

The UML diagrams for the 'Model' layer are organized into several packages: domain, repository, service, and GUI. The domain package contains entities like Room, Session, Proposal, PaperRecommendation, UserSection, Review, Message, User, and Conference. The repository package contains interfaces and implementations for ProposalRepository, ReviewRepository, PCMemberProposalRepository, UserConferenceRepository, and ConferenceRepository. The service package contains the Service class. The GUI package contains LoginView, CreateAccountView, AdminView, and UserView. There are also enumeration types for Availability, ReviewResult, and Role, and an exception hierarchy for ConferenceException and RuntimeException.

```

classDiagram
    package domain {
        class Room {
            +id: int
            +name: string
            +capacity: string
        }
        class Session {
            +sessionid: int
            +confereceid: int
            +topic: string
            +participantsLimit: int
            +roomid: int
        }
        class Proposal {
            +id: int
            +userConferenceid: int
            +abstractText: string
            +paperText: string
            +title: string
            +authors: string
            +keywords: string
            +accepted: boolean
            +fullPaperLocation: string
            +attributes
        }
        class PaperRecommendation {
            +id: int
            +reviewerid: int
            +proposalid: int
            +recommendation: string
        }
        class UserSection {
            +userid: int
            +sectionid: int
        }
        class Review {
            +pcMemberid: int
            +proposalid: int
            +reviewResult: ReviewResult
        }
        class Message {
            +text: string
            +proposalid: int
            +userid: int
            +username: string
        }
        class User {
            +id: Integer
            +name: String
            +password: String
            +email: String
            +fullname: string
            +affiliation: string
            +personalWebsite: string
            +domainOfInterest: string
        }
        class Conference {
            +id: Integer
            +name: String
            +date: Date
            +startDatePrice: Integer
            +submitPaperDeadline: Date
            +reviewPaperDeadline: Date
            +holdingPhaseDeadline: Date
            +withFullPaper: bool
        }
        class UserConference {
            +id: int
            +userid: Integer
            +conferenceid: Integer
            +role: Role
            +paid: Boolean
        }
        class Role {
            AUTHOR
            CHAIR
            CO_CHAIR
            REVIEWER
            SC_MEMBER
            SESSION_CHAIR
            LISTENER
            SPEAKER
        }
    }

    package repository {
        class ProposalRepository {
            +url: String
            +db_user: String
            +db_password: String
            +addProposal(proposal: Proposal)
            +updateProposal(proposal: Proposal)
            +getProposal(): List<Proposal>
            +getProposalsForPCMember(pcMemberid: int, conferenceid: int): List<Proposal>
            +getProposalsOfUser(userid: int): List<Proposal>
        }
        class ReviewRepository {
            +url: String
            +db_user: String
            +db_password: String
            +addPair(review: Review)
            +getAll(): List<Review>
        }
        class PCMemberProposalRepository {
            +url: String
            +db_user: String
            +db_password: String
            +addPair(): PCMemberProposal
        }
        class UserConferenceRepository {
            +url: String
            +db_user: String
            +db_password: String
            +getConferencesOfUser(userid: int): List<UserConference>
            +findUserByid(id: int): User
            +getUsers(): List<User>
            +addUser(user: User): void
        }
        class ConferenceRepository {
            +url: String
            +db_user: String
            +db_password: String
            +findConferencesByid(id: int): Conference
            +getConferences(): List<Conference>
            +addConference(conference: Conference): void
        }
    }

    package service {
        class Service {
            +userRepository: UserRepository
            +conferenceRepository: ConferenceRepository
            +userConferenceRepository: UserConferenceRepository
            +reviewRepository: ReviewRepository
            +pcMemberProposalRepository: PCMemberProposalRepository
            +addUserbyid(id: int): User
            +addUser(name: String, password: String, email: String): void
            +addUserbyid(id: int): User
            +findUserbyid(id: int): User
            +getConferences(): List<Conference>
            +addConference(name: String, date: Date, price: int): void
            +getConferencesOfUser(userid: int): List<UserConference>
            +getUsersOfConference(id: int): List<UserConference>
            +addUserToConference(id: int, conf: int, role: Role, paid: Boolean): void
            +usersWithnameAndPassword(username: String, password: String): List<User>
            +getUserByEmail(email: String): User
            +getUserName(username: String): String
            +getEmailOfUser(username: String): String
            +addProposal(userConference: UserConference, abstractText: string, paperText: string, title: string, authors: string, keywords: string, accepted: boolean)
            +updateProposal(userConference: UserConference, abstractText: string, paperText: string, title: string, authors: string, keywords: string, accepted: boolean)
            +getPCMemberProposals(pcMemberid: int, proposalid: int, availability: Availability)
            +getProposalsOfUser(userid: int): List<Proposal>
            +getProposalsForPCMember(pcMemberid: int, conferenceid: int)
        }
    }

    package GUI {
        class LoginView {
            +service: Service
            +root: GridPane
            +usernameField: TextField
            +passwordField: PasswordField
            +loginButton: Button
            +createAccountButton: Button
            +handleLogin(): void
            +handleCreateAccount(): void
        }
        class CreateAccountView {
            +service: Service
            +root: GridPane
            +emailField: TextField
            +usernameField: TextField
            +passwordField: PasswordField
            +confirmPasswordField: PasswordField
            +createAccountButton: Button
            +backToLoginButton: Button
            +handleLogin(): void
            +handleCreateAccount(): void
        }
        class AdminView {
            +service: Service
            +nameField: TextField
            +dateField: DatePicker
            +addConferenceButton: Button
            +conferenceListView: ListView<Conference>
            +chairNameField: TextField
            +coChairRadioButton: RadioButton
            +inviteChairButton: Button
            +logoutButton: Button
            +handleLogoutButton(): void
            +handleInviteChair(): void
            +handleCreateConference(): void
            +getSelectedConference(): Conference
        }
        class UserView {
            +user: User
            +service: Service
            +root: GridPane
            +logoutButton: Button
            +memberList: MemberList
            +conferenceListView: ListView<Conference>
            +handleLogout(): void
            +loadData(): void
        }
    }

    Room "1" -- "1..*" Session
    Session "1" -- "1..*" Proposal
    Proposal "1..*" -- "1" Review
    Proposal "1..*" -- "1..*" PaperRecommendation
    Proposal "1..*" -- "1..*" UserSection
    Review "1..*" -- "1..*" Message
    Message "1..*" -- "1..*" User
    User "1..*" -- "1..*" UserConference
    UserConference "1..*" -- "1..*" Conference
    UserConference "1..*" -- "1..*" Role
    
```

