Санкт-Петербургский политехнический университет Петра Великого

Институт информационных технологий и управления

Кафедра компьютерных систем и программных технологий

Отчет

о лабораторной работе

по дисциплине: «Распределенные системы»

на тему: «Разработка корпоративных распределенных web-приложений   
с помощью технологии Enterprise Java Beans»

Выполнил: студент группы 63501/2

Мурашко Денис Сергеевич

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель: Стручков Игорь Вячеславович

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2016 г.

Санкт-Петербург

2016

# Задание Система электронного голосования. Операции удаленного объекта: добавить кандидата, удалить кандидата, подать голос за кандидата, получить результат по кандидату. Сериализуемый объект: результат голосования (имя кандидата, количество голосов, процент от общего числа голосов).

# Анализ задания

Выполним объектно-ориентированный анализ предметной области: с помощью UML-диаграмм отобразим требуемые для выполнения задания понятия и связи между ними.

Для начала составим список понятий предметной области:

* Избиратель
* Кандидат
* Избирательная комиссия
* Голос
* Результат голосования

**Модель предметной области**



**Роли**

Избиратель

* подает голос за кандидата
* получает результаты по кандидату (количество голосов)

2. Кандидат

* получает результаты выборов
* поздравления при победе

3. Избирательная комиссия

* добавляет/удаляет кандидата
* начинает/заканчивает выборы
* объявляет результаты голосования

4. Голос

* принадлежность кандидату

5. Результат голосования

* кто победил в выборах/нет победителя

**Варианты использования**

**Избиратель**

1. Избиратель голосует за кандидата

* Избиратель логинится в систему - вводит Номер и серию паспорта
* Избиратель ставит Галочку напротив своего кандидата и кнопку Голосовать

2. Избиратель получает результаты голосования

* Избиратель логинится в систему - вводит Номер и серию паспорта
* Избиратель получает результаты - список кандидатов, количество голосов, победителя

**Кандидат**

1. Кандидат получает результаты голосования

2. Кандидат побеждает на выборах

* Система выводит сообщения о победе

**Избирательная комиссия**

1. Избирательная комиссия может добавлять и удалять кандидатов (до начала голосования)

* Ввести имя и фамилию кандидата, нажать кнопку "Добавить кандидата",
* Ввести имя и фамилию кандидата, нажать кнопку "Удалить кандидата",

3. Избирательная комиссия начинает/заканчивает выборы (кандидат или нет победителя)

* Избирательная комиссия комиссия нажимает на кнопку "Начать выборы"
* Избирательная комиссия комиссия нажимает на кнопку "Завершить выборы"

**Диаграмма вариантов использования**

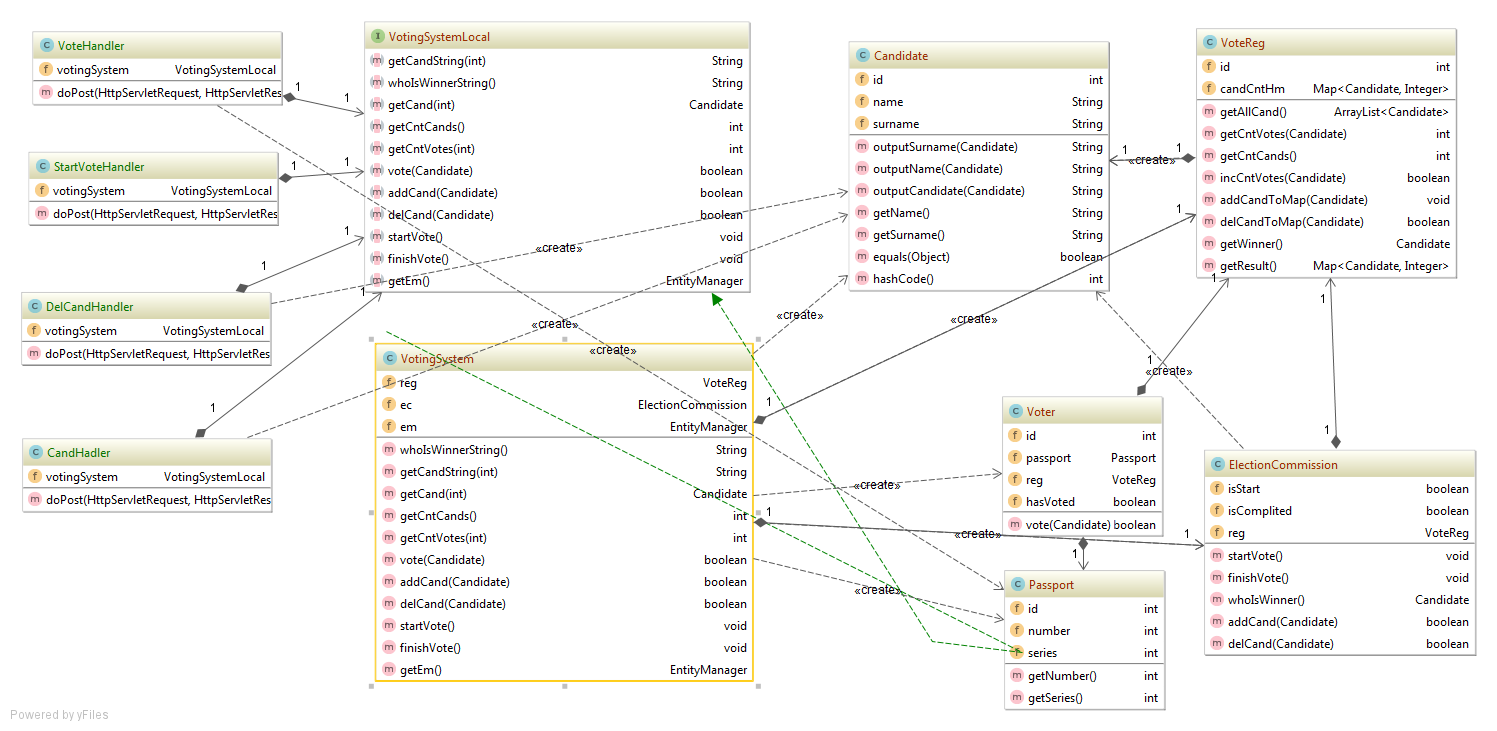
****

**Разработка**

**Объектно-ориентированное программирование**

**Диаграмма классов**

Ниже приведена диаграмма классов, в которой содержатся классы, соответствующие классам предметной области и интерфейсам будущей системы.



|  |  |  |
| --- | --- | --- |
| **Класс** | **Предназначение** | **Поля** |
| Passport | Сущностный класс для паспортов избирателей | Id – идентификатор, первичный ключ для соответствующей таблицы, number – номер, series – серия |
| Voter | Сущностный класс для избирателей | Id – идентификатор, первичный ключ для соответствующей таблицы, passport – паспорт избирателя,  reg – регистр учета голосов  hasVoted – признак того, что избиратель проголосовал |
| Candidate | Сущностный класс для кандидата | Id - идентификатор, первичный ключ для соответствующей таблицы,  surname – фамилия, name – имя |
| VoteReg | Сущностный класс для подсчета голосов | Id – идентификатор, первичный ключ для соответствующей таблицы,  candCntHm – HashMap: ключ – кандидат, значение – количество голосов у кандидата |
| ElectionCommission | Сущностный класс для избирательной комиссии | isStart – признак начала голосования,  isComplited– признак завершения выборов,  reg – регистр учета голосов |
| VotingSystem | Объединяющий класс  @Singleton | reg – регистр учета голосов,  ec – избирательная комиссия  em - EntityManager |
| VotingSystemLocal | Локальный Интерфейс класса VotingSystem |  |

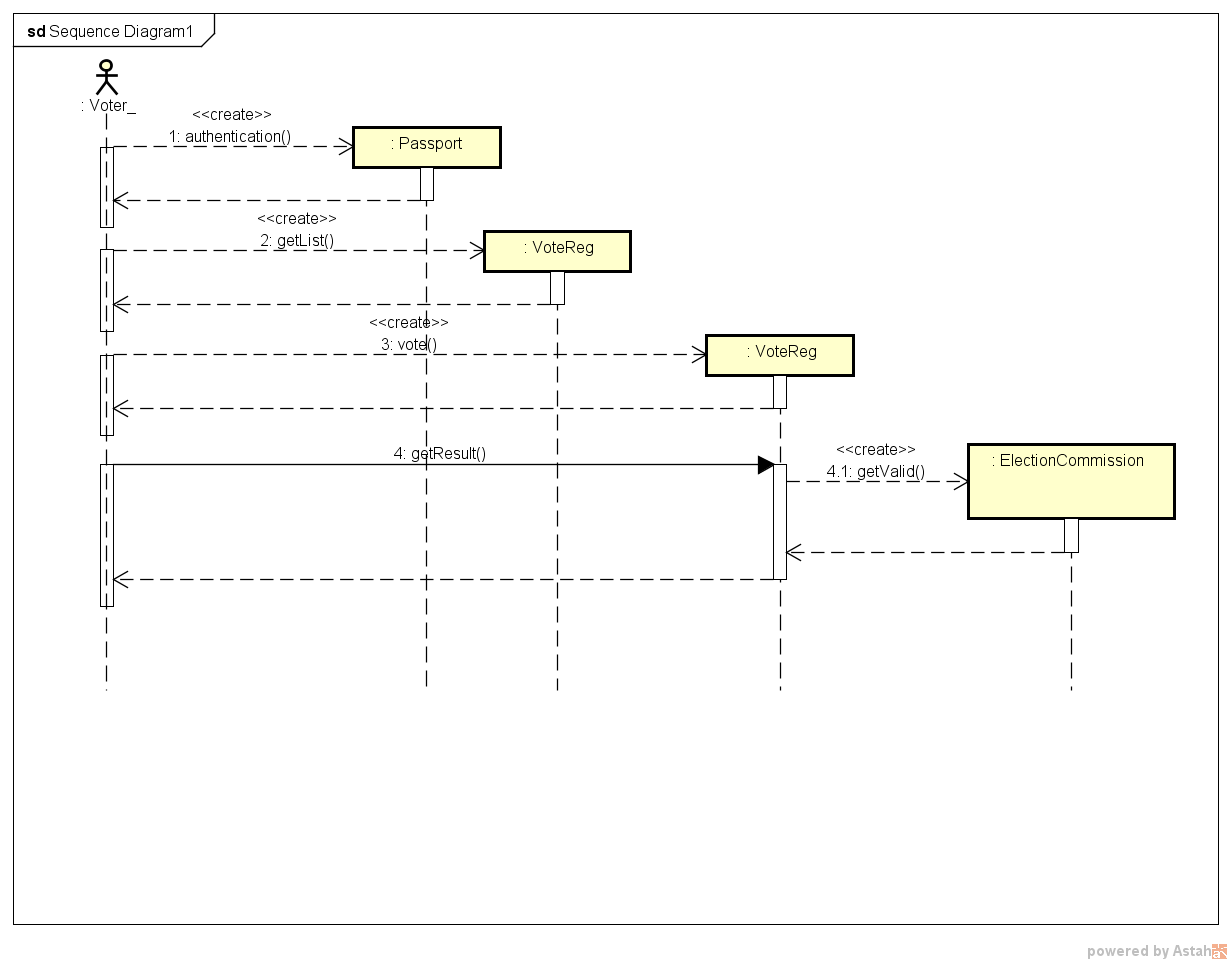
Классы VoteHandler, CandHadler, DelCandHandler, StartVoteHandler реализуют логику Web-интерфейса, которая основана на API Servlet.

Класс VotingSystem c аннотацией Singleton - гарантирует, что у класса есть только один экземпляр, и предоставляет к нему глобальную точку доступа.

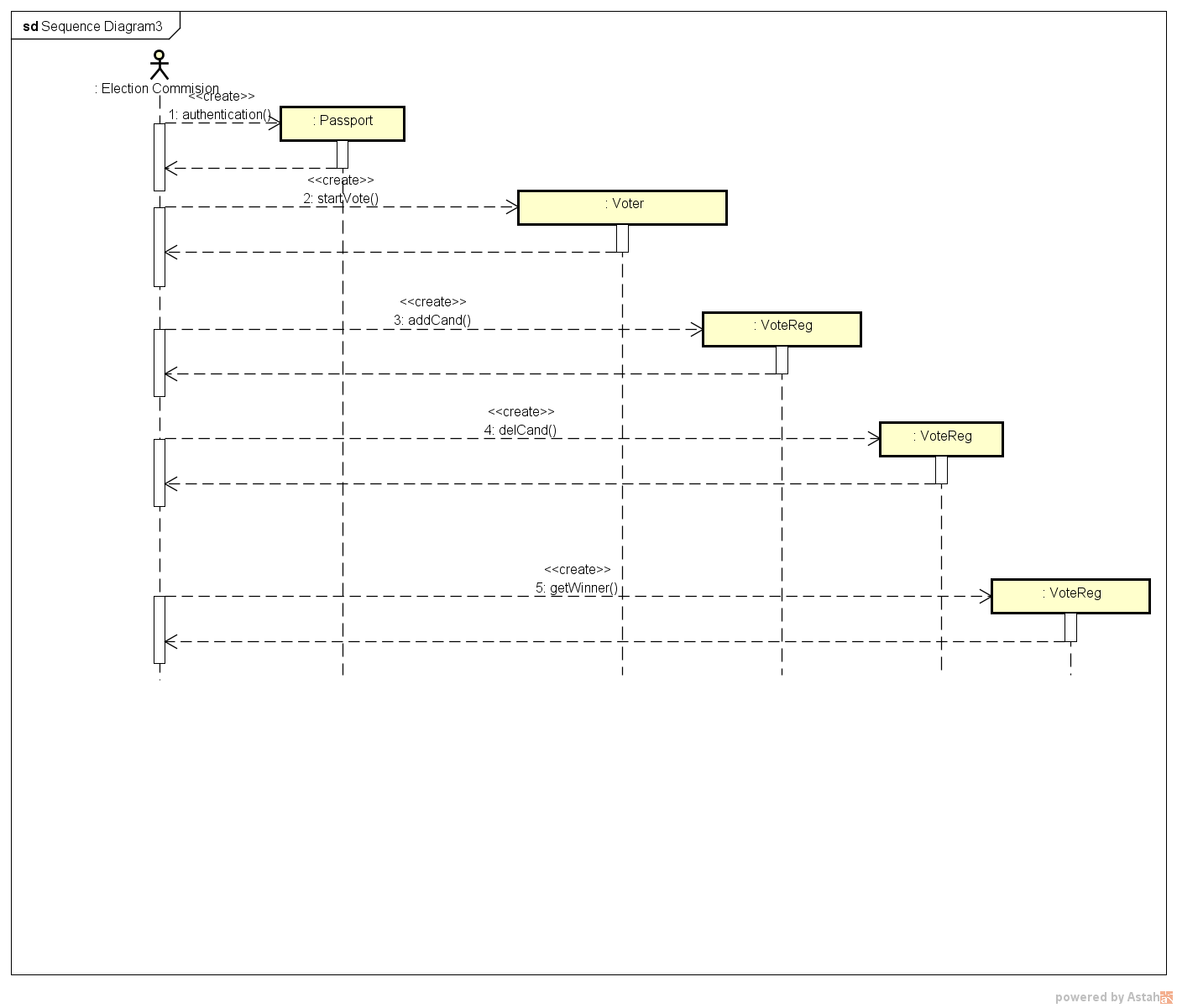
ORM реализован с помощью библиотеки Hibernate с помощью интерфейса JPA (Java Persistence API). Пример обращения к БД (запрос на наличие паспорта в БД):  
  
List<Passport> list = **votingSystem**.getEm().createNamedQuery(**"findPassport"**, Passport.**class**)  
 .setParameter(**"number"**, number)  
 .setParameter(**"series"**, serial)  
 .getResultList();

**Диаграммы последовательностей**

**Избиратель голосует за кандидата и получает результат**

****

**Избирательная комиссия начинает/заканчивает голосование, добавляет/удаляет кандидата и подводит результаты выборов**

****