Университет ИТМО

Кафедра ВТ

Курсовая работа

по ПИП

***Вариант 670***

Выполнил: Беспалов Влад

Татаринов Данил

Группа: P3211

Преподаватель: Алексей Письмак Евгеньевич

Санкт-Петербург

2017

Техническое задание на разработку интернет – сайта «MyBattlefield»

# Общие положения

Предметом разработки является интернет – сайт «MyBattlefield», основанный на игре «Battlefield 3», выпущенный компанией Electronic Arts в 2011 году.

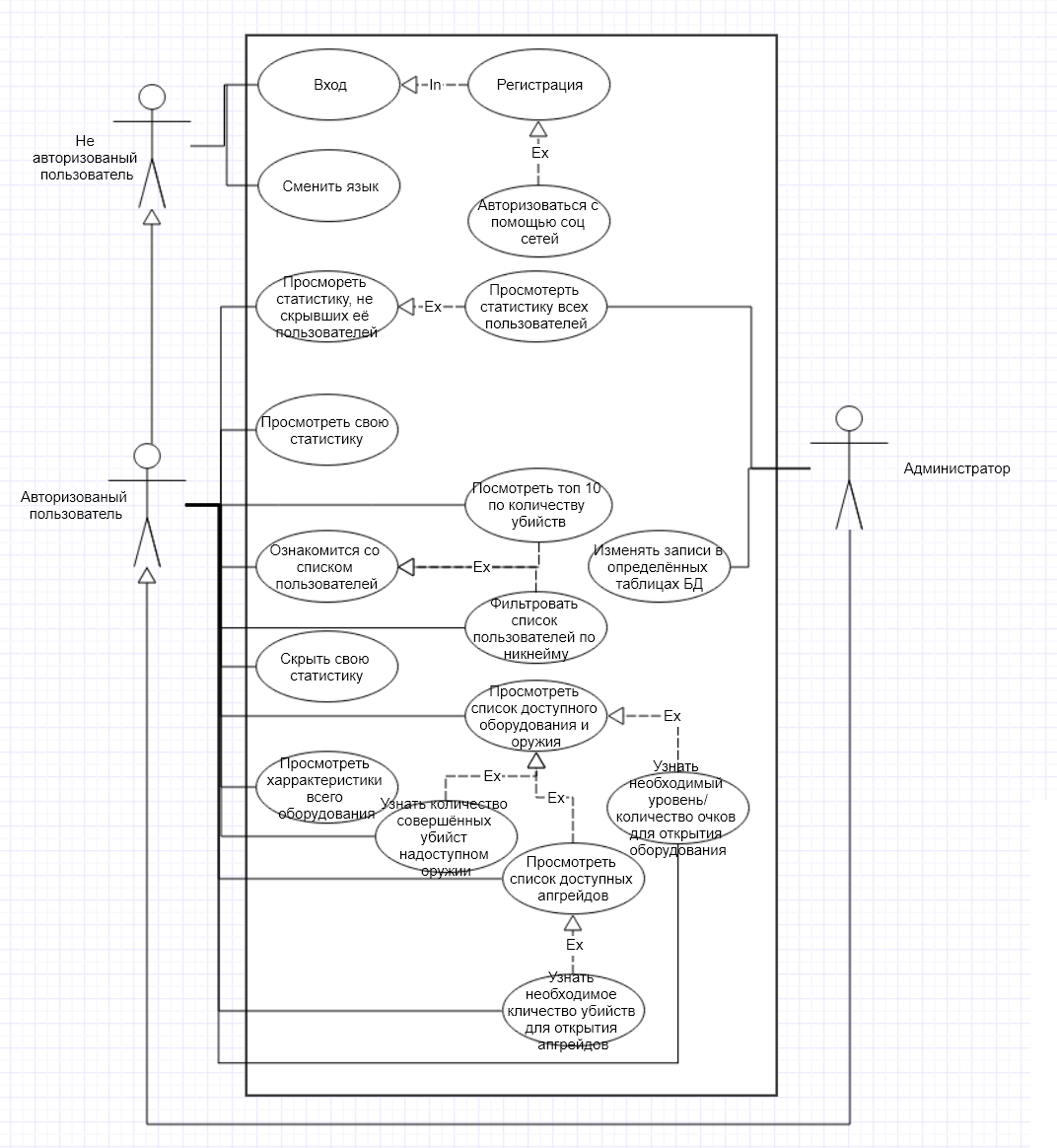
Назначение сайта:

* Размещение статистики пользователя в игре
* Предоставление возможности просмотра статистики других пользователей
* Размещение списка и описания характеристик всего предоставленного в игре оборудования
* Наглядное представление пользователю информации об открытом ему оборудовании и условий открытия остального оборудования

# Требования

|  |  |  |
| --- | --- | --- |
| ID | Описание | Важность |
| 1. Функциональные требования | | |
| 1.1. Требования не авторизованных пользователей | | |
| FR-1 | Приложение должно предоставлять пользователям возможность зарегистрироваться на сайте. | MH |
| FR-2 | Пользователь должен иметь возможность указать свой никнейм и установить пароль для своей учётной записи на сайте. | МH |
| FR-3 | Приложение должно предоставлять пользователю возможность авторизоваться в системе с помощью своего пароля и никнейма. | МH |
| FR-4 | Приложение должно предоставить пользователю возможность авторизоваться в системе с помощью социальных сетей. | SH |
| FR-6 | Приложение должно предоставить пользователю возможность сменить язык интерфейса. 2 доступных: Eng, Рус. | SH |
| 1.2. Требования авторизованных пользователей | | |
| FR-7 | Система должна предоставлять пользователю возможность ознакомиться со списком всех остальных зарегистрированных на сайте пользователей. | SH |
| FR-8 | Система должна предоставлять возможность поиска пользователей по никнейму в списке всех пользователей сайта. | SH |
| FR-9 | Система должна предоставлять пользователю возможность ознакомиться с топ 10 игроков по суммарному количеству убийств, среди не скрывших статистику своего аккаунта. | CH |
| FR-10 | Приложение должно предоставлять пользователю статистику набранного им уровня и очков по каждому из классов (Assault, Support, Engineer, Recon) в удобном интерфейсе. | MH |
| FR-11 | Система должна предоставлять возможность просмотра статистики других пользователей, за исключением тех, которые скрыли статистику своего аккаунта. | SH |
| FR-12 | Система должна предоставлять возможность скрытия своей статистики от других пользователей, за исключением тех, которые имею права администратора. | CH |
| FR-13 | Система должна в наглядном виде демонстрировать характеристики всего существующего в игре оборудования. | MH |
| FR-14 | Система должна предоставлять пользователю список доступных ему камуфляжей/специализаций/гаджетов/оружия в соответствии с набранным им уровнем и очками классов. | MH |
| FR-15 | Система должна предоставлять пользователю возможность просмотра количества совершённых им убийств из открытого им оружия. | MH |
| FR-16 | Система должна предоставлять возможность просмотра доступных пользователю апгрейдов для каждого открытого им вида оружия, в соответствии с количеством, совершённых им убийств посредством этого оружия. | MH |
| FR-18 | Система должна предоставлять пользователю возможность узнать требуемый уровень и количество классовых очков для открытия камуфляжей, специализаций, типов оружия, гаджетов. | MH |
|  | Система должна предоставлять пользователю возможность узнать требуемое количество убийств на открытых им видах оружия для открытия к ним определённых апгрейдов. | MH |
| 1.2.1 Требования пользователей с правами администратора | | |
| FR-20 | Система должна предоставлять администраторам возможность добавлять/изменять/удалять записи в базе данных в таблицах:  CAMOS  SPECIALIZATIONS  WEAPON  GADGETS  COMMON\_WEAPONS  CLASS\_WEPONS  UPGRADES | MH |
| FR-21 | Система должна предоставлять администраторам возможность просмотра статистики скрывших её пользователей. | MH |
| 2. Не функциональные требования | | |
| NFR-1 | Все данные сайта должны храниться в структурированном виде под управлением реляционной СУБД. | MH |
| NFR-2 | Информация о статистике пользователей должна извлекаться из базы данных ресурса bf3stats.com, посредством «Battlefield 3 Stats JSON API». | MH |
| NFR-3 | Уровень back-end должен быть основан на фреймворке Spring. | MH |
| NFR-4 | Уровень front-end должен быть построен на Google Web Toolkit (gwt) с использованием библиотеки Vaadin. | MH |
| NFR-5 | Все веб-интерфейсы системы должны быть адаптированы для отображения в 3 режимах:  "Десктопный" - для устройств, ширина экрана которых равна или превышает 1202 пикселей.  "Планшетный" - для устройств, ширина экрана которых равна или превышает 774, но меньше 1202 пикселей. | MH |

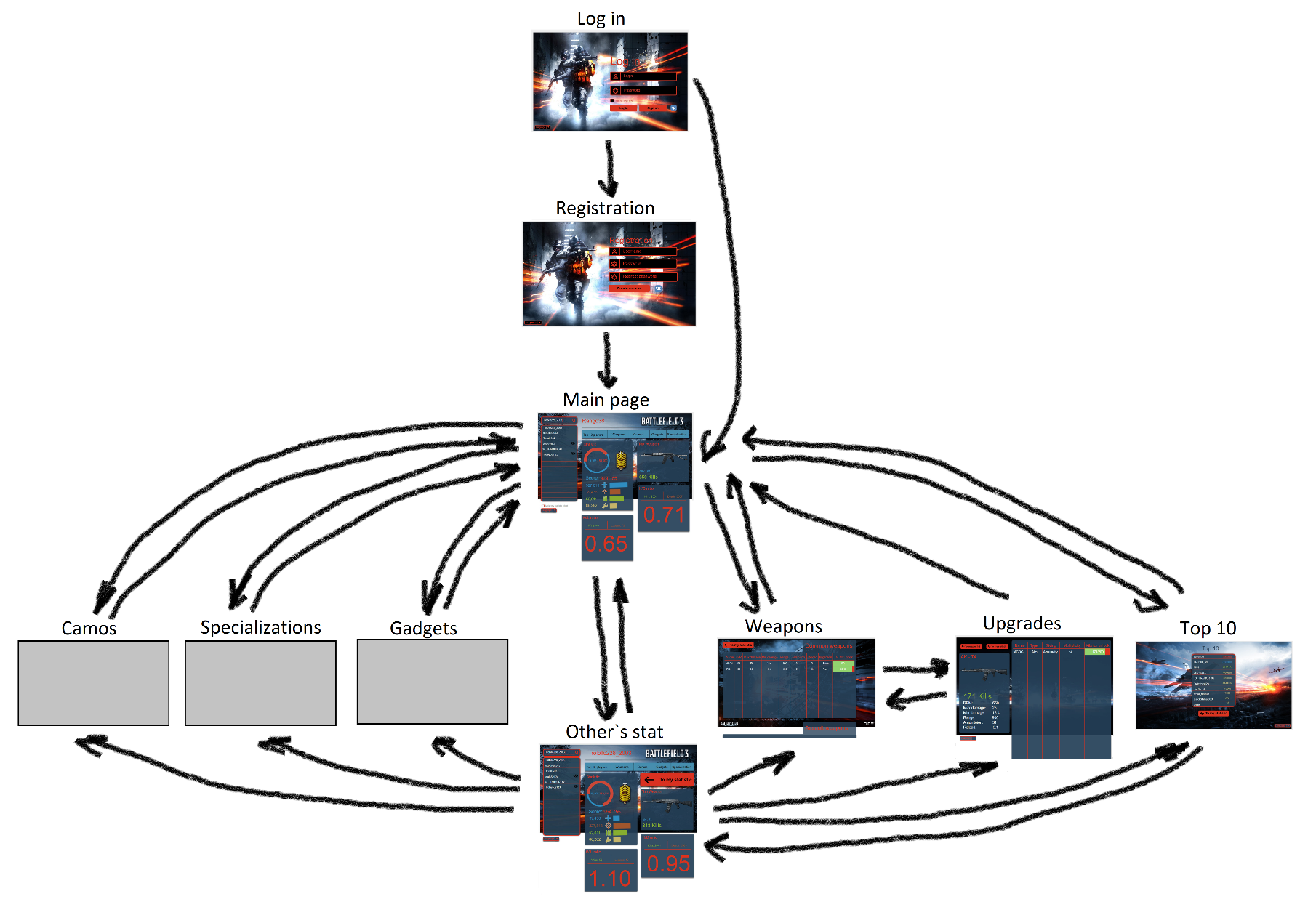
# Use-case диаграмма прецедентов



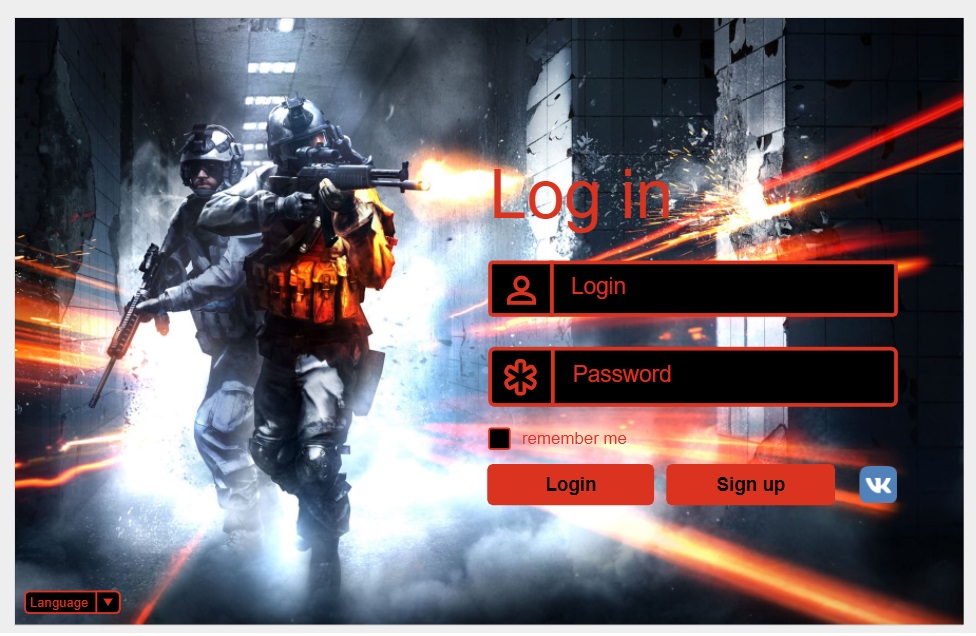
# Требования к интерфейсу сайта

Всего сайт должен содержать в себе 10 страниц:

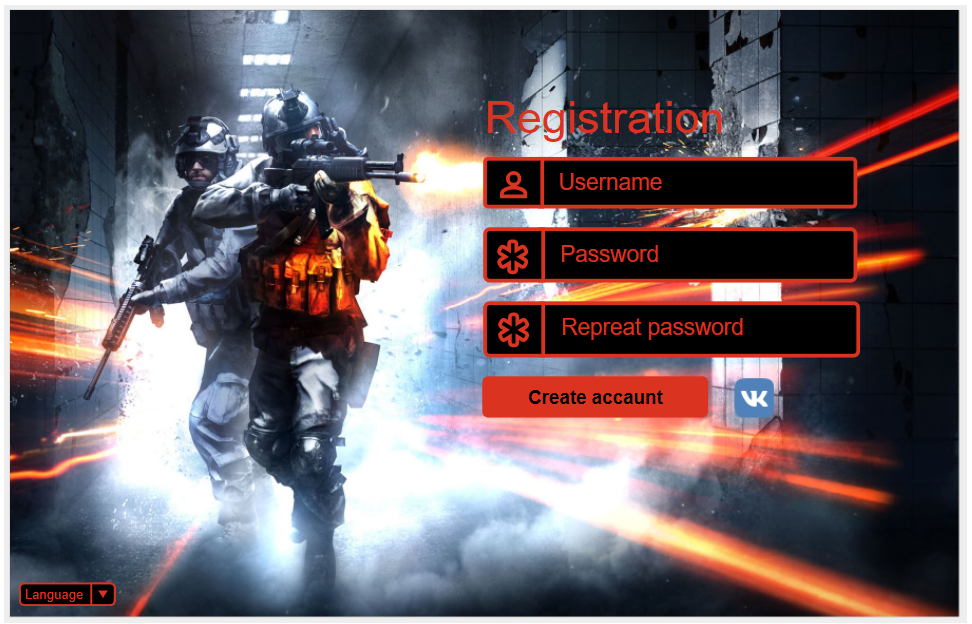
* Страница Log in
* Страница регистрации
* Главная страница сайта со статистикой пользователя
* Главная страница сайта со статистикой другого игрока
* Страница со списком оружия
* Страница со списком апгрейдов для конкретного оружия
* Страница со списком камуфляжей
* Страница со списком специализаций
* Страница со списком Гаджетов
* Страница с топом 10 игроков, зарегистрированных на сайте

Переход между страницами должен будет осуществляться посредством кнопок и элементов интерфейса, расположенных на страницах, в соответствии со следующей схемой.

## Log in

страница log in позволяет залогиниться на сайте, авторизоваться посредством социальной сети «Вконтакте» или перейти к регистрации

## Registration



Страница регистрации позволяет создать аккаунт пользователя на сайте «MyBattlefield» или авторизоваться посредством социальной сети «Вконтакте».

Необходимо указание верного username в соответствии с аккаунтом сети EA.

## Main page

Все элементы главной страницы должны поверх фона. При прокрутке страницы, фон должен оставаться на месте.

На правой части страницы должна располагаться статистика пользователя.

На странице должен располагаться radio button, позволяющий скрыть статистику пользователя от просмотра другими игроками.

На главной странице с правой стороны должен располагаться полный список всех зарегистрированных пользователей с возможностью прокрутки и поиска по нику. При нажатии на Никнейм пользователя, не скрывшего свою статистику должен осуществляться переход на страницу с его статистикой.

В шапке страницы, должны располагаться элементы, позволяющие перейти к страницам оборудования и странице топа игроков.

## Other`s stat

Страница со статистикой других пользователей должна содержать элемент для возврата к просмотру собственной статистики.

## Top 10

Страница с топом 10 лучших игроков должна предоставлять список десяти пользователей с самым большим количество совершённых убийств.

Напротив никнейма каждого игрока должно быть число его убийств.

При нажатии на никнейм должен осуществляться переход к статистике данного пользователя.

На странице должен быть элемент для возвращения на главную страницу.

## Weapons

На странице оружия присутствует кнопка возврата к главной странице и 5 таблиц с собственной прокруткой:

* Common weapons
* Assault weapons
* Support weapons
* Engineer weapons
* Recon weapons

В столбце LVL for unlock отображается уровень необходимый для открытия оружия и с помощью прогресс бара изображено сколько осталось набрать. При нажатии на элемент в стоке происходит переход на страницу апгрейдов данного оружия.

## Upgrades

На странице апгрейдов находится кнопки возврата к странице оружия и главной странице, показывается информация об оружии и демонстрируется лист апгрейдов, где указано необходимое количество убийств для открытия данного апгрейда.

## Camos, specializations, gadgets

Страницы камуфляжей, специализаций, гаджетов полностью повторяют структуру и дизайн страницы оружия.

# Архитектура

Пользователь взаимодействует с системой через браузер, который отправляет серверу запросы и получает ответы.

Сервер взаимодействует с базой данных системы посредством фреймворка Spring.

Данные в базе обновляются с помощью данных из базы bf3stats, используя bf3stats API.

Back-end системы построен на Spring, Front-end построен на GWT.

Уровень доступа к даным

# Пример класса типа Entity

import javax.persistence.\*;

@Entity

@Table(name = "player\_profiles", schema = "public", catalog = "postgres")

public class PlayerProfilesEntity {

private int playerId;

private String nickName;

private Integer lvl;

private Integer assaultPoints;

private Integer engineerPoints;

private Integer supportPoints;

private Integer reconPoints;

@Id

@Column(name = "player\_id")

public int getPlayerId() {

return playerId;

}

public void setPlayerId(int playerId) {

this.playerId = playerId;

}

@Basic

@Column(name = "nick\_name")

public String getNickName() {

return nickName;

}

public void setNickName(String nickName) {

this.nickName = nickName;

}

@Basic

@Column(name = "lvl")

public Integer getLvl() {

return lvl;

}

public void setLvl(Integer lvl) {

this.lvl = lvl;

}

@Basic

@Column(name = "assault\_points")

public Integer getAssaultPoints() {

return assaultPoints;

}

public void setAssaultPoints(Integer assaultPoints) {

this.assaultPoints = assaultPoints;

}

@Basic

@Column(name = "engineer\_points")

public Integer getEngineerPoints() {

return engineerPoints;

}

public void setEngineerPoints(Integer engineerPoints) {

this.engineerPoints = engineerPoints;

}

@Basic

@Column(name = "support\_points")

public Integer getSupportPoints() {

return supportPoints;

}

public void setSupportPoints(Integer supportPoints) {

this.supportPoints = supportPoints;

}

@Basic

@Column(name = "recon\_points")

public Integer getReconPoints() {

return reconPoints;

}

public void setReconPoints(Integer reconPoints) {

this.reconPoints = reconPoints;

}

@Override

public boolean equals(Object o) {

if (this == o) return true;

if (o == null || getClass() != o.getClass()) return false;

PlayerProfilesEntity that = (PlayerProfilesEntity) o;

if (playerId != that.playerId) return false;

if (nickName != null ? !nickName.equals(that.nickName) : that.nickName != null) return false;

if (lvl != null ? !lvl.equals(that.lvl) : that.lvl != null) return false;

if (assaultPoints != null ? !assaultPoints.equals(that.assaultPoints) : that.assaultPoints != null)

return false;

if (engineerPoints != null ? !engineerPoints.equals(that.engineerPoints) : that.engineerPoints != null)

return false;

if (supportPoints != null ? !supportPoints.equals(that.supportPoints) : that.supportPoints != null)

return false;

if (reconPoints != null ? !reconPoints.equals(that.reconPoints) : that.reconPoints != null) return false;

return true;

}

@Override

public int hashCode() {

int result = playerId;

result = 31 \* result + (nickName != null ? nickName.hashCode() : 0);

result = 31 \* result + (lvl != null ? lvl.hashCode() : 0);

result = 31 \* result + (assaultPoints != null ? assaultPoints.hashCode() : 0);

result = 31 \* result + (engineerPoints != null ? engineerPoints.hashCode() : 0);

result = 31 \* result + (supportPoints != null ? supportPoints.hashCode() : 0);

result = 31 \* result + (reconPoints != null ? reconPoints.hashCode() : 0);

return result;

}

@Override

public String toString(){

return String.format(

"Player[playerId=%d, nickName='%s', lvl='%d']",

playerId, nickName, lvl);

}

}

# Пример класса типа EntityRepository

import org.springframework.data.repository.CrudRepository;

import java.util.List;

public interface PlayerProfilesRepository extends CrudRepository<PlayerProfilesEntity,Integer>{

/\*\*

\* Запрос по имени для PlayerProfilesEntity

\* @param nickName искомый никнейм

\* @return первое PlayerProfilesEntity найденное по данному имени

\*/

PlayerProfilesEntity findByNickName(String nickName);

/\*\*

\* Запрос по уровню для UpgradesEntity

\* @param lvl уровень для поиска

\* @return все PlayerProfilesEntity уровень которых равен переданному уровню

\*/

List<PlayerProfilesEntity> findAllByLvl(int lvl);

}