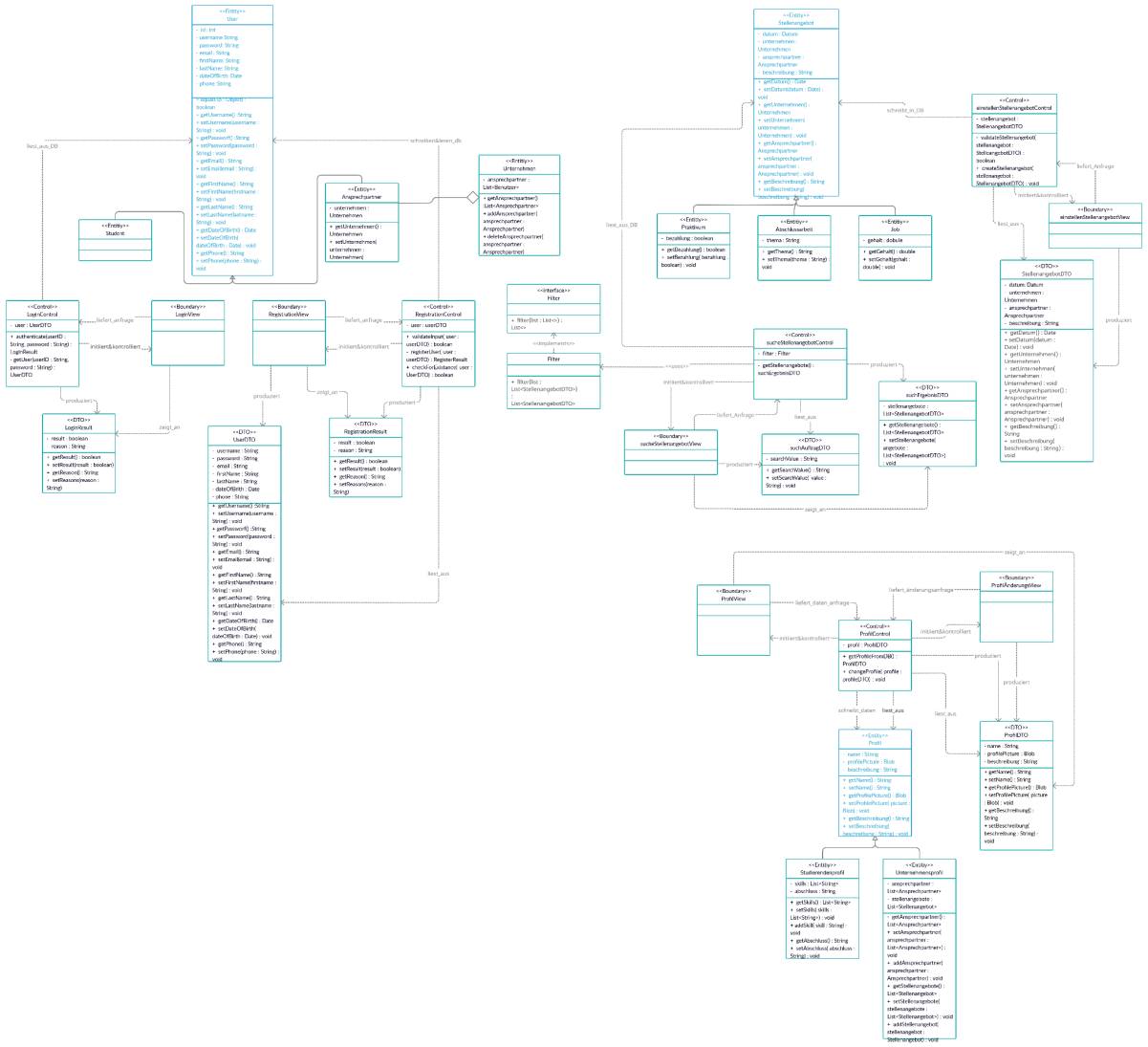
<<Entity>>
User
- id : int
- username:String
- password: String
- email : String
- firstName: String
- lastName: String
- dateOfBirth: Date
- phone: String
+ equals(o : Object) : boolean
+ getUsernameO :String
+ setUsernameusername : String) : void
+ getPasswordP() :String
+ setPasswordpassword : String) : void
+ getEmail() : String
+ setEmailemail : String) : void
+ getFirstName() : String
+ setFirstNameFirstname : String) : void
+ getLastName() : String
+ setLastNameLastname : String) : void
+ getDateOfBirth() : Date
+ setDateOfBirth(dateOfBirth : Date) : void
+ getPhone() : String
+ setPhone(phone : String) : void
liest_aus_DB
schreiben&lesen_db
<<Entity>>
Student
<<Entity>>
Ansprechpartner
- unternehmen : Unternehmen
+ getUnternehmen() : Unternehmen
+ setUnternehmen(unternehmen : Unternehmen)
<<Entity>>
Unternehmen
- ansprechpartner : List<Benutzer>
+ getAnsprechpartner() List<Ansprechpartner>
+ addAnsprechpartner(ansprechpartner : Ansprechpartner)
+ deleteAnsprechpartner(ansprechpartner : Ansprechpartner)
<<Entity>>
Stellenangebot
- datum : Datum
- unternehmen : Unternehmen
- ansprechpartner : Ansprechpartner
- beschreibung : String
+ getDatum() : Date
+ setDatum(datum : Date) : void
+ getUnternehmen() : Unternehmen
+ setUnternehmen(unternehmen : Unternehmen) : void
+ getAnsprechpartner() : Ansprechpartner
+ setAnsprechpartner(ansprechpartner : Ansprechpartner) : void
+ getBeschreibung() : String
+ setBeschreibung(beschreibung : String) : void
schreibt_in_DB
<<Control>>
einstellbarStellenangebotControl
- stellenangebot : StellenangebotDTO
- validateStellenangebot(stellenangebot : StellenangebotDTO) : boolean
- createStellenangebot(stellenangebot : StellenangebotDTO) : void
liefert_Anfrage
initiiert&kontrolliert
<<Boundary>>
einstellbarStellenangebotView
liest_aus_DB
<<Entity>>
Praktikum
- bezahlung : boolean
+ getBezahlung() : boolean
+ setBezahlung(bezahlung : boolean) : void
<<Entity>>
Abschlussarbeit
- thema: String
+ getThema() : String
+ setThema(thema : String) : void
<<Entity>>
Job
- gehalt : double
+ getGehalt() : double
+ setGehalt(gehalt : double) : void
liest_aus
<<DTO>>
StellenangebotDTO
- datum: Datum
- unternehmen : Unternehmen
- ansprechpartner : Ansprechpartner
- beschreibung : String
+ getDatum() : Date
+ setDatum(datum : Date) : void
+ getUnternehmen() : Unternehmen
+ setUnternehmen(unternehmen : Unternehmen) : void
+ getAnsprechpartner() : Ansprechpartner
+ setAnsprechpartner(ansprechpartner : Ansprechpartner) : void
+ getBeschreibung() : String
+ setBeschreibung(beschreibung : String) : void
produziert
<<Control>>
LoginControl
- user : UserDTO
+ authenticate(userID : String, password : String) : LoginResult
- getUserJ(userID : String, password : String) : UserDTO
liefert_anfrage
initiiert&kontrolliert
<<Boundary>>
LoginView
<<Boundary>>
RegistrationView
liefert_anfrage
initiiert&kontrolliert
<<Control>>
RegistrationControl
- user : userDTO
+ validateInput(user : userDTO) : boolean
- registerUser(user : user DTO) : RegisterResult
+ checkForExistance(user : UserDTO) : boolean
produziert
zeigt_an
produziert
<<DTO>>
LoginResult
- result : boolean
- reason : String
+ getResult() : boolean
+ setResult(result : boolean)
+ getReason() : String
+ setReason(reason : String)
zeigt_an
<<DTO>>
UserDTO
- username : String
- password : String
- email : String
- firstName : String
- lastName : String
- dateOfBirth : Date
- phone : String
+ getUsername() :String
+ setUsernameusername : String) : void
+ getPassword() :String
+ setPassword(password : String) : void
+ getEmail() : String
+ setEmailemail : String) : void
+ getFirstName() : String
+ setFirstNameFirstname : String) : void
+ getLastName() : String
+ setLastNameLastname : String) : void
+ getDateOfBirth() : Date
+ setDateOfBirth(dateOfBirth : Date) : void
+ getPhone() : String
+ setPhone(phone : String) : void
<<DTO>>
RegistrationResult
- result : boolean
- reason : String
+ getResult() : boolean
+ setResult(result : boolean)
+ getReason() : String
+ setReason(reason : String)
liest_aus
<<interface>>
Filter
+ filter(list : List<>) : List<>
<<implementiert>>
Filter
+ filter(list : List<Stellenangebot DTO>) : List<Stellenangebot DTO>
<<uses>>
<<Control>>
sucheStellenangebotControl
- filter : Filter
- getStellenangebote() : suchErgebnisDTO
produziert
initiiert&kontrolliert
liefert_Anfrage
liest_aus
<<DTO>>
suchFilterDTO
- vielleicht : List<Stellenangebot DTO>
+ getVielleicht() : List<Stellenangebot DTO>
+ setVielleicht(vielleicht : List<Stellenangebot DTO>) : void
<<Boundary>>
sucheStellenangebotView
produziert
<<DTO>>
suchAufrufDTO
- searchValue : String
+ getSearchValue() : String
+ setSearchValue(value : String) : void
zeigt_an

zeigt_an
<<Boundary>>
ProfilView
liefert_daten_anfrage
liefert_änderungsanfrage
<<Boundary>>
ProfilÄnderungsView
initiiert&kontrolliert
initiiert&kontrolliert
<<Control>>
ProfilControl
- profil : ProfilDTO
+ getProfileFromDB() : ProfilDTO
+ changeProfile(profile : profilDTO) : void
produziert
produziert
schreibt_daten
liest_aus
liest_aus
<<Entity>>
Profil
- name : String
- profilePicture : Blob
- beschreibung : String
+ getName() : String
+ setName() : String
+ getProfilePicture() : Blob
+ setProfilePicture(picture : Blob) : void
+ getBeschreibung() : String
+ setBeschreibung(beschreibung : String) : void
<<DTO>>
ProfilDTO
- name : String
- profilePicture : Blob
- beschreibung : String
+ getName() : String
+ setName() : String
+ getProfilePicture() : Blob
+ setProfilePicture(picture : Blob) : void
+ getBeschreibung() : String
+ setBeschreibung(beschreibung : String) : void
<<Entity>>
Studierendenprofil
- skills : List<String>
- abschluss : String
+ getSkills() : List<String>
+ setSkills(skills : List<String>) : void
+ addSkill(skill : String) : void
+ getAbschluss() : String
+ setAbschluss(abschluss : String) : void
<<Entity>>
Unternehmensprofil
- ansprechpartner : List<Ansprechpartner>
- stellenangebote : List<Stellenangebot>
+ getAnsprechpartner() : List<Ansprechpartner>
+ setAnsprechpartner(ansprechpartner : List<Ansprechpartner>) : void
+ addAnsprechpartner(ansprechpartner : Ansprechpartner) : void
+ getStellenangebote() : List<Stellenangebot>
+ setStellenangebote(stellenangebote : List<Stellenangebot>) : void
+ addStellenangebot(stellenangebot : Stellenangebot) : void

UserDTO
User
StellenausschreibungDTO
RecruitmentAdvertisingDTO
CompanyDTO
JobAdvertisement
MessageDTO
UserDTOImpl
StudentUserDTO
ContactPersonDTO
ContactPerson
StudentUser
StellenausschreibungDTOImpl
RecruitmentAdvertisingDTOImpl
CompanyDTOImpl
MessageDTOImpl
StudentUserDTOImpl
ContactPersonDTOImpl
AppView
Company
RegistrationDTO
Message
RegistrationResultDTO
Address
StudentProfileEditView
RegistrationControl
RegistrationDTOImpl
RegistrationResultDTOImpl
ReasonType
Utils
LoginResultDTO
RecruitmentAdvertisementFormularView
CompanyProfileEditView
InboxControl
RegistrationView
RoleDTO
InboxView
LoginResultDTOImpl
OS
CompanyRegisterForm
RegisterForm
RoleDTOImpl
CompanyProfileView
Globals
MessageRepository
UserFactory
AddressRepository
PageTitles
Pages
Errors
Roles
Countries
ContactPersonControl
StudentUserControl
ContactingControl
StudentProfileView
CompanyRepository
JobAdvertisementRepository
CompanyControl
LoginControl
ContactingView
LoginView
UserRepository
DatabaseUserFunction
JobFactory
RecruitmentAdvertisementControl
MainLayout
LayoutAlternative
CompanyFactory
StudentUserRepository
AddressControl
MessageFactory
MainView
CoiApplication
RecruitmentAdvertisementView
RegistrationSuccessfulView
JobAdvertisementView
DataProtectionView
ImpressumView
ContactView