

```

classDiagram
    class AbstractActiveRecord {
        <<Abstract>>
        ActiveRecord
        _id int
        #_saveFromDB
        #_update() void
        #_insert() void
        #_delete() void
        #_getById(STRING) ResultSet
        #connectiveDB(ResultSet) void
        #build(STRING) void
        #save() void
        #get_id() int
        #set_id(int) void
        #remove() void
    }
    class User {
        _lastName string
        _firstName string
        _login string
        _gender Gender
        _password string
        _role Role
        + User()
        + User(login:string, string, Gender:string)
        + User(login:string, string, string:string)
        + User(ResultSet)
        + User(id)
        + get_idName() string
        + set_idName(string) void
        + get_idName() string
        + set_idName(string) void
        + get_login() string
        + set_login(string) void
        + getGender() Gender
        + setGender(Gender) void
        + getPassword() string
        + setPassword(string) void
        + findByIdLoginAndFirst(string, string) User
        + findByIdLogin() ResultSet
        + findByIdLoginAndFirstName(string, string) User
        + findByIdLogin() User
        + findById(int) User
        + findById() List<User>
    }
    class Chat {
        _title string
        _owner string
        _description string
        _date_creation string
        _date_premption string
        + Chat(string, string, string, string, string)
        + Chat(string, string, string, string)
        + Chat(ResultSet)
        + Chat(id) void
        + addParticipant(string) void
        + getAllChatMembersLogin(string) Array<List<Chat>
        + getAllChatMembersLoginAndFirst(string) Array<List<Chat>
        + findChatByTitle(string) Chat
        + getAllMembers() ResultSet
        + get_title() string
        + set_title(string) void
        + getOwner() string
        + setOwner(string) void
        + getDescription() string
        + setDescription(string) void
        + getDate_creation() string
        + setDate_creation(string) void
        + getDate_premption() string
        + setDate_premption(string) void
    }
    class BDDConnector {
        -> singleton Connection
        BDDConnector()
        + getinstance() Connection
    }
    class Enums {
        <<Enumeration>>
        Gender
        Role
    }
    AbstractActiveRecord <|-- User
    AbstractActiveRecord <|-- Chat
    BDDConnector <|-- User
    BDDConnector <|-- Chat
    Enums <|-- Gender
    Enums <|-- Role
  
```

```
classDiagram
    class ValidationChat {
        - validationId: long
        + validateRequest(HttpServiceRequest, HttpServiceResponse) void
        + processRequest(HttpServiceRequest, HttpServiceResponse) void
    }
    class Validation {
        - validationId: long
        + validateRequest(HttpServiceRequest, HttpServiceResponse) void
        + processRequest(HttpServiceRequest, HttpServiceResponse) void
    }
    class SupprimerChat {
        - validationId: long
        + validateRequest(HttpServiceRequest, HttpServiceResponse) void
        + processRequest(HttpServiceRequest, HttpServiceResponse) void
    }
    class RedirectTutuhs {
        - validationId: long
        + validateRequest(HttpServiceRequest, HttpServiceResponse) void
        + processRequest(HttpServiceRequest, HttpServiceResponse) void
    }
    class Connexion {
        - validationId: long
        + validateRequest(HttpServiceRequest, HttpServiceResponse) void
        + processRequest(HttpServiceRequest, HttpServiceResponse) void
    }
    class ConnexionChat {
        - validationId: long
        + validateRequest(HttpServiceRequest, HttpServiceResponse) void
        + processRequest(HttpServiceRequest, HttpServiceResponse) void
    }
    class Logout {
        - validationId: long
        + validateRequest(HttpServiceRequest, HttpServiceResponse) void
        + processRequest(HttpServiceRequest, HttpServiceResponse) void
    }
    class MesChats {
        - validationId: long
        + validateRequest(HttpServiceRequest, HttpServiceResponse) void
        + processRequest(HttpServiceRequest, HttpServiceResponse) void
    }
    class ChatsArchives {
        - validationId: long
        + validateRequest(HttpServiceRequest, HttpServiceResponse) void
        + processRequest(HttpServiceRequest, HttpServiceResponse) void
    }
    class ModifierChats {
        - validationId: long
        + validateRequest(HttpServiceRequest, HttpServiceResponse) void
        + processRequest(HttpServiceRequest, HttpServiceResponse) void
    }
    class ValidationModificationChat {
        - validationId: long
        + validateRequest(HttpServiceRequest, HttpServiceResponse) void
        + processRequest(HttpServiceRequest, HttpServiceResponse) void
    }
    class HttpServer {
        + doGet(HttpServiceRequest, HttpServiceResponse) void
        + doPost(HttpServiceRequest, HttpServiceResponse) void
    }
    ValidationChat --> HttpServer
    Validation --> HttpServer
    SupprimerChat --> HttpServer
    RedirectTutuhs --> HttpServer
    Connexion --> HttpServer
    ConnexionChat --> HttpServer
    Logout --> HttpServer
    MesChats --> HttpServer
    ChatsArchives --> HttpServer
    ModifierChats --> HttpServer
    ValidationModificationChat --> HttpServer
```

```

classDiagram
    class ChatServer {
        +singleton : ChatServer
        -session : HashMap<String, Session>
        +StartUpConfigurator : class
        +ChatServer()
        +getinstance() : ChatServer
        +open(Session, String) void
        +close(Session) void
        +endError(Throwable) void
        +handleMessage(String, Session) void
        +sendMessage(String, Session) void
        +sendLeftSender(String) void
    }
    class EndPointConfigurator {
        +getinstanceInstance(Class <T>, <T>...) T
    }
    ChatServer --> EndPointConfigurator

```