/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package cta;

import java.io.Serializable;

import java.util.Date;

import javax.persistence.Basic;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import javax.persistence.NamedQueries;

import javax.persistence.NamedQuery;

import javax.persistence.Table;

import javax.persistence.Temporal;

import javax.persistence.TemporalType;

import javax.validation.constraints.NotNull;

import javax.validation.constraints.Size;

import javax.xml.bind.annotation.XmlRootElement;

/\*\*

\*

\* @author nafiy

\*/

@Entity

@Table(name = "CREDENTIAL")

@XmlRootElement

@NamedQueries({

@NamedQuery(name = "Credential.findAll", query = "SELECT c FROM Credential c")

, @NamedQuery(name = "Credential.findByCreId", query = "SELECT c FROM Credential c WHERE c.creId = :creId")

, @NamedQuery(name = "Credential.findByCreSignupdate", query = "SELECT c FROM Credential c WHERE c.creSignupdate = :creSignupdate")

, @NamedQuery(name = "Credential.findByCreUsername", query = "SELECT c FROM Credential c WHERE c.creUsername = :creUsername")

, @NamedQuery(name = "Credential.findByCrePasshash", query = "SELECT c FROM Credential c WHERE c.crePasshash = :crePasshash")

, @NamedQuery(name = "Credential.findByCreIdAndUserSurname", query = "SELECT c FROM Credential c WHERE c.creId = :creId AND c.creUserid.userSurname = :userSurname")

, @NamedQuery(name = "Credential.findByCreUserid", query = "SELECT c FROM Credential c WHERE c.creUserid.userId = :creUserid")

, @NamedQuery(name = "Credential.findByCreUsernameAndCrePasshash", query = "SELECT c FROM Credential c WHERE c.creUsername = :creUsername AND c.crePasshash = :crePasshash")})

public class Credential implements Serializable {

private static final long serialVersionUID = 1L;

@Id

@Basic(optional = false)

@NotNull

@Column(name = "CRE\_ID")

private Short creId;

@Column(name = "CRE\_SIGNUPDATE")

@Temporal(TemporalType.DATE)

private Date creSignupdate;

@Size(max = 25)

@Column(name = "CRE\_USERNAME")

private String creUsername;

@Size(max = 60)

@Column(name = "CRE\_PASSHASH")

private String crePasshash;

@JoinColumn(name = "CRE\_USERID", referencedColumnName = "USER\_ID")

@ManyToOne

private Users creUserid;

public Credential() {

}

public Credential(Short creId) {

this.creId = creId;

}

public Short getCreId() {

return creId;

}

public void setCreId(Short creId) {

this.creId = creId;

}

public Date getCreSignupdate() {

return creSignupdate;

}

public void setCreSignupdate(Date creSignupdate) {

this.creSignupdate = creSignupdate;

}

public String getCreUsername() {

return creUsername;

}

public void setCreUsername(String creUsername) {

this.creUsername = creUsername;

}

public String getCrePasshash() {

return crePasshash;

}

public void setCrePasshash(String crePasshash) {

this.crePasshash = crePasshash;

}

public Users getCreUserid() {

return creUserid;

}

public void setCreUserid(Users creUserid) {

this.creUserid = creUserid;

}

@Override

public int hashCode() {

int hash = 0;

hash += (creId != null ? creId.hashCode() : 0);

return hash;

}

@Override

public boolean equals(Object object) {

// TODO: Warning - this method won't work in the case the id fields are not set

if (!(object instanceof Credential)) {

return false;

}

Credential other = (Credential) object;

if ((this.creId == null && other.creId != null) || (this.creId != null && !this.creId.equals(other.creId))) {

return false;

}

return true;

}

@Override

public String toString() {

return "cta.Credential[ creId=" + creId + " ]";

}

}