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**

C:\Git\System-of-vote\classes.png

**Роли**

Избиратель

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

2. Кандидат

* получает результаты выборов

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

2. Избиратель получает результаты голосования

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

**Кандидат**

1. Кандидат получает результаты голосования

**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",

2. 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**

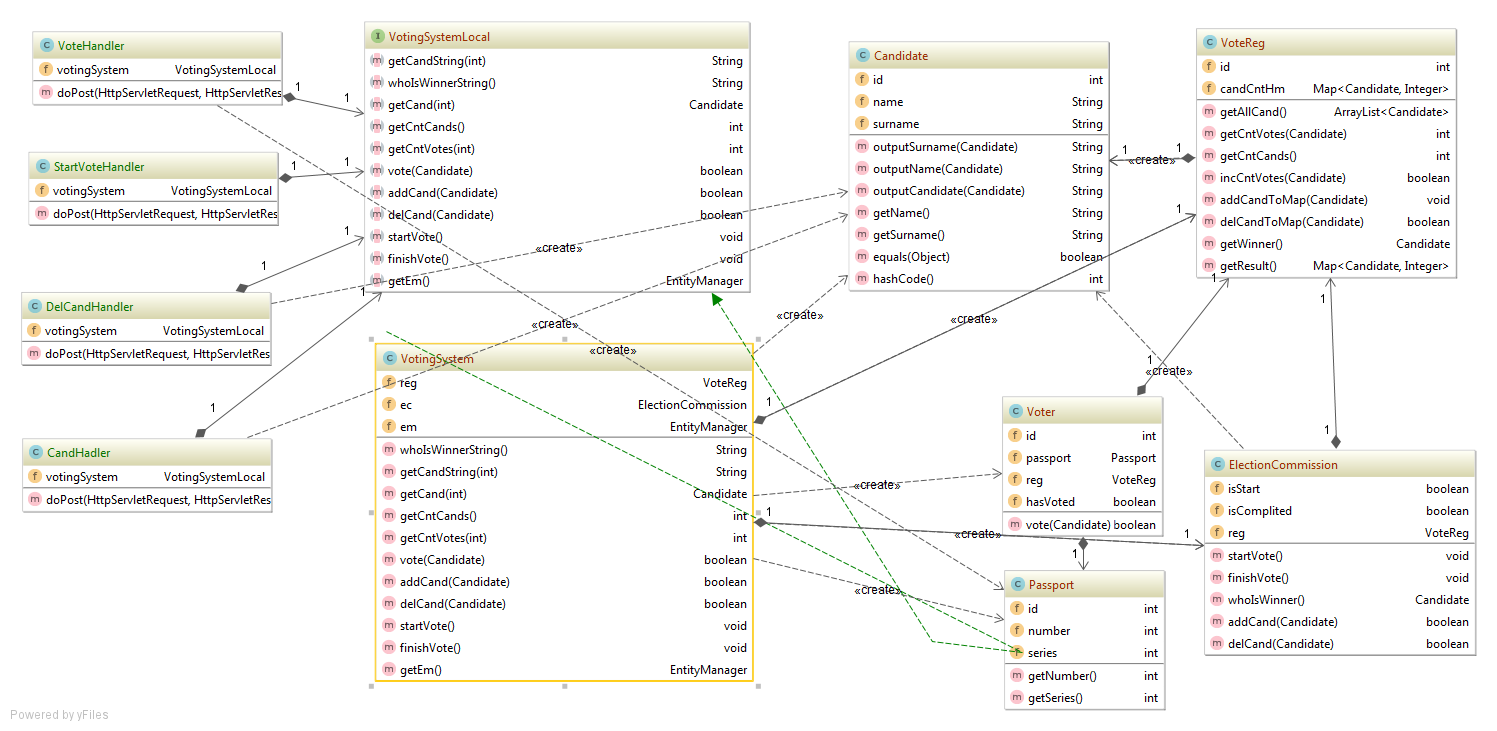
****

**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.

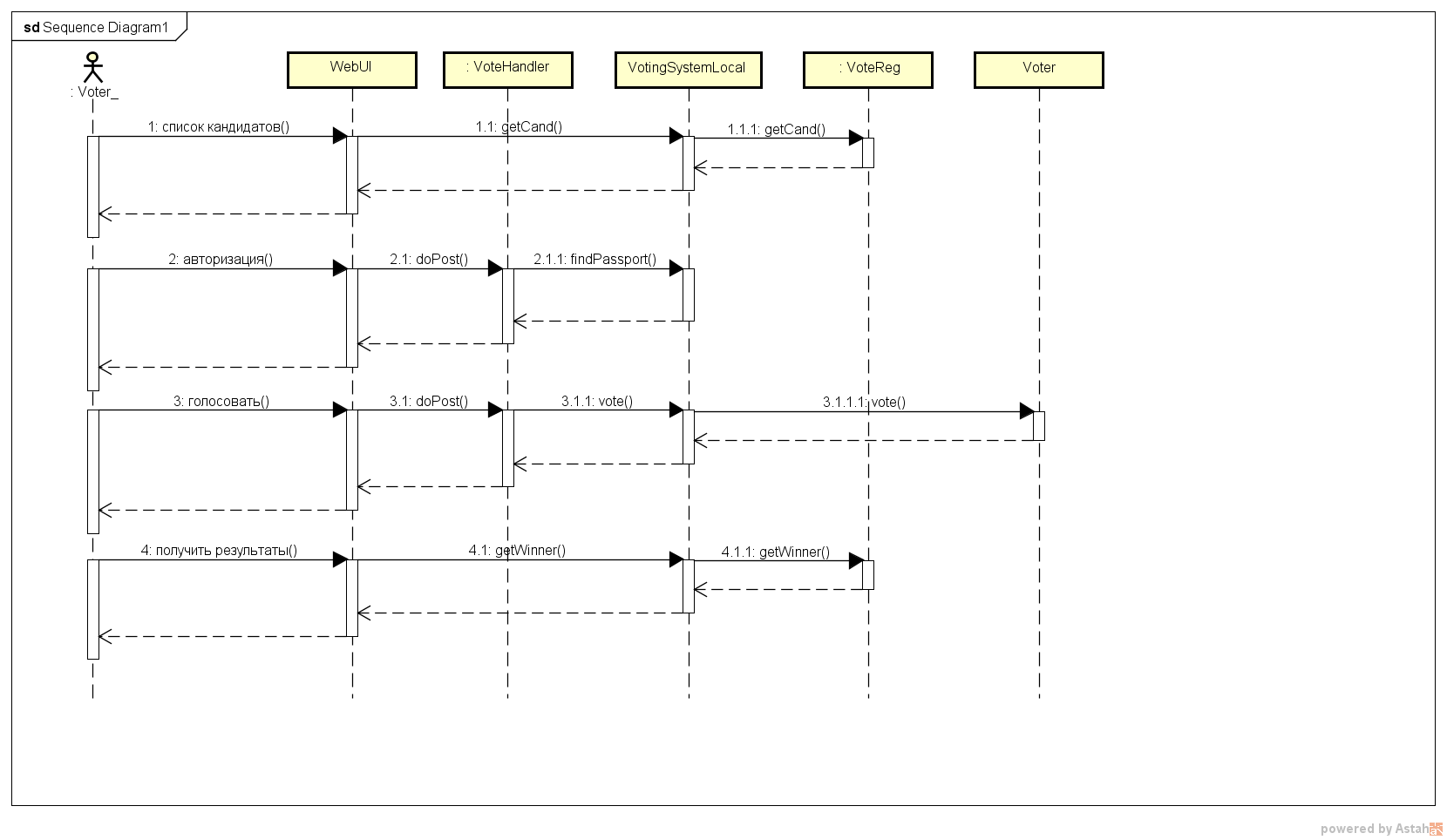
Класс VotingSystem c аннотацией Singleton - гарантирует, что у класса есть только один экземпляр, и предоставляет к нему глобальную точку доступа.

ORM реализован с помощью библиотеки Hibernate с помощью интерфейса JPA (Java Persistence API). Пример обращения к БД (запрос на наличие паспорта в БД):

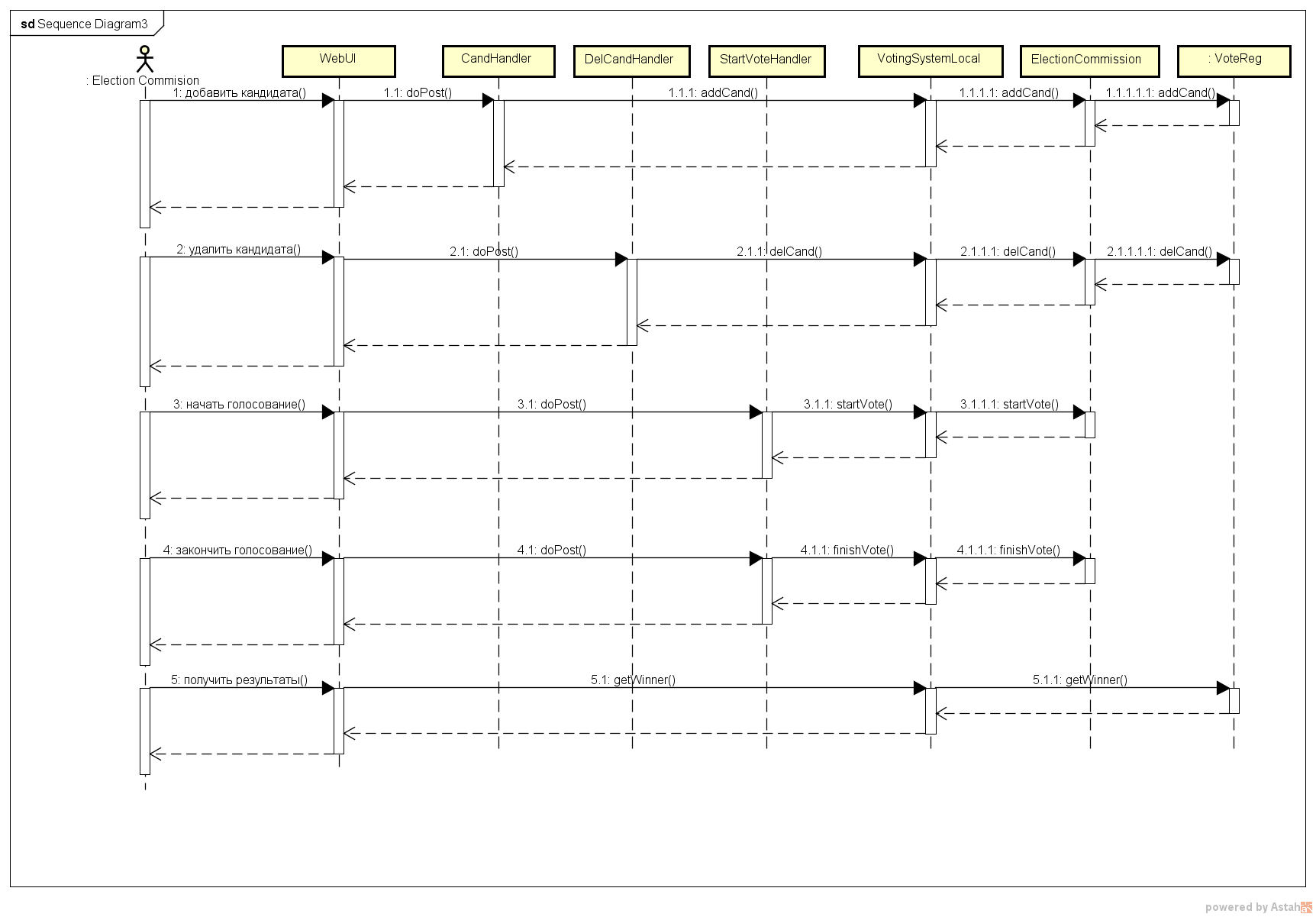
@NamedQuery(name = **"findPassport"**, query = **"SELECT u FROM Passport u WHERE u.number = :number and u.series = :series "**)

**Диаграммы последовательностей**

**Избиратель голосует за кандидата и получает результат**

****

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

****

**Методика и результаты тестирования**

Тестирование системы заключается в проверке работоспособности всех требуемых вариантов использования системы, описанных в разделе «Анализ задания».

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",

|  |  |
| --- | --- |
| Действие | Результат |
| 1. Вводим фамилию и имя 2. Нажимаем на кнопку «Add Candidate» | На страничке избирателя появился новый кандидат |
| C:\Users\Denis\YandexDisk\Скриншоты\2016-12-23_09-54-09.png | C:\Users\Denis\YandexDisk\Скриншоты\2016-12-23_09-55-23.png |

|  |  |
| --- | --- |
| Действие | Результат |
| 1. Вводим фамилию и имя 2. Нажимаем на кнопку «Delete Candidate» | На страничке избирателя исчез кандидат |
| C:\Users\Denis\YandexDisk\Скриншоты\2016-12-23_09-58-13.png | C:\Users\Denis\YandexDisk\Скриншоты\2016-12-23_10-01-25.png |

2.Избирательная комиссия начинает/заканчивает выборы

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

|  |  |
| --- | --- |
| Действие | Результат |
| 1. Нажимаем на кнопку «Start Vote» | 1.Избиратель может голосовать (см. п3). 2. Нельзя добавить/удалить кандидатов (как в п.1) – появляется сообщение «You cant add candidate!» 3. На страничке результатов в поле для победителя – «Wait Finish Vote...» |
| 1. Нажимаем на кнопку «Finish Vote» | 1. 1. Избиратель не может голосовать – появляется сообщение «Vote dont Start!» 2. 2. На страничке результатов в поле для победителя – выводится победитель (или, если равное количество голосов, выводится «ReVote, equals votes». |

3.Избиратель голосует за кандидата

|  |  |
| --- | --- |
| Действие | Результат |
| 1. Вводим паспотные данные 2. Выбираем кандидата, нажимаем кнопку «Vote» | 1.На страничке результатов отразился голос избирателя 2. При неверном вводе паспортных данных будет выведено сообщение «Wrong Passport Data!» |
| C:\Users\Denis\YandexDisk\Скриншоты\2016-12-23_10-24-48.png | C:\Users\Denis\YandexDisk\Скриншоты\2016-12-23_10-25-49.png |

4.Получение результатов голосования (для всех ролей)

|  |  |
| --- | --- |
| Действие | Результат |
| 1. Нажимаем на ссылку «Get Results» | 1.На страничке результатов выводятся кандидаты и соответствующие им голоса 2. В поле «Winner» выводится кандидат-победитель или «Wait Finish Vote...», или «ReVote, equals votes» |

**Инструкция по развертыванию приложения**

Для корректной работы требуется установить следующее ПО:

• Пакет Java Runtime Environment 8

• Сервер приложений WildFly 10

Для сборки проекта необходимо выполнить команду ./gradlew war:

./ gradlew war

clean

compileJava

processResources

classes

war

BUILD SUCCESSFUL

Total time : 0.946 secs

Собранный war-файл доступен по следующему пути: build/libs/Denis.war

После сборки необходимо установить и настроить WildFly. С помощью скрипта

wildfly-10.1.0.Final\bin\add-user.sh нужно добавить пользователя-администратора.

После добавления пользователя нужно запустить сам сервер приложений с помощью скрипта wildfly-10.1.0.Final\bin\standalone.sh

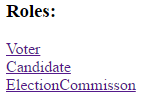
WildFly выведет на консоль адрес страницы для управления сервером приложения, следует зайти на нее. В пункте Deployment выберем пункт Start - далее необходимо нажать на кнопку Add и выбрать war-файл.

По умолчанию веб-сайт доступен по адресу: http://localhost:8080/Denis.

**Инструкция пользователя**

По умолчанию веб-сайт доступен по адресу: http://localhost:8080/Denis.

Главная страница Web приложения:



**Инструкция Избирателя (Voter)**

Если пользователь - избиратель, ему необходимо пройти в раздел Voter по соответствующей ссылке. Для дальнейшей работы пользователю необходимо аутентифицироваться. В верхней части страницы необходимо ввести паспортные данные (номер и серию). После этого выбрать кандидата для голосования и нажать кнопку «Vote».

После голосования пользователь может посмотреть результаты, пройдя по ссылке Get Results.

**Инструкция Кандидата (Candidate)**

Если пользователь - кандидат, ему необходимо пройти в раздел Candidate по соответствующей ссылке. Пользователь может посмотреть результаты, пройдя по ссылке Get Results.

**Инструкция Избирательной комиссии (ElectionCommisson)**

Если пользователь – Избирательная комиссия, ему необходимо пройти в раздел ElectionCommisson по соответствующей ссылке. Пользователь может начать/завершить голосование – для этого нужно нажать на кнопку Start Vote или Finish Vote соответственно. Пользователь может добавлять и удалять кандидатов до начала голосования – для этого требуется ввести в поля Surname и name фамилию и имя кандидата, нажать на кнопку Add Candidate или Delete Candidate. Пользователь может посмотреть результаты, пройдя по ссылке Get Results.

**Выводы**

В данной работе был проведен объектно-ориентированный анализ предметной области для задания, получена диаграмма предметной области и описание вариантов использования. Также выполнено объектно-ориентированное проектирование, создана диаграмма классов и последовательностей при помощи UML в соответствии с полученными ранее диаграммами. Были изучены технологии EJB (Enterprise Java Beans) и JPA (Java Persistence API). Благодаря использованию данных технологий было разработано приложение – система электронного голосования. В качестве сервера приложений использовался WildFly 10. Приложение удовлетворяет требованиям прозрачности, масштабируемости и открытости, во многом благодаря использованию технологий HTML, EJB и JPA.