

Санкт-Петербургский государственный политехнический университет

Институт информационных технологий и управления

Кафедра компьютерных систем и программных технологий

Курсовая работа

По предмету "Проектирование архитектур программного
обеспечения"

на тему:

Создание информационной системы

Выполнила: студентка группы 53501/3

Тарасова А. А.

Преподаватель: Зозуля А. В.

Санкт-Петербург

2015

Содержание

1	Сбор функциональных требований	3
1.1	Функциональные требования для клиента	3
1.2	Функциональные требования для экскурсовода	3
1.3	Функциональные требования для администратора	3
2	Разработка вариантов использования (обобщенная диаграмма прецен-	
	тов)	4
2.1	Варианты использования. Пользователь Клиент	4
2.2	Варианты использования. Пользователь Экскурсовод	5
2.3	Варианты использования. Пользователь Администратор	5
2.4	Диаграмма прецендов	6
3	Подробное описание всех вариантов использования (текстовое описание с	
	альтернативами)	7
4	Разработка статической объектной модели предметной области (диаграммы	
	классов)	8
5	Разработка динамической объектной модели предметной области (диаграм-	
	мы последовательности)	8
6	Проектирование слоя бизнес-логики (выбор архитектурного шаблона уров-	
	ня бизнес-логики)	8
7	Реализация слоя бизнес-логики (Java, NetBeans), unit-тестирование (JUnit)	8
8	Проектирование слоя источников данных (выбор архитектурного шаблона	
	уровня доступа к данным: DB + внешний сервис)	8
9	Реализация слоя источников данных (JavaDB, NetBeans), unit-тестирование	8
10	Проектирование сервисного слоя и слоя представления: GUI (Swing), внеш-	
	ний сервис	9
11	Реализация слоев представления, сервисного слоя, unit-тестирование сер-	
	висного слоя	9
12	Комплексное тестирование системы	9

1 Сбор функциональных требований

Разрабатываемая информационная система является сервисом, позволяющим просматривать предложения об экскурсиях по Санкт-Петербургу, бронировать экскурсии, а так же следить за текущими предложениями экскурсоводов.

Пользователями системы могут быть:

- **Клиенты** - обычные пользователи, которые хотят воспользоваться разрабатываемой информационной системой для выбора и заказа самых разнообразных экскурсий.
- **Экскурсоводы** - пользователь, который может предложить экскурсии.
- **Администратор** - пользователь, который занимается проверкой всей поступившей информации, модерацией, добавлением и редактированием персональных данных.

1.1 Функциональные требования для клиента

- Просмотр полного списка экскурсий с их описанием
- Бронирование экскурсии
- Внесение предоплаты за экскурсию
- Отмена брони экскурсии

1.2 Функциональные требования для экскурсовода

- Просмотр полного списка экскурсий с их описанием
- Добавление экскурсий и их описания

1.3 Функциональные требования для администратора

- Просмотр полного списка экскурсий с их описанием
- Модерация экскурсий, добавленных экскурсоводом
- Размещение и удаление экскурсий

2 Разработка вариантов использования (обобщенная диаграмма прецедентов)

2.1 Варианты использования. Пользователь Клиент

1. Просмотр полного списка экскурсий с их описанием

- (a) При запуске программы пользователь проходит авторизацию как *гость*
- (b) Пользователь использует функцию *список экскурсий*
- (c) Система отображает список со всеми экскурсиями и их описанием

2. Бронирование экскурсии

- (a) При запуске программы пользователь проходит авторизацию как *гость*
- (b) Пользователь использует функцию *список экскурсий*
- (c) Система отображает список со всеми экскурсиями и их описанием
- (d) Пользователь использует функцию *забронировать экскурсию*
- (e) Система отображает сообщение, что экскурсия экскурсия будет забронирована после внесения предоплаты

3. Внесение предоплаты за экскурсию

- (a) При запуске программы пользователь проходит авторизацию как *гость*
- (b) Пользователь использует функцию *список экскурсий*
- (c) Система отображает список со всеми экскурсиями и их описанием
- (d) Пользователь использует функцию *забронировать экскурсию*
- (e) Система отображает сообщение, что экскурсия будет забронирована после внесения предоплаты
- (f) Пользователь использует функцию *внести предоплату* и указывает данные для оплаты, имя пользователя и пароль.
- (g) Система выводит сообщение, что экскурсия забронирована..

4. Отмена брони экскурсии

- (a) При запуске программы пользователь проходит авторизацию как *пользователь*
- (b) Пользователь использует функцию *Мои экскурсии*

- (с) Система отображает список со всеми забронированными экскурсиями пользователя и их описанием
- (d) Пользователь использует функцию *отменить бронь экскурсии*
- (е) Система отображает сообщение, что в случае отмены брони менее чем за 48 часов - предоплата не возвращается.
- (f) Пользователь использует функцию *согласен с условиями*
- (g) Система выводит сообщение, что бронь на экскурсию отменена и предоплата будет возвращена. В случае если пользователь решил отменить бронь менее чем за 48 часов до начала мероприятия - предоплата не возвращается.

2.2 Варианты использования. Пользователь Экскурсовод

1. Просмотр полного списка экскурсий с описанием

- (a) При запуске программы пользователь проходит авторизацию как *гость*
- (b) Пользователь использует функцию *список экскурсий*
- (с) Система отображает список со всеми экскурсиями и их описанием

2. Добавление экскурсий и их описания

- (a) При запуске программы пользователь проходит авторизацию как *экскурсовод*
- (b) Пользователь использует функцию *предложить экскурсию*
- (с) Система отображает форму для заполнения
- (d) Пользователь использует функцию *добавить экскурсию*
- (е) Система выводит сообщение, что экскурсия будет доступна после модерации.

2.3 Варианты использования. Пользователь Администратор

1. Просмотр полного списка экскурсий с описанием

- (a) При запуске программы пользователь проходит авторизацию как *гость*
- (b) Пользователь использует функцию *список экскурсий*
- (с) Система отображает список со всеми экскурсиями и их описанием

2. Модерация экскурсий, добавленных экскурсоводами

- (a) Пользователь проходит авторизацию как *администратор*

(b) Пользователь использует функцию *список предложенных экскурсий*

(c) Пользователь использует функцию *разместить экскурсию*

2.4 Диаграмма прецедентов

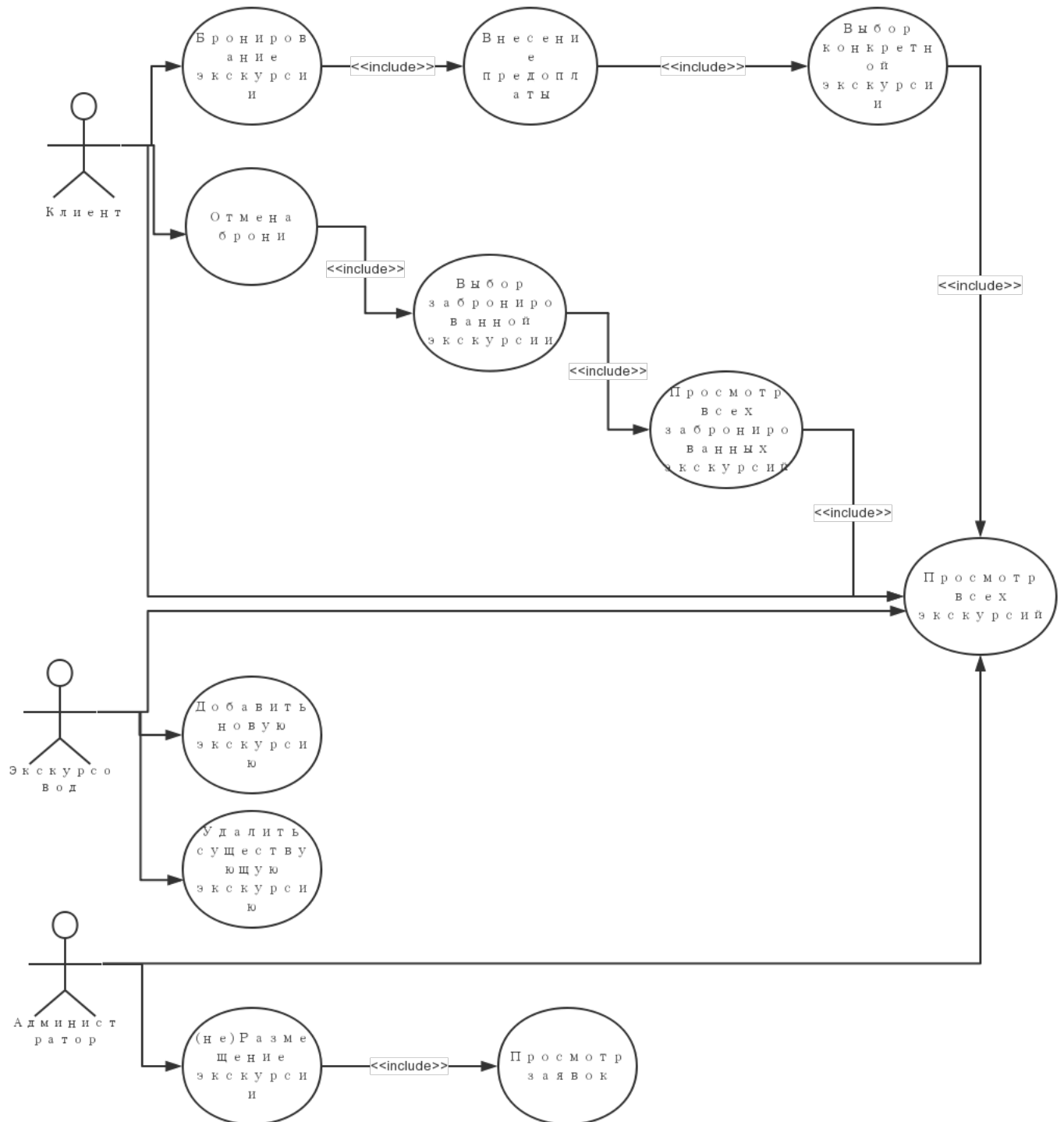


Рис. 1: Диаграмма прецедентов

3 Подробное описание всех вариантов использования (текстовое описание с альтернативами)

Заказ экскурсии

1. Клиент просматривает список экскурсий и выбирает экскурсию для бронирования
2. Система выводит полную информацию о текущей экскурсии и ее цене
3. Клиент вводит информацию, необходимую для бронирования экскурсии: данные банковской карты, логин и пароль.
4. Система подтверждает оплату
5. Система отправляет клиенту и экскурсоводу контакты друг друга

Альтернатива: постоянный клиент

Система выводит полную информацию о текущей экскурсии и ее цене, а также последние 4 цифры информации о банковской карточке

- 4 Разработка статической объектной модели предметной области (диаграммы классов)
- 5 Разработка динамической объектной модели предметной области (диаграммы последовательности)
- 6 Проектирование слоя бизнес-логики (выбор архитектурного шаблона уровня бизнес-логики)
- 7 Реализация слоя бизнес-логики (Java, NetBeans), unit-тестирование (JUnit)
- 8 Проектирование слоя источников данных (выбор архитектурного шаблона уровня доступа к данным: DB + внешний сервис)
- 9 Реализация слоя источников данных (JavaDB, NetBeans), unit-тестирование

Таблица пользователей:

- idUser
- userType
- userName
- numTelephone
- e-mail

Таблица экскурсий:

- idExcursion
- nameExcursion
- descriptionExcursion
- idGuide
- status //null-не проводилась валидация, 1 - разрешено, 0 - не разрешено

Таблица клиент-экскурсия

- idRecord
- idUser
- idExcursion

- 10 Проектирование сервисного слоя и слоя представления: GUI (Swing), внешний сервис
- 11 Реализация слоев представления, сервисного слоя, unit-тестирование сервисного слоя
- 12 Комплексное тестирование системы