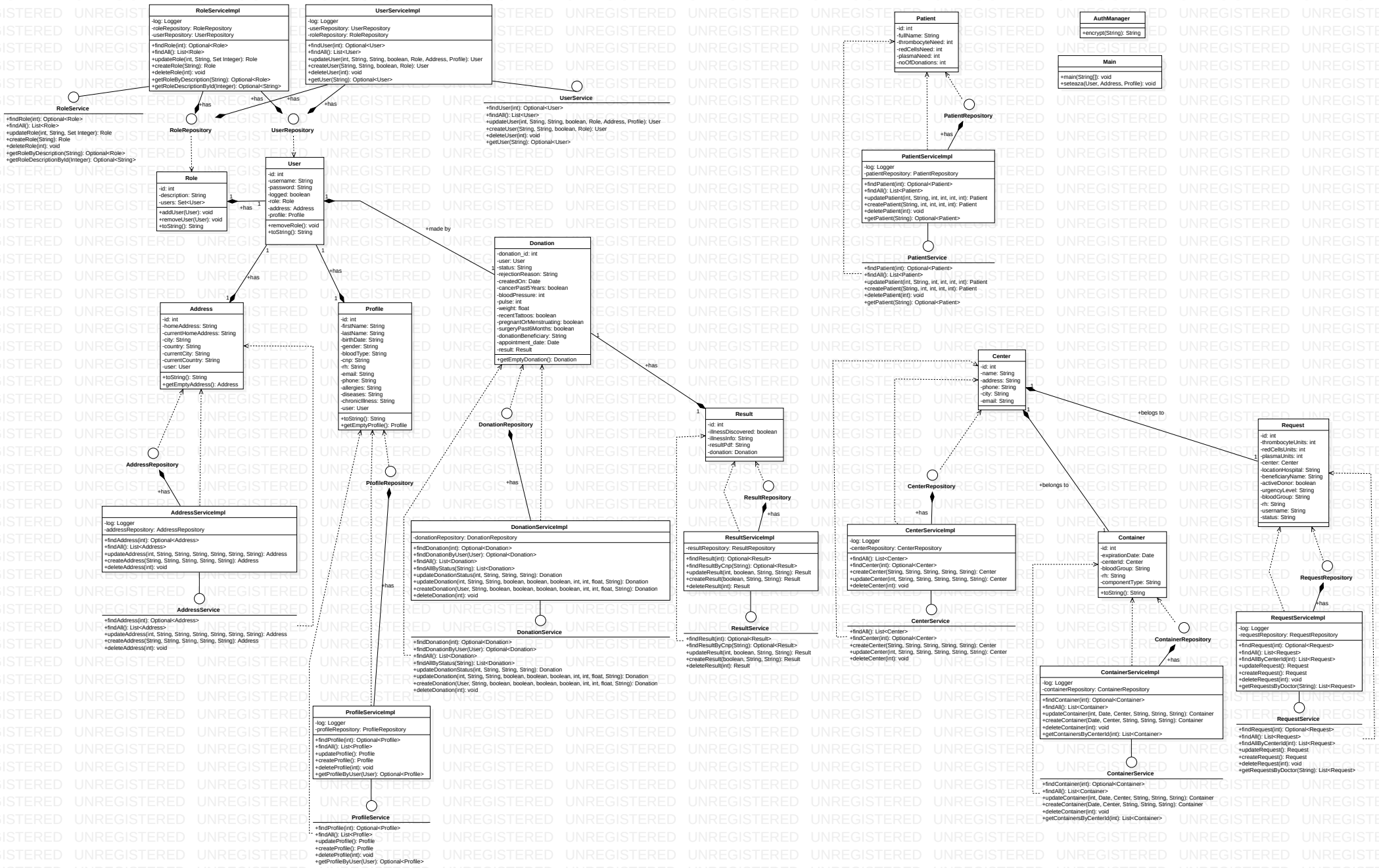
Model Main


RoleServiceImpl
-log: Logger
-roleRepository: RoleRepository
-userRepository: UserRepository
+findRole(int): Optional<Role>
+findAll(): List<Role>
+updateRole(int, String, Set Integer): Role
+createRole(String): Role
+deleteRole(int): void
+getRoleByDescription(String): Optional<Role>
+getRoleDescriptionById(Integer): Optional<String>
UserServiceImpl
-log: Logger
-userRepository: UserRepository
-roleRepository: RoleRepository
+findUser(int): Optional<User>
+findAll(): List<User>
+updateUser(int, String, String, boolean, Role, Address, Profile): User
+createUser(String, String, boolean, Role): User
+deleteUser(int): void
+getUser(String): Optional<User>
RoleService
+findRole(int): Optional<Role>
+findAll(): List<Role>
+updateRole(int, String, Set Integer): Role
+createRole(String): Role
+deleteRole(int): void
+getRoleByDescription(String): Optional<Role>
+getRoleDescriptionById(Integer): Optional<String>
UserService
+findUser(int): Optional<User>
+findAll(): List<User>
+updateUser(int, String, String, boolean, Role, Address, Profile): User
+createUser(String, String, boolean, Role): User
+deleteUser(int): void
+getUser(String): Optional<User>
+has
RoleRepository
UserRepository
Role
-id: int
-description: String
-users: Set<User>
+addUser(User): void
+removeUser(User): void
+toString(): String
User
-id: int
-username: String
-password: String
-logged: boolean
-role: Role
-address: Address
-profile: Profile
+removeRole(): void
+toString(): String
+made by
Address
-id: int
-homeAddress: String
-currentHomeAddress: String
-city: String
-country: String
-currentCity: String
-currentCountry: String
+toString(): String
+getEmptyAddress(): Address
Profile
-id: int
-firstName: String
-lastName: String
-birthDate: String
-gender: String
-bloodType: String
-cnp: String
-rh: String
-email: String
-phone: String
-allergies: String
-diseases: String
-chronicIllness: String
-user: User
+toString(): String
+getEmptyProfile(): Profile
Donation
-donation_id: int
-user: User
-status: String
-rejectionReason: String
-createdOn: Date
-cancerPast5Years: boolean
-bloodPressure: int
-pulse: int
-weight: float
-recentTattoos: boolean
-pregnantOrMenstruating: boolean
-surgeryPast6Months: boolean
-donationBeneficiary: String
-appointment_date: Date
-result: Result
+getEmptyDonation(): Donation
Result
-id: int
-illnessDiscovered: boolean
-illnessInfo: String
-resultPdf: String
-donation: Donation
AddressRepository
ProfileRepository
DonationRepository
ResultRepository
AddressServiceImpl
-log: Logger
-addressRepository: AddressRepository
+findAddress(int): Optional<Address>
+findAll(): List<Address>
+updateAddress(int, String, String, String, String, String, String): Address
+createAddress(String, String, String, String, String): Address
+deleteAddress(int): void
AddressService
+findAddress(int): Optional<Address>
+findAll(): List<Address>
+updateAddress(int, String, String, String, String, String, String): Address
+createAddress(String, String, String, String, String): Address
+deleteAddress(int): void
DonationServiceImpl
-donationRepository: DonationRepository
+findDonation(int): Optional<Donation>
+findDonationByUser(User): Optional<Donation>
+findAll(): List<Donation>
+findAllByStatus(String): List<Donation>
+updateDonationStatus(int, String, String, String): Donation
+updateDonation(int, String, String, boolean, boolean, boolean, int, int, float, String): Donation
+createDonation(User, String, boolean, boolean, boolean, boolean, int, int, float, String): Donation
+deleteDonation(int): void
DonationService
+findDonation(int): Optional<Donation>
+findDonationByUser(User): Optional<Donation>
+findAll(): List<Donation>
+findAllByStatus(String): List<Donation>
+updateDonationStatus(int, String, String, String): Donation
+updateDonation(int, String, String, boolean, boolean, boolean, int, int, float, String): Donation
+createDonation(User, String, boolean, boolean, boolean, boolean, int, int, float, String): Donation
+deleteDonation(int): void
ResultServiceImpl
-resultRepository: ResultRepository
+findResult(int): Optional<Result>
+findResultByCnp(String): Optional<Result>
+updateResult(int, boolean, String, String): Result
+createResult(boolean, String, String): Result
+deleteResult(int): Result
ResultService
+findResult(int): Optional<Result>
+findResultByCnp(String): Optional<Result>
+updateResult(int, boolean, String, String): Result
+createResult(boolean, String, String): Result
+deleteResult(int): Result
ProfileServiceImpl
-log: Logger
-profileRepository: ProfileRepository
+findProfile(int): Optional<Profile>
+findAll(): List<Profile>
+updateProfile(): Profile
+createProfile(): Profile
+deleteProfile(int): void
+getProfileByUser(User): Optional<Profile>
ProfileService
+findProfile(int): Optional<Profile>
+findAll(): List<Profile>
+updateProfile(): Profile
+createProfile(): Profile
+deleteProfile(int): void
+getProfileByUser(User): Optional<Profile>
Patient
-id: int
-fullName: String
-thrombocyteNeed: int
-redCellsNeed: int
-plasmaNeed: int
-noOfDonations: int
AuthManager
+encrypt(String): String
Main
+main(String[]): void
+seteaza(User, Address, Profile): void
PatientRepository
PatientServiceImpl
-log: Logger
-patientRepository: PatientRepository
+findPatient(int): Optional<Patient>
+findAll(): List<Patient>
+updatePatient(int, String, int, int, int, int): Patient
+createPatient(String, int, int, int, int): Patient
+deletePatient(int): void
+getPatient(String): Optional<Patient>
PatientService
+findPatient(int): Optional<Patient>
+findAll(): List<Patient>
+updatePatient(int, String, int, int, int, int): Patient
+createPatient(String, int, int, int, int): Patient
+deletePatient(int): void
+getPatient(String): Optional<Patient>
Center
-id: int
-name: String
-address: String
-phoning
-city: String
-email: String
+belongs to
Request
-id: int
-thrombocyteUnits: int
-redCellsUnits: int
-plasmaUnits: int
-center: Center
-locationHospital: String
-beneficiaryName: String
-activeDonor: boolean
-urgencyLevel: String
-bloodGroup: String
-rh: String
-username: String
-status: String
CenterRepository
RequestRepository
CenterServiceImpl
-log: Logger
-centerRepository: CenterRepository
+findAll(): List<Center>
+findCenter(int): Optional<Center>
+createCenter(String, String, String, String, String): Center
+updateCenter(int, String, String, String, String, String): Center
+deleteCenter(int): void
CenterService
+findAll(): List<Center>
+findCenter(int): Optional<Center>
+createCenter(String, String, String, String, String): Center
+updateCenter(int, String, String, String, String, String): Center
+deleteCenter(int): void
Container
-id: int
-expirationDate: Date
-centerId: Center
-bloodGroup: String
-rh: String
-componentType: String
+toString(): String
ContainerRepository
ContainerServiceImpl
-log: Logger
-containerRepository: ContainerRepository
+findContainer(int): Optional<Container>
+findAll(): List<Container>
+updateContainer(int, Date, Center, String, String, String): Container
+createContainer(Date, Center, String, String, String): Container
+deleteContainer(int): void
+getContainersByCenterId(int): List<Container>
ContainerService
+findContainer(int): Optional<Container>
+findAll(): List<Container>
+updateContainer(int, Date, Center, String, String, String): Container
+createContainer(Date, Center, String, String, String): Container
+deleteContainer(int): void
+getContainersByCenterId(int): List<Container>
RequestServiceImpl
-log: Logger
-requestRepository: RequestRepository
+findRequest(int): Optional<Request>
+findAll(): List<Request>
+findAllByCenterId(int): List<Request>
+updateRequest(): Request
+createRequest(): Request
+deleteRequest(int): void
+getRequestsByDoctor(String): List<Request>
RequestService
+findRequest(int): Optional<Request>
+findAll(): List<Request>
+findAllByCenterId(int): List<Request>
+updateRequest(): Request
+createRequest(): Request
+deleteRequest(int): void
+getRequestsByDoctor(String): List<Request>