

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Отчет по курсовой работе

по дисциплине «Программное обеспечение распределенных вычислительных
сетей»

«Информационная система для букмекерской конторы»

Работу выполнил студент группы №23541/3

Шаляпин Н.С. _____

Работу принял преподаватель

к.т.н., доц. Стручков И.В. _____

Санкт-Петербург

2017

Индивидуальное задание

Разработать информационную систему для букмекерской конторы.

В информационной системе предусмотрены следующие роли

- Букмекер - представитель букмекерской конторы, осуществляет контроль за компаниями и пользователями
- Компания - представитель компании, создающий события и исходы, на которые производятся ставки
- Пользователь - непосредственно участник ставок. Выбирает события и делает ставки на определенные исходы

Функциональные требования

1) Для пользователя:

- Пополнение средств
- Отображение доступных событий и исходов
- Создание ставки
- Пополнение средств
- Пополнение счета при успешном исходе ставки
- Снятие средств

2) Для компании:

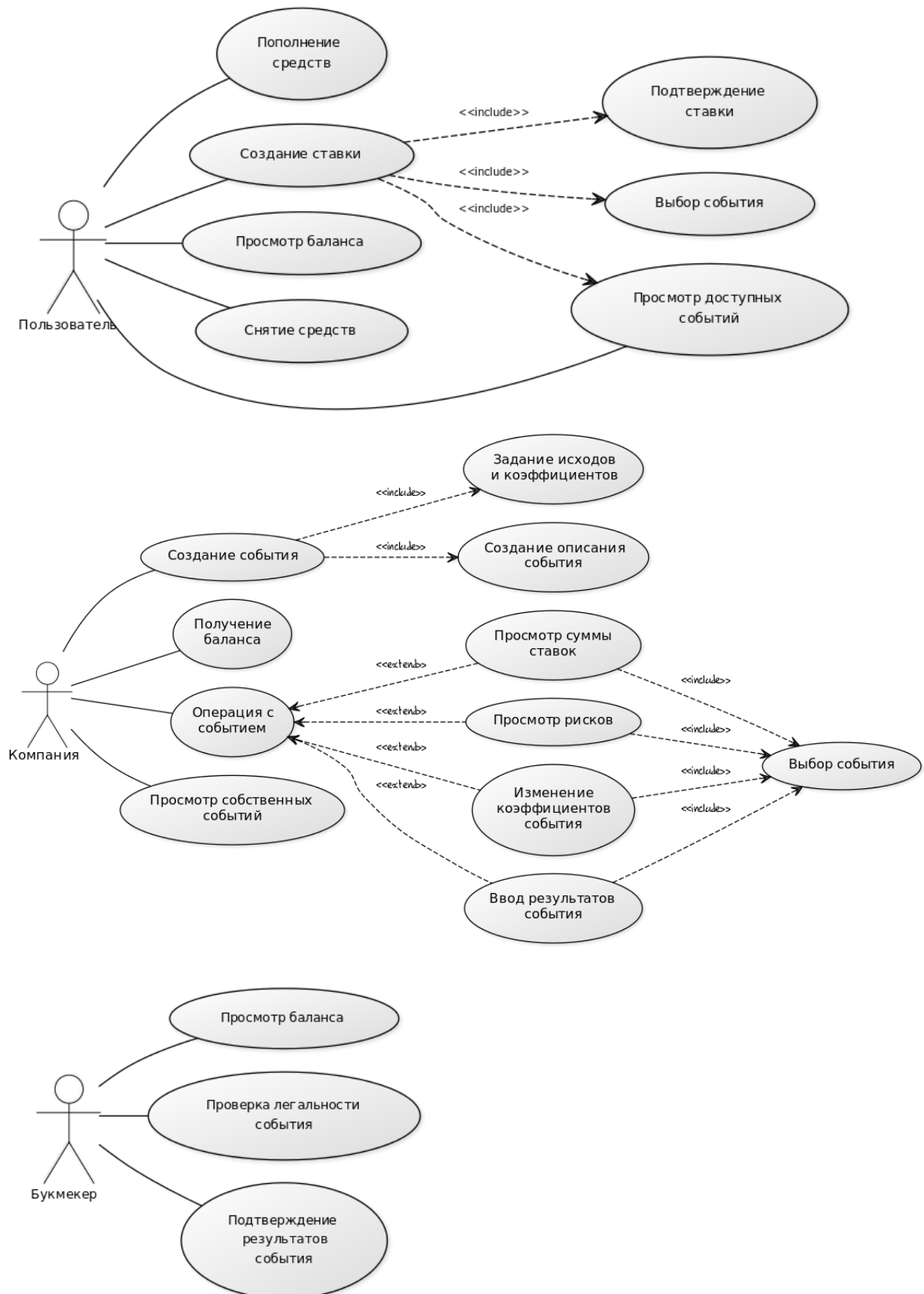
- Создания события с исходами
- Отображение коэффициентов события
- Отображение рисков
- Отображение общей суммы ставок
- Перерасчет прибыли при определении результатов исхода
- Просмотр баланса

3) Для букмекера:

- Подтверждение результатов события
- Проверка легальности события
- Просмотр баланса

Разработка вариантов использования

Обобщенная диаграмма прецедентов:



Подробное описание вариантов использования

1. Пользователь

1.1. Пополнение средств

1. Пользователь авторизуется
2. Система предоставляет ему реквизиты
3. Пользователь вносит средства используя иные инструменты

1.2. Просмотр баланса

1. Пользователь авторизуется
2. Система предоставляет информацию о балансе

1.3. Снятие средств

1. Пользователь авторизуется
2. Пользователь заказывает вывод средств на сторонние счета
3. Система проверяет наличие указанной суммы

Альтернатива: указанной суммы нет

Заявка отменяется

1. Средства списываются со счета
2. Пользователь ожидает поступления на стороннем счете

1.4. Просмотр доступных событий

1. Пользователь авторизуется
2. Система предоставляет информацию о доступных событиях, их исходов и коэф.

1.5. Создание ставки

1. Пользователь авторизуется
2. Система предоставляет информацию о доступных событиях, их исходов и коэф.
3. Пользователь выбирает одно из событий
4. Пользователь указывает исход и сумму ставки
5. Пользователь подтверждает создание ставки

6. Система проверяет наличие указанной суммы

Альтернатива: указанной суммы нет

Операция отменяется

1. Система регистрирует указанную ставку

2. Компания

2.1. Получение баланса

1. Представитель компании авторизуется
2. Система предоставляет информацию о балансе

2.2. Создание события

1. Представитель компании авторизуется
2. Представитель вводит описание события
3. Представитель вводит возможные исходы и коэффициенты
4. Представитель подтверждает создание нового исхода

2.3. Просмотр собственных событий

1. Представитель компании авторизуется
2. Система предоставляет информацию о созданных событиях

2.4. Операция над событием

1. Представитель компании авторизуется
2. Представитель просматривает список событий компании и выбирает одно из них

2.4.1 Просмотр суммы ставок

1. Выполняются действия из 2.4
2. Система предоставляет информацию о сумме ставок на каждый исход

2.4.2 Просмотр рисков

1. Выполняются действия из 2.4
2. Система предоставляет информацию о сумме рисков по каждому исходу

2.4.3 Изменение коэффициентов

1. Выполняются действия из 2.4
2. Представитель указывает новые коэффициенты для каждого исхода
3. Представитель подтверждает изменения

2.4.4 Ввод результатов события

1. Выполняются действия из 2.4
2. Представитель указывает победивший исход
3. Система блокирует событие для новых ставок и изменений
4. Событие ожидает подтверждение Букмекера (п 3.3)

3. Букмекер

3.1. Просмотр баланса

1. Представитель букмекерской конторы (букмекер) авторизуется
2. Система предоставляет информацию о текущем балансе

3.2. Проверка легальности события

1. Букмекер авторизуется
2. Букмекер выбирает событие
3. Букмекер запрашивает статус легальности события
4. Система осуществляет запрос к стороннему сервису и предоставление полученной информации

3.3. Подтверждение результатов события

1. Букмекер авторизуется
2. Букмекер выбирает событие
3. Букмекер подтверждает победивший исход

Альтернатива: букмекер не подтверждает правдивость данных и указывает другой исход

Система фиксирует указанный букмекером исход как победивший

1. Система выплачивает соответствующую сумму победившим Пользователям
2. Система начисляет комиссию на счет Букмекера

3. Система выплачивает Компании проигравшие ставки
4. Система меняет статус события на “выплачен”

Диаграмма классов сущностей в проекте.

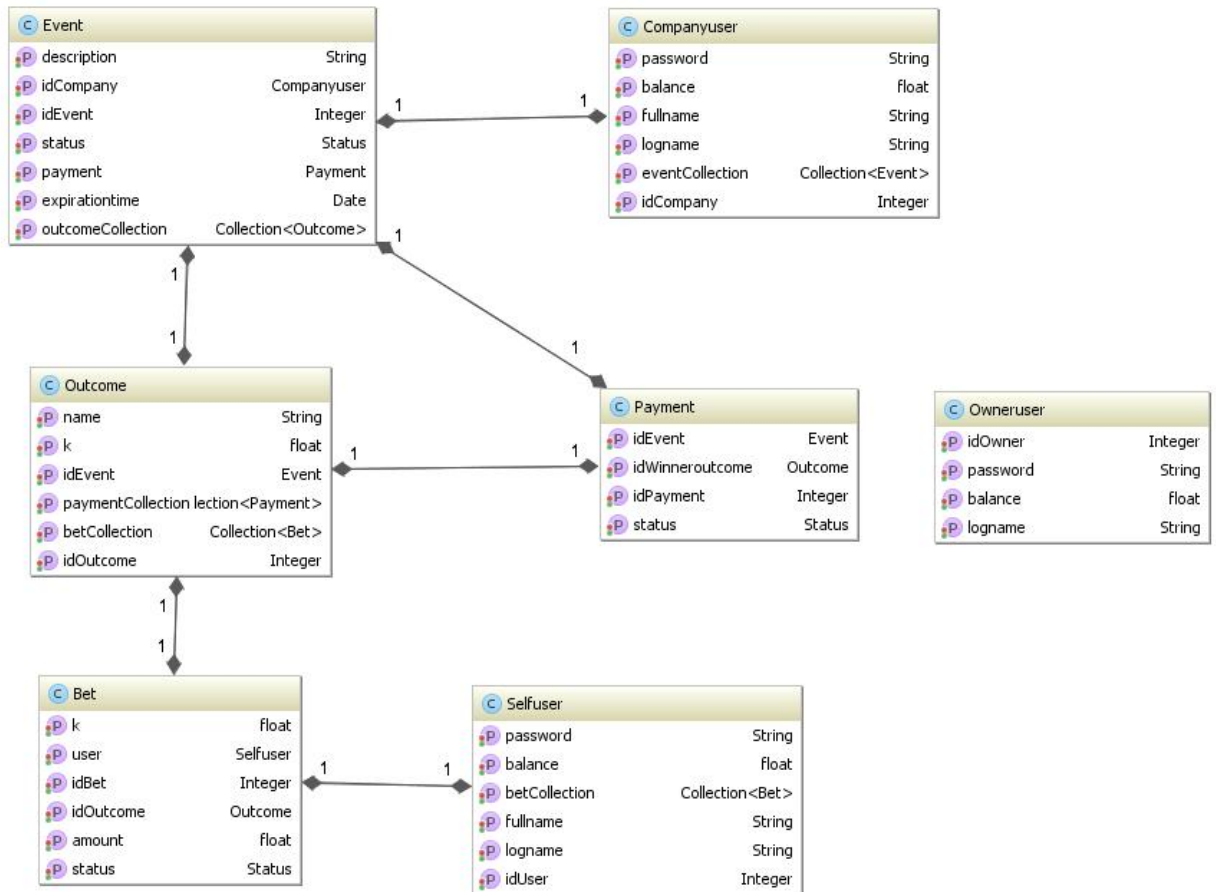


Рис. 2. Диаграмма классов.

Разработка динамической объектной модели предметной области

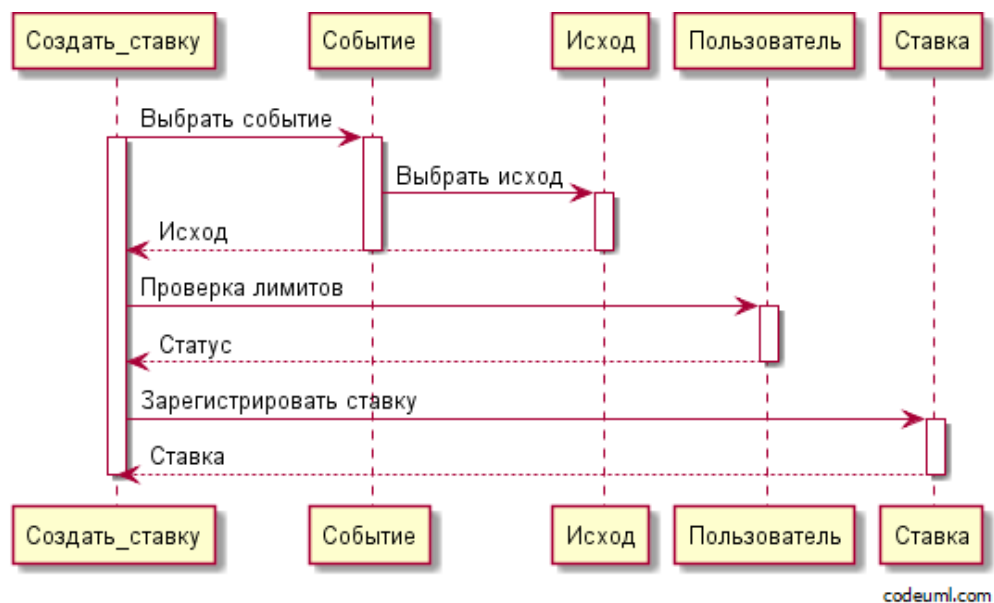


Рис. 3. Диаграмма последовательности «Создание ставки».

Реализация индивидуального задания

Объектно-ориентированное проектирование

Разработав и подробно описав варианты использования, разработав как статическую, так и динамическую объектные модели предметной области, необходимо определить используемые инструменты в проекте.

В качестве СУБД было решено использовать MySQL, так как это открытая, бесплатная и популярная СУБД.

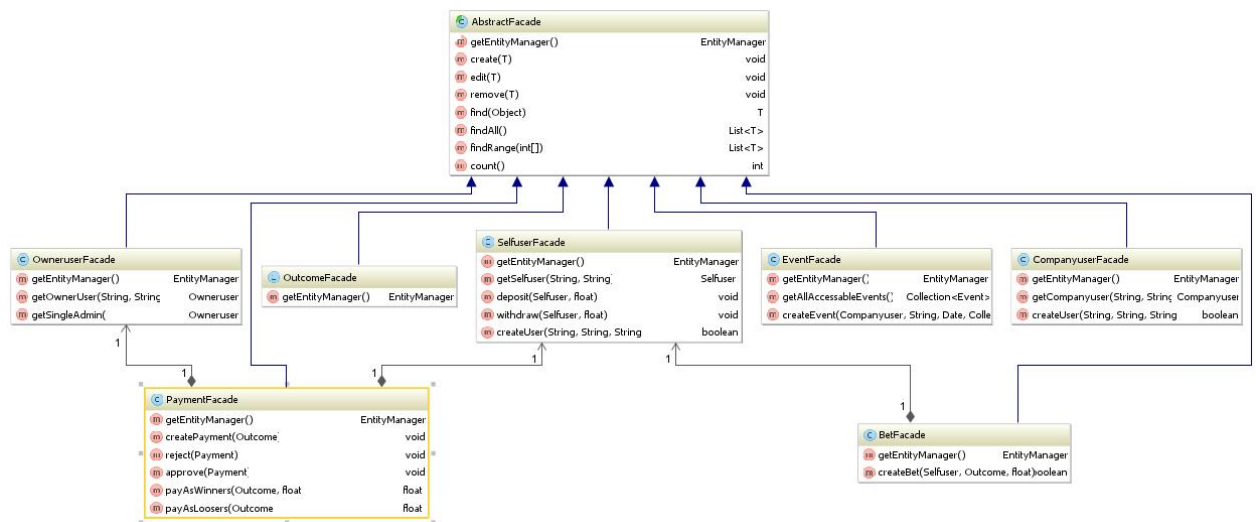
В качестве решения для организации бизнес-логики было решено использовать сеансовые компоненты без состояния (Stateless Session Beans), так как все временные данные можно хранить в БД, при этом в дальнейшем приложение может легко масштабироваться. Также Stateless Session Beans проще в использовании.

Так как для указанного решения уже была разработана БД, то слой доступа к данным был автоматически сгенерирован при помощи IDE NetBeans. Для каждой полученной сущности также были сгенерированы локальные фасады (Stateless Session Beans). Вся необходимая бизнес логика, которой не хватало была добавлена в указанные классы. Было решено не создавать удаленный интерфейсов для бизнес-логики. Весь компоненты слиты в один проект. Это позволяет получить

доступ к сущностям из более высокоуровневых слоев. Это удобно, например, в случае отображения данных или при подготовке создания объекта.

Описание программы

В ходе работы приложения было решено реализовать GUI при помощи JSF. Содержание и управление страницами осуществляется при помощи объектов Managed Beans. Указанные Managed Beans используют методы фасады сущностей.



Текст программы

Все исходники программы расположены в репозитории по адресу:

https://github.com/NikitaShalyapin/PO_RVS_EjbBet

Методика и результаты тестирования

Для тестирования web-приложений использовались такие браузеры как Google Chrome, Mozilla и Opera. Тестирование проводилось вручную. В качестве основы были использованы use cases из главы 1. Отдельно тестировалась корректность авторизации и ограничения доступа, а также устойчивость приложения к некорректным данным.

Тестирование показало, что разработанная реализация удовлетворяет основным требованиям.

Инструкция системного администратора

Для работы приложения понадобится СУБД MySQL.

Необходимо создать БД Bet и настроить доступ к ней для пользователя betadmin.

Для указанной БД выполнить скрипт Bet.sql.

Для развертки приложения необходим GlassFish Server версии 4.1.1.

Запуск домена:

```
asadmin start-domain domain1
```

Далее скачать исходники проекта с GitHub:

```
git clone https://github.com/NikitaShalyapin/PO_RVS_EjbBet
```

Собрать проект в подкаталоге EjbBet:

```
mvn package
```

Развернуть приложение на сервере GlassFish:

```
asadmin deploy EjbBet-1.0-SNAPSHOT.war
```

Также приложение можно развернуть, используя удаленный интерфейс GlassFish.

Инструкция пользователя

Окно регистрации. Можно выбрать тип пользователя (компания или пользователь):

Sign up form

User type:	<input type="text" value="user"/>
Full name:	<input type="text" value="The user"/>
Username:	<input type="text" value="user"/>
Password:	<input type="password" value="....."/>
<input type="button" value="OK"/>	

Главное окно пользователя:

User common information

Username: llama

Full name: The llama

Balance: 33.0

Actions

- [Show bets](#)
- [Show available events](#)

Deposit/withdraw simulation

Operation:

Amount:

Окно просмотра ставок пользователя. Можно также фильтровать список по статусу. Для этого необходимо установить checkbox.

All user bets

event	outcome	amount	rate	status
qwerty	Yes	100.0	2.0	Open
newevent	No	100.0	1.5	Paid
qaz	No	100.0	1.5	Paid
test event	Yes	1.0	2.0	Paid
test event	Yes	2.0	2.0	Paid
test event	Yes	3.0	2.0	Paid
test event	Yes	3.0	2.0	Paid
test event	Yes	5.0	2.0	Paid
Is this correct app?	Yes	100.0	1.1	Open

Only open: ☐

Вывод

В ходе разработки информационной системы для букмекерской конторы были рассмотрены все роли пользователей. Для каждой из ролей были определены функциональные требования, после чего разработаны и описаны варианты использования.

Разработав и подробно описав варианты использования, как статическую, так и динамическую объектные модели предметной области был спроектирован слой предметной области. Была изучена и использована спецификация Enterprise JavaBeans (EJB), а также framework JSF 2.0.

Дальнейшее расширение системы не составит труда. Это обусловлено подходом к проектированию приложения, разделению бизнес-логики, работу с данными и представление на разные слои. Наиболее уязвимым местом в указанном решении является изменения сущностей. Данное изменение может повлечь за собой необходимость рефакторинга всех вышележащих слоев.

Общее впечатление от платформы смешанное. Указанная технология в чистом виде является крайне неудобной и непредсказуемой, ее применимость по большей части зависит от уровня ее поддержки в используемых инструментах. Кроме того, в ходе работы было обнаружено несколько ошибок в системе. Краткий поиск показал, что некоторые из них находятся или находились в системе значительное время. Однако, следует отметить, что данная технология, при значительной автоматизации позволяет быстро и эффективно разрабатывать бизнес-приложения.