

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)

Кафедра информационных систем и программной инженерии

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Методические указания к курсовому проектированию

Составитель
В.В. Вершинин

Владимир 2013

1. ОБЩИЕ ПОЛОЖЕНИЯ	4
2. ЦЕЛЬ И ЗАДАЧИ КУРСОВОГО ПРОЕКТИРОВАНИЯ	4
3. ЭТАПЫ ВЫПОЛНЕНИЯ ПРОЕКТА И ИХ СОДЕРЖАНИЕ	4
3.1 Требования к порядку выполнения курсового проекта	4
3.2 Календарный план выполнения курсового проекта	5
3.3 Требования оформлению курсового проекта	5
4. СРОКИ И ПОРЯДОК ЗАЩИТЫ КУРСОВОГО ПРОЕКТА	6
5. ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ КУРСОВОГО ПРОЕКТИРОВАНИЯ	6
6. СПИСОК ЛИТЕРАТУРЫ	11
7. ПОРЯДОК ПРОВЕДЕНИЯ КУРСОВОГО ПРОЕКТИРОВАНИЯ	13
8. СОСТАВ И ОПИСАНИЕ ОСНОВНЫХ ЭТАПОВ РАЗРАБОТКИ ПРОГРАММНОЙ СИСТЕМЫ	14
9. ПРИМЕР РАЗРАБОТКИ ПРОГРАММНОЙ СИСТЕМЫ	17
2 АНАЛИЗ	18
2.1 Постановка задач	18
2.2 Описание предметной области	18
2.3 Сценарий взаимодействия пользователей с системой	19
3 ПРОЕКТИРОВАНИЕ	20
3.1 Диаграмма прецедентов	20
3.2 Диаграммы последовательностей	20
3.3 Диаграмма видов деятельности	22
3.4 Диаграммы классов	23
3.5 Диаграмма состояний	24
4 РАЗРАБОТКА	25
4.1 Общие принципы организации системы	25
4.2 Организация доступа к данным	25
4.3 Организация бизнес-логики	26
4.4 Организация веб-интерфейса	26

5 РЕАЛИЗАЦИЯ	27
5.1 Соединение с БД	27
5.2 Создание страниц.....	27
5.3 Маршрутизация	27
5.4 Ограничения целостности	27
5.5 Классы – сущности	28
5.6 Работа с сессиями.....	28
6 ИНСТРУКЦИЯ ПО УСТАНОВКЕ ПРИЛОЖЕНИЯ	29

1. ОБЩИЕ ПОЛОЖЕНИЯ

Общая тематика курсового проектирования по дисциплине «Технологии программирования» связана с рассмотрением и изучением вопросов создания, проектирования и разработки программ, имеющих признаки сложных систем и являющихся средством поддержки и автоматизации функций информационных систем (ИС), частью которых они и являются. Процесс выполнения курсового проекта представляется как взаимосвязь последовательных этапов, связанных с реализацией программной системы, программного комплекса или его части в соответствии с общепринятыми стандартами в области разработки software (например, RUP).

2. ЦЕЛЬ И ЗАДАЧИ КУРСОВОГО ПРОЕКТИРОВАНИЯ

Целью курсового проектирования является изучение процесса разработки сложных систем на примере конкретной программной системы, определяемой тематикой курсового проекта. Для достижения поставленной цели необходимо решить следующие задачи:

- изучение особенностей конкретной предметной области, относящихся к теме курсового проекта;
- анализ возможных подходов и методов решения с обоснованием выбранного подхода;
- разработка моделей (математической, структурной, информационной, данных) программной системы;
- выбор алгоритмов с учетом их точности и спецификой предметной области;
- разработка программного обеспечения;
- анализ полученных результатов работы ПО.

3. ЭТАПЫ ВЫПОЛНЕНИЯ ПРОЕКТА И ИХ СОДЕРЖАНИЕ

3.1 Требования к порядку выполнения курсового проекта

Порядок выполнения курсового проекта определяется общими требованиями к курсовым проектам, принятым в ВлГУ. Конечным результатом курсового проектирования является программная система, выполненная в соответствии требованиями и задачами проектной работы, изложенными в разделе «ЦЕЛЬ И ЗАДАЧИ КУРСОВОГО ПРОЕКТИРОВАНИЯ». При этом студент обязан в течении семестра докладывать о состоянии дел и об уровне завершенности курсового проекта. Все вопросы решаются в рабочем порядке, на занятиях по курсовому проектированию в соответствии с расписанием занятий. В случае возникновения задержек с выполнением курсового проектирования об этом предупреждается заранее или переносится на более поздние сроки в индивидуальном порядке. В течении семестра предусмотрено 3 рубежных контрольных мероприятия с представлением промежуточных материалов:

1. постановка задачи. Анализ задания. Анализ предметной области. Результаты анализа задания и предметной области (техническое задание на разработку ПС, словарь предметной области с перечислением сущностей (и их атрибутов) и характеристикой связей между ними;

2. участники системы и варианты ее использования (фактически это: диаграммы прецедентов взаимодействия с системой и диаграммы последовательностей, диаграмма классов *концептуального* уровня). Анализ прецедентов и формальные функциональные требования к системе;

3. архитектура разрабатываемой ПС (фактически это диаграмма классов, приближенная к реализации, диаграммы состояний, видов деятельности);

4. реализация программной системы (диаграмма модулей, диаграмма развертывания).

3.2 Календарный план выполнения курсового проекта

Курсовой проект выполняется студентами в самостоятельной форме в течении всего семестра. В рамках дисциплины предусмотрены практические занятия, на которых рассматриваются вопросы выполнения каждого этапа курсового проекта. В процессе работы над курсовым проектом студент самостоятельно изучает предметную область выбранной тематики и необходимые для этого литературные (представлены в разделе 5) источники. Конечным результатом выполнения проекта является отчет (пояснительная записка) по выполненной работе. Примерное распределение видов деятельности по неделям учебного семестра представлено в таблице.

Таблица

Примерный календарный план выполнения курсового проектирования

№пп.	Наименование выполняемой работы	Неделя
1.	Согласование тем курсовых проектов и выдача и оформление заданий	1 – 2 недели
2.	Изучение предметной области, изучение структуры и функций информационной системы. Предъявление требований к программной системе и выделение функций программной системы и вариантов ее использования. Составление словаря предметной области и синтез диаграммы прецедентов	3 – 5 недели
3.	Моделирование структуры объектов предметной области и их взаимодействия на концептуальном уровне (уровне предметной области). Распределение функций системы по ее структурным элементам. Составление спецификаций прецедентов системы. Синтез диаграмм последовательностей и диаграмм прецедентов	6 – 7 недели
4.	Проектирование структуры программной системы. Формирование диаграммы классов и диаграммы объектов программной системы. Синтез и построение диаграмм состояний и последовательностей	8 – 9 недели
5.	Реализация программной системы, составление тестов. Сохранение результатов работы в репозиторий проекта	10 – 15 недели
6.	Составление пояснительной записки и графического материала	16 неделя
7.	Сдача курсовых работ и их проверка. Защита	16 – 18 неделя

Допускается незначительное отклонение хода выполнения работ от календарного плана.

3.3 Требования оформлению курсового проекта

Общие требования к оформлению пояснительной записки курсового проекта определяются требованиями к порядку выполнения курсового проекта. Пояснительная записка оформляется в соответствии с требованиями (ГОСТ, ЕСКД, ЕСПД) и должна удовлетворять требованиям, изложенным в [8]. Общий объем пояснитель-

ной записки курсового проекта составляет 30-40 страниц машинописного текста, включая рисунки, таблицы и содержит обязательные части:

- 1) титульный лист;
- 2) лист задания;
- 3) аннотация (на русском и английском языках);
- 4) лист содержания (нумеруется как первый лист в составе пояснительной записки);
- 5) основные разделы записки (введение, анализ, проектирование, разработка, реализация, тестирование; эксплуатационная документация, заключение, список использованных источников);
- 6) приложения.

В отчете по курсовому проектированию должно быть как минимум одно приложение, содержащее исходный код разработанной программной системы. Графический материал вычерчивается на чертежах формата А1. Минимальное количество два листа. Структура и содержание графического материала определяется в листе задания на проект и согласовывается с преподавателем.

Примечание: допускается предоставление уменьшенных копий (до формата А4) чертежей, прикладываемых отдельными листами к пояснительной записке, при условии, что защита будет происходить с использованием электронных средств (презентация).

4. СРОКИ И ПОРЯДОК ЗАЩИТЫ КУРСОВОГО ПРОЕКТА

Срок сдачи курсового проекта происходит на последних неделях учебного семестра. После проверки назначается дата защиты проекта. Защита происходит в составе комиссии, состоящей из трех преподавателей. Общая продолжительность 5 минут. Результаты выполнения курсового проекта и его защита оценивается по пятибалльной системе. Оценка проставляется в ведомость и зачетную книжку студента.

5. ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ КУРСОВОГО ПРОЕКТИРОВАНИЯ

1. Программная система учета знаний учащихся.
2. Программная система автоматизации и учета подвижного состава автобусного предприятия.
3. *Название:* Программная система учета пациентов в информационной системе «Поликлиника» (2)
Краткая характеристика системы: учет и хранение обращений пациентов в поликлинику.
Минимальный набор функций:
 - поддержка (авторизация) различных типов пользователей (администратор, врач, пациент);
 - поддержка справочников диагнозов (адм.);
 - поддержка справочника врачей (адм.);

- выписка рецепта (врач);
- подача заявок и просмотр их состояния (пациент);
- назначение пациенту лечащего врача (система).

4. *Название:* Программная система движения материалов на оптовой базе

Краткая характеристика системы: учет и хранение данных об отпущенных материалах

Минимальный набор функций:

- поддержка (авторизация) различных типов пользователей (администратор, оптовик);
- поддержка справочников материалов и оптовиков (адм.);
- отгрузка/принятие материала (оптовик/система);
- просмотр остатков материалов (оптовик);
- просмотр сведений об операциях отгрузки и принятия товара за выбранный период (оптовик/система).

5. *Название:* Программная система автоматизации учета торговой деятельности компьютерного магазина

Краткая характеристика системы: автоматизация работы компьютерного магазина

Минимальный набор функций:

- поддержка (авторизация) различных типов пользователей (менеджер, продавец);
- поддержка справочника категорий товара (продавец);
- поддержка справочника поставщиков (продавец);
- списание остатков товара (менеджер/система);
- продажа товара продавцом (пациент);
- просмотр оборота продаж продавцом за выбранный период (менеджер).

6. *Название:* Программная система автоматизации документооборота малого предприятия

Краткая характеристика системы: учет поступления документов и их исполнение

Минимальный набор функций:

- поддержка (авторизация) различных типов пользователей (администратор, директор, начальник отдела);
- поддержка справочника отделов, пользователей (адм.);
- поддержка справочника типов документа
- добавление нового документа и его учет по исполнителям (директор/начальник/система);
- отчет по исполненным/неисполненным документам (директор/система);

7. *Название:* Программная система "Туристическое агентство"

Краткая характеристика системы: подбор туров для отдыхающих

Минимальный набор функций:

- поддержка (авторизация) различных типов пользователей (турменеджер, посетитель);

- поддержка справочника туров (турменеджер);
- план отдыха посетителя (аналог корзины);
- вывод статистики по странам отдыха (турменеджер/система);
- автоматическая скидка на «горящие туры» (система/турменеджер);

8. *Название:* Информационно-программная система «Рекламное агентство»

Краткая характеристика системы: автоматизация работы рекламного агентства

Минимальный набор функций:

- поддержка (авторизация) различных типов пользователей (администратор, менеджер, клиент);
- поддержка справочника менеджеров (администратор);
- поддержка справочника категорий рекламы (менеджер);
- поддержка справочника клиентов (менеджер);
- оформление заказа на размещение рекламы (менеджер/система);
- снятие рекламы с оборота по прошествии акции (система/менеджер).

9. *Название:* Программная система автоматизации учета расходных материалов на промышленном предприятии

Краткая характеристика системы: автоматизация учета расходных материалов по различным категориям и по различным структурным единицам

Минимальный набор функций:

- поддержка (авторизация) различных типов пользователей (администратор, начальник подразделения, учетчик);
- поддержка справочника пользователей, (администратор);
- поддержка справочника техники, на которую расходуется материал (начальник подразделения);
- поддержка справочника расходных материалов (учетчик);
- ввод новых расходных материалов (учетчик);
- оценка результата прохождения тестов (система);
- вывод сведений по израсходованным и оставшимся материалам (система/начальник подразделения/учетчик);
- вывод сведений по среднему расходу материала за выбранный период (система/учетчик).

10. *Название:* Программная система «Страховое агентство»

Краткая характеристика системы: организация учета страховой деятельности организации.

Минимальный набор функций:

- поддержка (авторизация) различных типов пользователей (администратор, агент, пользователь);
- поддержка справочника услуг страхования (администратор);
- регистрация договора (агент);
- поддержка справочника юр/физ лиц (агент);
- вывод сведений по страховым договорам за выбранный период (система/агент);
- поиск договоров по критериям (система/агент);
- автоматический вывод сведений по «проблемным» договорам (система).

11. *Название:* Программная система «Деканат университета»

Краткая характеристика системы и минимальный набор функций: обговариваются индивидуально

12. *Название:* Программная система «Дистанционное обучение студента» (тема допускает два исполнителя)

Краткая характеристика системы: организация дистанционного прохождения пользователем системы тестов

Минимальный набор функций:

- поддержка (авторизация) различных типов пользователей (администратор, лектор, студент);
- поддержка справочника лекторов (администратор);
- поддержка справочника предметов (лектор);
- поддержка справочника студентов (администратор);
- прохождение справочника тестов (лектор);
- включение студента в дистанционное прохождение тестов (лектор);
- прохождение тестов (система/студент);
- прохождение тестов (система/студент);
- оценка результата прохождения тестов (система);
- вывод сведений по итогам прохождения теста (система/лектор).

13. *Название:* Программная система «Кадровое агентство»

Краткая характеристика системы: организация работы кадрового агентства;

Минимальный набор функций:

- поддержка (авторизация) различных типов пользователей (администратор, менеджер, пользователь);
- поддержка справочника пользователей (администратор);
- поддержка справочника вакансий (менеджер);
- заполнение анкет от соискателей (менеджер);
- допуск пользователя к просмотру анкет соискателей (менеджер);
- подбор персонала для пользователя (менеджер);
- просмотр базы данных (анкет) соискателей: разовый допуск и допуск на N дней (менеджер/пользователь);

14. *Название:* Программная система «Кадастровое управление»

Краткая характеристика системы: организация работы кадастрового управления;

Минимальный набор функций:

- ввод данных в систему о кадастровых объектах (земельный участок, объект недвижимости и т.д.)
- работа со справочником типов объектов учета в Кадастровом управлении;
- самостоятельная регистрация заявителя в системе;
- подача заявки на регистрацию кадастрового объекта;
- рассмотрение заявок и регистрация (или отказ).
- просмотр сведений о кадастровых объектах и изменение сведений о них

15. *Название:* Регистрационная система «Паспортный стол»

Краткая характеристика системы: организация работы паспортно-визовой службы;

Минимальный набор функций:

- работа со справочником типов документов, регистрируемых в службе(паспорт, регистрация по месту и т.д.);
- регистрация заявителя в системе;
- подача заявки на выдачу/оформление документа;
- рассмотрение заявок на оформление документа (или отказ) и последующая выдача.
- поиск и просмотр сведений о заявках и изменение сведений о них.

16. *Название:* Программная система «База знаний отрасли»

Краткая характеристика системы: организация и поддержка работы системы хранения знания по отраслям;

Минимальный набор функций:

- работа следующих категорий пользователей в системе: администратор, менеджер, пользователи (читатели);
- работа со справочниками отраслей, разделов знаний в отрасли (например: тяжелая промышленность/горное дело/чёрная металлургия и т.п.)
- добавление статей в базу знаний;
- комментирование статей и их модерация (добавление к статье и отклонение).

17. *Название:* Подсистема учета расходных материалов промышленного предприятия

Краткая характеристика системы: организация учета расходных материалов в процессе работы промышленного предприятия;

Минимальный набор функций:

- работа следующих категорий пользователей в системе: администратор, заведующий склада, учетчики;
- работа со справочниками категорий материалов (например: инструмент, доска, бумага для принтера и т.п.);
- ввод поступлений материалов;
- создание, открытие и закрытие табеля оперативного расхода материала;
- вывод сведений об остатке материала на основании данных по оперативному расходу в виде таблицы рас.

18. *Название:* Программная система учета пассажирских перевозок информационной системы «Автотранспортное предприятие»

Краткая характеристика системы: поддержка работы автотранспортного предприятия и автоматизация деятельности;

Минимальный набор функций:

- работа следующих категорий пользователей в системе: администратор, диспетчер, водитель, пользователь (покупатель);
- работа со справочниками маршрутов, автотранспортных единиц, персонала;
- ввод маршрутов и распределение персонала (водитель + кондуктор) по ним;

- вход (+саморегистрация) покупателей и бронирование/покупка билета на рейс;
- подтверждение брони/покупки диспетчером
- просмотр актуальных сведений о маршрутах.

19. *Название:* Программная система муниципальной образовательной информационной сети

Краткая характеристика системы: организация работы кадрового агентства;

Минимальный набор функций:

- работа следующих категорий пользователей в системе: администратор, представитель информационной сети, сотрудник образовательного учреждения;
- работа со справочниками образовательных учреждений;
- добавление годовых отчетов от муниципального образовательного учреждения;
- утверждение отчета представителем сети (или отправка на доработку);
- фильтрация и вывод сведений по образовательным учреждениям.

Перечень основных требований КП

1. к процессу разработки

- использование хранилища репозитория проекта и документации к нему в SVN (Source Forge <http://sourceforge.net/> или Google Code <https://code.google.com/hosting/>) и средств планирования работы над проектом
- регулярное обновление статуса выполнения работы на сайте кафедры.

2. к работе с данными

- разделение слоя доступа к данным и слоя работы с ними

3. к реализации

- а) использование фонового кода и страниц разметки: **а6**) использование MVC;

4. к оформлению

- без помарок и исправлений;
- с учетом требований к оформлению ПЗ (ГОСТ и университетские стандарты)

5. к процедуре защиты

- полный перечень материалов;
- предварительная демонстрация;
- корректно оформленная презентация.

6. СПИСОК ЛИТЕРАТУРЫ

1. Липаев В.В. Проектирование программных систем. М.: Высш. шк, 1990.
2. Буч Г. Объектно-ориентированное проектирование / Пер. с англ. Конкорд, 1996.
3. Майерс Г. Надежность программного обеспечения. М.: Мир, 1980

4. Калянов Г.Н. CASE структурный системный анализ (автоматизация и применение). – М.: «ЛЮРИ», 1996.
5. Шмуллер Дж. Освой самостоятельно UML. 2-е изд. М.: изд. дом «Вильямс», 2002
6. Калберстон Р., Браун К., Кобб Г. Быстрое тестирование М.: изд. дом «Вильямс», 2002
7. Кармайл Э., Хейвуд Д. Быстрая и качественная разработка программного обеспечения. М.: изд. дом «Вильямс», 2003
8. Буланкин В.Б. «Стандарт предприятия» 2004 год

7. ПОРЯДОК ПРОВЕДЕНИЯ КУРСОВОГО ПРОЕКТИРОВАНИЯ

1. Постановка задачи и анализ предметной области

Цель: провести анализ предметной области для проектирования архитектуры приложения и перехода к следующим этапам разработки системы.

Задачи: исследовать предметную область, выделить основные сущности и их атрибуты; определить условия функционирования разрабатываемой системы и ее основные функции.

Результат: техническое задание на разработку системы; словарь предметной области; перечень требований к разрабатываемой программной системе и варианты ее использования (оформленные в виде диаграммы UseCase), концептуальная модель предметной области.

2. Участники системы и варианты ее использования (фактически это: диаграммы прецедентов взаимодействия с системой и диаграммы последовательностей, диаграмма классов *концептуального* уровня). Анализ прецедентов и формальные функциональные требования к системе.

При построении диаграмм вариантов использования нужно придерживаться следующих принципов (правил):

– не моделируйте связи между действующими лицами. По определению действующие лица находятся вне сферы действия системы. Это означает, что связи между ними также не относятся к ее компетенции.

– не соединяйте стрелкой два варианта использования непосредственно. Диаграммы данного типа описывают только сами варианты использования, а не порядок их выполнения. Для отображения порядка выполнения вариантов использования применяют диаграммы деятельности.

– Каждый вариант использования должен быть инициирован действующим лицом. Это означает, что всегда должна быть стрелка, начинающаяся на действующем лице и заканчивающаяся на варианте использования.

3. Архитектура разрабатываемой ПС (фактически это диаграмма классов, приближенная к реализации, диаграммы состояний, видов деятельности).

4. Реализация программной системы (диаграмма модулей, диаграмма развертывания).

8. Состав и описание основных этапов разработки программной системы

8.1. Анализ

Каждая предметная область оперирует со своими понятиями и терминами, а также объектами и связями между ними.. Для того, чтобы разрабатываемая система была адекватна предметной области, была понятна тем, кто с ней будет работать необходим качественный и всесторонний анализ предметной области.

В промышленной разработке ПС разработчику всегда приходится тесно общаться с заказчиком, для наиболее полного выявления фактов предметной области, выделения объектов системы и установления отношений между ними. Кроме этого на данном этапе нужно определить границы системы и выделить основные функции, которые необходимо будет реализовывать. Все эти моменты определяются результатами:

- обследования объекта заказчика, для которого и разрабатывается ПС;
- переговоров заказчика и исполнителя по конкретным вопросам;
- конкретных требований предъявляемых заказчиком к системе.

Итогом анализа является техническое задание на разрабатываемую систему. Формально этот документ описывает общее видение системы с описанием требований к ней и основных функций системы. Дополнительно там же оговариваются вопросы коммерческой составляющей для разработки системы, основные этапы и календарные сроки. В некоторых случаях по общей договоренности сторон в задание вносятся дополнительные пункты.

При выполнении анализа происходит формирование перечня понятий предметной области, основных сущностей и функций будущей системы.

В курсовой работе в качестве заказчика выступает преподаватель, выдавший . Задача студента на данном этапе провести анализ предметной области, основываясь на своих собственных знаниях, литературных источников и в ходе общения с преподавателем. Результат должен быть оформлен в виде краткой характеристики предметной области (которое является частью введения пояснительной записки). За описанием должен следовать словарь предметной области, состоящий из списка объектов и их характеристик (как совокупности свойства/атрибутов и операций над ним/поведения). Объект и его свойства определяются как существительные, а его поведения как глагольные формы русского языка. . Словарь логически связан с описанием предметной области. Всякие противоречия должны быть исключены Словарь является отправным моментом для последующих этапов проектирования и разработки системы..

8.2 Проектирование структуры системы

Все задачи, решаемые на данном этапе соотносятся с проработкой и синтезом архитектурных решений и решений, которые в дальнейшем будут использоваться при реализации системы. Из предыдущего этапа извлекается перечень понятий и перекладывается в систему классов и связей между ними. Как правило, существительные из словаря переходят в категорию классов, глагольные формы в операции

классов (над классами). Все упоминания одного класса в контексте другого класса выражаются в виде связей между классами типа ассоциация.

8.2.1 Проектирование диаграммы прецедентов

На данном этапе происходит локализаций основных функций системы. Результат оформляется в виде диаграммы прецедентов и спецификаций на каждый прецедент. Согласно методологии UML каждый прецедент будет тождественен бизнес-функции системы, выявленной на этапе анализа или же она описана в техническом задании. Диаграмма прецедентов дополняется спецификацией прецедентов и таблицей распределения функций системы по прецедентам, из которой четко и ясно следует, что все функции будут реализованы и учтены. Шаблон спецификации прецедента представлен в таблице 1

Таблица 1. Шаблон спецификации прецедента программной системы

Название раздела спецификации прецедента	Назначение
Название	Название прецедента, которое совпадает с именованием прецедента из диаграммы
Актер	Актер, запускающий прецедент или получающий результаты выполнения прецедента
Предусловия	Условия, которые должны быть выполнены, чтобы прецедент мог выполняться
Постусловия	Особые условия, которые должны быть удовлетворены после выполнения основного потока событий
Основной поток событий	Описание набора шагов, выполнение которых полностью реализует функцию (прецедент)
Альтернативные потоки событий	Набор альтернативных потоков событий, в которых происходит обработка особых ситуаций, не предусмотренных основным потоком событий

Формат описания разделов «Основной поток событий» и «Альтернативные потоки событий» представляется в виде последовательности пар: «действие пользователя – реакция системы» или «действие системы – реакция пользователя», в зависимости от того, кто инициирует взаимодействие в рамках описываемого прецедента.

8.2.2 Проектирование диаграммы классов.

Проектирование диаграммы классов начинается с анализа содержимого словаря предметной области, ТЗ и разработанной ранее диаграммы прецедентов. Результатом анализа является синтезированная диаграмма классов, которая ложится в основу разрабатываемой системы на этапе реализации. Для облегчения работ на данном этапе иногда представляют двухуровневое представление диаграммы классов. Верхний уровень соответствует первоначальному пониманию структуры будущего приложения, выраженному непосредственно в бизнес-понятиях предметной области (которые в свою очередь были отражены в словаре).

Уровень реализации соответствует представлению системы близком к уровню реализации системы в исходном коде. Тем не менее возможны определенные расхождения, т.к. при реализации могут быть добавлены вспомогательные классы (методы, атрибуты), не учтенные в проектировании. Однако, скорее всего это будут системные атрибуты, методы иногда даже вспомогательные классы и связи между ними. Все проектирование и документирование данного этапа представляется в соответствии с нотацией UML.

Диаграмма классов с комментариями к ней является результатом данного этапа работы.

8.2.3 Проектирование диаграмм видов деятельности ???

8.2.4 Проектирование диаграмм последовательностей и сотрудничества

Данный тип диаграмм частично соотносится с этапом реализации. Однако может формироваться и на данном этапе работы над проектом и уточняться позднее. Нотация описания диаграмм стандартная и основана на UML. Основное назначение данных диаграмм показать работу системы во времени (каждая из диаграмм устанавливает свой акцент на представлении работы системы (подробнее см., например [5]). Выносимые модели не должны противоречить ранее созданным диаграммам прецедентов и классов. Степень детализации диаграмм должна соответствовать последовательности вызовов методов (передачи управления) классов, которые взаимодействуют в работе. Взаимодействующие классы должны быть соединены соответствующими типами связей на диаграмме классов. По отдельно взятой диаграмме последовательностей (сотрудничества) должна четко прослеживаться логика реализации одного прецедента. Таким образом минимальное число диаграмм последовательностей (сотрудничеств) должно соответствовать числу прецедентов системы.

8.2.5 Проектирование диаграмм состояний

Проектирование данного типа диаграмм должно начинаться с понимания того, что такое состояние и как оно ассоциируется с целевым объектом, для которого она создается. Важность этого следует из того, что не для всех объектов можно разработать данную диаграмму. Диаграмма состояний не проектируется для объектов, состояния которых не различимо в системе или которые могут находиться только в дихотомическом состоянии, т.е. в состоянии работает (создан) / не работает (уничтожен, не используется). Явным признаком наличия состояний у объекта является наличие некоторого знакового атрибута (или совокупности атрибутов), который может находиться в некотором значении из определенного диапазона. Каждое значение характеризует объект и отличает конкретным значением состояние от состояния на протяжении всего времени его жизни. Например, атрибут "Статус" объекта "Покупка" будет характеризовать текущее состояние заказа на всем протяжении жизни объекта и может принимать следующие значения: "Оформлена", "Оплачена", "Формируется", "Доставлена" и т.д. Полный перечень состояний объекта определяет все его возможные поведения и закладывается на данном этапе. В тоже время ошибочным следует понимать ситуации когда в качестве состояния выделяют промежуточ-

ные состояния. По этой причине частой ошибкой является выделение состояния для объекта, которая по сути является переходом от состояния к состоянию. Такой переход может быть сколь угодно длительным может даже оставить объект в прежнем состоянии. Для вышеприведенного примера неправильным пониманием состояния будет состояние "Редактируется", т.к. оно зачастую сводится к пониманию того, что существует некоторая экранная форма, в которой пользователь редактирует состав покупки (добавляет туда товары). Скорее всего, если не оговорено отдельно, такое состояние будет соответствовать переходу объекта из начального состояния, когда пользователь начал формирование покупки путем добавления к ней товаров и нажатия в интерфейсе кнопки "Оформить", в результате такого события (а оно в действительности связывается с переходом объекта в новое состояние и даже отмечается на диаграмме рядом с переходом) объект действительно переходит в состояние "Оформлена".

8.3 Реализация системы

Согласно классическому представлению о ЖЦ разработки ПС реализация идет после этапа проектирования. Однако практика показывает, что календарные сроки данного этапа могут быть сдвинуты ближе к началу этапа проектирования, а иногда и до его полного завершения. Некоторые практики разработки ПС даже приветствуют это (XP, Agile и т.п.). Реализация опирается на выбранные до этого стандарты программирования, технологии и платформы. Если система дорабатывается новым функционалом, то как правило при отсутствии какого либо рефакторинга принимаются стандарты, с применением которых разработка велась ранее.

Ключевым моментом данного этапа является командная разработка функционала системы с дроблением функций системы для реализации ее отдельным участником (программистом). Координацию работ осуществляет старший программист или какой либо руководитель проекта по части разработки. Эти особенности свойственны сложным проектам, с большим функционалом.

Для данного этапа проектируются (разрабатываются) следующие виды диаграмм: модулей, развертывания.

9. Пример разработки программной системы

Согласно классическому ЖЦ разработки ПО выделяются следующие основные этапы:

Анализ / Проектирование / Кодирование / Тестирование / Внедрение и сопровождение. Далее показан пример (как результат) последовательного решения основных вопросов на каждом их этапов.

2 АНАЛИЗ

2.1 Постановка задач

Цель данного курсового проекта – разработка программной информационной системы «Система учета заказов полиграфического производства» с использованием современных информационных технологий.

Система должна быть реализована на платформе .NET.

Разрабатываемая программная система должна выполнять следующие функции:

- Регистрация и авторизация заказчика на сайте предприятия;
- Регистрация менеджера администратором;
- Оформление заказа на печать полиграфической продукции;
- Подтверждение/опровержение корректности заказа;
- Контроль последовательности выполнения стадий заказа;
- Оплата заказа при помощи платежа с кредитных карт;
- Возможность отмены заказа до момента начала передачи его на исполнение;

2.2 Описание предметной области

Полиграфия — это отрасль промышленности, занимающаяся размножением печатной продукции, а именно книжно-журнальной, деловой, газетной, этикеточной и упаковочной продукции.

Заказ - документ, направляемый покупателем (заказчиком) продавцу, в котором выражено твердое намерение купить определенный товар услуги и изложены все необходимые условия для совершения покупки: количество, качество товара, цена, срок поставки и др.

Статус заказа – определенное состояние объекта(заказа) в системе, которое входит в совокупность стабильных значений.

Процесс обработки заказа имеет большое значение в производстве полиграфической продукции. Этот факт объясняется минимум двумя причинами:

1. От времени, затрачиваемого на обработку заказа, зависит то, насколько быстро предприятие приступит к печати.

2. От способа оформления заказа зависит степень удобства и, как следствие заинтересованности заказчика.

Обеспечить удобство для заказчика, а также сократить время на обработку заказа можно с помощью дистанционной работы, для обеспечения которой в рамках курсового проекта разрабатывается прототип программной системы.

2.3 Сценарий взаимодействия пользователей с системой

Взаимодействие пользователя с системой осуществляется через web-браузер.

Пользователи подразделяются по ролям на: администратора, менеджера, пользователя. Обязательным условием авторизации является наличие учетной записи в базе информационной системы. Каждая роль отличается собственным набором функций.

Пользователь:

- Авторизоваться;
- Зарегистрироваться;
- Разместить заказ;
- Отменить заказ;
- Оплатить заказ;

Менеджер:

- Авторизоваться;
- Проверить заказ;
- Назначать статус «выполняется», «выполнен», «отгружен».
- Назначать сроки оплаты и выполнения заказа;

Администратор:

- добавление нового пользователя, менеджера, администратора;

3 ПРОЕКТИРОВАНИЕ

3.1 Диаграмма прецедентов

На рисунке 1 представлена диаграмма прецедентов для прототипа программной системы «Система учета заказов полиграфического производства»

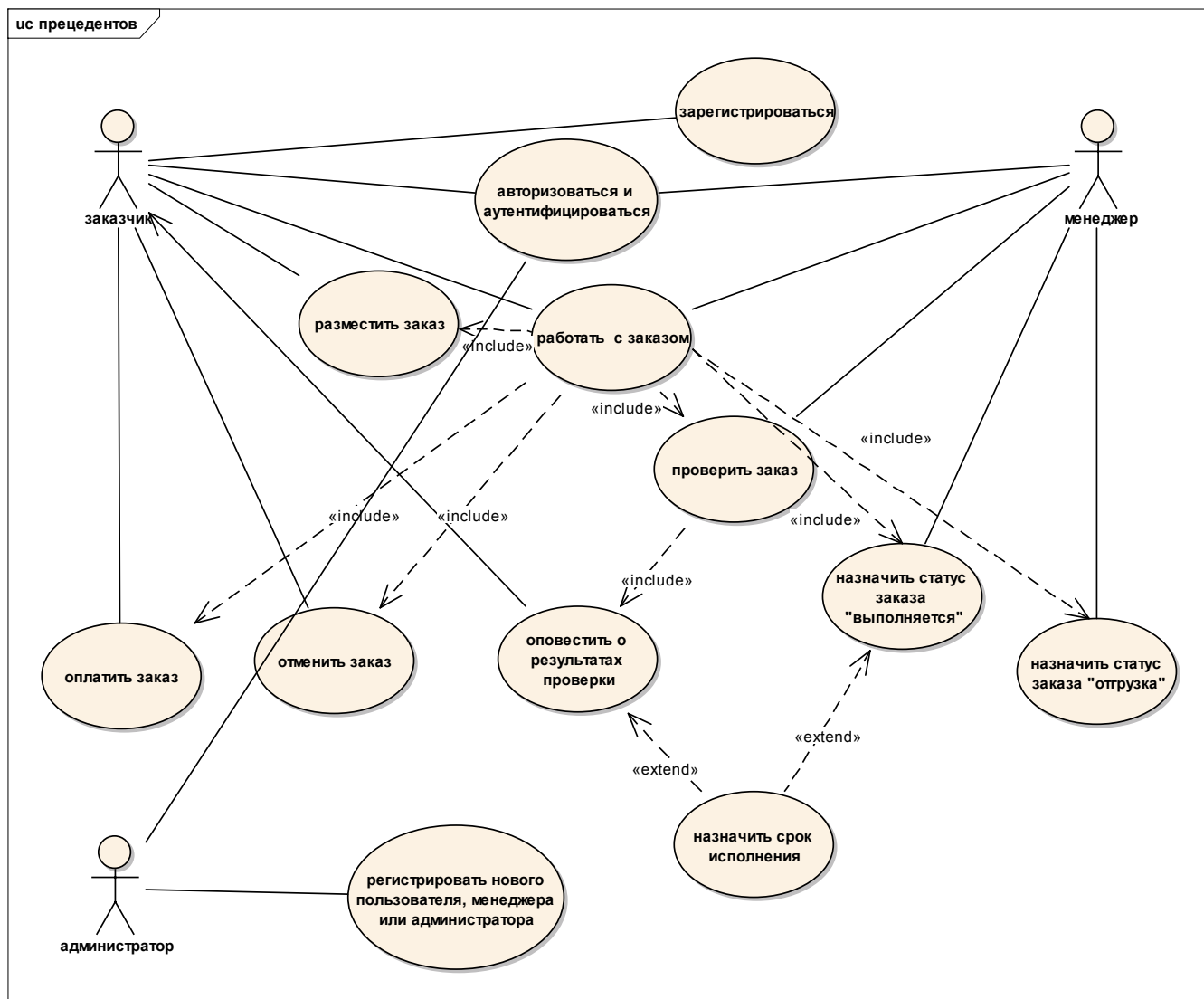


Рисунок 1 - Диаграмма прецедентов.

3.2 Диаграммы последовательностей

На рисунке 2 представлена диаграмма последовательностей для прецедента «Разместить заказ».

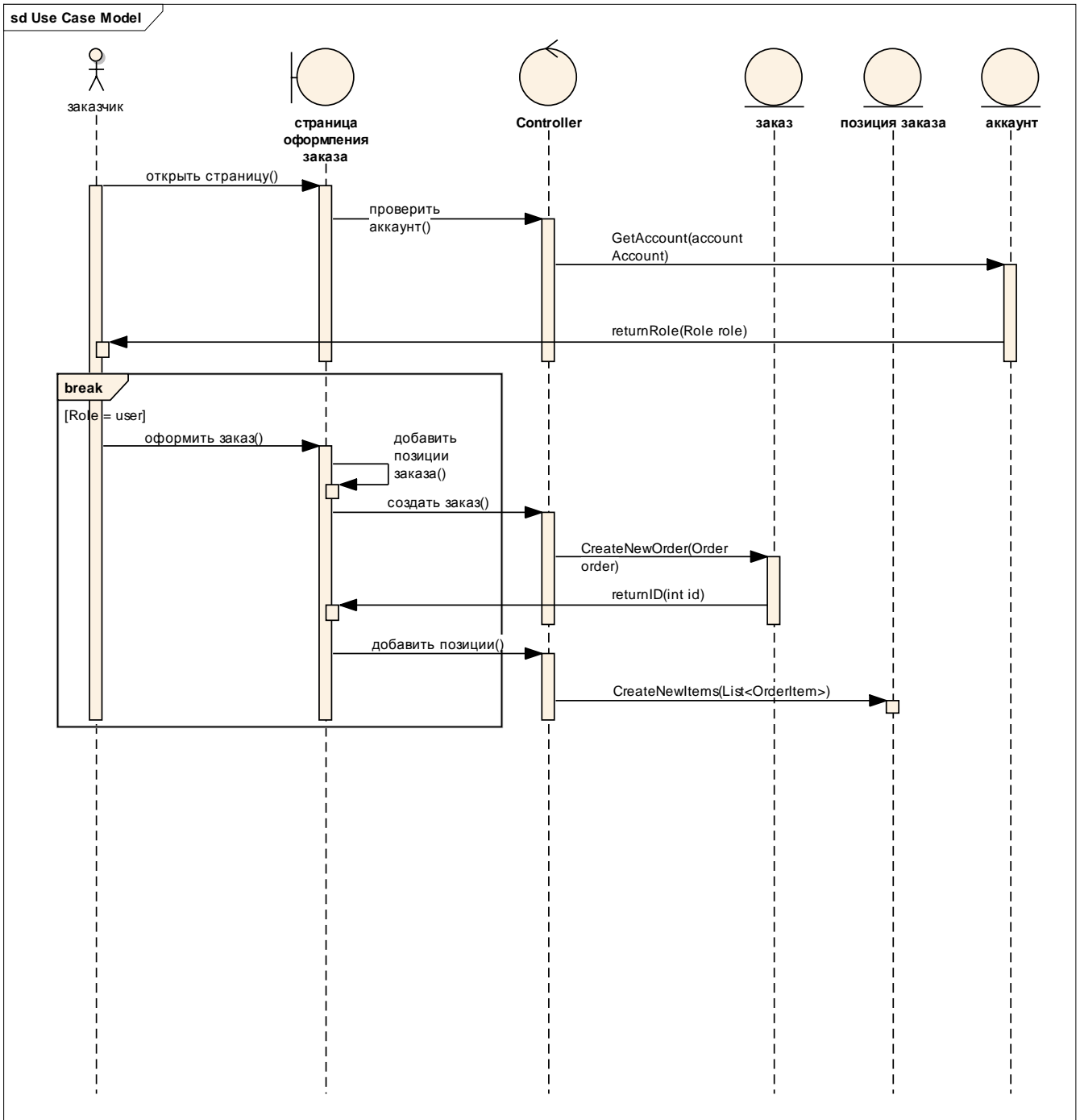


Рисунок 2 - Диаграмма последовательностей для прецедента «Разместить заказ».

3.3 Диаграмма видов деятельности

Диаграмма видов деятельности(рисунк 3) построена для прецедента «Оплатить заказ», участниками которой являются: менеджер, заказчик, система.

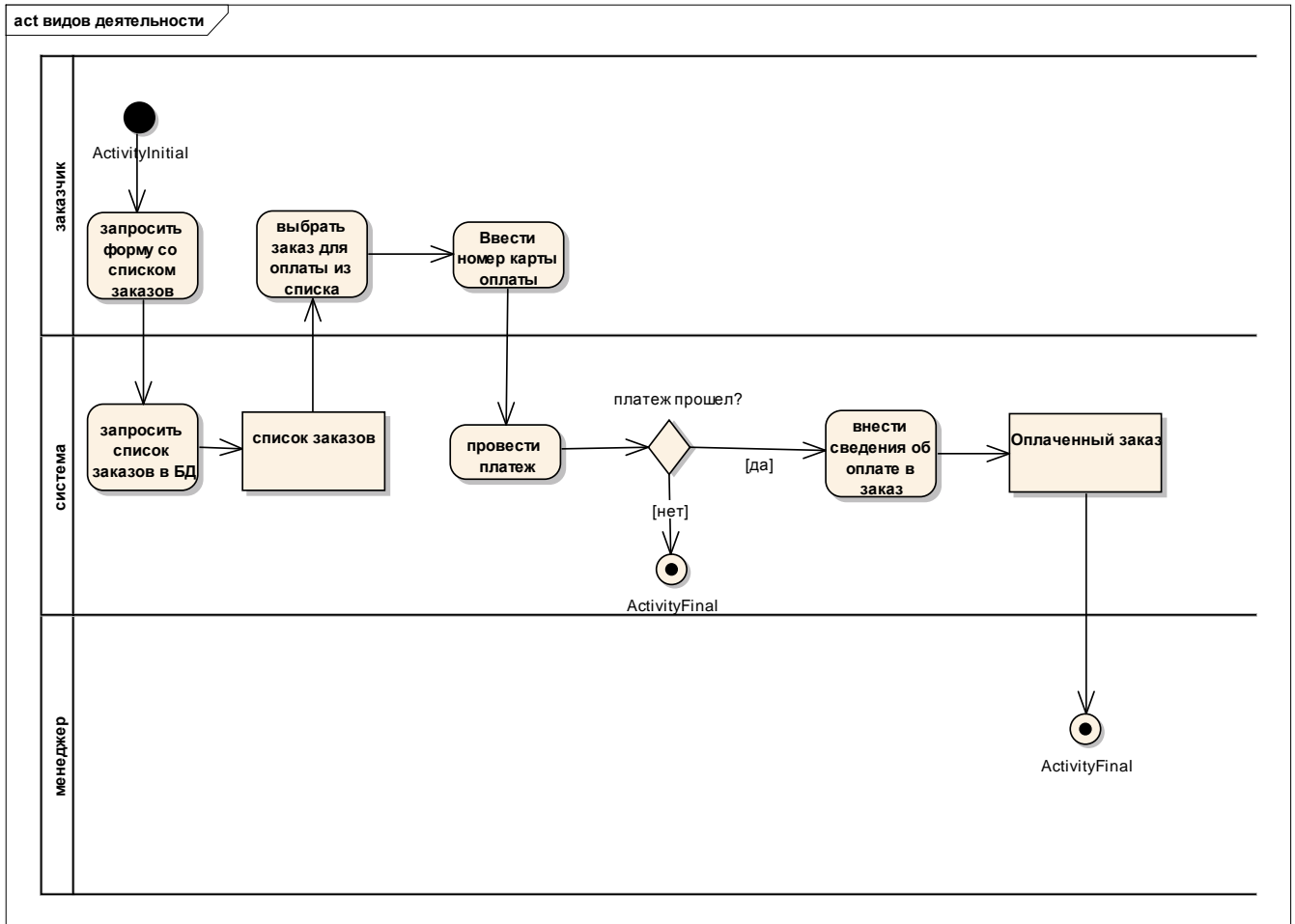


Рисунок 3 - Диаграмма видов деятельности для прецедента «Оплатить заказ».

3.4 Диаграммы классов.

Разработаны две диаграммы классов: концептуального уровня и уровня реализации. Они представлены на рисунках 4 и 5 соответственно.

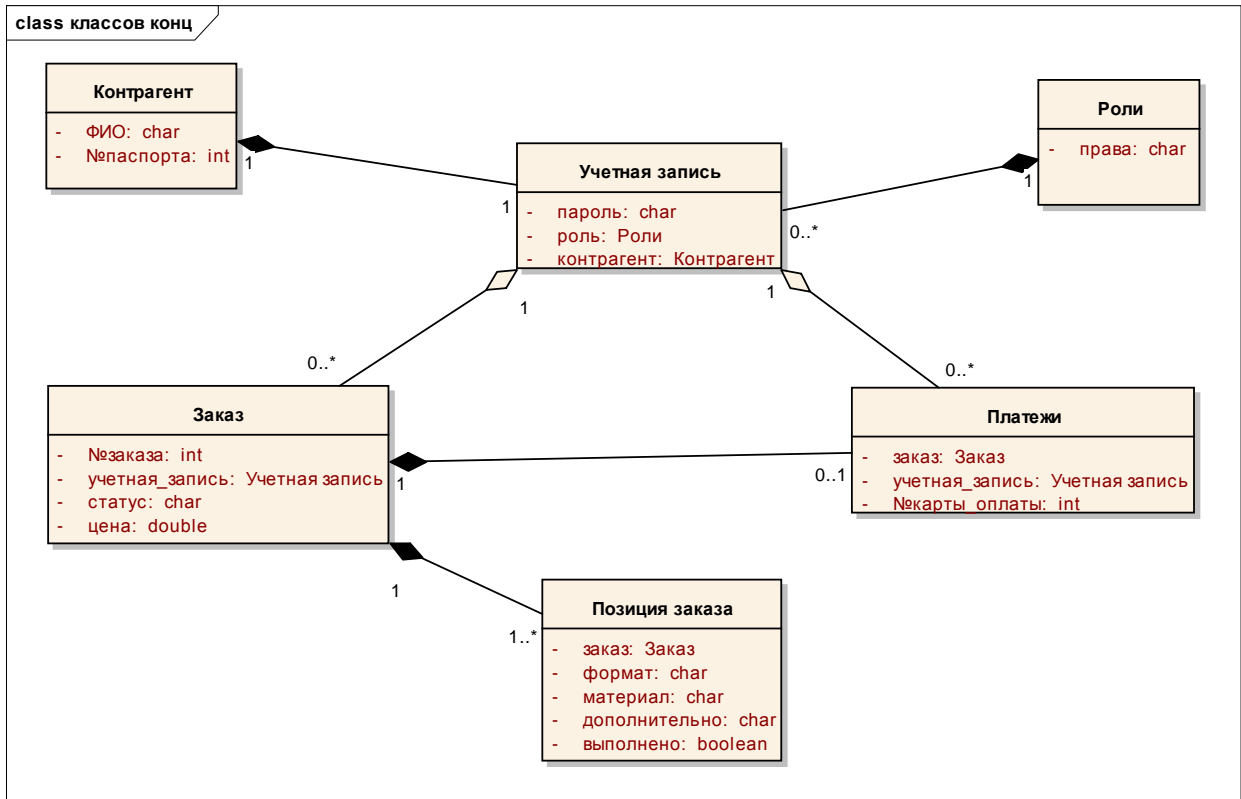
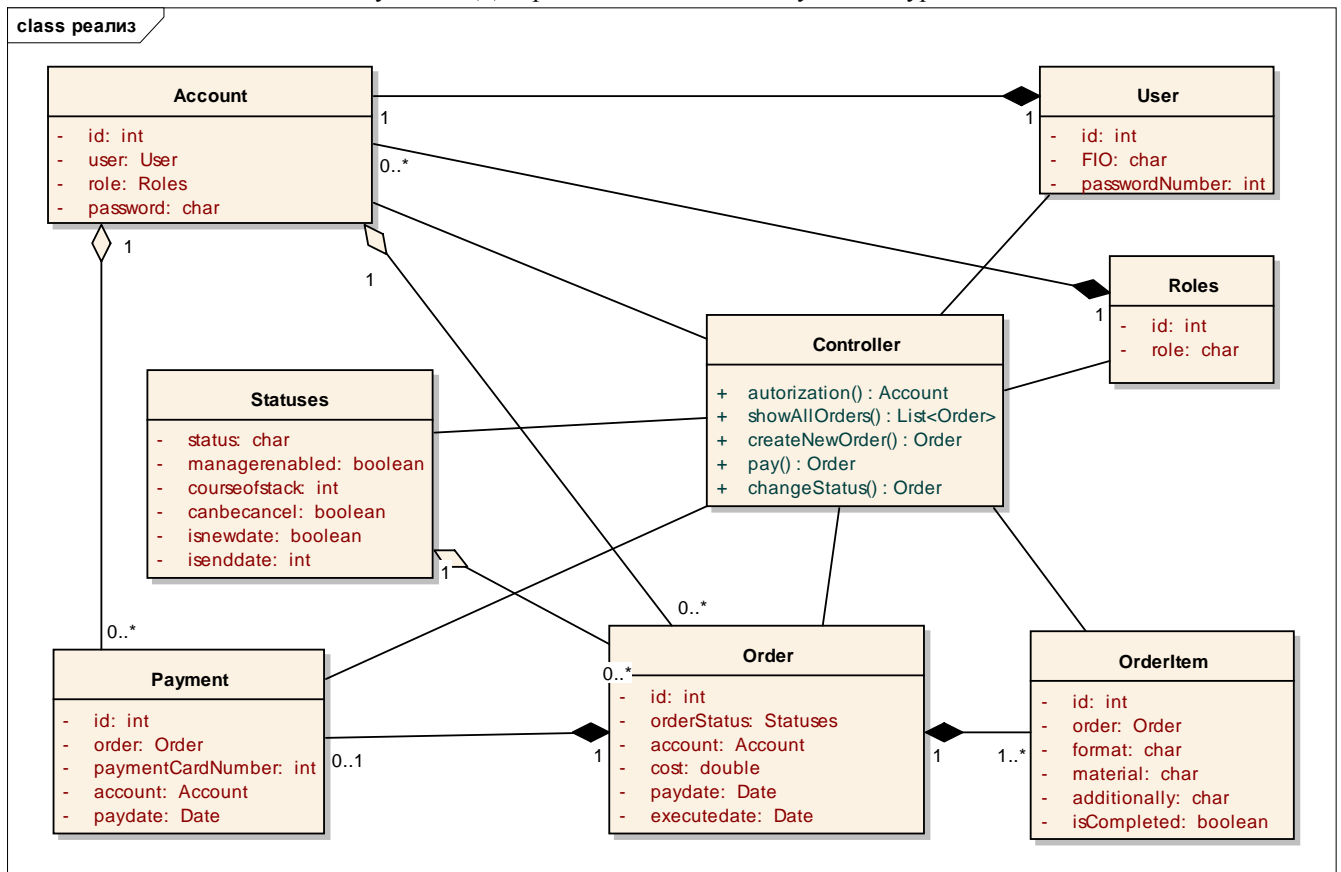


Рисунок 4 - Диаграмма классов концептуального уровня.



В качестве контроллера выступает code behind каждой созданной страницы приложения «polygraph».

3.5 Диаграмма состояний

Для объекта «заказ» была составлена диаграмма состояний, представленная на рисунке 6.

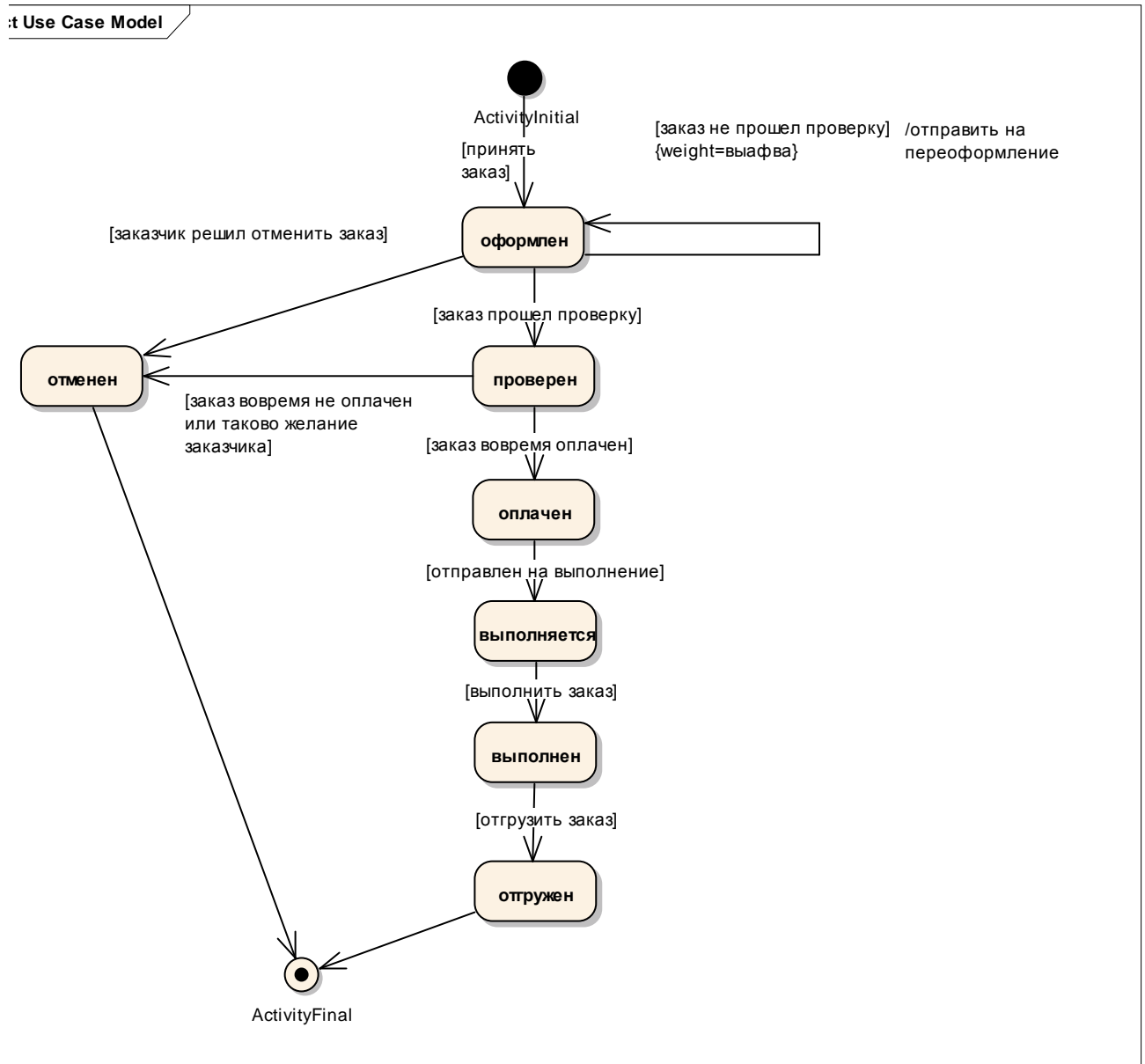


Рисунок 6 - Диаграмма состояний объекта «заказ»

4 РАЗРАБОТКА

4.1 Общие принципы организации системы

Разрабатываемая система представляет из себя веб-приложение, построенное на базе архитектуры ASP.NET.

Среда создания и обработки страниц и компонент ASP .NET — среда программирования, исполняемая на web-сервере для динамической генерации и отображения ASP .NET web-страниц. Страница ASP .NET может быть запрошена любым браузером или устройством, в ответ ASP .NET генерирует разметку (HTML). Как правило, можно отображать одну и ту же страницу в различных браузерах, так как ASP .NET генерирует разметку с учетом типа браузера, запрашивающего страницу. Разумеется, можно создать web-страницу ASP .NET Web, ориентированную на конкретный браузер. ASP .NET поддерживает компоненты для создания web-приложений для мобильных устройств (сотовых телефонов, КПК и других).

Страницы ASP .NET являются объектно-ориентированными. При разработке страницы можно работать с HTML-элементами, используя свойства, методы и события. ASP .NET предлагает следующую модель разделения функциональности на серверную и клиентскую части: события, происходящие на клиентской стороне, обрабатываются исполняемым на сервере кодом. Кроме того, ASP .NET автоматически управляет состоянием страницы и компонент.

Среда создания страниц и компонент позволяет создавать повторно используемые пользовательские элементы управления, инкапсулируя в них необходимый UI и функциональность.

Кроме того, среда создания страниц и компонент ASP .NET предоставляет темы и скины для придания единообразного вида всему приложению, отдельным страницам или для настройки внешнего вида отдельных элементов управления. Скин — это описание внешнего вида элементов управления, относящихся к определенному классу (например, использовать зеленый шрифт для всех кнопок) или обладающих другим общим признаком. Тема — это набор скинов, применяемых как единое целое. Темы и скины описываются XML-файлами. Темы и скины можно применять как программно, так и с помощью визуального редактора свойств. Существует возможность создавать собственные темы и скины.

4.2 Организация доступа к данным

Была составлена модель данных предметной области, в которой выделены следующие сущности, хранимые в базе данных приложения:

- Данные об аккаунтах(Accounts)
- Данные о пользователях (Users)
- Данные о заказах (Orders)

- Данные о позициях заказов (OrderItem)
- Данные о ролях (Roles)
- Данные о платежах (Payment)
- Данные о статусах заказа(Statuses)

Разработанная система включает в себя БД «Database», которая хранится на сервере MS SQL Server 2008. Схема представлена на рисунке 7.

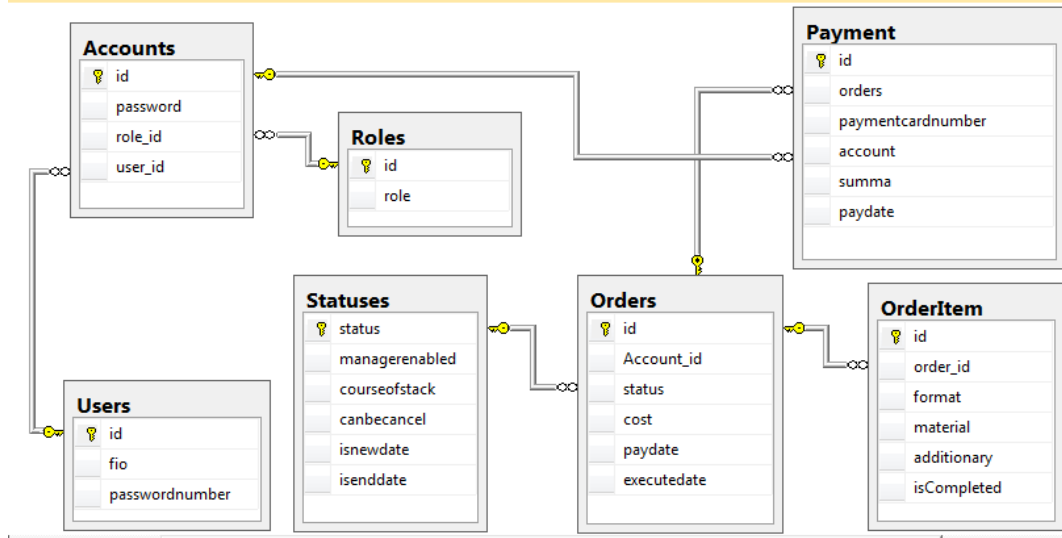


Рисунок 7 - Модель данных.

4.3 Организация бизнес-логики

Для организации бизнес-логики используются файлы `code behind` управления страницами `*.aspx.cs`, в экземплярах которых реализуются методы обработки бизнес логики для конкретной страницы.

Так как проверка предоставления прав пользователя, менеджера, администратора реализовано не с помощью стандартных средств ASP.NET, алгоритм предоставления прав посетителю вписан в обработчик `Page_Load` каждой страницы.

4.4 Организация веб-интерфейса

В приложении используется технология ASP.NET для декларативного модульного описания шаблона страниц.

ASP.NET позволяет избавиться от недостатков, присущих скриптовым языкам. В ASP.NET HTML-код страницы и программный код, выполняемый на сервере (серверный код), выделены в разные файлы `*.aspx` и `*.aspx.cs` соответственно. Содержимое ASP.NET-страниц помимо тэгов HTML включает в себя так называемые серверные элементы управления – специальные тэги, которые имеют объектное представление в серверном коде.

5 РЕАЛИЗАЦИЯ

5.1 Соединение с БД

Взаимодействие приложения с БД, описанной в разделе 4.2, осуществляется с помощью соединения, прописанного в файле web.config.

```
<connectionStrings>
  <add name="ConnectionStringDB" connectionString="Data
  Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\Database.mdf;Integrated
  Security=True;User Instance=True"
  providerName="System.Data.SqlClient" />
</connectionStrings>
```

5.2 Создание страниц

Создаем веб-страницы, каждая из которых сопровождается файлом с фоновым кодом:

Autorization.aspx	Autorization.aspx.cs
CommitNewOrder.aspx	CommitNewOrder.aspx.cs
ManagerOrdersPage.aspx	ManagerOrdersPage.aspx.cs
NewOrder.aspx	NewOrder.aspx.cs
OrdersPage.aspx	OrdersPage.aspx.cs
Registration.aspx	Registration.aspx.cs
SuccessOrder.aspx	SuccessOrder.aspx.cs

5.3 Маршрутизация

Приложения на базе ASP.NET используют непосредственную навигацию, описываемую в файлах поведения веб-страниц. Предоставляются различные возможности для настройки маршрутизации, возможно использование регулярных выражений.

5.4 Ограничения целостности

Для обеспечения целостности данных, а также возможности использования сложных SELECT-запросов используется Database diagram «Diagram2» базы данных «Database.mdf», на которой отображаются все связи между таблицами.

Ограничение PRIMARY KEY для таблицы или столбца означает, что группа из одного или нескольких столбцов образуют первичный ключ таблицы, который является единственным.

Ограничение FOREIGN KEY... REFERENCES... для таблицы и ограничение REFERENCES... для столбца определяют внешний ключ таблицы. Ограничение REFERENCES... для столбца определяет простой внешний ключ, т.е. ключ, состоящий из одной колонки. Ограничение FOREIGN KEY...

REFERENCES... для таблицы может определять как простой, так и сложный внешний ключ, т.е. ключ, состоящий из нескольких колонок таблицы. Столбец или группа столбцов таблицы, на которую ссылается внешний ключ, должна иметь ограничения PRIMARY KEY или UNIQUE. Столбцы, на которые ссылается внешний ключ, должны иметь тот же тип данных, что и столбцы, входящие в состав внешнего ключа. Таблица может иметь ссылку на себя. Ограничение внешнего ключа нарушается, если значения, присутствующие во внешнем ключе, не совпадают со значениями соответствующего ключа родительской таблицы ни для одной строки из родительской таблицы. Операции, приводящие к нарушению ограничения внешнего ключа, отвергаются.

Ограничение NOT NULL (3) столбца не допускает появления в столбце NULL-значений, что обязательно для значений первичного и внешнего ключа.

5.5 Классы – сущности

Для хранения в сессии текущего аккаунта и корзины были созданы классы:

- Accounts.cs
- OrderItems.cs

Набор атрибутов идентичен одноименным таблицам в базе данных.

5.6 Работа с сессиями

Методы Session_Start() Session_End() содержат инструкции, которые должны выполняться при начале и завершении сессии соответственно. В данной курсовой работе модель хранения состояния – InProc. Состояние сеанса хранится в адресном пространстве процесса ASP.NET. Информация о настройках сессии указана в файле web.config:

```
<system.web>
  <sessionState
    mode="InProc"
    sqlConnectionString="Data Source=localhost;Integrated Security=SSPI"
    cookieless="true"
    timeout="30" />
</system.web>
```

6 ИНСТРУКЦИЯ ПО УСТАНОВКЕ ПРИЛОЖЕНИЯ

Для использования приложения следует выполнить следующие действия:

1. Открыть Visual Studio или Visual Web Developer.
2. Выполнить File => Open WebSite.
3. Указать директорию расположения папки «polygraph»

Файлы БД уже находятся в указанной папке. Web.config уже содержит строку настройки подключения к базе данных.

