

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Е.Н. ЭГОВ

**РАЗРАБОТКА ПРОГРАММЫ УЧЕТА ЛИТЕРАТУРЫ
В АБСТРАКТНОМ УНИВЕРСИТЕТЕ
практикум по дисциплине
«Технологии программирования» и
«Разработка профессиональных приложений»**

Ульяновск

УлГТУ

2022

УДК 004.9 : 338(076)

ББК 32.973.26 – 018.2я73

К69

Рекомендовано научно-методической комиссией факультета информационных систем и технологий в качестве практикума.

Эгов, Евгений Николаевич

К69

РАЗРАБОТКА ПРОГРАММЫ УЧЕТА ЛИТЕРАТУРЫ В АБСТРАКТНОМ УНВЕРСИТЕТЕ : практикум / Е. Н. Эгов. – Ульяновск : УлГТУ, 2024. – 84 с.

Практикум адресован студентам для выполнения и оформления курсовой работы по дисциплине «Технологии программирования» и «Разработка профессиональных приложений». Предоставлены варианты заданий для индивидуальной и парной разработки. Рекомендации и требования к курсовой работе разработаны в соответствии с рабочей программой дисциплины. Предназначены для студентов, обучающихся по направлению 09.03.04 «Программная инженерия (профиль Программная инженерия)»

Работа подготовлена на кафедре «Информационные системы».

УДК 004.9 : 338(076)

ББК 32.973.26 – 018.2я73

© Эгов Е. Н., 2022

© Оформление. УлГТУ, 2022

СОДЕРЖАНИЕ

содержание.....	3
Этап №0. Постановка задания	5
Этап №1. Формулировка ТЗ.....	10
Этап №2. Составление диаграмм	14
Этап №3. Составление схемы БД.....	20
Этап №4. Представление проектов	24
Этап №5. Реализация модулей контрактов и хранения данных	25
Этап №6. Реализация (частичная) модуля бизнес-логики.....	26
Этап №7. Реализация прототипа интерфейса пользователя.....	27
Этап №8. Демонстрация кода программы.....	28
Этап №9. Реализация (полная) модуля бизнес-логики	29
Этап №10. Реализация интерфейса пользователя	30
Этап №10.1 Реализация дополнительного задания	30
Этап №11. Оформление записки и презентации	31
Этап №12. Защита курсовой работы.....	32
Темы курсовой работы	33
Список использованных источников	64
Приложение 1 Титульный лист	65
Приложение 2 Задание на курсовую работу	66
Приложение 3 Отзыв руководителя.....	67
Приложение 4 Шаблон технического задания для роли «Работник (первая роль)»	68
Приложение 5 Шаблон технического задания для роли «Кладовщик (вторая роль)»	72

Приложение 6 Шаблон технического задания для роли «Исполнитель (третья роль)».....	76
Приложение 7 Шаблон технического задания для роли «Поручитель (четвертая роль)».....	80

ЭТАП №0.

ПОСТАНОВКА ЗАДАНИЯ

Общее задание (90 баллов):

- разработать программный продукт, согласно заданию по варианту;
- проект разбить на ряд уровней (многоуровневая архитектура, концепция DAL), как минимум, включающих уровни: контрактов, хранение данных, бизнес-логику, интерфейс пользователя;
- в качестве хранилища данных использовать СУБД;
- предусмотреть проверку данных, вводимых пользователем;
- предусмотреть авторизацию при входе в систему;
- все этапы реализации выкладывать на git.

Дополнительное задание (10 баллов):

- визуализация хранимых данных в виде диаграмм и графиков.

Работа в команде:

Каждая тема разделена на 6 ролей. Первая и вторая роли должны обращаться к общему хранилищу данных. Третья и четвертые роли также должны обращаться к общему хранилищу данных, но отличному от хранилища для первой-второй ролей. Пятая и шестая роли реализуются аналогично. ОДИН студент реализует ОДНУ роль в рамках курсовой работы. Студенты могут работать в паре, либо независимо друг от друга. В зависимости от выбранной стратегии работы техническое задание на курсовую работу будет отличаться:

- для парного программирования модули бизнес-логики и хранения данных будут общими, а уровни интерфейса пользователя различными (общие уровни пишутся совместно, а интерфейс пользователя каждый пишет свой);
- для индивидуальной работы реализуется работа одной из ролей, но, так как для полноценного функционирования роли требуются

данные, генерируемые другой ролью, то требуется в проекте реализовать механизмы имитации работы другой роли для генерации данных, необходимых для полноценной работы.

Этапы сдачи обязательного задания (после выдачи задания):

1. Формулировка ТЗ под свой вариант. Поиск аналогов.
2. Составление диаграмм (Use-case, диаграмма последовательностей и т.д.).
3. Составление БД (ER-диаграмма).
- 4. Представление проекта, оформление первой главы записки – конец 4 недели (30 баллов).**
5. Реализация модулей контрактов и хранения данных.
6. Реализация (частичная) модуля бизнес-логики.
7. Реализация прототипа интерфейса пользователя.
- 8. Демонстрация кода программы – конец 10 недели (30 баллов).**
9. Реализация (полная) модуля бизнес-логики.
10. Реализация интерфейса пользователя
 - а. Реализация дополнительного задания (если требуется, 10 баллов)
11. Подготовка презентации, оформление записки.
- 12. Демонстрация программы, защита курсовой работы – зачетная неделя (30 баллов).**

При пропуске сроков сдачи этапа баллы будут снижаться.

Этапы 4, 8, 12 принимаются преподавателем в установленные сроки.

Задание:

Все задания на курсовую делятся на 6 ролей. Студенты могут выполнять задания парами 1-2, 3-4 или 5-6 роли. Первая пара реализует взаимодействие ролей Работник-Кладовщик, вторая пара – Исполнитель-Поручитель, третья пара – Поставщик-Заказчик.

Для работника (первая роль) будет требоваться реализовать следующий функционал:

- формирование записи (CRUD) по «сущности 1» с выбором из списка «сущности 5» нескольких записей к ней;
- получение списка по «сущности 4» на основе выбранных записях «сущности 1» в формате doc/xls;
- формирование записей (CRUD) «сущности 2»;
- привязывание записей «сущности 2» к записям «сущности 1»;
- формирование записей (CRUD) «сущности 3» с привязкой записи «сущности 2»;
- получение отчета за период по записям из «сущности 1» с расшифровкой по записям «сущности 4», относящихся к записям «сущность 2», относящихся к записям «сущности 1» в формате pdf на почту и вывод отчета на форму.

Для кладовщика (вторая роль) будет требоваться реализовать следующий функционал:

- формирование записи (CRUD) по «сущности 5» с выбором из списка «сущности 4» нескольких записей к ней;
- получение списка по «сущности 2» на основе выбранных записях «сущности 5» в формате doc/xls;
- формирование записей (CRUD) по «сущности 4»;
- привязывание записей «сущности 4» к записям «сущности 2»;
- формирование записей (CRUD) «сущности 6» с привязкой записи «сущности 5»;
- получить отчет за период по движению записей «сущности 4» в формате pdf на почту и вывод отчета на форму;

Для исполнителя (третья роль) будет требоваться реализовать следующий функционал:

- формирование записи (CRUD) по «сущности 7»;
- формирование записи (CRUD) по «сущности 8» с выбором из списка «сущности 7» нескольких записей к ней;
- формирование записи (CRUD) по «сущности 9» с выбором из списка «сущности 7» нескольких записей к ней;

- получение списка по «сущности 12» на основе выбранных записях «сущности 7» в формате doc/xls;
- привязка к записи «сущности 8» записи «сущности 11»;
- получение отчета за период по записям «сущности 7», связанных с записями «сущности 8» и «сущности 9» в формате pdf на почту и вывод отчета на форму.

Для поручителя (четвертая роль) будет требоваться реализовать следующий функционал:

- формирование записи (CRUD) по «сущности 10»;
- формирование записи (CRUD) по «сущности 11» с привязкой записи по «сущности 10»;
- формирование записи (CRUD) по «сущности 12» с выбором из списка «сущности 10» нескольких записей к ней;
- получение списка по «сущности 8» на основе выбранных записях «сущности 10» в формате doc/xls;
- привязка к записи «сущности 12» записи «сущности 9»;
- получение отчета за период по «сущностям 11» и «сущностям 12», связанных между собой, в формате pdf на почту и вывод отчета на форму;

Для поставщика (пятая роль) будет требоваться реализовать следующий функционал:

- формирование записи (CRUD) по «сущности 13»;
- формирование записи (CRUD) по «сущности 14» с выбором из списка «сущности 13» нескольких записей к ней;
- формирование записи (CRUD) по «сущности 15» с выбором из списка «сущности 13» записи к ней;
- получение списка по «сущности 18» на основе выбранных записях «сущности 13» в формате doc/xls;
- привязка к записи «сущности 14» нескольких записей из списка «сущности 16»;

- получение отчета за период по записям «сущности 15», связанных с записями «сущности 17» в формате pdf на почту и вывод отчета на форму.

Для заказчика (шестая роль) будет требоваться реализовать следующий функционал:

- формирование записи (CRUD) по «сущности 16»;
- формирование записи (CRUD) по «сущности 17» с выбором из списка «сущности 16» нескольких записей к ней;
- формирование записи (CRUD) по «сущности 18» с выбором из списка «сущности 17» записи к ней;
- получение списка по «сущности 15» на основе выбранных записях «сущности 17» в формате doc/xls;
- привязка к записи «сущности 17» нескольких записей из списка «сущности 13»;
- получение отчета за период по «сущностям 15» и «сущностям 18», связанных между собой, в формате pdf на почту и вывод отчета на форму;

ЭТАП №1.

ФОРМУЛИРОВКА ТЗ

В качестве примера рассмотрим работу университета со студентами, сотрудниками и издательством. В качестве «сущности 1» и будут выступать обучения, формируемые студентами. «Сущность 2» – это курсы, которые относятся к разным обучениям. «Сущность 3» – это оценки, получаемые студентами на курсах. «Сущность 4» – литература, используемая при проведении занятий. «Сущность 5» – это сами проводимые занятия (проводятся в рамках обучений). «Сущность 6» – выставление баллов на занятии. Для второй пары: «сущность 7» – занятия студентов; «сущность 8» – заявки на получение литературы для занятий; «сущность 9» – записи посещения читального зала для получения информации к занятию; «сущность 10» – литература, выдаваемая для чтения студентам; «сущность 11» – записи выдачи литературы в читальном зале; «сущность 12» – записи передачи на руки литературы в библиотеке. Для третьей пары: «сущность 13» – производимая издательством литература; «сущность 14» – поставки литературы университету; «сущность 15» – акты изготовления литературы; «сущность 16» – заявки на закупку; «сущность 17» – записи поступления литературы по заявкам; и «сущность 18» – акты приема литературы в университете на основе записей поступления.

- Роль – Работник (Студент):
 - создать обучение и выбирать в него занятия;
 - получить список литературы по выбранным обучениям, в формате doc/xls;
 - создать курсы;
 - привязать выбранные курсы к выбранному обучению;
 - создать оценки на основе выставленных баллов по занятиям, проводимым за курс (с выбором курса, за который получается оценка);
 - получить отчет за период по пройденным обучениям с расшифровкой по используемой в рамках курсов литературе и полученных оценках в формате pdf на почту или на форме.

- Роль – Кладовщик (Преподаватель):
 - создать занятие и выбирать в него литературу;
 - получить список курсов по выбранным занятиям, в формате doc/xls;
 - CRUD литература;
 - привязать выбранную литературу к выбранным курсам;
 - создать отметки выставления баллов студентам, на проводимом занятии;
 - получить отчет за период по литературе с указанием по курсам и занятиям, на которых она использовалась в формате pdf на почту или на форме.
- Роль – Исполнитель (Студент):
 - создать занятия;
 - создать запись заявки на получение литературы в библиотеке и выбирать под какие занятия требуется литература;
 - создать запись посещения читального зала для чтения литературы и выбирать под какие занятия требуется литература;
 - получить список записей выдачи литературы по выбранным занятиям, в формате doc/xls;
 - привязывать запись посещения читального зала к записи передачи литературы на руки;
 - получить отчета за период по занятиям, для которых требовалось посещения читального зала и получения литературы на руки, в формате pdf на почту или на форме.
- Роль – Поручитель (Библиотекарь):
 - создать литературу;
 - создать запись выдачи литературы в читальном зале и выбирать в него литературу;
 - создать запись передачи на руки литературы в библиотеке и выбирать в него литературу;
 - получить список заявок студентов по выбранной литературе, в формате doc/xls;

- привязывать запись выдачи литературы в читальном зале к заявке студента;
- получить отчет за период по литературе, указываемой в записях выдачи литературы и передачи на руки, в формате pdf на почту или на форме.
- Роль – Поставщик (Издательство):
 - создать литературу;
 - создать запись поставки литературы, выбирать литературу для поставки;
 - создать запись изготовления литературы с указанием записи литературы, которую изготовили;
 - получить список записей актов приемки литературы по выбранным записям изготовления, в формате doc/xls;
 - привязывать запись поставки к записям заявок на поставки;
 - получить отчета за период по изготовлению литературы, поставленной университету, в формате pdf на почту или на форме.
- Роль – Заказчик (Университет):
 - создать заявки на поставки;
 - создать запись поступления литературы и выбирать по каким заявкам она поставлялась;
 - создать актов получения литературы с указанием по какой поставке она пришла;
 - получить список записей актов изготовления литературы по выбранным позициям поставок литературы, в формате doc/xls;
 - привязывать запись поступления литературы к литературе;
 - получить отчет за период по литературе, изготовленной и полученной (оформлены через акты), в формате pdf на почту или на форме.

Техническое задание (ТЗ) должно включать в себя следующие пункты:

1. Цель разрабатываемого программного продукта.

2. Описание всех возможных действий для роли.
3. Выбор реализации интерфейса для роли.
4. Обзор существующих программных решений.

Примеры технического задания:

Для роли работника – Приложение 4.

Для роли кладовщика – Приложение 5.

Для роли исполнителя – Приложение 6.

Для роли поручителя – Приложение 7.

Для роли поставщика – Приложение 8.

Для роли заказчика – Приложение 9.

Красным отмечено то, что требуется заменить, исходя из логики своего варианта, а также по количеству студентов, выполняющих работу.

В ТЗ рассмотреть 2-3 аналога из существующих программных решений.
Указать их достоинства и недостатки.

ЭТАП №2. СОСТАВЛЕНИЕ ДИАГРАММ

В рамках разрабатываемой системы необходимо составить ряд UML диаграмм:

- диаграмма вариантов использования (use-case);
- диаграмма последовательности (sequence);
- диаграмма состояний (state-machine);

Рассмотрим для чего нужны эти диаграммы и что на них должно быть.

Диаграмма вариантов использования

Диаграмма вариантов использования отражает отношения между актёрами и прецедентами и является составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Основная цель построения диаграммы вариантов использования для нашего проекта – определить действующих лиц (актёров), их взаимодействие с системой и ожидаемую функциональность системы;

Работа над диаграммой может начаться с текстового описания, полученного при работе с заказчиком. При этом нефункциональные требования (например, конкретный язык или система программирования) при составлении модели прецедентов опускаются.

Так как в плане интерфейсов пользователей будут разрабатываться две различные программы (для парного программирования), то для каждой роли строится своя, отдельная, диаграмма вариантов использования.

Рассмотрим, как может выглядеть диаграмма (рисунок 2.1).

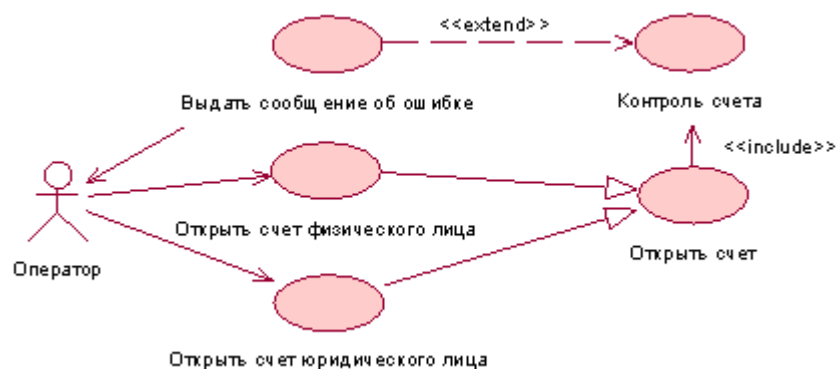


Рисунок 2.1 – Пример диаграммы вариантов использования

На данной диаграмме представлен один актер – оператор. Он имеет 2 возможных действия в системе: открыть счет физического лица или открыть счет юридического лица. Оба этих действия имеют общую часть функционала – действие «Открыть счет». При выполнении действия «Открыть счет» также требуется выполнить действие «Контроль счета». У этого действия имеется необязательное действие «Выдать сообщение об ошибке».

ПРИМЕР ДЕМОНСТРИРУЕТ КАК МОЖЕТ ВЫГЛЯДЕТЬ ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ. ОНА НЕ СООТВЕТСТВУЕТ ТОМУ, ЧТО ТРЕБУЕТСЯ СДЕЛАТЬ ПО КУРСОВОЙ РАБОТЕ. В ДИАГРАММЕ ПО КУРСОВОЙ РАБОТЕ ТРЕБУЕТСЯ ОТРАЗИТЬ ВСЕ ВОЗМОЖНЫЕ ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ, ОПИСАННЫЕ В ТЗ.

Для разрабатываемого проекта в качестве возможных действий в системе могут выступать:

- Авторизация
- Регистрация
- Получение списка. К ней дополнительный функционал в виде добавления, редактирования и удаления.
- Формирование отчетов. К ним дополнительный функционал в виде отправки их на почту.
- И т.д.

Диаграмма последовательности

Это диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл какого-либо определённого объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актёров (действующих лиц) ИС в рамках какого-либо определённого прецедента (отправка запросов и получение ответов). Рассмотрим пример (рисунок 2.2)

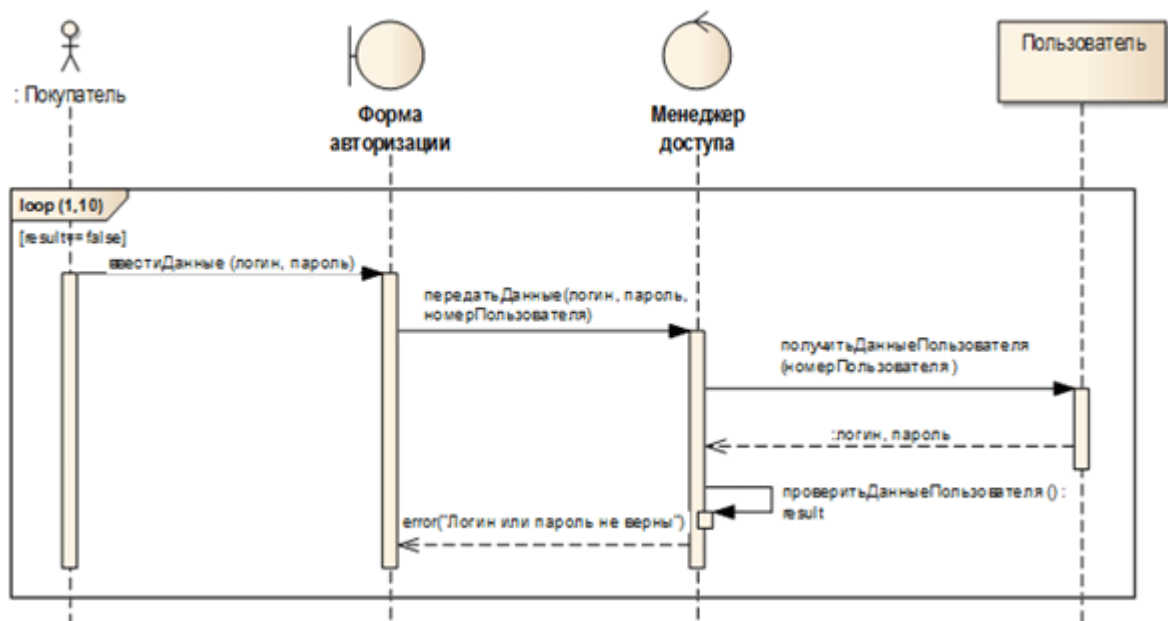


Рисунок 2.2 – Пример диаграммы последовательности

На данной диаграмме есть один актер – Покупатель, одна форма – форма авторизации, класс-контроллер (менеджер доступа) и сущность (пользователь). В диаграмме отображена последовательность действий при неудачной авторизации. Пользователь (покупатель) вводит на форму авторизации данные. Форма передает данные контролу. Контрол на основе переданных данных запрашивает у сущности (например, в БД) данные по пользователю и сверяет их. Если они не совпадают, то контрол возвращает на форму ошибку. Форма отображает эти данные пользователю (на диаграмме этого участка не представлено).

ПРИМЕР ДЕМОНИСТРИРУЕТ КАК МОЖЕТ ВЫГЛЯДЕТЬ ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ. ОНА НЕ СООТВЕТСТВУЕТ ТОМУ, ЧТО ТРЕБУЕТСЯ СДЕЛАТЬ ПО КУРСОВОЙ РАБОТЕ. В ДИАГРАММЕ ПО КУРСОВОЙ РАБОТЕ ТРЕБУЕТСЯ ОТРАЗИТЬ ПРОЦЕСС ВЗАИМОДЕЙСТВИЯ ЭЛЕМЕНТОВ СИСТЕМЫ, ОТ РАБОТЫ ПОЛЬЗОВАТЕЛЯ С ФОРМОЙ, ДО РАБОТЫ С БД.

В рамках курсовой работы для каждой роли следует реализовать последовательность:

- Работник
 - процесс формирования записи «сущности 1» с выбором записей «сущности 2»;

- Кладовщик
 - процесс формирования записи «сущности 5» с выбором записей «сущности 4»;
- Исполнитель
 - процесс формирования записи «сущности 8» с выбором записей «сущности 7»;
- Поручитель
 - процесс формирования записи «сущности 11» с выбором записей «сущности 10»;
- Поставщик
 - процесс формирования записи «сущности 14» с выбором записей «сущности 13»;
- Заказчик
 - процесс формирования записи «сущности 17» с выбором записей «сущности 16»;

Диаграмма состояний

Диаграмма состояний — это, по существу, ориентированный граф для конечного автомата из теории автоматов со стандартизированными условными обозначениями, в котором вершины обозначают состояния, а дуги показывают переходы между двумя состояниями. Диаграмма может определять множество систем от компьютерных программ до бизнес-процессов. Используются следующие условные обозначения:

- круг, обозначающий начальное состояние;
- окружность с маленьким кругом внутри, обозначающая конечное состояние (если есть);
- скруглённый прямоугольник, обозначающий состояние;
- стрелка, обозначающая переход;
- толстая горизонтальная линия с либо множеством входящих линий и одной выходящей, либо одной входящей линией и множеством выходящих, обозначающая объединение и разветвление соответственно.

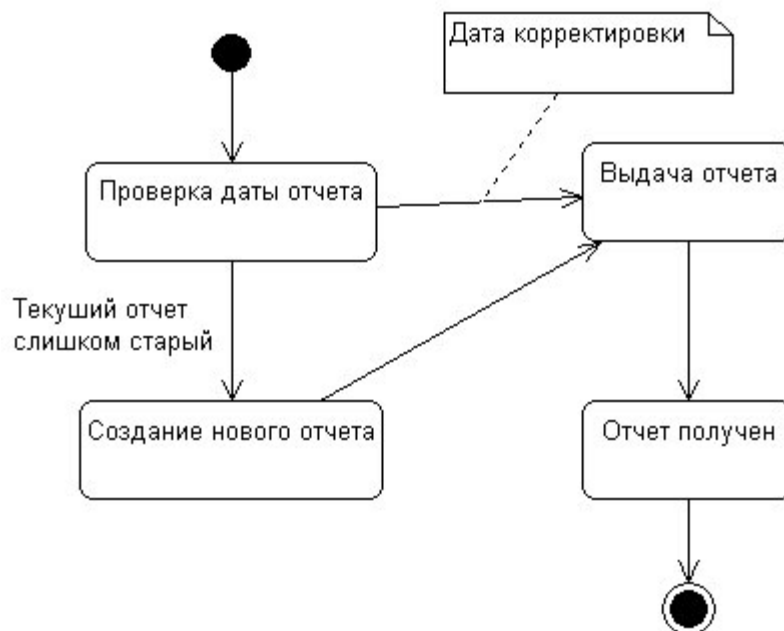


Рисунок 2.3 – Пример диаграммы состояний

Процесс начинается с начальной точки, затем следует самый первый переход в состояние «Проверка даты отчета». Далее возможен ряд переходов после чего процесс приходит в конечную точку. В поведении объекта в системе можно выделить *действия*, отображаемые переходами, и *деятельности*, отображаемые состояниями. Хотя и то и другое – это процессы, реализуемые, как правило, некоторым методом класса «Отчет», они трактуются различным образом. Действия связаны с переходами и рассматриваются, как мгновенные и непрерываемые. Деятельности связаны с состояниями и могут длиться достаточно долго. Деятельность может быть прервана в результате наступления некоторого события. В данном случае действия – это переходы к

ПРИМЕР ДЕМОНСТРИРУЕТ КАК МОЖЕТ ВЫГЛЯДЕТЬ ДИАГРАММА СОСТОЯНИЙ. ОНА НЕ СООТВЕТСТВУЕТ ТОМУ, ЧТО ТРЕБУЕТСЯ СДЕЛАТЬ ПО КУРСОВОЙ РАБОТЕ. В ДИАГРАММЕ ПО КУРСОВОЙ РАБОТЕ ТРЕБУЕТСЯ ОТРАЗИТЬ ПРОЦЕСС ФОРМИРОВАНИЯ ДОКУМЕНТОВ РАЗЛИЧНОГО ФОРМАТА С ВЫБОРКАМИ РАЗЛИЧНЫХ ДАННЫХ.

В рамках курсовой работы для каждой роли следует реализовать последовательность:

- Работник
 - процесс формирования списка записей «сущности 4» по выбранным записям «сущности 1» в формате doc/ xls;
- Кладовщик
 - процесс формирования списка записей «сущности 2», по выбранным записям «сущности 5», в формате doc/xls;
- Исполнитель
 - процесс формирования списка записей «сущности 11» по выбранным записям «сущности 7» в формате doc/xls;
- Поручитель
 - процесс формирования списка записей «сущности 8», по выбранным записям «сущности 10», в формате doc/xls;
- Поставщик
 - процесс формирования списка записей «сущности 18» по выбранным записям «сущности 13» в формате doc/xls;
- Заказчик
 - процесс формирования списка записей «сущности 15», по выбранным записям «сущности 17», в формате doc/xls;

ЭТАП №3.

СОСТАВЛЕНИЕ СХЕМЫ БД

Основная цель – составление схемы базы данных, какие таблицы будут в БД, какие в них будут поля и как таблицы будут связаны между друг другом. Для этих целей создают ER-диаграмму.

Рассмотрим пример ER-диаграммы на рисунке 3.1.

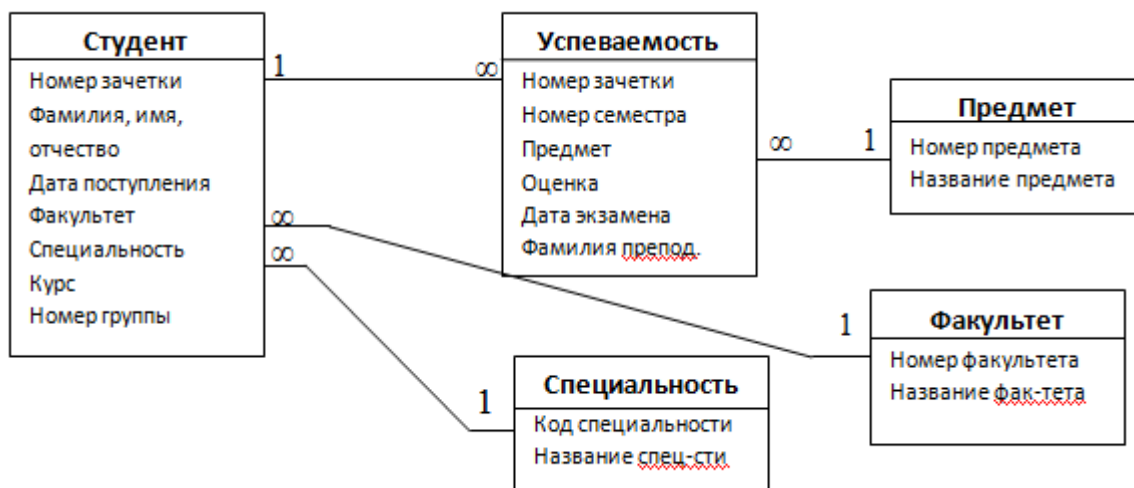


Рисунок 3.1 – Пример ER-диаграммы

На данной схеме представлены 5 сущностей (таблицы БД):

- Студент;
- Успеваемость;
- Предмет;
- Специальность;
- Факультет.

Под названием каждой сущности указываются атрибуты – данные сущности, которые мы хотим хранить. Линиями обозначаются связанные между собой сущности. Цифры или символы на концах линий обозначают тип связи:

- Один к одному. На концах линий стоят цифры 1. Для записи из одной сущности может быть (или должна быть) одна и только одна запись в другой сущности.
- Один ко многим. На одном конце стоит цифра 1, на другом знак бесконечности. Для записи из одной сущности (рядом с которой цифра 1) может быть несколько записей в другой сущности (рядом с которой стоит знак бесконечности).

- Многие ко многим. На обоих концах линии стоят знаки бесконечности. Означает, что для записи из одной сущности может быть несколько записей в другой сущности, и для записи из другой сущности может быть несколько записей из первой сущности.

Рассмотрим на примере схемы с рисунка 3.1, как это выглядит.

На данной схеме нет примера связи **один к одному**, так как эта связь крайне редко применяется. По сути, это означает что описание одной сущности разнесено в разные таблицы. Если бы, например, для сущности «Факультет» была отдельная сущность «История создания факультета», где хранилась бы информация о дате создания, кем был создан и т.п. В таком случае, для любого факультета из таблицы «Факультет» была бы одна запись в таблице «История создания факультета».

Один ко многим. В примере у нас есть сразу 2 записи такого типа:

- Специальность-Студент;
- Факультет-Студент.

Связь говорит о том, что к одному факультету может относиться сразу несколько студентов. Но каждый студент может относиться только к **ОДНОМУ** факультету. Для специальностей аналогично. Представленная на рисунке схема не совсем верна, правильное будет, чтобы специальности относились к факультетам (один ко многим, на один факультет несколько специальностей), а студенты относились бы только к специальностям и через специальность относились бы уже к факультету (рисунок 3.2).



Рисунок 3.2 – Обновлённая ER-диаграмма

Связь многие ко многим, по сути, представляет собой связь между 3 сущностями (одна из сущностей является связующей). На схеме такая связь представлена в виде связи Студент-Предмет. Много студентов сдают много предметов. Для этого создана отдельная сущность «Успеваемость». В сущности-связи, помимо ссылок на записи связываемых сущностей могут быть дополнительные атрибуты, как представлено на схеме в примере. Минимально в сущности-связи должно быть 2 поля: ссылка на сущность 1 и ссылка на сущность 2.

В базе данных для связи между таблицами требуется, чтобы в каждой таблице был атрибут (колонка), в которой для каждой записи было бы уникальное значение в рамках таблицы. Рассмотрим на примере студента. Можно ли в качестве такого атрибута рассматривать, например, ФИО? Ответ: нет. Мы не можем исключать случай, что в университете могут одновременно учиться полные тезки. В данном случае уникальным будет атрибут «Номер зачетки», так как нет двух студентов с одинаковым номером зачетной книжки. В общем случае для таблицы создают атрибут «Идентификатор» (сокращенно Id) и каждой добавляемой записи в таблицу в этот атрибут проставляют значение, которого нет ни у какой другой записи в этой таблице (уникальность требуется только в рамках таблицы, в других таблицах могут также быть поля Id и в них могут быть такие же значения, что и в поле Id у данной таблицы). При создании связи (один ко многим) в таблице (рядом с ней будет знак ∞), к которой будет относиться «другая таблица» (у нее цифра 1) требуется создать поле-значение уникального поля из «другой таблицы» (рисунок 3.3).



Рисунок 3.3 – ER-диаграмма с указанием полей-связей

ПРИМЕР ДЕМОНСТРИРУЕТ КАК МОЖЕТ ВЫГЛЯДЕТЬ ER-ДИАГРАММА. ОНА НЕ СООТВЕТСТВУЕТ ТОМУ, ЧТО ТРЕБУЕТСЯ СДЕЛАТЬ ПО КУРСОВОЙ РАБОТЕ. В ДИАГРАММЕ ПО КУРСОВОЙ РАБОТЕ ТРЕБУЕТСЯ ОТРАЗИТЬ СУЩНОСТИ, КОТОРЫЕ СЛЕДУЕТ ХРАНИТЬ, КАК ОНИ СВЯЗАНЫ МЕЖДУ СОБОЙ, И КАКИЕ В НИХ БУДУТ ПОЛЯ.

В рамках курсовой работы требуется составить ER-диаграмму сущностей, которые будут использоваться в разрабатываемом проекте. Требуется дать им названия, указать названия и **типы** атрибутов, а также проставить связи между сущностями.

ЭТАП №4.

ПРЕДСТАВЛЕНИЕ ПРОЕКТОВ

В рамках недели студенческой науки каждая группа разработчиков должна представить модель разрабатываемого программного продукта. В рамках доклада (первая глава записки курсовой работы) следует осветить следующие пункты:

- описание предметной области;
- рассмотрение аналогов;
- диаграммы, описывающие поведение системы;
- диаграмма базы данных;
- применяемые технологии;
- распределение обязанностей (для парных работ, кто за что отвечает).

Лучше работы будут выбраны для публикации в сборнике.

Прием работ может проводиться как в очном виде (онлайн или офлайн), так и в заочном, на усмотрение преподавателя. От каждой группы разработчиков необходимо представить доклад в виде текстового документа в случае заочного участия, либо презентацию в случае очного участия. По результатам будут выставлены замечания по оформлению доклада.

ЭТАП №5.

РЕАЛИЗАЦИЯ МОДУЛЕЙ КОНТРАКТОВ И ХРАНЕНИЯ ДАННЫХ

На данном этапе будет создано несколько слоев будущего проекта, а именно (при желании, первые 2 слоя можно объединить в один):

- слой моделей;
- слой контрактов;
- слой хранения данных.

В слое моделей объявляются модели, которые описывают сущности предметной области. Они строятся на основе ER-диаграммы базы данных, спроектированной на 3 этапе и с учетом замечаний, полученных на 4 этапе. Модели могут представляться как классами, так и интерфейсами, это уже делается на усмотрение команды разработчиков.

Слой контрактов состоит из интерфейсов, которые будут использовать другие слои для взаимодействия между собой. Требуется реализовать возможность «общения» слоя пользователя со слоем бизнес-логики, а также слоя бизнес-логики со слоем хранения данных.

На данном этапе требуется реализовать модуль с реализацией интерфейсов под конкретный вариант хранения данных. При разработке слоя хранения данных будет применяться EntityFramework, подход Code-First для создания БД, поэтому требуется создать модели (все модели наследуются от моделей, описанных в слое моделей), на основе которых будет создана БД с указанием через атрибуты связанность между моделями.

Важно, на что будет делаться акцент при проверке – это как составляются запросы к базе данных для получения связанных данных. Запросы должны извлекать и передавать в слой бизнес-логики только те данные из общего объема, которые требуются по условию. Например, если требуется извлечь из таблицы «Сущности 1» записи, связанные с одной или несколькими записями «Сущности 3», то это означает, что нужно передавать указания, по каким записям «Сущности 3» нужно извлекать записи из «Сущности 1» и делать запрос к базе данных таким образом, чтобы сразу получить только нужные записи «Сущности 1».

ЭТАП №6.

РЕАЛИЗАЦИЯ (ЧАСТИЧНАЯ) МОДУЛЯ БИЗНЕС-ЛОГИКИ

На данном этапе требуется реализовать модуль бизнес-логики. Полностью логику реализовывать пока не требуется, так как некоторые вещи, например, отправка на почту еще не изучалась в рамках работы в семестре. Однако, механизм CRUD (Create-Read-Update-Delete) для сущностей реализовать уже можно, что и необходимо сделать в рамках этапа.

Создание отчетов, отправка информации по почте и прочее будет делаться уже на последующих этапах.

ЭТАП №7.

РЕАЛИЗАЦИЯ ПРОТОТИПА ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ

На данном этапе разработки каждый из разработчиков должен разработать свой модуль взаимодействия с пользователем. Для этого необходимо создать проект для работы с формами или web-проект.

На данном этапе следует разработать макеты будущих форм. Требуется создать формы (web-страницы) и разместить на них элементы. Какие потребуются формы, какие элементы на них потребуются, описано в ТЗ. Никакой логики писать пока не требуется, это будет делаться на последующих этапах.

ЭТАП №8.

ДЕМОНСТРАЦИЯ КОДА ПРОГРАММЫ

Прием работ может проводится как в очном виде (онлайн или офлайн), так и в заочном, на усмотрение преподавателя. От команды требуется предоставить ссылку на репозиторий, где располагается проект. Будет оцениваться, в первую очередь, модуль хранения данных и разработанные там запросы к базе данных. Также будет оцениваться вклад каждого участника команды (в случае парного программирования) на основе его коммитов в репозиторий.

По результатам будут выставлены замечания по оформлению кода и запросов.

ЭТАП №9.

РЕАЛИЗАЦИЯ (ПОЛНАЯ) МОДУЛЯ БИЗНЕС-ЛОГИКИ

На данном этапе доделывается модуль бизнес-логики, прописывается код для создания отчетов, отправки данных на почту. Данный этап можно и нужно делать совместно с 10 этапом, чтобы выполнять одновременно доработку и отладку.

ЭТАП №10.

РЕАЛИЗАЦИЯ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ

На данном этапе доделывается модуль интерфейса пользователя, для созданных форм прописывается логика и выполняется связь с модулем бизнес-логики. Данный этап можно и нужно делать совместно с 9 этапом, чтобы выполнять одновременно доработку и отладку.

ЭТАП №10.1

РЕАЛИЗАЦИЯ ДОПОЛНИТЕЛЬНОГО ЗАДАНИЯ

На данном этапе реализуется дополнительное задание, в рамках которого необходимо добавить в программу формы с выводом различных графиков по сущностям. Какие графики будут выводиться, определяется разработчиками. Оцениваться будет адекватность составляемых графиков, насколько они могут быть полезны для задач аналитики. Этап реализуется по необходимости, если требуется добрать баллов.

ЭТАП №11.

ОФОРМЛЕНИЕ ЗАПИСКИ И ПРЕЗЕНТАЦИИ

На основе разработанного программного продукта, оформляется пояснительная записка и подготавливается презентация (если будет объявлена очная защита с презентацией). Пояснительная записка оформляется **ИНДИВИДУАЛЬНО** каждым студентом на основе того набора функций, который он реализовывал.

Основной текст записки: Times New Roman, 14 шрифт, полуторный интервал, отступ первой строк 1.27.

Пояснительная записка должна содержать следующие пункты:

- Титульный лист (приложение 1).
- Задание на курсовую работу (приложение 2).
- Отзыв руководителя (приложение 3).
- Введение. Описывается актуальность задачи.
- Первая глава. Теоретическая. Описывается предметная область, аналоги, ТЗ. Приводятся диаграммы с их описанием, согласно ТЗ и варианту
- Вторая глава. Приводится руководство пользователя для разработанного проекта.
- Третья глава. Приводится руководство программиста для разработанного проекта.
- Заключение. Выводы по проделанной работе, какие технологии были применены и освоены.
- Список литературы. **Не менее 10** источников с ссылками в тексте.
- Приложение. Листинг кода (8 шрифт, 3 колонки).

Записка загружается в электронном виде в LMS преподавателю для проверки.

ЭТАП №12.
ЗАЩИТА КУРСОВОЙ РАБОТЫ

К защите курсовой работы допускаются студенты, выполнившие и сдавшие все предыдущие этапы. В рамках защиты каждая группа разработчиков представляет презентацию с информацией о проделанной работе и разработанном программном продукте, если объявлена очная защита с представлением проекта.

Сдается распечатанная и подписанная пояснительная записка к курсовой работе.

ТЕМЫ КУРСОВОЙ РАБОТЫ

В темах курсовых работ представлены краткие описания того, что требуется сделать. Развернутые описания представлены в приложениях для примера, рассматриваемого в методичке (обучение студентов). В 0 этапе представлено описание заданий для каждой роли с указанием сущностей, с которыми ей необходимо будет работать. В описании к теме курсовой есть подчеркнутые слова – это и есть сущности для конкретного варианта. В приложении к выбранной роли красным цветом выделены сущности по рассматриваемому примеру. Вам вместо этих сущностей нужно будет подставить свои из описания к варианту. **Синим** цветом помечены элементы, которые нужно будет дописать **самостоятельно**.

Например, если выбран первый вариант и первая роль (вариант 1.1 и, соответственно, роль Клиент), то здесь в качестве «сущности 1», описываемой в задании для Клиента будет выступать «визит». Второй сущностью будет «оплата» (вторая сущность практически у всех ролей и во всех вариантах одинаковая). В приложении 4 описывается ТЗ к роли «Клиент». Следует взять за шаблон это ТЗ и заменить все выделенные красным слова «обучение» на «визит». Также, потребуется переработать цель программы (первый абзац), скорректировать по логике варианта общие фразы и дополнить то, что выделено синим цветом.

№	Тема	Описание
---	------	----------

1. Салон красоты «Вы ужасны»

1.1	Салон красоты «Вы ужасны». Работник	Программа «Салон красоты «Вы ужасны». Работник» на платформе ASP.NET предназначена для ведения учета оказываемых услуг. Пользователи могут создавать <u>заказы (1)</u> , <u>процедуры (2)</u> , которые они будут получать в рамках заказа, и проставлять <u>оценки (3)</u> за процедуры. Программа предоставляет возможность получения отчетов по заказам и услугам к ним. Пользователи могут создавать заказы и выбирать к ним косметику. Могут создавать процедуры. Могут связывать процедуры с заказами. Могут создавать оценки и привязывать их к процедурам. Могут получать списки косметики по выбранным процедурам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по заказам с расшифровкой по услугам и оценкам за период.
1.2	Салон красоты «Вы ужасны». Кладовщик	Программа «Салон красоты «Вы ужасны». Кладовщик» предназначена для ведения учета услуг. Пользователи могут формировать список предлагаемых <u>услуг (4)</u> и списки <u>косметики (5)</u> к ним, а также <u>трудозатраты (6)</u> на оказания услуг клиентам. Программа предоставляет возможность получения отчетов по оказываемым услугам. Пользователи могут создавать услуги. Могут связывать услуги и процедуры. Могут создавать косметику и выбирать к ним услуги. Могут создавать трудозатраты и привязывать их к косметике. Могут получать списки процедур по выбранной косметике в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по услугам с расшифровкой по заказам и трудозатратам за период.

№	Тема	Описание
---	------	----------

1. Салон красоты «Вы ужасны»

1.3	Салон красоты «Вы ужасны». Исполнитель	Программа «Салон красоты «Вы ужасны». Исполнитель» предназначена для ведения учета процедур. Пользователи указывают <u>процедуры (7)</u> , под которые они могут <u>покупать (8)</u> косметику и <u>посещения (9)</u> процедур. Программа предоставляет возможность получения отчетов по выполненным процедурам. Пользователи могут создавать процедуры. Могут создавать покупки и выбирать к ним процедуры. Могут создавать посещения и выбирать к ним процедуры. Могут связывать посещения с записями выдачи. Могут получать списки выдачи по выбранным процедурам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по процедурам с расшифровкой по посещениям и чекам за период.
1.4	Салон красоты «Вы ужасны». Поручитель	Программа «Салон красоты «Вы ужасны». Поручитель» предназначена для ведения учета косметики. Пользователи пробивают <u>чеки (11)</u> на покупаемую <u>косметику (10)</u> и <u>выдачу (12)</u> косметики специалистам при посещении клиентами салона. Программа предоставляет возможность получения отчетов по чекам и выдачам. Пользователи могут создавать косметику. Могут создавать чеки и выбирать к ним косметику. Могут связывать чеки с покупками. Могут создавать записи выдачи и выбирать к ним косметику. Могут получать списки покупок по выбранной косметике в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по косметике с расшифровкой по чекам и посещениям за период.

№	Тема	Описание
---	------	----------

2. Школа «Опять учиться»

2.1	Школа «Опять учиться». Работник	Программа «Школа «Опять учиться». Работник» на платформе ASP.NET предназначена для ведения учета кружков. Пользователи указывают свои <u>интересы (1)</u> , <u>занятия (2)</u> , на которых они хотят их осваивать, и проставлять свои <u>достижения (3)</u> за занятия. Программа предоставляет возможность получения отчетов по интересам. Пользователи могут создавать интересы и выбирать к ним материалы. Могут создавать занятия. Могут связывать занятия с интересами. Могут создавать достижения и привязывать их к занятиям. Могут получать списки материалов по выбранным занятиям в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по интересам с расшифровкой по кружкам и достижениям за период.
2.2	Школа «Опять учиться». Кладовщик	Программа «Школа «Опять учиться». Кладовщик» предназначена для ведения учета кружков. Пользователи могут формировать список проводимых <u>кружков (4)</u> и списки <u>материалов (5)</u> , которые требуются на них, а также <u>медальки (6)</u> школьникам за успешную работу на кружках. Программа предоставляет возможность получения отчетов по проводимым кружкам. Пользователи могут создавать кружки. Могут связывать кружки и занятия. Могут создавать материалы и выбирать к ним кружки. Могут создавать медальки и привязывать их к материалам. Могут получать списки занятий по выбранным материалам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по кружкам с расшифровкой по интересам и медалькам за период.

№	Тема	Описание
---	------	----------

2. Школа «Опять учиться»

2.3	Школа «Опять учиться». Исполнитель	Программа «Школа «Опять учиться». Исполнитель» предназначена для ведения учета интересов. Пользователи указывают <u>интересы (7)</u> , изготавливают дома <u>поделки (8)</u> и <u>изделия (9)</u> на занятиях. Программа предоставляет возможность получения отчетов по интересам школьников. Пользователи могут создавать интересы. Могут создавать поделки и выбирать к ним интересы. Могут создавать изделия и выбирать к ним интересы. Могут связывать изделия с занятиями. Могут получать списки занятий по выбранным интересам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по интересам с расшифровкой по изделиям и заданиям на поделки за период.
2.4	Школа «Опять учиться». Поручитель	Программа «Школа «Опять учиться». Поручитель» предназначена для ведения учета материалов. Пользователи выдают <u>задания на поделки (11)</u> и проводят <u>занятия (12)</u> , выдавая школьника <u>материалы (10)</u> под это. Программа предоставляет возможность получения отчетов по заданиям и занятиям. Пользователи могут создавать материалы. Могут создавать задания на поделки и выбирать к ним материалы. Могут связывать задания на поделки с поделками. Могут создавать занятия и выбирать к ним материалы. Могут получать списки поделок по выбранным материалам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по материалам с расшифровкой по заданиям на поделки и изделиям за период.

№	Тема	Описание
---	------	----------

3. Столовая «Рога и копыта»

3.1	Столовая «Рога и копыта». Работник	Программа «Столовая «Рога и копыта». Работник» на платформе ASP.NET предназначена для ведения учета обедов. Пользователи указывают свои <u>обеда (1)</u> , <u>заказы (2)</u> , которые они совершали, и какими <u>приборами (3)</u> пользовались. Программа предоставляет возможность получения отчетов по обедам. Пользователи могут создавать обеда и выбирать к ним продукты. Могут создавать заказы. Могут связывать заказы с обедами. Могут создавать столовые приборы и привязывать их к заказам. Могут получать списки продуктов по выбранным заказам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по блюдам с расшифровкой по поварам и приборам за период.
3.2	Столовая «Рога и копыта». Кладовщик	Программа «Столовая «Рога и копыта». Кладовщик» предназначена для ведения учета поваров. Пользователи могут формировать список <u>поваров (4)</u> и списки <u>продуктов (5)</u> , которые им требуются, а также <u>блюда (6)</u> , которые они готовят. Программа предоставляет возможность получения отчетов по поварам. Пользователи могут создавать поваров. Могут связывать поваров и заказы. Могут создавать продукты и выбирать к ним поваров. Могут создавать блюда и привязывать их к продуктам. Могут получать списки заказов по выбранным продуктам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по поварам с расшифровкой по обедам и блюдам за период.

№	Тема	Описание
---	------	----------

3. Столовая «Рога и копыта»

3.3	Столовая «Рога и копыта». Исполнитель	Программа «Столовая «Рога и копыта». Исполнитель» предназначена для ведения учета официантов. Пользователи указывают <u>официантов (7)</u> , которые обслуживают заказы, <u>заказы (8)</u> и <u>алкогольные карты (9)</u> для клиентов. Программа предоставляет возможность получения отчетов по официантам. Пользователи могут создавать официантов. Могут создавать заказы и выбирать к ним официантов. Могут создавать алкогольные карты и выбирать к ним официантов. Могут связывать алкогольные карты с напитками. Могут получать списки напитков по выбранным официантам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по официантам с расшифровкой по алкогольным картам и блюдам за период.
3.4	Столовая «Рога и копыта». Поручитель	Программа «Столовая «Рога и копыта». Поручитель» предназначена для ведения учета продуктов. Пользователи, используя <u>продукты (10)</u> , делают <u>блюда (11)</u> к заказам и <u>напитки (12)</u> для карт. Программа предоставляет возможность получения отчетов по продуктам и блюдам. Пользователи могут создавать продукты. Могут создавать блюда и выбирать к ним продукты. Могут связывать блюда с заказами. Могут создавать напитки и выбирать к ним продукты. Могут получать списки заказов по выбранным продуктам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по продуктам с расшифровкой по блюдам и алкогольным картам за период.

№	Тема	Описание
---	------	----------

4. Турфирма «Иван Сусанин»

4.1	Турфирма «Иван Сусанин». Работник	<p>Программа «Турфирма «Иван Сусанин». Работник» на платформе ASP.NET предназначена для ведения учета путешествий. Пользователи создают свои <u>путешествия (1)</u>, <u>экскурсии (2)</u> в них и какие <u>места (3)</u> они посетили. Программа предоставляет возможность получения отчетов по путешествиям. Пользователи могут путешествия и выбирать к ним туры. Могут создавать экскурсии. Могут связывать экскурсии с путешествиями. Могут создавать места и привязывать их к экскурсиям. Могут получать списки туров по выбранным экскурсиям в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по путешествиям с расшифровкой по гидам и местам за период.</p>
4.2	Турфирма «Иван Сусанин». Кладовщик	<p>Программа «Турфирма «Иван Сусанин». Кладовщик» предназначена для ведения учета гидов. Пользователи указывают <u>гидов (4)</u>, проводящих экскурсии и <u>туры (5)</u>, а также <u>остановки (6)</u> клиентов в турах. Программа предоставляет возможность получения отчетов по задействованным гидам. Пользователи могут создавать гидов. Могут связывать гидов и экскурсии. Могут создавать туры и выбирать к ним гидов. Могут создавать остановки и привязывать их к турам. Могут получать списки экскурсий по выбранным турам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по гидам с расшифровкой по путешествиям и остановкам за период.</p>

№	Тема	Описание
---	------	----------

4. Турфирма «Иван Сусанин»

4.3	Турфирма «Иван Сусанин». Исполнитель	Программа «Турфирма «Иван Сусанин». Исполнитель» предназначена для ведения учета туров. Пользователи указывают <u>туры (7)</u> , для <u>экскурсий (8)</u> и больших <u>экскурсионных групп (9)</u> . Программа предоставляет возможность получения отчетов по турам. Пользователи могут создавать туры. Могут создавать экскурсии и выбирать к ним туры. Могут создавать экскурсионные группы и выбирать к ним туры. Могут связывать группы с местами для посещения. Могут получать списки мест для посещения по выбранным турам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по турам с расшифровкой по экскурсионным группам и гидам за период.
4.4	Турфирма «Иван Сусанин». Поручитель	Программа «Турфирма «Иван Сусанин». Поручитель» предназначена для ведения учета поездок. Пользователи, нанимают <u>гидов (11)</u> , указывают <u>места для посещения (12)</u> к экскурсиям и <u>поездки (10)</u> для больших групп. Программа предоставляет возможность получения отчетов по гидам и местам. Пользователи могут создавать поездки. Могут создавать гидов и выбирать к ним поездки. Могут связывать гидов с экскурсиями. Могут создавать места для посещения и выбирать к ним поездки. Могут получать списки экскурсий по выбранным поездкам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по поездкам с расшифровкой по гидам и экскурсионным группам за период.

№	Тема	Описание
---	------	----------

5. Гостиница «Принцесса на горошине»

5.1	Гостиница «Принцесса на горошине». Работник	Программа «Гостиница «Принцесса на горошине». Работник» на платформе ASP.NET предназначена для ведения учета конференций. Пользователи создают <u>конференции (1)</u> , <u>семинары (2)</u> в них и <u>участников (3)</u> семинаров. Программа предоставляет возможность получения отчетов по конференциям. Пользователи могут создавать конференции и выбирать к ним номера. Могут создавать семинары. Могут связывать семинары с конференциями. Могут создавать участников и привязывать их к семинарам. Могут получать списки номеров по выбранным семинарам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по конференциям с расшифровкой по обедам и участникам за период.
5.2	Гостиница «Принцесса на горошине». Кладовщик	Программа «Гостиница «Принцесса на горошине». Кладовщик» предназначена для ведения учета обедов. Пользователи указывают <u>обеда (4)</u> , подаваемые на семинары и в номера, <u>номера (5)</u> под конференции, а также <u>постояльцев (6)</u> номеров. Программа предоставляет возможность получения отчетов по предоставляемым обедам. Пользователи могут создавать обеда. Могут связывать обеда и семинары. Могут создавать номера и выбирать к ним обеда. Могут создавать постояльцев и привязывать их к номера. Могут получать списки семинаров по выбранным номерам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по обедам с расшифровкой по конференциям и постояльцам за период.

№	Тема	Описание
---	------	----------

5. Гостиница «Принцесса на горошине»

5.3	Гостиница «Принцесса на горошине». Исполнитель	Программа «Гостиница «Принцесса на горошине». Исполнитель» предназначена для ведения учета участников. Пользователи указывают <u>участников (7)</u> , <u>планы питания для них (8)</u> и <u>конференции (9)</u> . Программа предоставляет возможность получения отчетов по участникам. Пользователи могут создавать участников. Могут создавать планы питания и выбирать к ним участников. Могут создавать конференции и выбирать к ним участников. Могут связывать конференции с бронями по конференциям. Могут получать списки броней конференций по выбранным участникам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по участникам с расшифровкой по конференциям и номерам за период.
5.4	Гостиница «Принцесса на горошине». Поручитель	Программа «Гостиница «Принцесса на горошине». Поручитель» предназначена для ведения учета обедов. Пользователи, указывают <u>номера (11)</u> , <u>брони по конференциям (12)</u> и <u>обеда (10)</u> , подаваемые в номера. Программа предоставляет возможность получения отчетов по номерам и конференциям. Пользователи могут создавать обеда. Могут создавать номера и выбирать к ним обеда. Могут связывать номера с планами питания. Могут создавать брони по конференциям и выбирать к ним обеда. Могут получать списки планов питания по выбранным обедам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по обедам с расшифровкой по номерам и конференциям за период.

№	Тема	Описание
---	------	----------

6. Поликлиника «Будьте больны»

6.1	Поликлиника «Будьте больны». Работник	Программа «Поликлиника «Будьте больны». Работник» на платформе ASP.NET предназначена для ведения учета пациентов. Врачи указывают <u>пациентов (1)</u> , выписывают <u>рецепты (2)</u> им и под какие <u>болезни (3)</u> . Программа предоставляет возможность получения отчетов по пациентам. Пользователи могут создавать пациентов и выбирать к ним процедуры. Могут создавать рецепты. Могут связывать рецепты с пациентами. Могут создавать болезни и привязывать их к рецептам. Могут получать списки процедур по выбранным рецептам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по пациентам с расшифровкой по лекарствам и болезням за период.
6.2	Поликлиника «Будьте больны». Кладовщик	Программа «Поликлиника «Будьте больны». Кладовщик» предназначена для ведения учета лекарств. Пользователи указывают используемые <u>лекарства (4)</u> , в каких <u>процедурах (5)</u> они используются, а также <u>описание процедур для врачей (6)</u> . Программа предоставляет возможность получения отчетов по предоставляемым лекарствам. Пользователи могут создавать лекарства. Могут связывать лекарства и рецепты. Могут создавать процедуры и выбирать к ним лекарства. Могут создавать описание процедур и привязывать их к процедурам. Могут получать списки рецептов по выбранным процедурам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по лекарствам с расшифровкой по пациентам и описаниям процедур за период.

№	Тема	Описание
---	------	----------

6. Поликлиника «Будьте больны»

6.3	Поликлиника «Будьте больны». Исполнитель	Программа «Поликлиника «Будьте больны». Исполнитель» предназначена для ведения учета болезней. Пользователи описывают <u>болезни (7)</u> , <u>курсы приема препаратов (8)</u> и <u>симптоматику (9)</u> . Программа предоставляет возможность получения отчетов по болезням. Пользователи могут создавать болезни. Могут создавать курсы приема препаратов и выбирать к ним болезни. Могут создавать записи симптоматики и выбирать к ним болезни. Могут связывать симптоматику с рецептами. Могут получать списки рецептов по выбранным болезням в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по болезням с расшифровкой по симптомам и лекарствам за период.
6.4	Поликлиника «Будьте больны». Поручитель	Программа «Поликлиника «Будьте больны». Поручитель» предназначена для ведения учета процедур. Пользователи, указывают <u>лекарства (11)</u> , <u>рецепты по ним (12)</u> и <u>процедуры (10)</u> , для уменьшения симптоматики болезней. Программа предоставляет возможность получения отчетов по лекарствам и рецептам. Пользователи могут создавать процедуры. Могут создавать лекарства и выбирать к ним процедуры. Могут связывать лекарства с курсами приема препаратов. Могут создавать рецепты и выбирать к ним процедуры. Могут получать списки курсов приема препаратов по выбранным процедурам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по процедурам с расшифровкой по лекарствам и симптомам за период.

№	Тема	Описание
---	------	----------

7. Университет «Все отчислены»

7.1	Университет «Все отчислены». Работник	Программа «Университет «Все отчислены». Работник» на платформе ASP.NET предназначена для ведения учета обучений. Пользователи указывают <u>планы обучения (1)</u> , прописывают <u>студентов (2)</u> и <u>аттестацию (3)</u> по ним. Программа предоставляет возможность получения отчетов по планам обучения. Пользователи могут создавать обучения и выбирать к ним преподавателей. Могут создавать студентов. Могут связывать студентов с обучениями. Могут создавать аттестации и привязывать их к студентам. Могут получать списки преподавателей по выбранным студентам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по обучениям с расшифровкой по дисциплинам и аттестациям за период.
7.2	Университет «Все отчислены». Кладовщик	Программа «Университет «Все отчислены». Кладовщик» предназначена для ведения учета дисциплин. Пользователи указывают проводимые <u>дисциплины (4)</u> , какие <u>преподаватели (5)</u> их ведут, а также <u>промежуточные ведомости (6)</u> . Программа предоставляет возможность получения отчетов по проводимым дисциплинам. Пользователи могут создавать дисциплины. Могут связывать дисциплины и студентов. Могут создавать ведомости и привязывать их к преподавателям. Могут создавать преподавателей и выбирать к дисциплины. Могут получать списки студентов по выбранным преподавателям в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по дисциплинам с расшифровкой по планам обучения и ведомостям за период.

№	Тема	Описание
---	------	----------

7. Университет «Все отчислены»

7.3	Университет «Все отчислены». Исполнитель	Программа «Университет «Все отчислены». Исполнитель» предназначена для ведения учета студентов. Пользователи указывают <u>студентов (7)</u> , <u>результаты испытаний (8)</u> и <u>ведомости (9)</u> . Программа предоставляет возможность получения отчетов по студентам. Пользователи могут создавать студентов. Могут создавать результаты испытаний и выбирать к ним студентов. Могут создавать ведомости и выбирать к ним студентов. Могут связывать ведомости с дисциплинами. Могут получать списки дисциплин по выбранным студентам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по студентам с расшифровкой по ведомостям и занятиям за период.
7.4	Университет «Все отчислены». Поручитель	Программа «Университет «Все отчислены». Поручитель» предназначена для ведения учета типов отчетности. Пользователи, указывают <u>типы отчетности (10)</u> , как их применять на <u>занятиях (11)</u> и в <u>дисциплинах (12)</u> . Программа предоставляет возможность получения отчетов по занятиям и дисциплинам. Пользователи могут создавать записи типов отчетности. Могут создавать занятия и выбирать к ним типы отчетности. Могут связывать занятия с результатами испытаний. Могут создавать дисциплины и выбирать к ним типы отчетности. Могут получать списки результатов испытаний по выбранным типам отчетности в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по типам отчетности с расшифровкой по занятиям и ведомостям за период.

№	Тема	Описание
---	------	----------

8. Автоцентр «Корыто»

8.1	Автоцентр «Корыто». Работник	<p>Программа «Автоцентр «Корыто». Работник» на платформе ASP.NET предназначена для ведения учета покупок. Пользователи создают <u>покупки (1)</u>, прописывают необходимые <u>предпродажные работы (2)</u> и <u>пожелания (3)</u> к ним. Программа предоставляет возможность получения отчетов по покупкам. Пользователи могут создавать покупки и выбирать к ним машины. Могут создавать предпродажные работы. Могут связывать работы с покупками. Могут создавать пожелания и привязывать их к работам. Могут получать списки машин по выбранным продажным работам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по покупкам с расшифровкой по комплектациям и пожеланиям за период.</p>
8.2	Автоцентр «Корыто». Кладовщик	<p>Программа «Автоцентр «Корыто». Кладовщик» предназначена для ведения учета комплектаций. Пользователи указывают продаваемые <u>машины (5)</u>, какими <u>комплектациями (4)</u> их можно оснащать, а также <u>особенности (6)</u> конкретных машин. Программа предоставляет возможность получения отчетов по используемым комплектациям. Пользователи могут создавать комплектации. Могут связывать комплектации и предпродажные работы. Могут создавать машины и выбирать к ним комплектации. Могут создавать особенности и привязывать их к машинам. Могут получать списки предпродажных работ по выбранным машинам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по комплектациям с расшифровкой по покупкам и особенностям за период.</p>

№	Тема	Описание
---	------	----------

8. Автоцентр «Корыто»

8.3	Автоцентр «Корыто». Исполнитель	<p>Программа «Автоцентр «Корыто». Исполнитель» предназначена для ведения учета продаж. Пользователи создают <u>продажи (7)</u>, <u>предпродажные работы (8)</u> и <u>сотрудников (9)</u>, которые выполняют подготовку машин. Программа предоставляет возможность получения отчетов по продажам. Пользователи могут создавать продажи. Могут создавать предпродажные работы и выбирать к ним продажи. Могут создавать сотрудников и выбирать к ним продажи. Могут связывать сотрудников с осмотрами. Могут получать списки осмотров по выбранным продажам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по продажам с расшифровкой по сотрудникам и комплектациям за период.</p>
8.4	Автоцентр «Корыто». Поручитель	<p>Программа «Автоцентр «Корыто». Поручитель» предназначена для ведения учета машин. Пользователи, указывают продаваемые <u>машины (10)</u>, <u>комплектации (11)</u> к ним и проводимые <u>осмотры (12)</u>. Программа предоставляет возможность получения отчетов по комплектациям и осмотрам. Пользователи могут создавать машины. Могут создавать комплектации и выбирать к ним записи машины. Могут связывать комплектации с предпродажными работами. Могут создавать осмотры и выбирать к ним машины. Могут получать списки предпродажных работ по выбранным машинам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по машинам с расшифровкой по комплектациям и сотрудникам за период.</p>

№	Тема	Описание
---	------	----------

9. Магазин компьютерной техники «Ты ж программист»

9.1	Магазин компьютерной техники «Ты ж программист». Работник	Программа «Магазин компьютерной техники «Ты ж программист». Работник» на платформе ASP.NET предназначена для ведения учета покупок. Пользователи создают <u>покупки (1)</u> , прописывают необходимые <u>сборки (2)</u> и <u>комментарии (3)</u> к сборкам. Программа предоставляет возможность получения отчетов по покупкам. Пользователи могут создавать покупки и выбирать к ним товары. Могут создавать сборки. Могут сборки с покупками. Могут создавать комментарии и привязывать их к сборкам. Могут получать списки товаров по выбранным сборкам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по покупкам с расшифровкой по комплектующим и комментариям за период.
9.2	Магазин компьютерной техники «Ты ж программист». Кладовщик	Программа «Магазин компьютерной техники «Ты ж программист». Кладовщик» предназначена для ведения учета комплектующих. Пользователи указывают продаваемые <u>товары (5)</u> , <u>комплектующие (4)</u> , а также <u>заказы на товары (6)</u> . Программа предоставляет возможность получения отчетов по комплектующим. Пользователи могут создавать комплектующие. Могут связывать комплектующие и сборки. Могут создавать товары и выбирать к ним комплектующие. Могут создавать заказы на товары и привязывать их к товарам. Могут получать списки сборок по выбранным товарам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по комплектующим с расшифровкой по покупкам и заказам на товары за период.

№	Тема	Описание
---	------	----------

9. Магазин компьютерной техники «Ты ж программист»

9.3	Магазин компьютерной техники «Ты ж программист». Исполнитель	Программа «Магазин компьютерной техники «Ты ж программист». Исполнитель» предназначена для ведения учета заказов. Пользователи формируют <u>заказы (7)</u> , из <u>партий товаров (8)</u> и <u>заявок (9)</u> . Программа предоставляет возможность получения отчетов по заказам. Пользователи могут создавать заказы. Могут создавать партии товаров и выбирать к ним заказы. Могут создавать заявки и выбирать к ним заказы. Могут связывать заявки со сборками. Могут получать списки сборок по выбранным заказам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по заказам с расшифровкой по заявкам и товарам за период.
9.4	Магазин компьютерной техники «Ты ж программист». Поручитель	Программа «Магазин компьютерной техники «Ты ж программист». Поручитель» предназначена для ведения учета комплектующих. Пользователи, указывают из имеющихся <u>комплектующих (10)</u> , делают готовые <u>товары (11)</u> и персональные <u>сборки (12)</u> по заявкам. Программа предоставляет возможность получения отчетов по товарам и сборкам. Пользователи могут создавать комплектующие. Могут создавать товары и выбирать к ним комплектующие. Могут связывать товары с партиями товаров. Могут создавать сборки и выбирать к ним комплектующие. Могут получать списки партий товаров по выбранным комплектующим в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по комплектующим с расшифровкой по товарам и заявкам за период.

№	Тема	Описание
---	------	----------

10. Завод «Иди работать»

10.1	Завод «Иди работать». Работник	<p>Программа «Завод «Иди работать». Работник» на платформе ASP.NET предназначена для ведения учета заготовок. Пользователи указывают <u>заготовки (1)</u>, из которых будут делаться изделия, прописывают <u>планы производства (2)</u> и <u>этапы выполнения (3)</u>. Программа предоставляет возможность получения отчетов по заготовкам. Пользователи могут создавать заготовки и выбирать к ним изделия. Могут создавать производства. Могут связывать производства с заготовками. Могут создавать этапы выполнения и привязывать их к производствам. Могут получать списки изделий по выбранным производствам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по заготовкам с расшифровкой по станкам и этапам выполнения за период.</p>
10.2	Завод «Иди работать». Кладовщик	<p>Программа «Завод «Иди работать». Кладовщик» предназначена для ведения учета станков. Мастера указывают имеющиеся <u>станки (4)</u>, производимые <u>изделия (5)</u>, а также <u>требования к ним (6)</u>. Программа предоставляет возможность получения отчетов по используемым станкам. Пользователи могут создавать станки. Могут связывать станки и планы производства. Могут создавать изделия и выбирать к ним станки. Могут создавать требования и привязывать их к изделиям. Могут получать списки планов производства по выбранным изделиям в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по станкам с расшифровкой по заготовкам и требованиям за период.</p>

№	Тема	Описание
---	------	----------

10. Завод «Иди работать»

10.3	Завод «Иди работать». Исполнитель	Программа «Завод «Иди работать». Исполнитель» предназначена для ведения учета деталей. Пользователи формируют <u>детали (7)</u> , в каких <u>изделиях (8)</u> они используются, и какие <u>производства (9)</u> будут задействованы в изготовлении. Программа предоставляет возможность получения отчетов по изделиям. Пользователи могут создавать детали. Могут создавать изделия и выбирать к ним детали. Могут создавать производства и выбирать к ним детали. Могут связывать производства с цехами. Могут получать списки цехов по выбранным деталям в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по деталям с расшифровкой по производствам и станкам за период.
10.4	Завод «Иди работать». Поручитель	Программа «Завод «Иди работать». Поручитель» предназначена для ведения учета работников. Пользователи, указывают <u>работников (10)</u> , <u>станки (11)</u> , на которых они работают и <u>цеха (12)</u> , в которых они работают. Программа предоставляет возможность получения отчетов по станкам и цехам. Пользователи могут создавать работников. Могут создавать станки и выбирать к ним работников. Могут связывать станки с изделиями. Могут создавать цеха и выбирать к ним работников. Могут получать списки изделий по выбранным работникам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по работникам с расшифровкой по станкам и производствам за период.

№	Тема	Описание
---	------	----------

11. Ветклиника «Айболит»

11.1	Ветклиника «Айболит». Работник	Программа «Ветклиника «Айболит». Работник» на платформе ASP.NET предназначена для ведения учета визитов. Пользователи создают записи <u>визитов (1)</u> , <u>животных (2)</u> , которые в рамках визита будут обследоваться, и <u>прививки (3)</u> животного. Программа предоставляет возможность получения отчетов по визитам. Пользователи могут создавать визиты и выбирать к ним услуги. Могут создавать животных. Могут связывать животных с визитами. Могут создавать прививки и привязывать их к животным. Могут получать списки услуг по выбранным животным в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по визитам с расшифровкой по медикаментам и прививкам за период.
11.2	Ветклиника «Айболит». Кладовщик	Программа «Ветклиника «Айболит». Кладовщик» предназначена для ведения учета медикаментов. Пользователи указывают имеющиеся <u>медикаменты (4)</u> , предлагаемые <u>услуги (5)</u> , а также <u>рекомендации (6)</u> по ним. Программа предоставляет возможность получения отчетов по используемым медикаментам. Пользователи могут создавать медикаменты. Могут связывать медикаменты и животных. Могут создавать услуги и выбирать к ним медикаменты. Могут создавать рекомендации и привязывать их к услугам. Могут получать списки животных по выбранным услугам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по медикаментам с расшифровкой по визитам и рекомендациям за период.

№	Тема	Описание
---	------	----------

11. Ветклиника «Айболит»

11.3	Ветклиника «Айболит». Исполнитель	Программа «Ветклиника «Айболит». Исполнитель» предназначена для ведения учета животных. Пользователи указывают <u>животных (7)</u> , какие лекарства для них <u>покупаются (8)</u> и их <u>посещения (9)</u> врачей. Программа предоставляет возможность получения отчетов по животным. Пользователи могут создавать животных. Могут создавать покупки и выбирать к ним животных. Могут создавать посещения и выбирать к ним животных. Могут связывать посещения с услугами. Могут получать списки услуг по выбранным животным в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по животным с расшифровкой по посещениям и лекарствам за период.
11.4	Ветклиника «Айболит». Поручитель	Программа «Ветклиника «Айболит». Поручитель» предназначена для ведения учета медикаментов. Пользователи, указывают имеющиеся <u>медикаменты (10)</u> , <u>лекарства (11)</u> на их основе и оказываемые <u>услуги (12)</u> . Программа предоставляет возможность получения отчетов по лекарствам и услугам. Пользователи могут создавать медикаменты. Могут создавать лекарства и выбирать к ним медикаменты. Могут связывать лекарства с покупками. Могут создавать услуги и выбирать к ним медикаменты. Могут получать списки покупок по выбранным медикаментам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по медикаментам с расшифровкой по лекарствам и посещениям за период.

№	Тема	Описание
---	------	----------

12. Банк «Вы банкрот»

12.1	Банк «Вы банкрот». Работник	Программа «Банк «Вы банкрот». Работник» на платформе ASP.NET предназначена для ведения учета клиентов. Клерки указывают <u>клиентов (1)</u> , их <u>вклады (2)</u> и <u>пополнения (3)</u> этих вкладов. Программа предоставляет возможность получения отчетов по клиентам. Пользователи могут создавать клиентов и выбирать к ним кредитные программы. Могут создавать вклады. Могут связывать вклады с клиентами. Могут создавать пополнения и привязывать их к вкладам. Могут получать списки кредитных программ по выбранным вкладам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по клиентам с расшифровкой по валютам и пополнениям за период.
12.2	Банк «Вы банкрот». Кладовщик	Программа «Банк «Вы банкрот». Кладовщик» предназначена для ведения учета валют. Пользователи указывают с какими <u>валютами (4)</u> возможна работа, предлагаемые <u>кредитные программы (5)</u> , а также <u>сроки (6)</u> по ним. Программа предоставляет возможность получения отчетов по используемым валютам. Пользователи могут создавать валюты. Могут связывать валюты и вклады. Могут создавать кредитные программы и выбирать к ним валюту. Могут создавать сроки и привязывать их к кредитным программам. Могут получать списки вкладов по выбранным кредитным программам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по валютам с расшифровкой по клиентам и срокам за период.

№	Тема	Описание
---	------	----------

12. Банк «Вы банкрот»

12.3	Банк «Вы банкрот». Исполнитель	Программа «Банк «Вы банкрот». Исполнитель» предназначена для ведения учета банковских карт. Пользователи указывают <u>карты (7)</u> , <u>заявки (8)</u> на получение с них наличных и <u>операции (9)</u> на пополнения. Программа предоставляет возможность получения отчетов по картам. Пользователи могут создавать карты. Могут создавать заявки и выбирать к ним карты. Могут создавать операции и выбирать к ним карты. Могут связывать операции с переводами денег. Могут получать списки переводов денег по выбранным картам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по картам с расшифровкой по операциям и выдачам наличности за период.
12.4	Банк «Вы банкрот». Поручитель	Программа «Банк «Вы банкрот». Поручитель» предназначена для ведения учета счетов. Пользователи, указывают имеющиеся <u>счета (10)</u> , <u>выдачу наличных (11)</u> на их основе и <u>переводы денег (12)</u> . Программа предоставляет возможность получения отчетов по выдачам и переводам. Пользователи могут создавать счета. Могут создавать записи выдачи наличности и выбирать к ним счета. Могут связывать записи выдачи наличности с заявками. Могут создавать переводы денег и выбирать к ним счета. Могут получать списки заявок по выбранным счетам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по счетам с расшифровкой по выдачам наличности и операциям за период.

№	Тема	Описание
---	------	----------

13. Юридическая фирма «Вас обманут»

13.1	Юридическая фирма «Вас обманут». Работник	Программа «Юридическая фирма «Вас обманут». Работник» на платформе ASP.NET предназначена для ведения учета дел. Клиент указывают свои <u>дела (1)</u> , <u>заседания (2)</u> по ним и <u>постановления (3)</u> . Программа предоставляет возможность получения отчетов по делам. Пользователи могут создавать дела и выбирать к ним услуги. Могут создавать заседания. Могут связывать заседания с делами. Могут создавать постановления и привязывать их к заседаниям. Могут получать списки услуг по выбранным заседаниям в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по делам с расшифровкой по юристам и постановлениям за период.
13.2	Юридическая фирма «Вас обманут». Кладовщик	Программа «Юридическая фирма «Вас обманут». Кладовщик» предназначена для ведения учета юристов. Пользователи указывают своих <u>юристов (4)</u> , оказываемые <u>услуги (5)</u> , а также <u>соглашения (6)</u> по ним. Программа предоставляет возможность получения отчетов по используемым юристам. Пользователи могут создавать юристов. Могут связывать юристов и заседания. Могут создавать услуги и выбирать к ним юристов. Могут создавать соглашения и привязывать их к услугам. Могут получать списки заседаний по выбранным услугам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по юристам с расшифровкой по делам и соглашениям за период.

№	Тема	Описание
---	------	----------

13. Юридическая фирма «Вас обманут»

13.3	Юридическая фирма «Вас обманут». Исполнитель	Программа «Юридическая фирма «Вас обманут». Исполнитель» предназначена для ведения учета клиентов. Пользователи указывают <u>клиентов (7)</u> , их <u>визиты (8)</u> и <u>дела (9)</u> . Программа предоставляет возможность получения отчетов по клиентам. Пользователи могут создавать клиентов. Могут создавать визиты и выбирать к ним клиентов. Могут создавать дела и выбирать к ним клиентов. Могут связывать дела со слушаниями. Могут получать списки слушаний по выбранным клиентам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по клиентам с расшифровкой по делам и консультациям за период.
13.4	Юридическая фирма «Вас обманут». Поручитель	Программа «Юридическая фирма «Вас обманут». Поручитель» предназначена для ведения учета юристов. Пользователи, указывают имеющихся <u>юристов (10)</u> , оказываемые ими <u>консультации (11)</u> и <u>слушания (12)</u> , на которых они присутствуют. Программа предоставляет возможность получения отчетов по консультациям и слушаниям. Пользователи могут создавать юристов. Могут создавать консультации и выбирать к ним юристов. Могут связывать консультации с визитами. Могут создавать слушания и выбирать к ним юристов. Могут получать списки визитов по выбранным юристам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по юристам с расшифровкой по консультациям и делам за период.

№	Тема	Описание
---	------	----------

14. СТО «Руки-крюки»

14.1	СТО «Руки-крюки». Работник	Программа «СТО «Руки-крюки». Работник» на платформе ASP.NET предназначена для ведения учета ТО. Работники указывают доступные <u>ТО (1)</u> , <u>машины (2)</u> , которые их проходили, и <u>записи сервисов (3)</u> машин. Программа предоставляет возможность получения отчетов по ТО. Пользователи могут создавать записи ТО и выбирать к ним работы. Могут создавать машины. Могут связывать машины с ТО. Могут создавать записи сервисов и привязывать их к машинам. Могут получать списки работ по выбранным машинам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по ТО с расшифровкой по запчастям и записям сервисов за период.
14.2	СТО «Руки-крюки». Кладовщик	Программа «СТО «Руки-крюки». Кладовщик» предназначена для ведения учета запчастей. Пользователи указывают <u>работы (5)</u> , которые выполняются в сервисе, <u>запчасти (4)</u> , необходимые для работ и <u>продолжительность работ (6)</u> по ним. Программа предоставляет возможность получения отчетов по используемым запчастям. Пользователи могут создавать запчасти. Могут связывать запчасти и машины. Могут создавать работы и выбирать к ним запчасти. Могут создавать продолжительности работ и привязывать их к работам. Могут получать списки машин по выбранным работам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по запчастям с расшифровкой по ТО и продолжительностям работ за период.

№	Тема	Описание
---	------	----------

14. СТО «Руки-крюки»

14.3	СТО «Руки-крюки». Исполнитель	Программа «СТО «Руки-крюки». Исполнитель» предназначена для ведения учета машин. Пользователи указывают <u>машины (7)</u> , их <u>неисправности (8)</u> и проводимые <u>ТО (9)</u> . Программа предоставляет возможность получения отчетов по машинам. Пользователи могут создавать машины. Могут создавать неисправности и выбирать к ним машины. Могут создавать записи ТО и выбирать к ним машины. Могут связывать записи ТО с работами. Могут получать списки работ по выбранным машинам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по машинам с расшифровкой по ТО и ремонтам за период.
14.4	СТО «Руки-крюки». Поручитель	Программа «СТО «Руки-крюки». Поручитель» предназначена для ведения учета запчастей. Пользователи, указывают имеющиеся <u>запчасти (10)</u> , выполняемые <u>ремонт (11)</u> и оказываемые <u>работы (12)</u> . Программа предоставляет возможность получения отчетов по ремонтам и работам. Пользователи могут создавать запчасти. Могут создавать ремонт (11) и выбирать к ним запчасти. Могут связывать ремонт (11) с неисправностями. Могут создавать работы и выбирать к ним запчасти. Могут получать списки неисправностей по выбранным запчастям в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по запчастям с расшифровкой по ремонтам и ТО за период.

№	Тема	Описание
---	------	----------

15. Фабрика мебели «Мягкое место»

15.1	Фабрика мебели «Мягкое место». Работник	<p>Программа «Фабрика мебели «Мягкое место». Работник» на платформе ASP.NET предназначена для ведения учета мебельных модулей. Менеджеры на основе <u>мебельных модулей (1)</u>, формируют <u>гарнитуры (2)</u> и их <u>продажи (3)</u>. Программа предоставляет возможность получения отчетов по мебельным модулям. Пользователи могут создавать мебельные модули и выбирать к ним мебель. Могут создавать гарнитуры. Могут связывать гарнитуры с модулями. Могут создавать продажи и привязывать их к гарнитурам. Могут получать списки мебели по выбранным гарнитурам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по мебельным модулям с расшифровкой по материалам и продажам за период.</p>
15.2	Фабрика мебели «Мягкое место». Кладовщик	<p>Программа «Фабрика мебели «Мягкое место». Кладовщик» предназначена для ведения учета материалов. Плотники указывают <u>мебель (5)</u>, <u>материалы (4)</u>, необходимые для их создания, и <u>области применения (6)</u>. Программа предоставляет возможность получения отчетов по используемым материалам. Пользователи могут создавать материалы. Могут связывать материалы и гарнитуры. Могут создавать мебель и выбирать к ним материалы. Могут создавать области применения и привязывать их к мебели. Могут получать списки гарнитур по выбранной мебели в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по материалам с расшифровкой по модулям и областям применения за период.</p>

№	Тема	Описание
---	------	----------

15. Фабрика мебели «Мягкое место»

15.3	Фабрика мебели «Мягкое место». Исполнитель	Программа «Фабрика мебели «Мягкое место». Исполнитель» предназначена для ведения учета салонов продаж. Пользователи указывают <u>салоны продаж (7)</u> , <u>заказы (8)</u> на мебель и продаваемые <u>гарнитуры (9)</u> . Программа предоставляет возможность получения отчетов по продажам. Пользователи могут создавать салоны продаж. Могут создавать заказы и выбирать к ним салоны продаж. Могут создавать гарнитуры и выбирать к ним салоны продаж. Могут связывать гарнитуры с модулями для гарнитуров. Могут получать списки модулей для гарнитуров по выбранным салонам продаж в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по салонам продаж с расшифровкой по гарнитурам и мебели за период.
15.4	Фабрика мебели «Мягкое место». Поручитель	Программа «Фабрика мебели «Мягкое место». Поручитель» предназначена для ведения учета материалов. Пользователи, указывают имеющиеся <u>материалы (10)</u> , производимую из них <u>мебель (11)</u> и <u>модули для гарнитуров (12)</u> . Программа предоставляет возможность получения отчетов по мебели и модулям. Пользователи могут создавать материалы. Могут создавать мебель и выбирать к ним материалы. Могут связывать мебель с заказами. Могут создавать модули для гарнитуров и выбирать к ним материалы. Могут получать списки заказов по выбранным материалам в формате текстового документа или электронной таблицы. Могут получать сведения на почту или в программе по материалам с расшифровкой по мебели и гарнитурам за период.

Список использованных источников

1. Pro Git. 2nd Edition [Электронный ресурс] / Режим доступа: <https://git-scm.com/book/ru/v2>.
2. METANIT.COM. Сайт о программировании [Электронный ресурс] / Режим доступа: <https://metanit.com/sharp/>. – Загл. с экрана.
3. ProfessorWeb. .Net & Web Programming [Электронный ресурс] / Режим доступа: <https://professorweb.ru/>. – Загл. с экрана.
4. Tiberiu Covaci, Rod Stephens, Vincent Varallo, Gerry O'Brien. MCSD Certification Toolkit (Exam 70-483) // Published by John Wiley & Sons, Inc. – 2013. – 656p.
5. MCTS Self-Paced Training Kit (Exam 70-536): Microsoft .NET Framework–Application Development Foundation, Second Edition eBook // Published by Microsoft Press. – 2009. – 829 p.
6. Бабищ А.В. Введение в UML [Электронный ресурс]/ Бабищ А.В.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 198 с.— Режим доступа: https://www.intuit.ru/goods_store/ebooks/8359.
7. Informicus. Диаграммы вариантов использования [Электронный ресурс] / Режим доступа: <http://www.informicus.ru/default.aspx?SECTION=6&id=73&subdivisionid=4>. – Загл. с экрана.
8. Теория и практика UML. Диаграмма последовательности[Электронный ресурс] / Режим доступа: http://it-gost.ru/articles/view_articles/94. – Загл. с экрана.

ПРИЛОЖЕНИЕ 1

Титульный лист

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет ИСТ

Кафедра «Информационные системы»

Дисциплина «Разработка профессиональных приложений»

КУРСОВАЯ РАБОТА

Тема указать свою тему по приказу

Выполнил студент

подпись

/Фамилия И.О./

инициалы, фамилия

Курс второй

Группа ПИБД-21

Направление/специальность 09.03.04 «Программная инженерия» (профиль «Программная инженерия»)

Руководитель

доцент

должность, ученая степень, ученое звание

Эгов Е.Н.

фамилия, имя, отчество

Дата сдачи:

«___» _____ 2020 г.

Дата защиты:

«___» _____ 2020 г.

Оценка:

Ульяновск
2020 г.

ПРИЛОЖЕНИЕ 2

Задание на курсовую работу

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет ИСТ
Кафедра «Информационные системы»
Дисциплина «Разработка профессиональных приложений»

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студенту ПИБД-21 Фамилия И.О.
группа фамилия, инициалы

Тема работы указать свою тему по приказу

Срок сдачи законченной работы «__» _____ 2020 г.

Исходные данные к работе: описание задания по теме, утвержденной распоряжением деканата ФИСТ

Рекомендуемая литература: курс лекций по дисциплине «Технологии программирования», методические указания к лабораторным работам по дисциплине «Технологии программирования», интернет-источники.

Содержание пояснительной записки (перечень подлежащих разработке вопросов)

Введение. Описание актуальности задачи.

Первая глава. Описание предметной области, поиск аналогов, ТЗ. Представление диаграмм с их описанием.

Вторая глава. Представление руководства пользователя для разработанного проекта.

Третья глава. Представление руководства программиста для разработанного проекта.

Перечень графического материала (с точным указанием обязательных чертежей)

Диаграммы UML: диаграмма вариантов использования (use-case), диаграмма последовательности (sequence), диаграмма состояния (state-machine).

ER-диаграмма.

Скриншоты разработанного программного продукта.

Руководитель доцент _____ /Эгов Е.Н./
должность подпись инициалы, фамилия

«__» _____ 2020 г.

Студент _____ /Фамилия И.О./
подпись инициалы, фамилия

«__» _____ 2020 г.

Отзыв руководителя

ОТЗЫВ

« » 2020 г.

ПРИЛОЖЕНИЕ 4

Шаблон технического задания для роли «Работник (первая роль)»

Техническое задание на разработку программного продукта «Учет литературы. Работник»

Программа «Учет литературы. Работник» предназначена для ведения учета выдаваемой студентам литературы для изучения на занятиях. Студенты могут создавать обучения и курсы, на основе которых им будут выдаваться литература и проставлять оценки за курс. Программа предоставляет возможность получения отчетов по выдаваемой литературе для студентов.

Основной функционал программы:

- Регистрация. Для регистрации пользователь должен заполнить следующие данные:

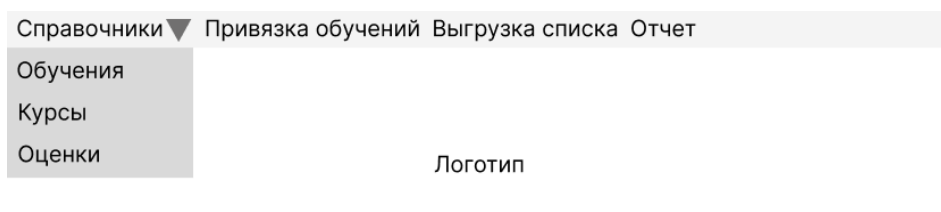
Данные от пользователя. Указать требования к каждому полю. Пример: логин. Логин должен быть уникальным, иметь длину не более 50 символов. Обязателен к заполнению.

The screenshot shows a registration form with the title 'Регистрация' at the top. It contains three input fields: 'Поле 1:', 'Поле 2:', and 'Поле N:'. Below the fields are two buttons: 'Сохранить' (Save) and 'Отмена' (Cancel).

- Авторизация. Пользователь вводит данные. Если такой пользователь есть в системе, то происходит переход к основной форме. Если такого пользователя нет в системе, выводится сообщение об этом с просьбой ввести заново данные.

The screenshot shows a login form with the title 'Вход в систему'. It contains two input fields: 'Логин:' and 'Пароль: *****'. Below the fields are two buttons: 'Вход' (Login) and 'Регистрация' (Registration).

- Основная форма. Состоит из пунктов меню и логотипа. Через пункты меню можно перейти на формы работы с **обучениями**, **курсами** (в том числе и **привязка курсов**), **оценками**, на форму получения списка **литературы** и на форму получения отчета. Дополнительно, если реализовано, то вызов формы рекомендательной подсистемы.



- Формирование **обучений** (CRUD). Имеется форма со списком всех **обучений** пользователя (которые создавал авторизовавшийся пользователь) и кнопки для создания, изменения и удаления **обучения**. Пользователь может создать **запись обучения**, включающую следующие поля:

[Данные для записи \(2-3 поля\).](#)

Имеется возможность выбирать в создаваемую запись доступные **занятия** из списка (несколько) и сохранить изменения.

- Формирование **курсов** (CRUD). Имеется форма со списком всех **курсов** пользователя и кнопки для создания, изменения и удаления **курса**. Запись **курса** включает в себя следующие поля:
- [Данные для записи \(2-3 поля\).](#)

- Привязка **курсов** к **обучениям**. Пользователь может выбрать **курс**, указать семестр, выбрать **обучения** (из тех, что создавал сам пользователь) и связать их.

Привязка обучений

Выбранный курс обучения из списка
▼

Семестр

☐ Обучение 1
☐ Обучение 2
☐ Обучение 3
☐ Обучение 4

Привязать

Отмена

- Формирование **оценок от преподавателя** (CRUD). Имеется форма со списком всех **оценок** пользователя и кнопки для создания, изменения и удаления **оценки**. Запись **оценки** включает в себя следующие поля:

Данные для записи (2-3 поля).

Имеется возможность выбрать запись **курса** из списка (выпадающий список, только курсы, из тех, что создавал сам пользователь) к записи **оценки**.

- Получение списка. Пользователь может получить список **литературы** по выбранным **обучениям** (из тех, что создавал сам пользователь). Пользователь отмечает интересующие его **обучения**, указывает формат файла (doc или xls), указывает имя и место сохранения файла и получает интересующий его список.

Выгрузка списка

☐ Обучение 1
☐ Обучение 2
☐ Обучение 3
☐ Обучение 4

Word

Excel

Отмена

При выводе в doc-файл формат документа будет следующим:

указать как будет выглядеть документ, можно приложить скрин.

При выводе в xls-файл формат документа будет следующим:

указать как будет выглядеть документ, можно приложить скрин.

- Пользователь может получить сведения по полученной им литературе и полученным оценкам за период по различным обучениям (из тех, что создавал сам пользователь). Для этого он должен указать интересующий его период, выбрать вариант получения сведений (отправка на почту или вывод на форму) и получить отчет.

		Отчет			
С	Компонент выбора даты	по	Компонент выбора даты	Показать	На почту
Вывод табличного представления отчета					

Указать, как будет выглядеть отчет, можно приложить скрин.

Если проект разрабатывается индивидуально, то указать следующие пункты:

- Метод создания литературы. Определить, когда должен вызываться метод. Описать как будет выполняться работа метода (либо случайно, либо по какому-то алгоритму).
- Метод создания занятий. Определить, когда должен вызываться метод. Описать как будет выполняться работа метода (либо случайно, либо по какому-то алгоритму).
- Метод привязки литературы к занятиям. Метод должен привязывать существующую литературу к созданным занятиям. Определить, когда должен вызываться метод. Описать как будет выполняться работа метода (либо случайно, либо по какому-то алгоритму).

В качестве интерфейса пользователя будет выступать Web-клиент, разработанный на ASP.NET.

Шаблон технического задания для роли «Кладовщик (вторая роль)»

Техническое задание на разработку программного продукта «Учет литературы. Кладовщик»

Программа «Учет литературы. Кладовщик» предназначена для ведения учета выдаваемой студентам литературы для проведения занятий. Преподаватели могут формировать список проводимых занятий и списки литературы к ним, а также акты выставления оценок на занятия. Программа предоставляет возможность получения отчетов по выдаваемой литературе.

Основной функционал программы:

- Регистрация. Для регистрации пользователь должен заполнить следующие данные:

Данные от пользователя. Указать требования к каждому полю. Пример: логин. Логин должен быть уникальным, иметь длину не более 50 символов. Обязателен к заполнению.

The screenshot shows a registration form with the title 'Регистрация' at the top. It contains three input fields: 'Поле 1:', 'Поле 2:', and 'Поле N:'. Below the fields are two buttons: 'Сохранить' (Save) and 'Отмена' (Cancel).

- Авторизация. Пользователь вводит данные. Если такой пользователь есть в системе, то происходит переход к основной форме. Если такого пользователя нет в системе, выводится сообщение об этом с просьбой ввести заново данные.

The screenshot shows a login form with the title 'Вход в систему'. It contains two input fields: 'Логин:' and 'Пароль: *****'. Below the fields are two buttons: 'Вход' (Login) and 'Регистрация' (Registration).

- Основная форма. Состоит из пунктов меню и логотипа. Через пункты меню можно перейти на формы работы с **литературой** (в том числе и **привязка курсов**), **занятиями**, **баллами**, на форму получения списка **курсов** и на форму получения отчета по **литературе**. Дополнительно, если реализовано, то вызов формы рекомендательной подсистемы.

- Формирование **литературы** (CRUD). Имеется форма со списком всей **литературы** пользователя и кнопки для создания, изменения и удаления **литературы**. По **литературе** хранятся следующие сведения:

Данные для записи (2-3 поля).

- Привязка **литературы** к **курсам**. Пользователь может выбрать **литературу** (из тех, что создавал сам пользователь), указать количество, выбрать **курсы** и связать их.

- Формирование **занятиям** (CRUD). Имеется форма со списком всех **занятий** пользователя и кнопки для создания, изменения и удаления **занятия**. По **занятиям** хранятся следующие сведения:

Данные для записи (2-3 поля).

Имеется возможность выбрать в создаваемую запись доступную **литературу** (из тех, что создавал сам пользователь) из списка (несколько) и сохранить изменения.

Работа с занятием

Поле 1:

Поле 2:

Выбранная литература

Доступная литература

- Формирование **отметок выставления баллов** (CRUD). Имеется форма со списком всех **отметок** пользователя и кнопки для создания, изменения и удаления **отметки**. Запись **отметки** включает в себя следующие поля:

[Данные для записи \(2-3 поля\).](#)

Имеется возможность выбрать запись **занятия** из списка (выпадающий список, только занятия, **из тех, что создавал сам пользователь**) к записи **отметки**.

- Получение списка. Пользователь может получить список **курсов** по выбранным **занятиям** (**из тех, что создавал сам пользователь**). Пользователь отмечает интересующие его **занятия**, указывает формат файла (doc или xls), указывает имя и место сохранения файла и получает интересующий его список.

Выгрузка списка

☐ Занятие 1
☐ Занятие 2
☐ Занятие 3
☐ Занятие 4

При выводе в doc-файл формат документа будет следующим:

[указать как будет выглядеть документ, можно приложить скрин.](#)

При выводе в xls-файл формат документа будет следующим:

[указать как будет выглядеть документ, можно приложить скрин.](#)

- Пользователь может получить сведения по полученной им **литературе** (из тех, что создавал сам пользователь) за период, с указанием на каких **занятиях** и **курсах** она использовалась. Для этого он должен указать интересующий его период, выбрать вариант получения сведений (отправка на почту или вывод на форму) и получить отчет.

Отчет

С Компонент выбора даты по Компонент выбора даты Показать На почту

Вывод табличного представления отчета

Указать, как будет выглядеть отчет, можно приложить скрин.

Если проект разрабатывается индивидуально, то указать следующие пункты:

- Метод создания **курсов**. Определить, когда должен вызываться метод. Описать как будет проходить формирования (либо рандомно, либо по какому-то алгоритму).
- Метод привязки **курсов** к **обучениям**. Определить, когда должен вызываться метод. Описать как будет проходить формирования (либо рандомно, либо по какому-то алгоритму).
- Метод формирования **обучений**. Метод должен формировать **обучения** и выбирать туда **занятия**. Определить, когда должен вызываться метод. Описать как будет проходить формирования (либо рандомно, либо по какому-то алгоритму).

В качестве интерфейса пользователя будет выступать **desktop-приложение**, разработанное на **WPF**.

ПРИЛОЖЕНИЕ 6

Шаблон технического задания для роли «Исполнитель (третья роль)»

Техническое задание на разработку программного продукта «Учет литературы. Исполнитель»

Программа «Учет литературы. Исполнитель» предназначена для ведения учета выдаваемой студентам в библиотеке литературы для чтения. Студенты указывают занятия, под которые им нужна литература, составляют заявки на получение литературы в библиотеке и отметке о посещении читального зала и какой литературой они там пользовались. Программа предоставляет возможность получения отчетов по заявкам на литературу и посещениям читального зала.

Основной функционал программы:

- Регистрация. Для регистрации пользователь должен заполнить следующие данные:

Данные от пользователя. Указать требования к каждому полю.
Пример: логин. Логин должен быть уникальным, иметь длину не более 50 символов. Обязателен к заполнению.

The screenshot shows a registration form with the title 'Регистрация' at the top. It contains three input fields: 'Поле 1:', 'Поле 2:', and 'Поле N:'. Below the fields are two buttons: 'Сохранить' (Save) and 'Отмена' (Cancel).

- Авторизация. Пользователь вводит данные. Если такой пользователь есть в системе, то происходит переход к основной форме. Если такого пользователя нет в системе, выводится сообщение об этом с просьбой ввести заново данные.

The screenshot shows a login form with the title 'Вход в систему' at the top. It contains two input fields: 'Логин:' and 'Пароль: *****'. Below the fields are two buttons: 'Вход' (Login) and 'Регистрация' (Registration).

- Основная форма. Состоит из пунктов меню и логотипа. Через пункты меню можно перейти на формы работы с **занятиями**, **заявками**, **посещениями** (в том числе и **привязка посещений**), на форму получения списка **занятий** и на форму получения отчета по **занятиям**. Дополнительно, если реализовано, то вызов формы рекомендательной подсистемы.

- Формирование **занятия** (CRUD). Имеется форма со списком всей **занятий** пользователя и кнопки для создания, изменения и удаления **занятия**. По **занятию** хранятся следующие сведения:

Данные для записи (2-3 поля).

- Формирование **заявок на литературу** (CRUD). Имеется форма со списком всех **заявок** пользователя и кнопки для создания, изменения и удаления **заявки**. По **заявке** хранятся следующие сведения:

Данные для записи (2-3 поля).

Имеется возможность выбрать в создаваемую запись доступные **занятия** (из тех, что создавал сам пользователь) из списка (несколько) и сохранить изменения.

- Формирование **записи посещения читального зала** (CRUD). Имеется форма со списком всех **посещений** пользователя и

кнопки для создания, изменения и удаления **посещения**. По **посещению** хранятся следующие сведения:

Данные для записи (2-3 поля).

Имеется возможность выбрать в создаваемую запись доступные **занятия** (из тех, что создавал сам пользователь) из списка (несколько) и сохранить изменения.

- Привязка **записи посещения** (из тех, что создавал сам пользователь) к **записи выдачи литературы**. Пользователь может выбрать **запись посещения**, **запись выдачи** и связать их.

Привязка посещения

Выбранное посещение из списка ▼	
Выбранная выдача из списка ▼	
Привязать	Отмена

- Получение списка. Пользователь может получить список **записей выдачи литературы** по выбранным **занятиям** (из тех, что создавал сам пользователь). Пользователь отмечает интересующие его **занятия**, указывает формат файла (doc или xls), указывает имя и место сохранения файла и получает интересующий его список.

Выгрузка списка

<input type="checkbox"/>	Занятие 1
<input type="checkbox"/>	Занятие 2
<input type="checkbox"/>	Занятие 3
<input type="checkbox"/>	Занятие 4

Word	Excel	Отмена
------	-------	--------

При выводе в doc-файл формат документа будет следующим:

[указать как будет выглядеть документ, можно приложить скрин.](#)

При выводе в xls-файл формат документа будет следующим:

[указать как будет выглядеть документ, можно приложить скрин.](#)

- Пользователь может получить сведения за период по **занятиям** (из тех, что создавал сам пользователь), для которых требовалось **посещение читального зала** и **передача литературы**. Для этого он

должен указать интересующий его период, выбрать вариант получения сведений (отправка на почту или вывод на форму) и получить отчет.

		Отчет			
С	Компонент выбора даты	по	Компонент выбора даты	Показать	На почту
Вывод табличного представления отчета					

Указать, как будет выглядеть отчет, можно приложить скрин.

Если проект разрабатывается индивидуально, то указать следующие пункты:

- Метод создания **литературы**. Метод может вызываться при старте программы, проверять, есть ли **литература** в базе данных и, если их нет, создавать ее там (**создаваться может случайно, либо загружать из файла** (путь до файла можно прописать жестко в коде)).
- Метод создания **записей выдачи литературы**. Определить, когда должен вызываться метод. Описать как будет проходить создание (**либо случайно, либо по определенному алгоритму**).
- Метод привязки **литературы** к **записям выдачи литературы**. Определить, когда должен вызываться метод. Описать как будет проходить формирование (**либо случайно, либо по какому-то алгоритму**).

В качестве интерфейса пользователя будет выступать **Web-клиент**, разработанный на **Web API**.

Шаблон технического задания для роли «Поручитель (четвертая роль)»

Техническое задание на разработку программного продукта «Учет литературы. Поручитель»

Программа «Учет литературы. Поручитель» предназначена для ведения учета выдаваемой библиотекарем литературы студентам для занятий. Библиотекари могут формировать литературу, записи ее выдачи студентам на занятия и в читальном зале. Программа предоставляет возможность получения отчетов по выдаваемой студентам литературе.

Основной функционал программы:

- Регистрация. Для регистрации пользователь должен заполнить следующие данные:

Данные от пользователя. Указать требования к каждому полю. Пример: логин. Логин должен быть уникальным, иметь длину не более 50 символов. Обязателен к заполнению.



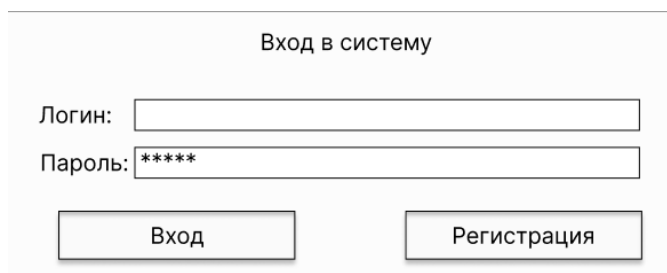
Регистрация

Поле 1:

Поле 2:

Поле N:

- Авторизация. Пользователь вводит данные. Если такой пользователь есть в системе, то происходит переход к основной форме. Если такого пользователя нет в системе, выводится сообщение об этом с просьбой ввести заново данные.



Вход в систему

Логин:

Пароль:

- Основная форма. Состоит из пунктов меню и логотипа. Через пункты меню можно перейти на формы работы с **литературой**, **выдачами**, **передачами** (в том числе и **привязка передач**), на форму получения списка **заявок** и на форму получения отчета по **литературе**. Дополнительно, если реализовано, то вызов формы рекомендательной подсистемы.

Справочники ▼	Привязка передач	Выгрузка списка	Отчет
Литература			
Выдачи			
Передачи			

Логотип

- Формирование **литературы** (CRUD). Имеется форма со списком всей **литературы** пользователя и кнопки для создания, изменения и удаления **литературы**.

По **литературе** хранятся следующие сведения:

Данные для записи (2-3 поля).

- Формирование **записи выдачи литературы** (CRUD). Имеется форма со списком всех **записей выдач** пользователя и кнопки для создания, изменения и удаления **записи выдачи**. По **записи** хранятся следующие сведения:

Данные для записи (2-3 поля).

Имеется возможность выбрать в создаваемую запись доступную **литературу** (из тех, что создавал сам пользователь) из списка (несколько) и сохранить изменения.

Работа с выдачей

Поле 1:

Поле 2:

Выбранная литература	<<	Доступная литература
	>>	
Сохранить		Отмена

- Формирование **записи передачи в руки литературы** (CRUD).
Имеется форма со списком всех **записей передач** пользователя и кнопки для создания, изменения и удаления **записи передачи**. По **записи** хранятся следующие сведения:

Данные для записи (2-3 поля).

Имеется возможность выбрать в создаваемую запись доступную **литературу** (из тех, что создавал сам пользователь) из списка (несколько) и сохранить изменения.

- Привязка **записи передачи литературы** к **записи заявки студента**.
Пользователь может выбрать **запись передачи** (из тех, что создавал сам пользователь), **запись заявки** и связать их.

Привязка передачи

Выбранная передача из списка ▼

Выбранная заявка из списка ▼

Привязать Отмена

- Получение списка. Пользователь может получить список **заявок студентов** по выбранной **литературе** (из тех, что создавал сам пользователь). Пользователь отмечает интересующую его **литературу**, указывает формат файла (doc или xls), указывает имя и место сохранения файла и получает интересующий его список.

Выгрузка списка

☐ Литература 1
☐ Литература 2
☐ Литература 3
☐ Литература 4

Word Excel Отмена

При выводе в doc-файл формат документа будет следующим:

указать как будет выглядеть документ, можно приложить скрин.

При выводе в xls-файл формат документа будет следующим:

указать как будет выглядеть документ, можно приложить скрин.

- Пользователь может получить сведения за период по **литературе** (из тех, что создавал сам пользователь), которая указывалась в **записях передачи** и **посещениях читального зала**. Для этого он должен указать интересующий его период, выбрать вариант получения сведений (отправка на почту или вывод на форму) и получить отчет.

Отчет

С Компонент выбора даты по Компонент выбора даты

Вывод табличного представления отчета

Указать, как будет выглядеть отчет, можно приложить скрин.

Если проект разрабатывается индивидуально, то указать следующие пункты:

- Метод создания **занятий**. Метод может вызываться при старте программы, проверять, есть ли **занятия** в базе данных и, если их нет, создавать ее там (создаваться может случайно, либо загружать из файла (путь до файла можно прописать жестко в коде)).
- Метод создания **записей заявок студентов**. Определить, когда должен вызываться метод. Описать как будет проходить создание (либо случайно, либо по определенному алгоритму).
- Метод привязки **занятий** к **записям заявок**. Определить, когда должен вызываться метод. Описать как будет проходить формирования (либо случайно, либо по какому-то алгоритму).

В качестве интерфейса пользователя будет выступать desktop-приложение, разработанное на WPF.

РАЗРАБОТКА ПРОГРАММЫ УЧЕТА ЛИТЕРАТУРЫ В
АБСТРАКТНОМ УНИВЕРСИТЕТЕ:
практикум к выполнению
курсовых работ по дисциплинам
«Технологии программирования» и «Разработка профессиональных
приложений»

Автор ЭГОВ Евгений Николаевич

Усл. печ. л. 3

УлГТУ, 432027, г. Ульяновск, ул. Сев. Венец, д. 32.