению в базе данных системы.

Логическая модель строится в несколько этапов с постепенным приближением к оптимальному для данных условий варианту. Эффективность такой модели зависит от того, насколько близко она отображает изучаемую предметную область. К предметной области относятся объекты, а также характеристики данных объектов, их свойства, взаимодействие и взаимное влияние. Таким образом, при построении логической модели данных сначала выявляются те объекты, которые интересуют пользователей проектируемой базы данных. Затем для каждого объекта формулируются характеристики и свойства, достаточно полно описывающие данный объект. Эти характеристики в дальнейшем будут отражены в базе данных как соответствующие поля.

Одним из наиболее удобных инструментов унифицированного представления данных, независимого от реализующего его программного

обеспечения, является диаграмма «Сущность-связь». Категории «сущность» и «связь» объявляются основополагающими, и разделение их производится на этапе создания конкретных представлений некоторой предметной области.

Для дипломного проекта была разработана диаграмма «Сущность-связь», которая представлена в ПРИЛОЖЕНИИ В.

Для разрабатываемого программного продукта было выделено 12 сущностей:

- учащийся – содержит информацию об учащихся учебного заведения;
- учебная группа – содержит перечень учебных групп;
- специальность – содержит перечень специальностей учебного заведения;
- факультет – содержит перечень факультетов учебного заведения;
- группа здоровья – содержит перечень групп здоровья для назначения учащимся;
- журнал обращений – содержит информацию об обращениях учащихся в здравпункт;
- журнал учета справок – содержит перечень справок учащихся;
- справка – содержит информацию о справке учащегося;
- медосмотр – содержит перечень проведенных медосмотров учащихся;
- тип медосмотра – содержит перечень возможных типов медосмотров;
- прививка учащийся – содержит информацию о прививках по каждому учащемуся;
- прививка – содержит информацию о доступных прививках для учащихся.

Подробное описание данных сущностей представлено в таблице 6.1.

Таблица 6.1 – Сущности, их атрибуты, предполагаемый тип и назначение

Название сущности	Название атрибута	Предполагаемый тип	Назначение
1	2	3	4
Учебная_группа	ID_группы	int	Первичный ключ
	Название	varchar	Название учебной группы
	ID_специальности	int	Внешний ключ
Специальность	ID_специальности	int	Первичный ключ
	Название	varchar	Название специальности

Продолжение табл. 6.1

1	2	3	4
	ID_факультета	int	Внешний ключ
Факультет	ID_факультета	int	Первичный ключ
	Название	varchar	Название факультета
Учащийся	ID_учащегося	int	Первичный ключ
	Фамилия	varchar	Фамилия учащегося
	Имя	varchar	Имя учащегося
	Отчество	varchar	Отчество учащегося
	Паспорт	varchar	Паспортные данные учащегося
	Дата рождения	datetime	Дата рождения учащегося
	ID_группы	int	Внешний ключ
	Страна	varchar	Страна проживания учащегося
	Область	varchar	Область проживания учащегося
	Город	varchar	Город проживания учащегося
	Улица	varchar	Улица, где проживает учащегося
	Номер дома	int	Номер дома, где проживает учащийся
	Номер квартиры	int	Номер квартиры, где проживает учащийся
	Место жительства	varchar	Место жительства учащегося
Группа здоровья	ID_Группы_здоровья	int	Первичный ключ
	Название	varchar	Название группы здоровья
Журнал_учета_справок	ID_журнала_учета_справок	int	Первичный ключ
	ID_справки	int	Внешний ключ

Продолжение табл. 6.1

1	2	3	4
	ID_учащегося	int	Внешний ключ
Журнал_обращений	ID_журнала_обращений	int	Первичный ключ
	Дата	datetime	Дата обращения в здравпункт
	Диагноз	varchar	Поставленный диагноз
	Направление	varchar	Направление к врачу
	Назначения	varchar	Назначение фельдшера
	ID_учащегося	int	Внешний ключ
Справка	ID_справки	int	Первичный ключ
	Дата начала	datetime	Дата начала больничного
	Дата окончания	datetime	Дата окончания больничного
	Количество пропущенных дней	int	Количество пропущенных учебных дней
	ID_журнала_обращений	int	Внешний ключ
	Классификация справки	varchar	Классификация справки
Прививка_учащийся	ID_прививка_учащийся	int	Первичный ключ
	ID_прививки	int	Внешний ключ
	Дата	datetime	Дата прививки
	Где выполнена	varchar	Место выполнения прививки
	ID_учащегося	int	Внешний ключ
Прививка	ID_прививки	int	Первичный ключ
	Название	varchar	Название прививки
	Вид	varchar	Вид прививки
	МИБП	varchar	Доза препарата

Продолжение табл. 6.1

1	2	3	4
	Серия	varchar	Серия прививки
	Доза	varchar	Доза прививки
	Реакция	varchar	Реакция на выполненную прививку
Медосмотр	ID_медосмотра	int	Первичный ключ
	ID_тип_медосмотра	int	Внешний ключ
	Флюорография	varchar	Диагноз в результате прохождения флюорографии
	Гинеколог	varchar	Диагноз в результате прохождения гинеколога
	Кожные заболевания	varchar	Диагноз в результате прохождения осмотра на кожные заболевания
	Дата	datetime	Дата прохождения медосмотра
	ID_учащегося	int	Внешний ключ
	Примечание	varchar	Примечания врача
Тип_медосмотра	ID_медосмотра	int	Первичный ключ
	Наименование	varchar	Наименование типа медосмотра

### 6.3 Разработка диаграммы развертывания системы

Физическое представление программной системы не может быть полным, если отсутствует информация о том, на какой платформе и на каких вычислительных средствах она реализована. Если создается простая программа, которая может выполняться локально на компьютере пользователя, не используя никаких распределенных устройств и сетевых ресурсов, то необходимости в разработке дополнительных диаграмм нет. Однако при создании корпоративных или распределенных приложений требуется визуализировать сетевую инфраструктуру программной системы.

Сложные программные системы могут реализовываться в сетевом варианте, на различных вычислительных платформах и технологиях доступа

к распределенным базам данных. Наличие локальной корпоративной сети требует решения целого комплекса дополнительных задач рационального размещения компонентов по узлам этой сети, что определяет общую производительность программной системы.

Для представления общей конфигурации и топологии распределенной программной системы в UML предназначены диаграммы развертывания.

Диаграмма развертывания применяется для представления общей конфигурации и топологии распределенной программной системы и содержит изображение размещения компонентов по отдельным узлам системы. Кроме того, диаграмма развертывания показывает наличие физических соединений – маршрутов передачи информации между аппаратными устройствами, задействованными в реализации системы. Диаграмма развертывания предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе ее исполнения. При этом представляются только компоненты-экземпляры программы, являющиеся исполняемыми файлами или динамическими библиотеками.

Диаграмма развертывания содержит графические изображения процессоров, устройств, процессов и связей между ними. В отличие от диаграмм логического представления, диаграмма развертывания является единой для системы в целом, поскольку должна всецело отражать особенности ее реализации. Разработка диаграммы развертывания, как правило, является последним этапом спецификации модели программной системы. Разработка диаграммы развертывания начинается с идентификации всех аппаратных, механических и других типов устройств, которые необходимы для выполнения системой всех своих функций. В первую очередь специфицируются вычислительные узлы системы, обладающие памятью и/или процессором. При этом используются имеющиеся в языке UML стереотипы, а в случае их отсутствия разработчики могут определить новые стереотипы. Отдельные требования к составу аппаратных средств могут быть заданы в форме ограничений, свойств и помеченных значений.

Дальнейшее построение диаграммы развертывания связано с размещением всех исполняемых компонентов диаграммы по узлам системы. Если отдельные исполняемые компоненты оказались не размещенными, то подобная ситуация должна быть исключена введением в модель дополнительных узлов, содержащих процессор и память [6].

Графическое отображение диаграммы развертывания представлено в ПРИЛОЖЕНИИ Г.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		
						23

## ЗАКЛЮЧЕНИЕ

В процессе прохождения преддипломной практики были изучены особенности работы сотрудников здравпунктов и проанализированы функциональные требования к разрабатываемой системе.

Во избежание типичных ошибок в реализации информационной системы, на начальном этапе разработки были найдены и проанализированы близкие к функциональным требованиям программного продукта аналоги и прототипы. В результате проведенного анализа исходных данных были сформулированы требования к разрабатываемому приложению.

В ходе преддипломной практики было изучено:

- методы проектирования программного обеспечения;
- среды, используемые для его разработки;
- стандарты, применяемые при проектировании программного обеспечения, оформлении программной документации.

Также были решены следующие задачи:

- изучена предметная область, определен перечень задач, подлежащих решению в процессе разработки программного средства;
- разработано техническое задание на программное средство;
- проведен анализ существующих аналогов и прототипов;
- выбраны языки программирования разработки программного обеспечения;
- разработана необходимая техническая документация.

Разрабатываемая информационная система должна удовлетворять и нефункциональным требованиям, выдвинутым в техническом задании.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		24

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Кухта С.В., Бураченко И.Б., Чеботарев С.П. Методические указания по организации и прохождению производственных (технологической и преддипломной) практик для студентов специальности 1-40 01 01 «Программное обеспечение информационных технологий».

2 ГОСТ 19.003-80 ЕСПД. Схемы алгоритмов и программ. Обозначения условные графические. – М.: Изд-во стандартов, 1982.

3 ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению. – М.: Изд-во стандартов, 1982.

4 Адам Ф. LINQ: язык интегрированных запросов в C# 2010 для профессионалов – М.: Вильямс, 2011. – 656 с.

5 Коннолли, С.Т., Бегг, К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика: Учебник. – 3-е изд., перераб. с англ. – М.: Издательский дом «Вильямс», 2003. – 1440 с.

6 Павловская Т. C#. Программирование на языке высокого уровня – П.: Питер, 2014. – 432 с.

7 Архитектура «клиент-сервер». Сетевые технологии. [Электронный ресурс]. – Режим доступа: <http://www.4stud.info/networking/lecture5.html>. – Дата доступа: 04.04.2020.

8 Введение в язык C#. [Электронный ресурс]. – Режим доступа: [https://msdn.microsoft.com/library/windows/desktop/z1zx9t92\(v=vs.120\).aspx](https://msdn.microsoft.com/library/windows/desktop/z1zx9t92(v=vs.120).aspx). – Дата доступа: 05.04.2020.

9 Microsoft SQL Server – Википедия. [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://ru.wikipedia.org/wiki/Microsoft_SQL_Server). – Дата доступа: 05.04.2020.

						Лист
						25
Изм.	Лист	№ докум.	Подпись	Дата		



# ПРИЛОЖЕНИЕ А

(обязательное)

## Техническое задание

### Введение

Наименование продукта: информационная система «Здравпункт среднего специального или высшего учреждения образования»

Разрабатываемая информационная система позволит улучшить работу фельдшера учебного заведения за счет автоматизации процесса учета обращений учащихся, их медосмотров, вакцинации и заболеваний.

### А.1 Основания для разработки

Основанием для разработки программного продукта является задание на выполнение дипломного проекта, выданное учащейся УО «Полоцкий государственный университет» группы 16ИТз Журавлёвой Каралине Славомировне, утвержденное приказом №402ув от 13.06.2018.

Тема: «Информационная система «Здравпункт среднего специального или высшего учреждения образования».

### А.2 Назначение разработки

#### А.2.1 Функциональное назначение

Функциональным назначением информационной системы является предоставление возможности работникам здравпункта вести учет обращений, медосмотров, вакцинации и заболеваний учащихся.

#### А.2.1 Эксплуатационное назначение

Разрабатываемый программный продукт может эксплуатироваться сотрудниками здравпункта среднего специального или высшего учреждения образования. Пользователями программы могут являться фельдшеры и администраторы здравпункта.

### А.3 Требования к программе или программному изделию

#### А.3.1 Требования к функциональным характеристикам

Разрабатываемая информационная система должна обеспечивать выполнение следующих функций:

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		
						26

- учет личных сведений учащихся;
- ведение журнала учета справок учащихся;
- ведение журнала амбулаторного приема учащихся;
- ведение журнала медосмотра учащихся;
- ведение журнала прививок учащихся;
- формирование и вывод на печать необходимых отчетов.

### А.3.2 Требования к надежности

Информационная система должна стабильно функционировать при бесперебойной работе персонального компьютера. При возникновении сбоя в работе аппаратуры, восстановление нормального функционирования конфигурации должно производиться при запуске системы после перезагрузки операционной системы.

Информационная система должна обеспечить следующие методы защиты данных:

- контроль корректности вводимых данных;  
в случае возникновения ошибок при вводе пользователь должен быть проинформирован о них соответствующими сообщениями;
- обеспечение целостности данных;
- входные и выходные данные должны корректно отображаться и быть понятны пользователю.

### А.3.3 Условия эксплуатации

Условия эксплуатации разрабатываемой информационной системы определяются требованиями к помещениям для ПЭВМ, а также необходимостью соблюдения правил охраны труда при работе с компьютером. Информационная система должна эффективно функционировать на любом персональном компьютере. В обязательные требования входит наличие операционной системы семейства Windows, установленных библиотек, необходимых для корректной работы приложения.

Все пользователи информационной системы должны обладать квалификацией оператора ЭВМ.

### А.3.4 Требования к составу и параметрам технических средств

Минимальные аппаратные требования рабочей станции клиента:

- процессор с частотой 1.7 ГГц;
- 1024 Мб оперативной памяти;

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		27

– не менее 300 Мб свободного места на жестком диске;

Приложение должно быть установлено на компьютер с операционной системой Windows 7 и выше.

#### А.3.5 Требования к маркировке и упаковке

Установочный файл информационной системы, включающий все необходимые данные и библиотеки для нормального функционирования системы, и инструкция по установке, а также пакет технической и эксплуатационной документации должны быть записаны на оптический носитель информации.

#### А.3.6 Требования к транспортированию и хранению

Требования к транспортировке и хранению не предъявляются.

### А.4 Требования к программной документации

Разработанная информационная система должна быть укомплектована следующими программными документами:

- описание программы;
- программа и методика испытаний;
- руководство пользователя и администратора.

Требования к перечисленным программным документам устанавливаются государственными стандартами ЕСПД.

#### А.5 Стадии и этапы разработки

Стадии и этапы разработки представлены в табл. А.1.

Таблица А.1 – Стадии и этапы разработки

№	Название этапов дипломного проектирования	Отчетность
1	2	3
1	Разработка технического задания	Техническое задание
2	Проектирование программного средства	Проектная документация
3	Реализация программного средства	Тексты программных компонентов
4	Комплексное тестирование	Тесты и результаты тестирования
5	Разработка программной документации	Программная документация

Продолжение табл. А.1

1	2	3
6	Разработка и оформление графических материалов	Программная документация
7	Расчет себестоимости и цены программного средства	Программная документация
8	Выполнение раздела по охране труда	Программная документация
9	Нормоконтроль	Программная документация

### А.6 Порядок контроля и приемки

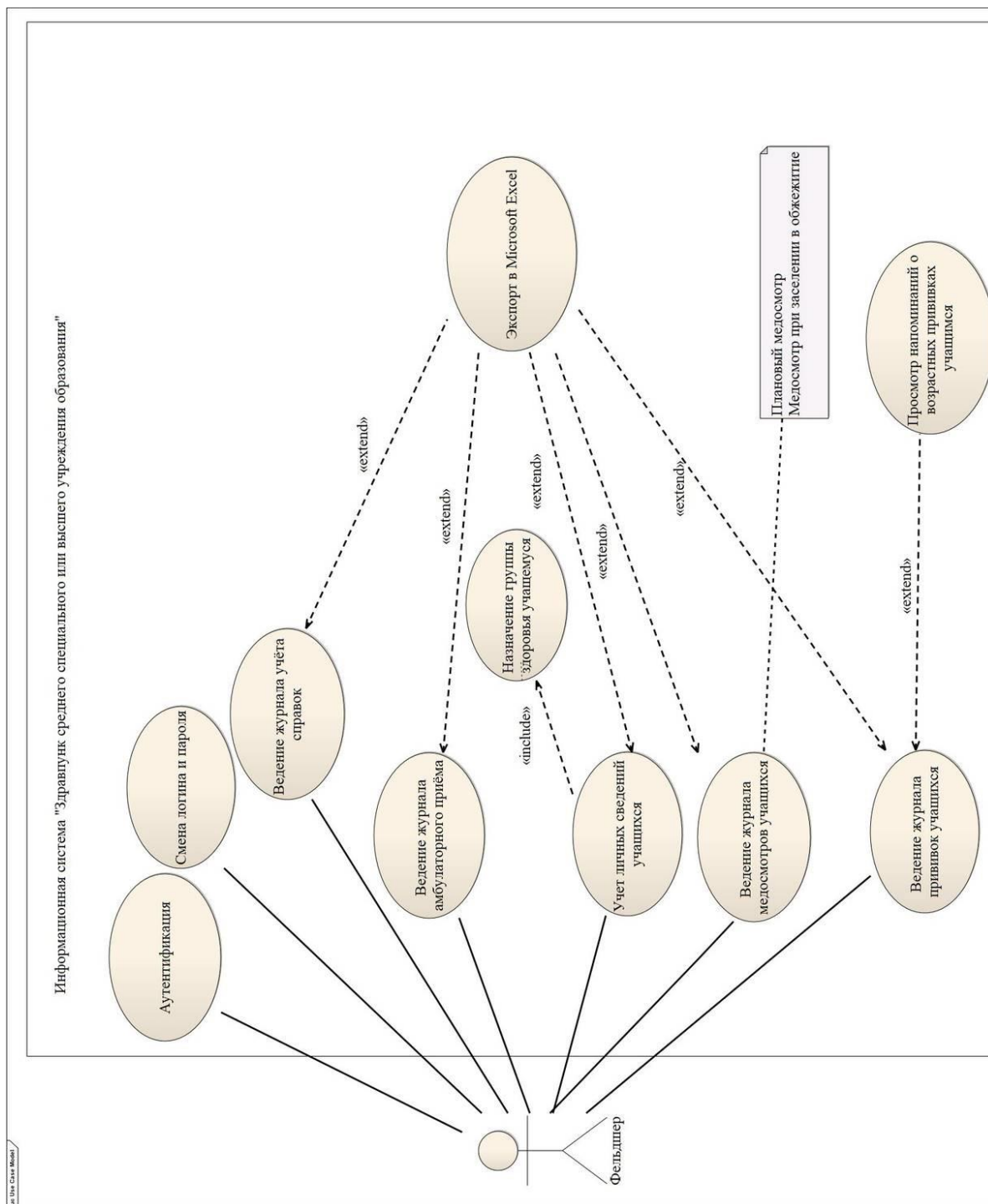
Контроль и приемка информационной системы осуществляется руководителем дипломного проектирования Данченко Еленой Владимировной. В процессе выполнения работы возможно уточнение отдельных требований технического задания по взаимному согласованию руководителя и исполнителя.

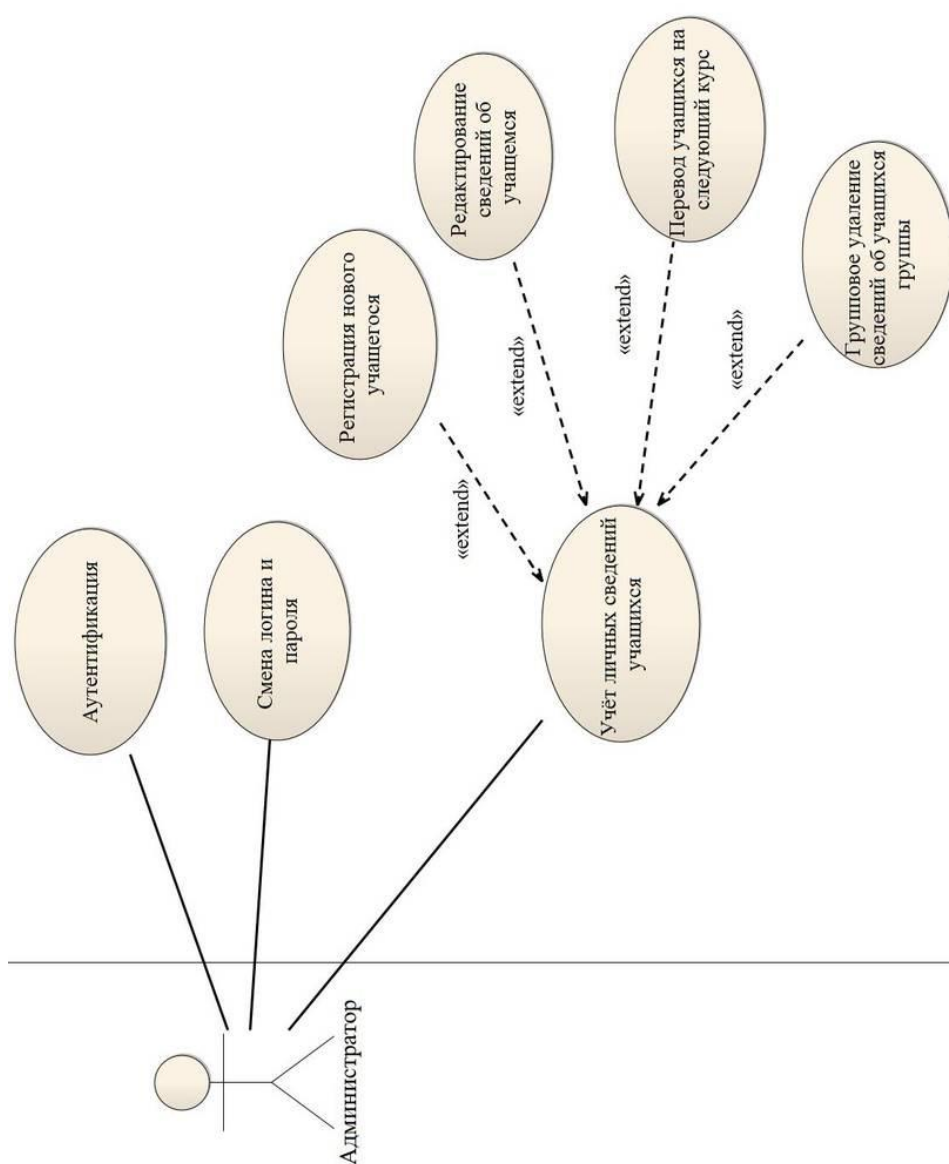
						Лист
						29
Изм.	Лист	№ докум.	Подпись	Дата		

# ПРИЛОЖЕНИЕ Б

(справочное)

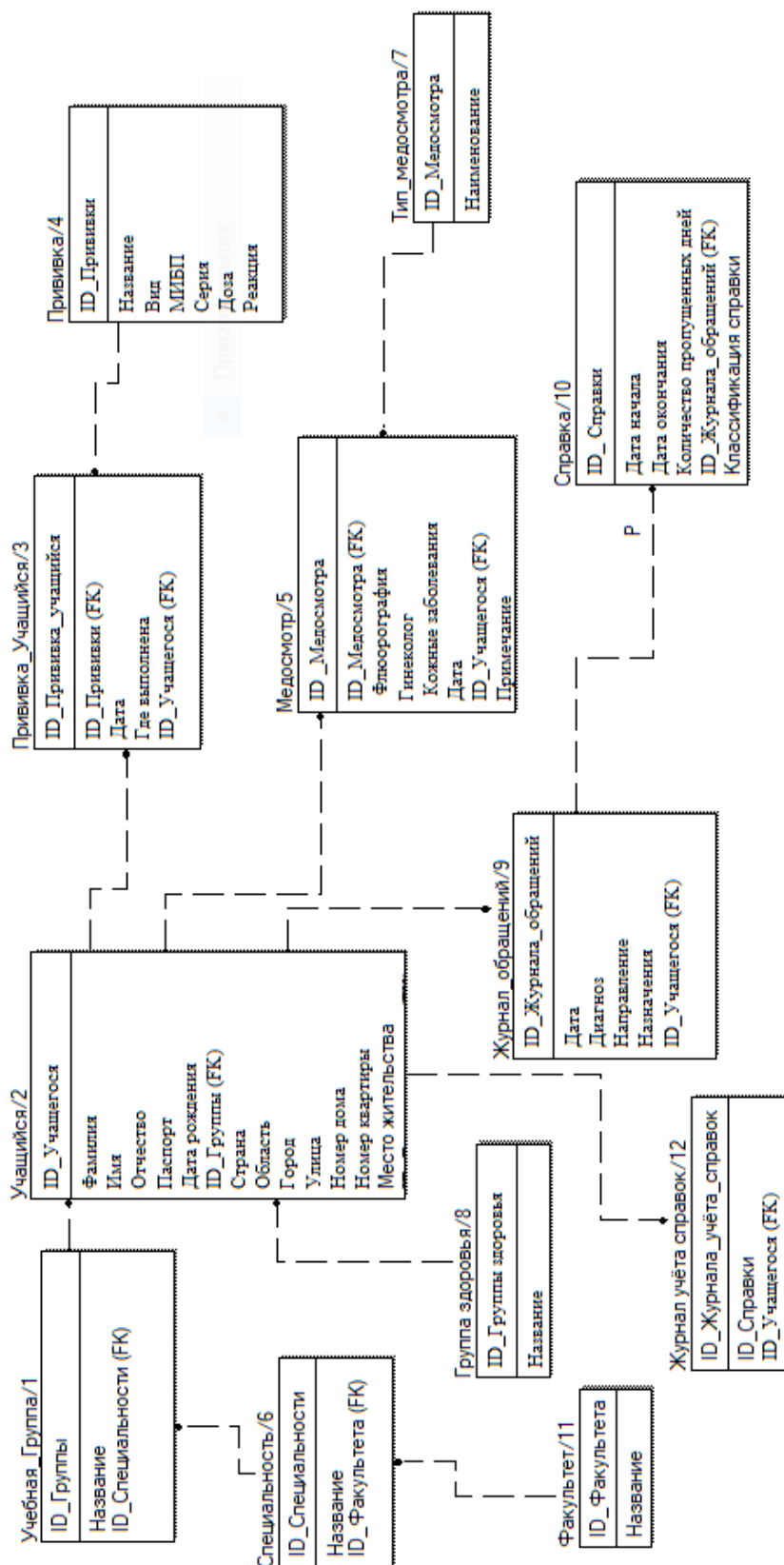
## Диаграмма вариантов использования





(справочное)

### Диаграмма «Сущность-связь»



# ПРИЛОЖЕНИЕ Г

(справочное)

## Диаграмма развертывания

