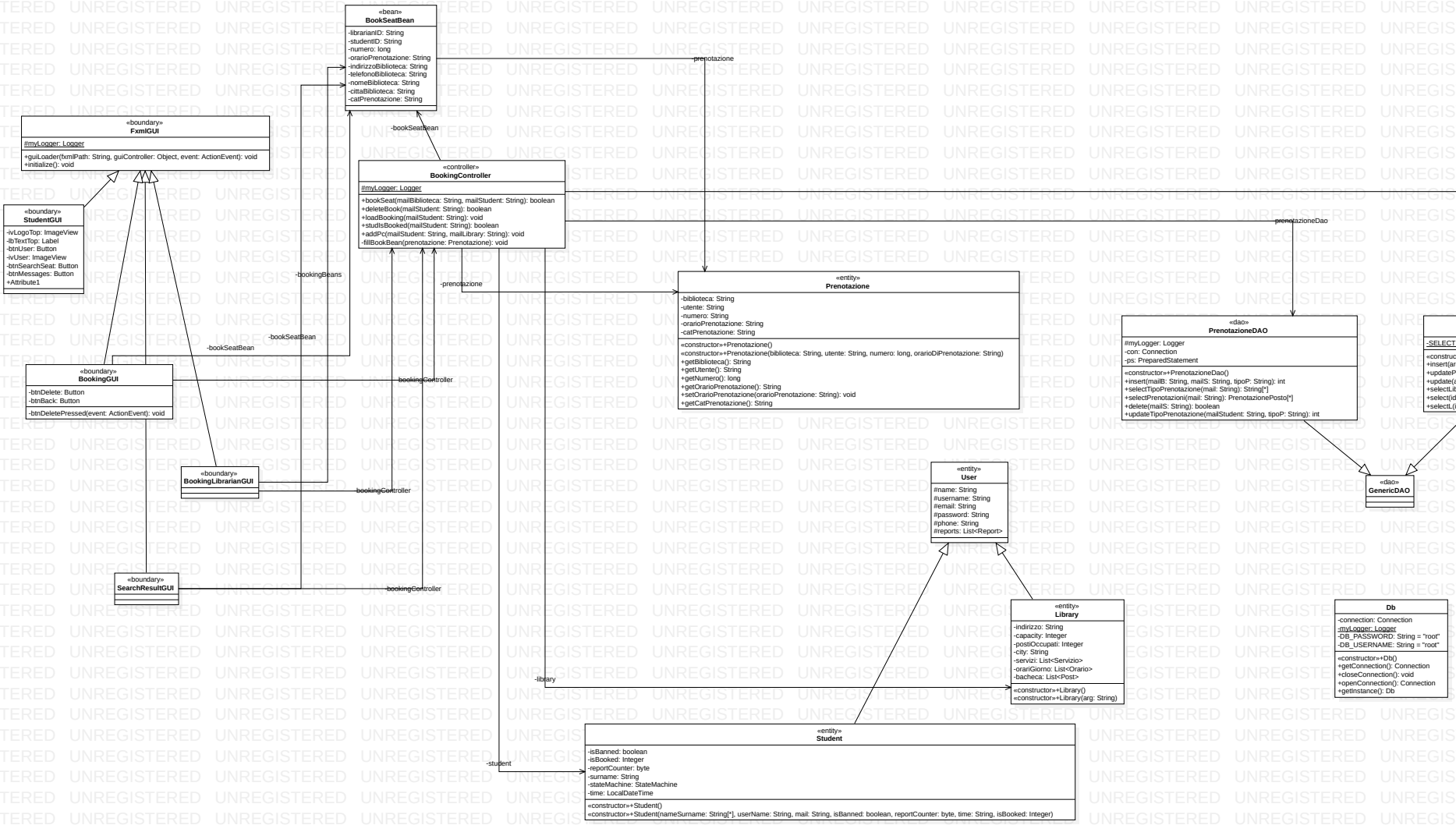
Model.mdj
«bean»
BookSeatBean
-librarianID: String
-studentID: String
-numero: long
-orarioPrenotazione: String
-indirizzoBiblioteca: String
-telefonoBiblioteca: String
-nomeBiblioteca: String
-cittaBiblioteca: String
-catPrenotazione: String
«boundary»
FxmlGUI
#myLogger: Logger
+guiLoader(fxmlPath: String, guiController: Object, event: ActionEvent): void
+initialize(): void
«boundary»
StudentGUI
-lvLogoTop: ImageView
-lbTextTop: Label
-btnUser: Button
-lvUser: ImageView
-btnSearchSeat: Button
-btnMessages: Button
+Attribute1
«controller»
BookingController
#myLogger: Logger
+bookSeat(mailBiblioteca: String, mailStudent: String): boolean
+deleteBook(mailStudent: String): boolean
+loadBooking(mailStudent: String): void
+studIsBooked(mailStudent: String): boolean
+addPc(mailStudent: String, mailLibrary: String): void
-fillBookBean(prenotazione: Prenotazione): void
«boundary»
BookingGUI
-btnDelete: Button
-btnBack: Button
-btnDeletePressed(event: ActionEvent): void
«boundary»
BookingLibrarianGUI
«boundary»
SearchResultGUI
«entity»
Prenotazione
-biblioteca: String
-utente: String
-numero: String
-orarioPrenotazione: String
-catPrenotazione: String
«constructor»+Prenotazione()
«constructor»+Prenotazione(biblioteca: String, utente: String, numero: long, orarioDiPrenotazione: String)
+getBiblioteca(): String
+getUtente(): String
+getNumero(): long
+getOrarioPrenotazione(): String
+setOrarioPrenotazione(orarioPrenotazione: String): void
+getCatPrenotazione(): String
«dao»
PrenotazioneDAO
#myLogger: Logger
-con: Connection
-ps: PreparedStatement
«constructor»+PrenotazioneDao()
+insert(mailB: String, mailS: String, tipoP: String): int
+selectTipoPrenotazione(mail: String): String[*]
+selectPrenotazioni(mail: String): PrenotazionePosto[*]
+delete(mailS: String): boolean
+updateTipoPrenotazione(mailStudent: String, tipoP: String): int
«dao»
LibraryDAO
-SELECT_FAILED: String = "Select biblioteca fallito."
«constructor»+LibraryDao()
+insert(arg: String): int
+updatePost(post: int, mail: String, tipo: String): int
+update(attr: String, newValue: String, entityId: String): int
+selectLibraryFromId(id: String): Library
+select(id1: String, id2: String): Library[*]
+selectL(id1: String, id2: String): Library
«dao»
GenericDAO
«entity»
User
#name: String
#username: String
#email: String
#password: String
#phone: String
#reports: List<Report>
«entity»
Library
-indirizzo: String
-capacity: Integer
-postiOccupati: Integer
-city: String
-servizi: List<Servizio>
-orariGiorno: List<Orario>
-bacheca: List<Post>
«constructor»+Library()
«constructor»+Library(arg: String)
Db
-connection: Connection
-myLogger: Logger
-DB_PASSWORD: String = "root"
-DB_USERNAME: String = "root"
«constructor»+Db()
+getConnection(): Connection
+closeConnection(): void
+openConnection(): Connection
+getInstance(): Db
«entity»
Student
-isBanned: boolean
-isBooked: Integer
-reportCounter: byte
-surname: String
-stateMachine: StateMachine
-time: LocalDateTime
«constructor»+Student()
«constructor»+Student(nameSurname: String[*], userName: String, mail: String, isBanned: boolean, reportCounter: byte, time: String, isBooked: Integer)
-prenotazione
-bookSeatBean
-libraryDao
-prenotazioneDao
-bookingBeans
-prenotazione
-bookSeatBean
-bookSeatBean
-bookingController
-bookingController
-bookingController
-library
-student