

Оглавление

Этап анализа и планирования требований (Начало):	4
1.Постановка задачи (задание по варианту 4).	4
2.Спецификация основных проектных требований, ключевых характеристик и главных ограничений (перечень требований-кандидатов, функциональных и нефункциональных требований).	4
3.Модель предметной области (диаграмма классов предметной области, глоссарий понятий, бизнес-модели).	5
4.Описание возможной архитектуры.....	7
5.Перечень критических рисков (и рекомендации по управлению).	9
6.Перечень экранных форм и их сложность; экспертные оценки скорости разработки и масштабных факторов; затраты, длительность и стоимость разработки (по модели СОСОМО-2 этапа композиции приложения).	10
Этап проектирования (Развитие).	12
7.Уточненная диаграмма (ы) прецедентов.	12
8.Расширенные описания основных прецедентов.....	13
9.Прототип пользовательского интерфейса	15
10.Модель проектирования.....	19
11.Итерации.....	21
Этап построения (Конструирование):	23
12.Уточненная диаграмма (ы) прецедентов.	23
Спецификация прецедентов	23
13.Модель анализа	24
14.Модель проектирования.....	26
15.Тестирование	27
Этап внедрения (Переход)	27
Список литературы	28

Этап анализа и планирования требований (Начало):

1. Постановка задачи.

Выполнить разработку кабинета клиента АИС «Круглосуточная пицца на дом». Для проектирования использовать этапы и модели RUP процесса.

По результатам проектирования получить работающую программу с паттернами: domain model в качестве бизнес-логики, active record для работы с БД.

2. Спецификация основных проектных требований, ключевых характеристик и главных ограничений (перечень требований-кандидатов, функциональных и нефункциональных требований).

Система должна выполнять следующие функции (функциональные требования):

1. Ведение базы данных, хранящую информацию о пользователях и их заказах;
2. Оформление новых заказов;
3. Просмотр информации о ранее оформленных заказах.

К системе предъявляются следующие нефункциональные требования:

Разработка ведется по RUP-технологии, на языке C++ с использованием среды QT.

Интерфейс программы должен содержать стартовое меню.

Система эксплуатируется на ОС Windows7 (клиентская рабочая станция).

В качестве СУБД используется SQLite. БД размещается на рабочей станции клиента.

Максимальный размер файла базы данных не должен превышать 20 ТБ.

Минимальные системные требования для клиентской рабочей станции:

- ОС Windows7;
- Поддержка SQLite.v3 и выше;
- 4Гб оперативной памяти;
- процессор частотой 2,2ГГц;
- жесткий диск объемом 22ТБ свободного места;
- манипулятор «мышь»;
- клавиатура;
- сетевой адаптер;
- монитор.

3. Модель предметной области (диаграмма классов предметной области, глоссарий понятий, бизнес-модели).

Предметная область описана двумя сущностями.

На рисунке 1 представлена диаграмма классов предметной области.

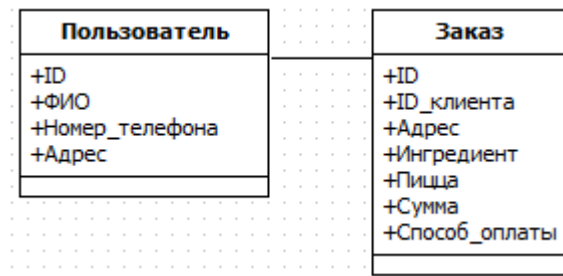


Рис. 1. Диаграмма классов предметной области

Сущность Пользователь необходима для хранения информации о клиентах пиццерии.

ID – уникальный номер клиента;

ФИО – фамилия, имя, отчество пользователя;

Номер_телефона – номер телефона пользователя;

Адрес – закрепленный за клиентом адрес доставки.

Сущность Заказ необходима для хранения общей информации о заказах и клиенте, который его оформил:

ID – уникальный номер заказа;

ID_клиента – уникальный номер клиента;

Адрес – адрес доставки данного заказа;

Ингредиент – дополнительные ингредиенты по желанию клиента;

Пицца – пиццы, которые пользователь может заказать;

Сумма – сумма заказа;

Способ_оплаты – способ, которым клиент оплачивает свой заказ.

Выявленные актеры.

Клиент пиццерии.

Выявленные прецеденты, их приоритеты и описание (кратко).

1. Добавление заказа
2. Учет заказов

Прецедент «Добавление заказа»

Краткое описание: позволяет пользователю делать новые заказы и изменять адрес доставки.

Приоритет: высокий

Прецедент «Учет заказов»

Краткое описание: позволяет пользователю увидеть всю информацию о его заказах.

Приоритет: высокий

Диаграмма основных прецедентов.

На рисунке 2 представлена диаграмма прецедентов.

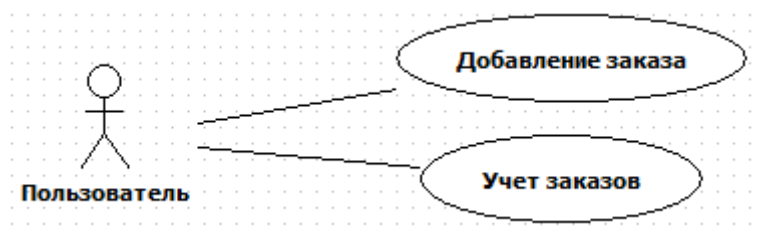


Рис. 2. Диаграмма прецедентов

4. Описание возможной архитектуры

(перечень архитектурно-значимых прецедентов, обобщенные механизмы проектирования, системное ПО, используемые компоненты и/или каркасы). По каждой из выбранных технологий (обобщенные механизмы проектирования, системное ПО и т.д.) необходимо привести варианты, их достоинства и недостатки и обосновать выбор.

При разработке используется IDE QtCreator, СУБД SQLite.

IDE Qt Creator преимущества:

- большая библиотека для работы с различными СУБД;
- комплексное решение разработки приложений и графических интерфейсов;
- бесплатный (если вести некоммерческую разработку);
- кроссплатформенная разработка;
- создание UI возможно при помощи нескольких технологий: декларативного языка QML, C++, HTML5, а также любых их комбинаций (QtQuick, QtWidgwtS);
- быстрый старт разработки: удобные инсталляторы, готовые инструменты, возможность создания прототипов сразу после установки;
- детальная документация API, примеры и обучение, техподдержка.

СУБД SQL Lite достоинства:

- Открытое ПО соответствующее стандарту SQL;
- Большое количество дополнений;
- файловая структура - вся база данных состоит из одного файла, поэтому её очень легко переносить на разные машины;
- бесплатная.

Среда StarUML будет использоваться для построения диаграмм. StarUML предоставляет максимальные возможности адаптации к требованиям пользователя, предлагая настройку параметров, которые определяют методологию создания программного обеспечения, проектную платформу и язык разработки. Удобный и простой интерфейс пользователя. Большое количество инструментов для построения UML-диаграмм.

На рисунке 3 представлена диаграмма развертывания.

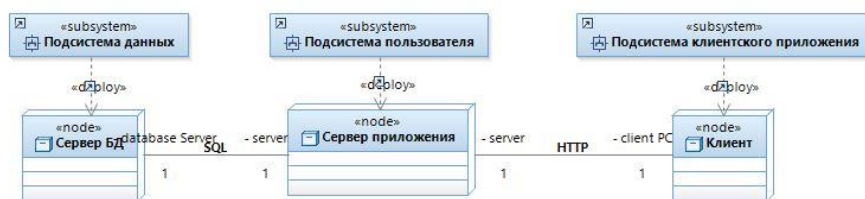


Рис. 3. Диаграмма развертывания

5. Перечень критических рисков (и рекомендации по управлению).

1) Внутренние сложности календарного планирования

При планировании сдачи этапов разработки применяются чрезмерно оптимистичные оценки возможностей разработчиков и аналитиков. Результат — существенное расхождение в планируемых и реальных сроках реализации проекта.

2) Увеличение требований со стороны заказчика в ходе реализации проекта

При реализации проекта у заказчика могут измениться требования к конечному продукту или к этапу разработки. Результат – необходимость перерабатывать весь план разработки и вносить изменения. Это влечет дополнительные трудозатраты.

3) Текучка кадров

Исходя из этого, необходимо увеличивать объем целевых коммуникаций между членами команды, чтобы потеря любого из сотрудников не оказалась критичной. Разработка не должна быть «замкнута» на конкретной личности.

4) Малая «комментируемость» кода

Данный риск вытекает из риска «Текучки кадров». Если исходный код приложения содержит малое количество комментариев, то при уходе сотрудника часть проекта, которую он разрабатывал, становится трудна для понимания и разбора.

5) Низкая производительность

Крупная задача может долго откладываться. Следует разбивать задачу на короткие этапы, что постоянно создает ощущение скорого дедлайна.

6. Перечень экранных форм и их сложность; экспертные оценки скорости разработки и масштабных факторов; затраты, длительность и стоимость разработки (по модели COSOMO-2 этапа композиции приложения).

Перечень экранных форм:

1. Стартовая форма приложения;
2. Форма для оформления нового заказа (изменение адреса доставки);
3. Форма для просмотра истории заказов пользователя;
4. Форма для просмотра информации о пиццерии.

Расчет затрат, длительности и стоимости разработки по модели COSOMO-2 этапа композиции приложения.

Таблица 1. Данные по экранам и отчетам

Тип объекта	Количество представлений	Количество таблиц	Сложность
Стартовая форма	0	0	простой
Форма для добавления заказа	4	5	средний
Форма просмотра заказов	1	2	простой
Форма для просмотра информации	1	0	простой

Таблица 2. Оценка количества объектных указателей

Тип объекта	Сложность	Количество	Вес	Итого
Экран	простой	3	x1	3
Экран	средний	1	x2	2
Объектные указатели				5

В проекте используются только вновь созданные компоненты (разработка является абсолютно новой) %REUSE = 0.

$$NOP = OP * \frac{100 - \%REUSE}{10} = 5$$

Опытность разработчика очень низкая PROD = 2, а зрелость среды разработки низкая PROD = 7.

Общий коэффициент продуктивности продукта:

$$PROD = \frac{2 + 7}{2} = 4,5$$

Затраты рассчитываются по формуле 1.1.

$$\text{ЗАТРАТЫ} = \frac{NOP}{PROD} [\text{чел. – мес.}] \quad (1.1)$$

$$\text{ЗАТРАТЫ} = \frac{5}{4,5} = 1,11 [\text{чел. – мес.}]$$

Стоимость рассчитывается по формуле 1.2.

$$\text{СТОИМОСТЬ} = \text{ЗАТРАТЫ} \times \text{РАБ_КОЭФ} \quad (1.2)$$

$$\text{СТОИМОСТЬ} = 1,11 \times 30\,000 = 33\,300 \text{ рублей}$$

Значение показателя степени вычисляется по формуле 1.3.

$$B = 1,01 + 0,01 \sum_{i=1}^5 W_i \quad (1.3)$$

Оценка масштабных факторов приведена в таблице 3.

Оценки принимают 6 значений: от очень низкой (5) до сверхвысокой (0).

Таблица 3. Оценка масштабных факторов

Масштабный фактор (W_i)	Значение
Предсказуемость PREC	5
Гибкость разработки FLEX	3
Разрешение архитектуры /риска RESL	2
Связность группы TEAM	1
Зрелость процесса PMAT	3
Общее количество	14

$$B = 1,01 + 0,01 \times 14 = 1,15$$

$$\text{Длительность (TDEV)} = 3,0 * \text{ЗАТРАТЫ}^{(0,33+0,2(B-1,01))} * \frac{SCEDPercentage}{100}$$

В соответствии с риском внутренних сложностей календарного планирования, процент увеличения номинального графика положим 15%, следовательно, SCEDPercentage = 115.

$$TDEV = 3,0 * 1,11^{(0,33+0,2(1,15-1,01))} * \frac{115}{100} = 3,6 \text{ месяца}$$

Этап проектирования (Развитие).

7. Уточненная диаграмма (ы) прецедентов.

На рисунке 4 представлена диаграмма прецедентов.

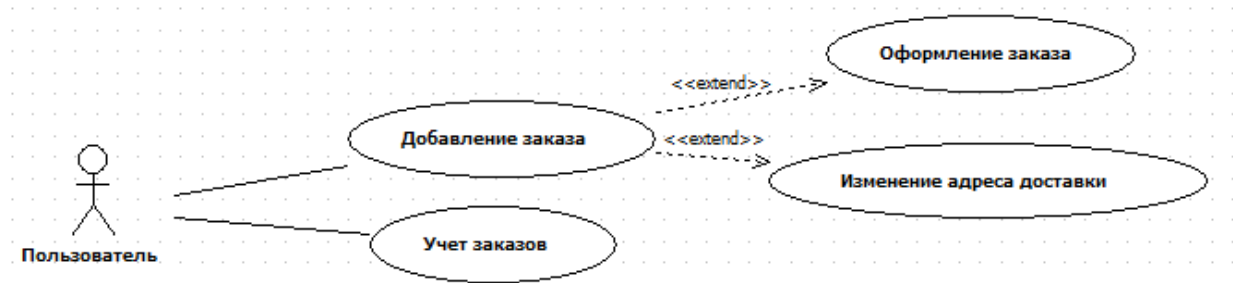


Рис. 4. Уточненная диаграмма прецедентов

8. *Расширенные описания основных прецедентов*

Прецедент «Добавление заказа»

Краткое описание: позволяет пользователю вносить новые заказы и изменять адрес доставки.

Предусловие: перед началом этого прецедента пользователь должен запустить приложение и открыть форму для заполнения данных по заказу.

Главный поток:

Этот прецедент начинается, когда пользователь вносит данные заказу. Пользователю предлагается внести данные по заказу в БД, а также изменить адрес доставки. Запись данных в БД осуществляется для новых заказов пользователя.

Подпотоки:

S-2: добавление заказа. Система проверяет, верно ли заполнено поле «Адрес доставки», и сообщает пользователю о результатах проверки с помощью диалогового окна (E-3).

S-3: сохранение данных. (E-4) Система сохраняет данные в БД и сообщает пользователю о результатах сохранения (E-5). Возврат к началу прецедента.

Альтернативные потоки:

E-3: поле «Адрес доставки» заполнено неверно. Пользователь может вернуться к заполнению данных или прекратить прецедент.

E-4. поле «Пицца» не заполнено. Пользователь может вернуться к заполнению данных или прекратить прецедент.

E-5: система не может сохранить данные. Пользователь может повторить заполнение данных или прекратить прецедент.

Прецедент «Учет заказов»

Краткое описание: позволяет пользователю увидеть всю информацию о его заказах.

Предусловие: перед началом этого прецедента пользователь должен запустить приложение и открыть форму для просмотра истории заказов.

Главный поток:

Этот прецедент начинается, когда пользователь запрашивает историю заказов, нажав на кнопку «Просмотреть». Пользователю предоставляется отчет по всей истории его заказов.

1. Пользователь нажал на кнопку «Просмотреть», выполняется подпоток S-1: сформировать отчет;
2. Если выбрано действие «Назад», прецедент заканчивается.

Подпотоки:

S-1: сформировать отчет. Система проверяет наличие заказов в БД и сообщает пользователю о результатах поиска с помощью диалогового окна (Е-1). Система формирует отчет и выводит его в специальное поле. Возврат к началу прецедента.

Альтернативные потоки:

Е-1:заказов нет в БД. Пользователь может вернуться и прекратить прецедент.

9. Прототип пользовательского интерфейса

Прототип пользовательского интерфейса содержит 5 форм:

Макет стартовой формы представлен на рисунке 5.

Рис. 5. Стартовая экранная форма

Макет формы добавления заказа представлен на рисунке 6.

Рис. 6. Формы добавления заказа

Макет формы просмотра информации о пиццерии представлен на рисунке 7.

Information about our pizzeria

Назад

This is a wireframe of a mobile application screen. It features a header bar at the top. Below the header is a large rectangular area containing the text 'Информация о нашей пиццерии' (Information about our pizzeria). At the bottom of the screen is a button labeled 'Назад' (Back).

Рис. 7. Форма для просмотра информации о пиццерии

Макет формы просмотра истории заказов представлен на рисунке 8.

Просмотреть

Назад

This is a wireframe of a mobile application screen. It features a header bar at the top. Below the header is a button labeled 'Просмотреть' (View). Below this button is a large rectangular area, likely intended for a list of orders. At the bottom of the screen is a button labeled 'Назад' (Back).

Рис. 8. Форма для просмотра данных

Макет формы для изменения адреса доставки в добавлении заказа представлен на рисунке 9.

Введите адрес

Сохранить

Назад

This is a wireframe of a mobile application screen. It features a header bar at the top. Below the header is a text input field with the placeholder text 'Введите адрес' (Enter address). Below the input field are two buttons: 'Сохранить' (Save) and 'Назад' (Back).

Рис. 9. Форма для изменения адреса доставки

Обзорная диаграмма классов анализа представлена на рисунке 10.

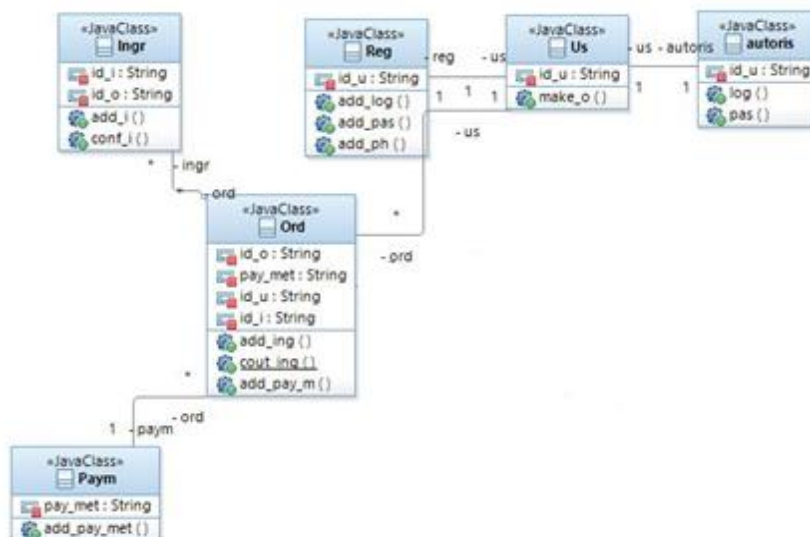


Рис. 10. Диаграмма классов анализа

Обзорная диаграмма пакетов анализа представлена на рисунке 11.

`$(project)` Key Abstractions

Significant Analysis Classes

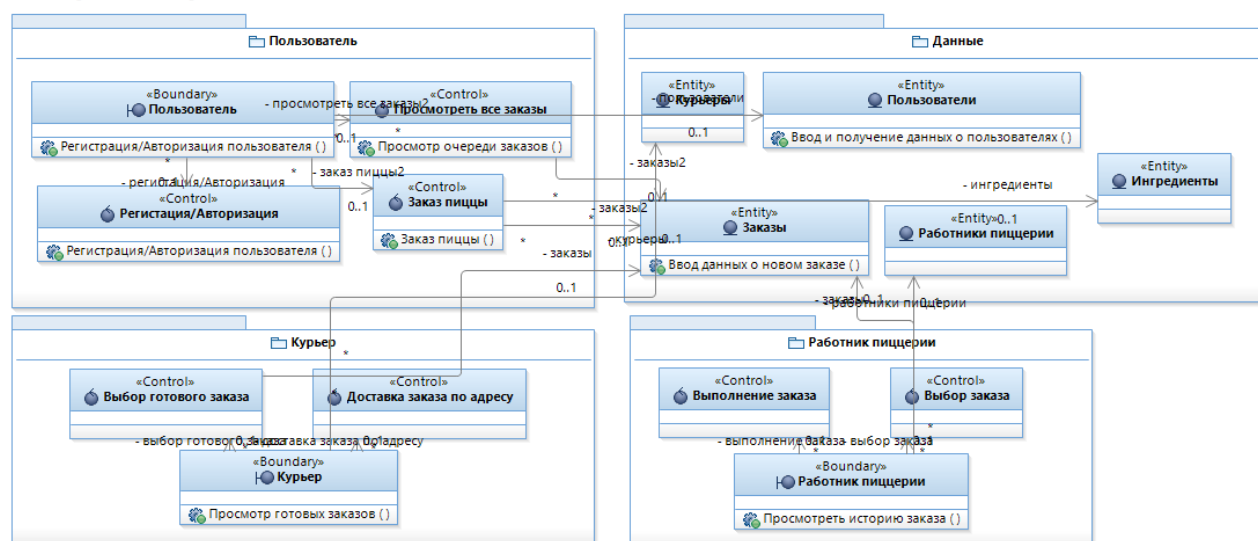


Рис. 11. Диаграмма пакетов анализа

Диаграммы последовательностей прецедентов «Добавление заказа» и «Учет заказов» представлены на рисунках 12 и 13 соответственно.

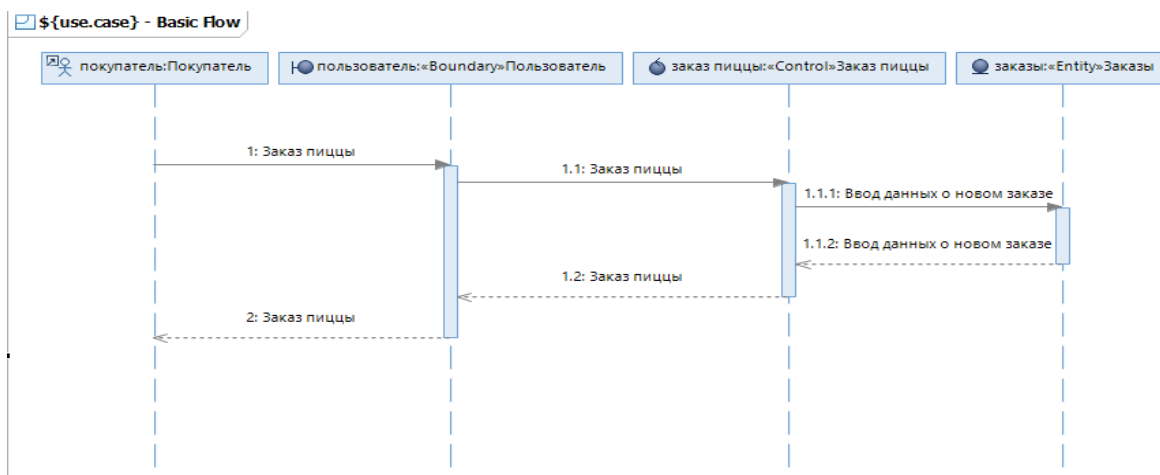


Рис. 12. Диаграмма последовательности «Добавление заказа»

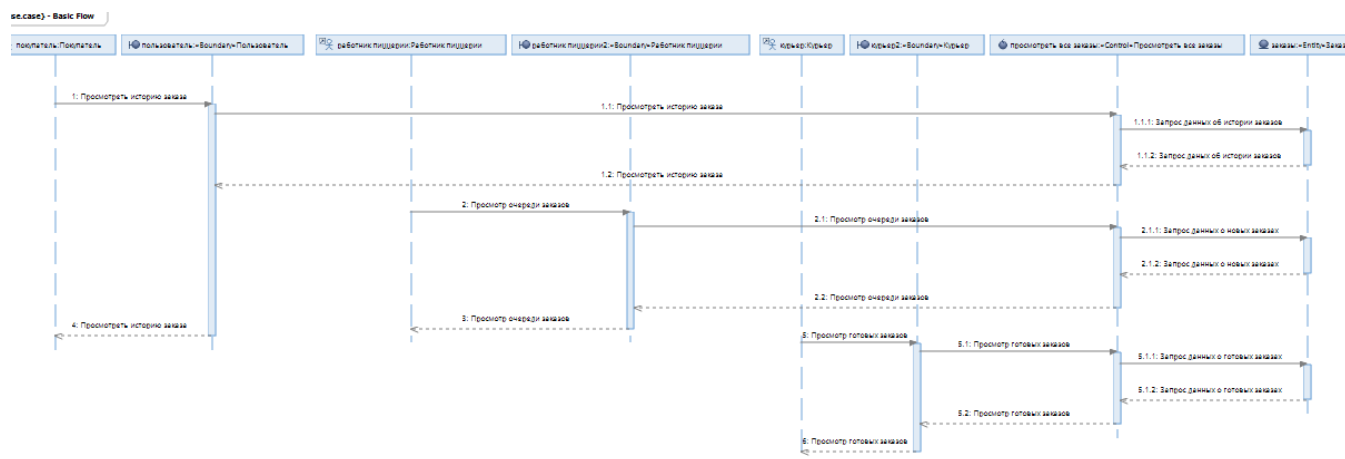


Рис. 13. Диаграмма последовательности «Учет заказов».

10. Модель проектирования

Трассировка классов анализа и проектирования представлена на рисунке 14.

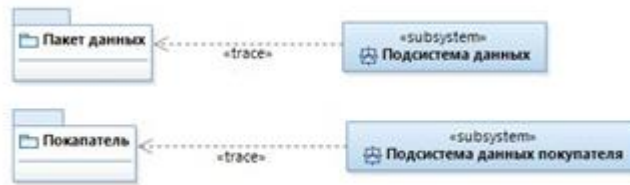


Рис. 14. Трассировка

На рисунке 15 представлено распределение подсистем на уровни.

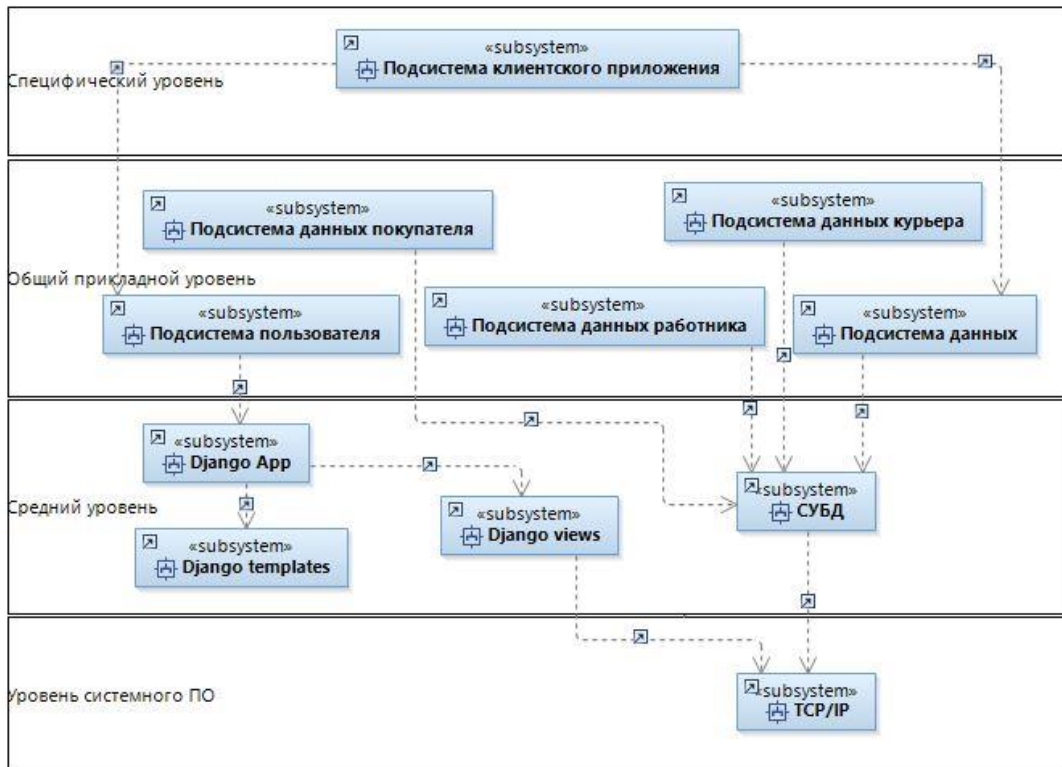


Рис. 15. Распределение подсистем по уровням

Диаграммы компонентов и развертывания для реализации архитектуры

На рисунке 16 представлена трассировка подсистем в компоненты.

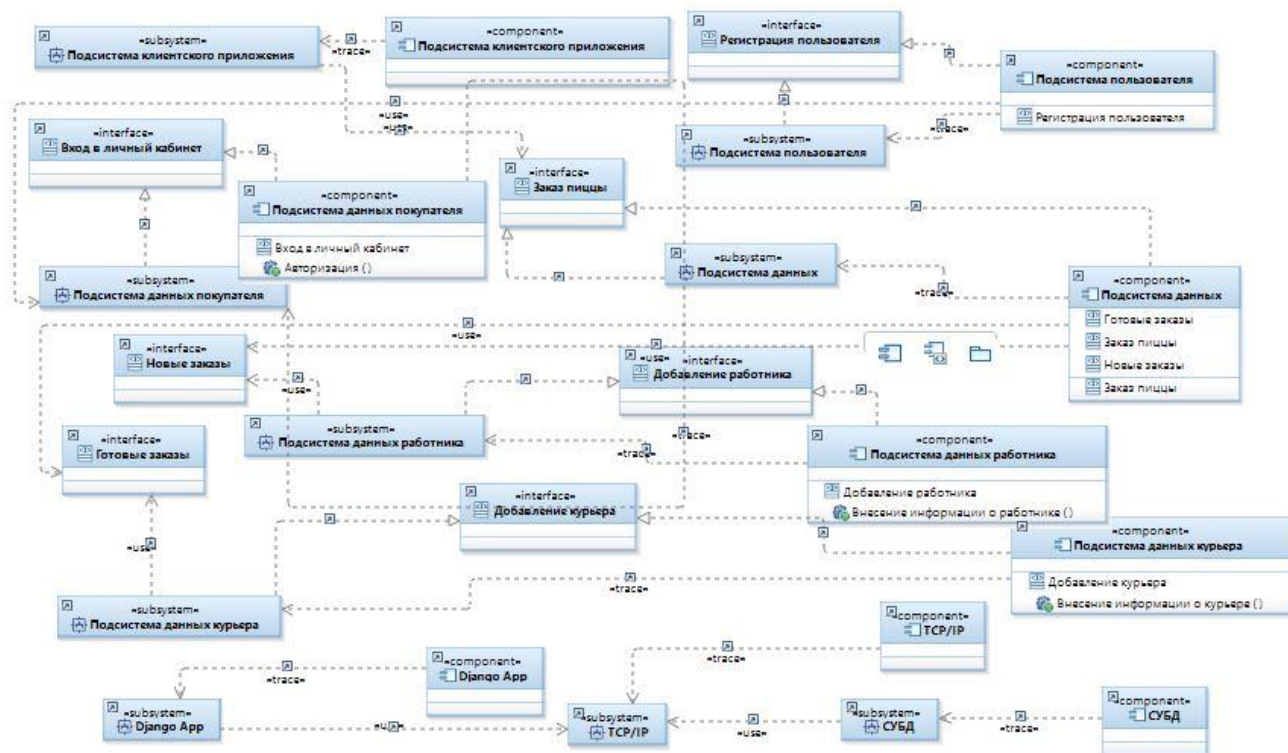


Рис. 16. Трассировка подсистем в компоненты

На рисунке 17 представлена диаграмма развертывания.



Рис. 17. Диаграмма развертывания

11. Итерации

В рамках работы будут реализованы следующие итерации проекта:

1. Реализация основных функций системы по работе с данными: просмотр информации, добавление записей, изменение записей, удаление записей.

2. Реализация остального функционала: поиск записей, формирование отчета.

Оценку качества будем производить по следующим метрикам:

- WMC (Weighted Methods Per Class) — Взвешенные методы на класс
- NOC (Number of Children) — Количество детей
- CBO (Coupling Between Object Classes) — Сцепление между классами объектов
- RFC (Response For a Class) — Отклик для класса
- LCOM (Lack of Cohesion in Methods) — недостаток связности в методах
- CS (Class Size) — Размер класса
- NOO (Number of Operations Overridden by a Subclass) — Количество операций, переопределяемых подклассом
- NOA (Number of Operations Added by a Subclass) — Количество операций, добавленных подклассом
- SI (Specialization Index) — Индекс специализации
- OS (Average Operation Size) — Средний размер операции
- NP (Average Number of Parameters Per Operation) — Среднее количество параметров на операцию
- DIT (Depth of Inheritance Tree) — Высота дерева наследования
- NOM (Number of Methods) — Суммарное количество методов, определенных во всех классах системы
- NC (Number of Classes) — Общее количество классов в системе
- LOC (Sum Lines of Code) — Суммарное количество строк кода

Первая итерация:

Первая итерация включает в себя разработку следующих прецедентов:

- добавление заказа.
- изменение адреса доставки.

Оценка качества для первой итерации представлена в таблицах 4, 5.

Табл.4. Оценка качества

Метрика	Добавление заказа	Изменение адреса доставки
WMC	15	3
DIT	0	0
NOC	0	0
CBO	4	1
RFC	19	4
LCOM	0	0
CS	21	6
NOO	0	0
NOA	0	0
SI	0	0
OS	5	1
NP	1	0.5
LOC	250	100

Табл.5. Оценка качества для всей подсистемы

DIT	0
NOM	53
NC	9
LOC	956

Вторая итерация:

Вторая итерация включает в себя разработку следующих прецедентов:

- учет заказов.

Оценка качества для второй итерации представлена в таблицах 6, 7.

Табл.6. Оценка качества

Метрика	Учет заказов
WMC	3
DIT	0
NOC	0
CBO	1
RFC	4
LCOM	0
CS	6
NOO	0
NOA	0
SI	0
OS	1
NP	0
LOC	100

Табл.7. Оценка качества для всей подсистемы

DIT	1
NOM	56
NC	10
LOC	1156

Этап построения (Конструирование):

12. Уточненная диаграмма (ы) прецедентов.

Уточненная диаграмма прецедентов представлена на рисунке 18.

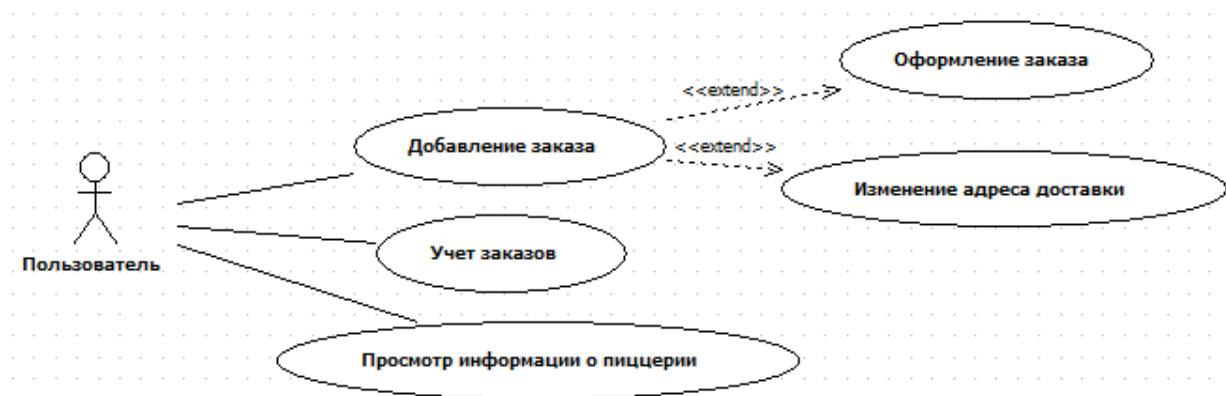


Рис. 18. Диаграмма прецедентов

Спецификация прецедентов

Прецедент «Добавление заказа»

Краткое описание: позволяет пользователю вносить новые заказы и изменять адрес доставки.

Предусловие: перед началом этого прецедента пользователь должен запустить приложение и открыть форму для заполнения данных по заказу.

Главный поток:

Этот прецедент начинается, когда пользователь вносит данные заказу. Пользователю предлагается внести данные по заказу в БД, а также изменить адрес доставки. Запись данных в БД осуществляется для новых заказов пользователя.

1. Пользователь выбрал действие «Выбрать пиццу» и наименование пиццы, выполняется подпоток S-1: выбрать пиццу;
2. Пользователь выбрал действие «Выбрать ингредиент» и наименование дополнительного ингредиента, выполняется подпоток S-2: выбрать ингредиент;
3. Пользователь выбрал действие «Выбрать способ оплаты» и способ оплаты заказа, выполняется подпоток S-3: выбрать способ оплаты;
4. Пользователь выбрал действие «Оформить заказ» выполняется подпоток S-4: сохранение данных;
5. Если выбрано действие «Назад», прецедент заканчивается.

Подпотоки:

S-1: выбрать пиццу. Система проверяет, выбрана ли пиццы и сообщает пользователю о результатах проверки с помощью диалогового окна (Е-1). Возврат к пункту 2 (главный поток).

S-2: выбрать ингредиент. Система проверяет, выбран ли ингредиент и сообщает пользователю о результатах проверки с помощью диалогового окна (Е-2). Возврат к пункту 3 (главный поток).

S-3: выбрать способ оплаты. Система проверяет, выбран ли способ оплаты и сообщает пользователю о результатах проверки с помощью диалогового окна (Е-3). Возврат к пункту 4 (главный поток).

S-4: сохранение данных. Система сохраняет данные в БД и сообщает пользователю о результатах сохранения (Е-5). Возврат к началу прецедента.

Альтернативные потоки:

Е-1: поле «Пицца» не заполнено. Пользователь может повторить пункт 1 (главный поток) или прекратить прецедент.

Е-2: поле «Ингредиент» не заполнено. Пользователь может повторить пункт 2 (главный поток) или прекратить прецедент.

Е-3: поле «Способ оплаты» не заполнено. Пользователь может повторить пункт 3 (главный поток) или прекратить прецедент.

Е-4: система не может сохранить данные. Пользователь может повторить пункт 4 или прекратить прецедент.

Прецедент «Учет заказов»

Краткое описание: позволяет пользователю увидеть всю информацию о его заказах.

Предусловие: перед началом этого прецедента пользователь должен запустить приложение и открыть форму для просмотра истории заказов.

Главный поток:

Этот прецедент начинается, когда пользователь запрашивает историю заказов, нажав на кнопку «Просмотреть». Пользователю предоставляется отчет по всей истории его заказов.

1. Пользователь нажал на кнопку «Просмотреть», выполняется подпоток S-1: сформировать отчет;
2. Если выбрано действие «Назад», прецедент заканчивается.

Подпотоки:

S-1: сформировать отчет. Система проверяет наличие заказов в БД и сообщает пользователю о результатах поиска с помощью диалогового окна (Е-1). Система формирует отчет и выводит его в специальное поле. Возврат к началу прецедента.

Альтернативные потоки:

Е-1:заказов нет в БД. Пользователь может вернуться и прекратить прецедент.

13. Модель анализа

Обзорная диаграмма классов анализа, представлена на рисунке 19.

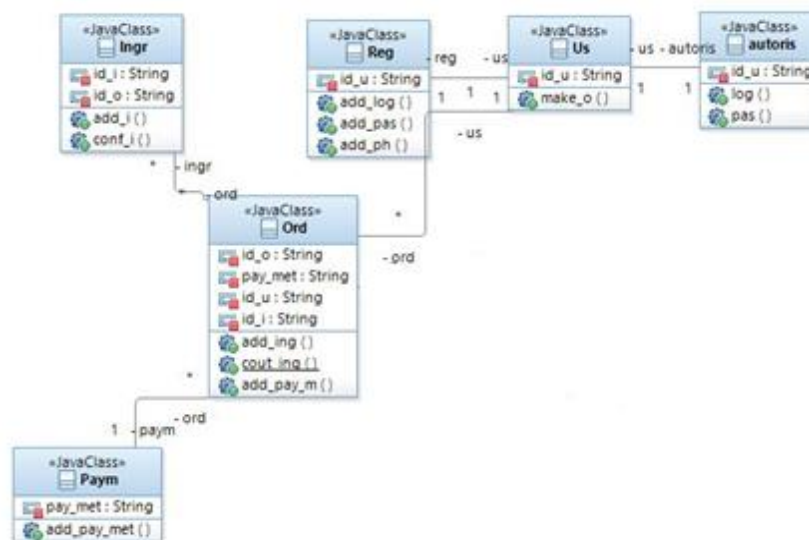


Рис. 19 Диаграмма классов

14. Модель проектирования

Диаграмма классов, распределенных по пакетам, представлена на рисунке 20.

Significant Analysis Classes

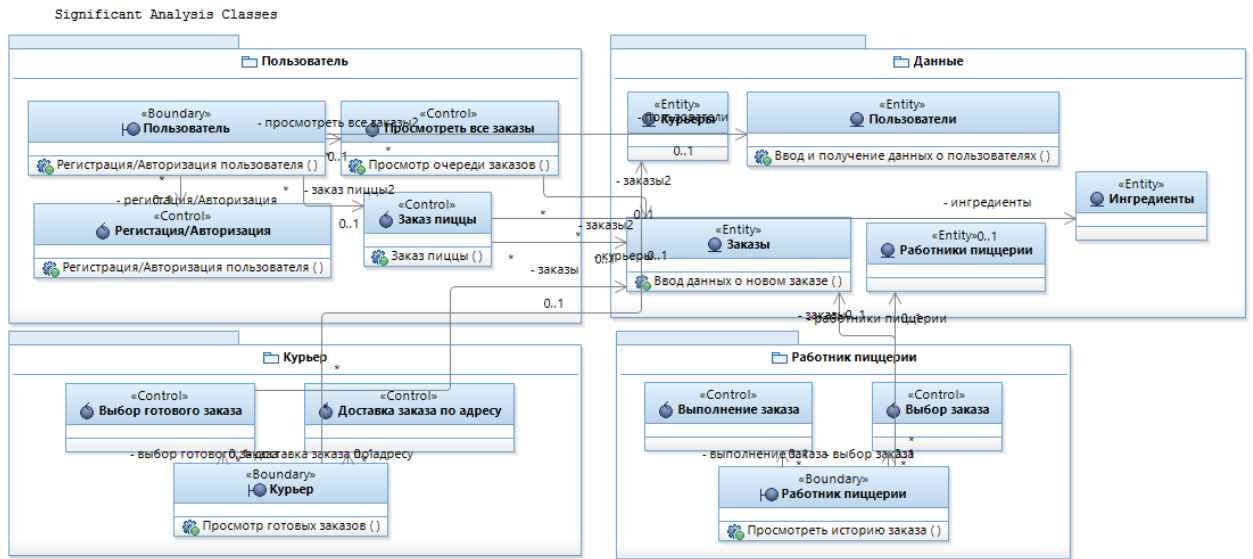


Рис. 20. Диаграмма классов в пакетах

Диаграммы последовательностей (рис. 12, 13), трассировка подсистем в компоненты (рис. 16) и развертывания (рис. 17) остались прежними. Диаграмма трассировки подсистем в компоненты выполнена в нотации пакетов и компонентов, а новые классы вошли в существующие пакеты.

15. Тестирование

Тесты проводились с помощью библиотеки QTest.

- 1) Добавление новых заказов – внесение данных в БД системы.
- 2) Изменение адреса доставки заказа – изменение данных в БД системы.
- 3) Просмотр истории заказов – вывод всех данных по истории заказов клиента в системе.
- 4) Просмотр информации о пиццерии – вывод информации о пиццерии.

Модульные тесты — для каждого модуля системы проводятся модульные тесты.

Интеграционные тесты — после каждого внедрения модуля проводится функциональное тестирование этого модуля внутри системы.

Системные тесты — после каждой итерации проводится тестирование всей системы.

Этап внедрения (Переход)

Для корректной работы системы должно быть установлено следующее ПО:

На одной машине:

- СУБД SQL Lite
- Windows 7 и выше
- Полный пакет Qt библиотек, содержащийся в поставке

Перечень документации для заказчика:

- Техническое задание
- Программа и методика испытаний

Перечень документации для пользователя:

- Описание системы
- Руководство по установке и эксплуатации системы
- Руководство пользователя
- Руководство системного администратора

Список литературы

1. Унифицированный процесс разработки программного обеспечения: учебное пособие /Виноградова М.В., Белоусова В.И. – М.: МГТУ им.Н.Э. Баумана. – 2015 г.
<http://ebooks.bmstu.ru/catalog/193/book1303.html>
2. Орлов С.А. Технологии разработки программного обеспечения. - СПб: Питер, - 2002 г.
3. Конспекты лекций Виноградовой М.В. по курсу «Технологии разработки ПО».- 2018 г.
4. - дата обращения 24.02.2019.