Министерство образования и науки

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Факультет инфокоммуникационных технологий

Кафедра программных систем

Отчет

по лабораторной работе № 1

# Проектирование архитектуры приложения

**Система управления проектами**

по дисциплине «Архитектура программного обеспечения инфокоммуникационных систем»

**Выполнил: студент группы K4220**

**Проверил: к.т.н., доцент Н.А. Осипов**

Санкт-Петербург

2017

## Цель работы.

– освоение процесса моделирования предметной области для проектирования архитектуры прикладных программных систем, разработки функциональных моделей прикладных программных систем, проектирования пользовательского интерфейса, разработки технического задания (ТЗ) на программные системы на этапе разработки прикладных программных систем

– моделирование предметной области"(Business Modeling),

– определение требований к программной системе" (Requirements) в соответствии с Rational Unified Process.

## Задачи, решаемые при выполнении работы.

### Создание архитектуры приложения.

### Определить элементы, отображаемые на UML-схемах.

## Объект исследования.

Архитектура приложения.

## Метод экспериментального исследования.

Имитационное визуально ориентированное моделирование.

## Рабочие формулы и исходные данные.

Visual Studio Ultimate предоставляет шаблоны для UML-диаграммы: вариантов использования.

Кроме того, можно создавать схемы слоев, которые помогают определить структуру системы.

UML-схемы моделирования и схемы слоев могут существовать только внутри проекта моделирования.

* Все проекты моделирования содержат общую UML-модель и несколько UML-диаграмм. Каждая диаграмма является представлением части модели. UML-модель содержит все элементы, отображаемые на UML-схемах, и может просматриваться с помощью обозревателя моделей UML.

## Схема работы.

### Создание архитектуры

Основное назначение архитектуры состоит в организации компонентов с целью обеспечения определенной функциональности. Такую организацию функциональности часто называют группировкой компонентов по «функциональным областям». На рис. 1.1 представлена типовая архитектура приложения, компоненты которого сгруппированы по функциональным областям.

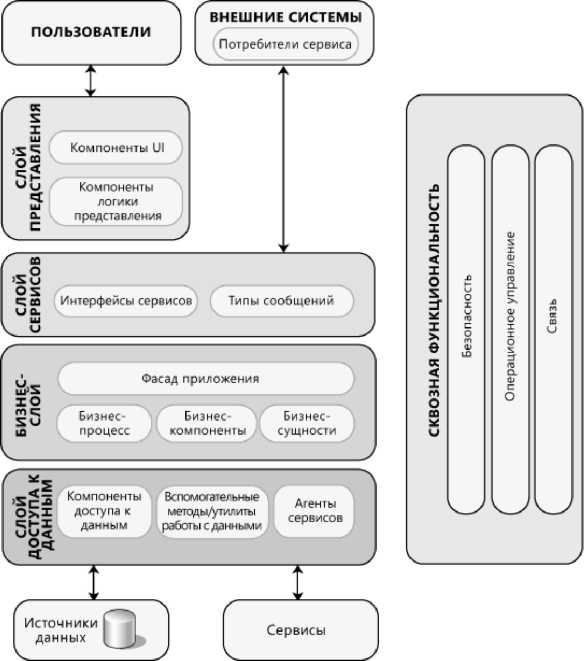


Рис. 1.1 Типовая архитектура приложения

Функциональные области используются не только для группировки компонентов, некоторые из них посвящены взаимодействию и организации совместной работы компонентов. Ниже приводятся рекомендации по различным функциональным областям, которыми необходимо руководствоваться при проектировании архитектуры собственного приложения.

## Окончательные результаты.

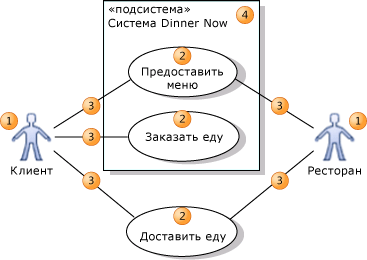
### Разработка схемы вариантов использования.

Схема вариантов использования является основным инструментом, используемым для описания пользовательских требований. Она описывает отношения между требованиями, пользователями и основными компонентами. Однако схема вариантов использования не описывает требования подробно, их можно представить на отдельных схемах или в документах, которые можно связать с каждым вариантом использования.

#### Чтение схем вариантов использования



В следующих таблицах описаны элементы, которые можно использовать на схеме вариантов использования, и их основные свойства.



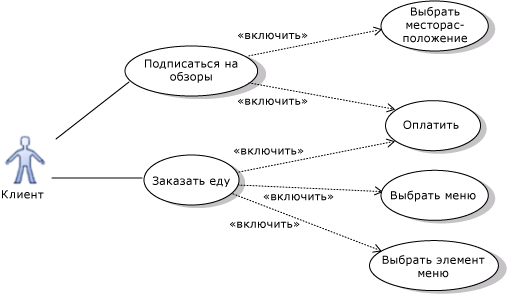
* *Актер* (1) — в терминах VS называется *субъектом*, это класс лиц, организаций, устройств или внешних программных компонентов, взаимодействующих с системой. Примерами субъектов являются следующие: Клиент, Ресторан, Датчик температуры, Устройство авторизации кредитных карт.
* *Вариант использования* (2) представляет действия, совершаемые одним или несколькими субъектами для достижения определенной цели. Примерами вариантов использования являются следующие: Заказ еды, Обновление меню, Обработка платежа.

На схеме вариантов использования они ассоциированы (3) с субъектами, выполняющими их.

* *Система (4)* — это любой объект в разработке. Системой может быть небольшой программный компонент, субъектами которой являются другие программные компоненты, полное приложение или крупный распределенный набор приложений, развернутых на нескольких компьютерах и устройствах. Примерами подсистем являются следующие: "Веб-сайт для заказа еды", "Бизнес по доставке еды", "Веб-сайт, версия 2".

Схема вариантов использования может показывать, какие варианты использования поддерживаются системой или ее подсистемами.

#### Структурирование вариантов использования



Цель и сценарии включенного варианта использования должны иметь смысл независимо друг от друга, чтобы их можно было включать в варианты использования, создаваемые позже.

Разделение вариантов использования на включающие и включенные части позволяет достичь следующих целей:

* Структурировать описания вариантов использования по уровню детализации.
* Избежать дублирования общих сценариев в разных вариантах использования.

## Выводы и анализ результатов работы.

Правильная архитектура снижает бизнес-риски, связанные с созданием технического решения; обладает значительной гибкостью, чтобы справляться с естественным развитием технологий, как в области оборудования и ПО, так и пользовательских сценариев и требований.

Архитектор должен учитывать общий эффект от принимаемых проектных решений, обязательно присутствующие компромиссы между атрибутами качества (такими как производительность и безопасность) и компромиссы, необходимые для выполнения пользовательских, системных и бизнес-требований.