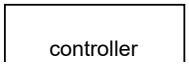
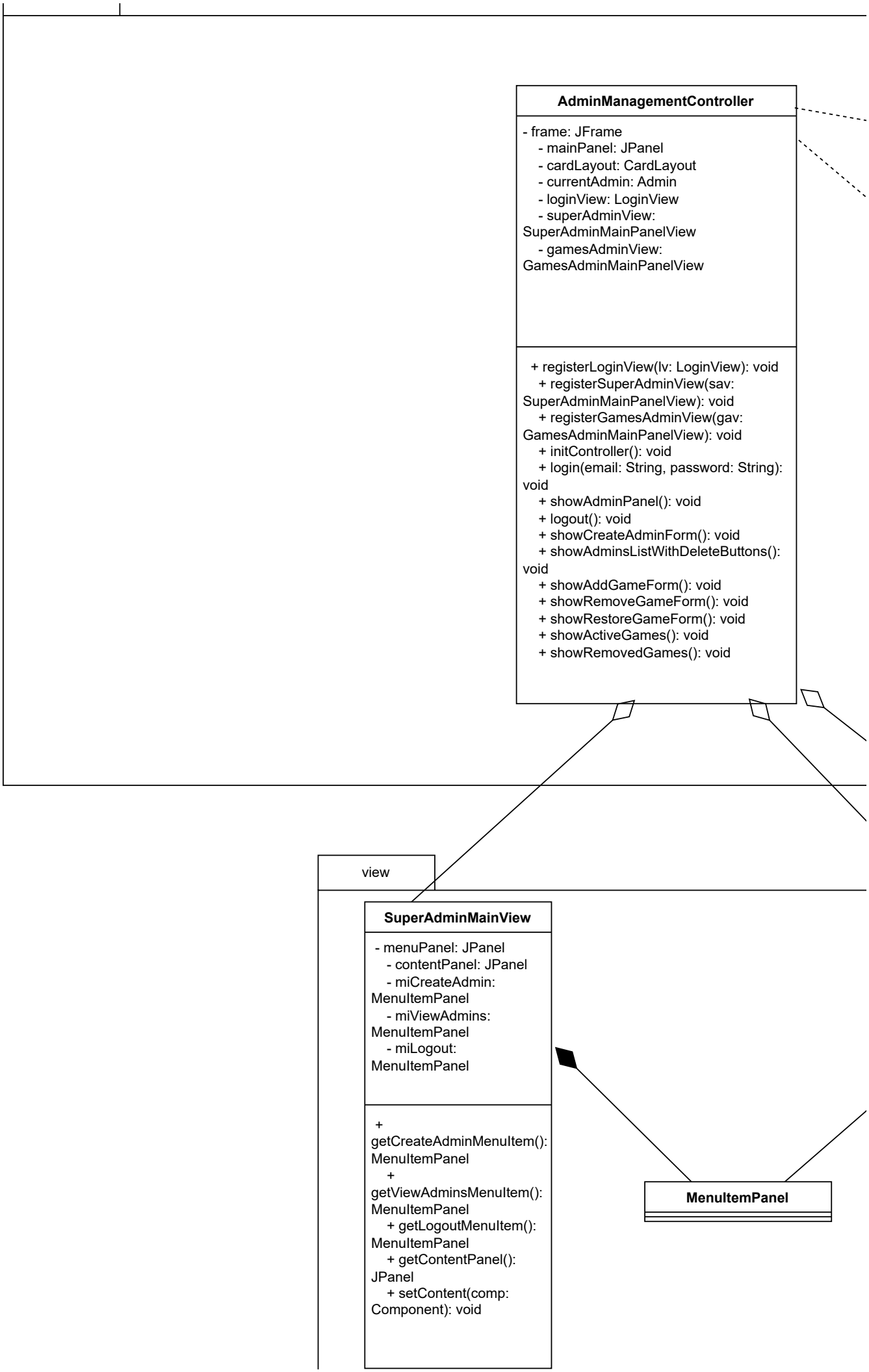
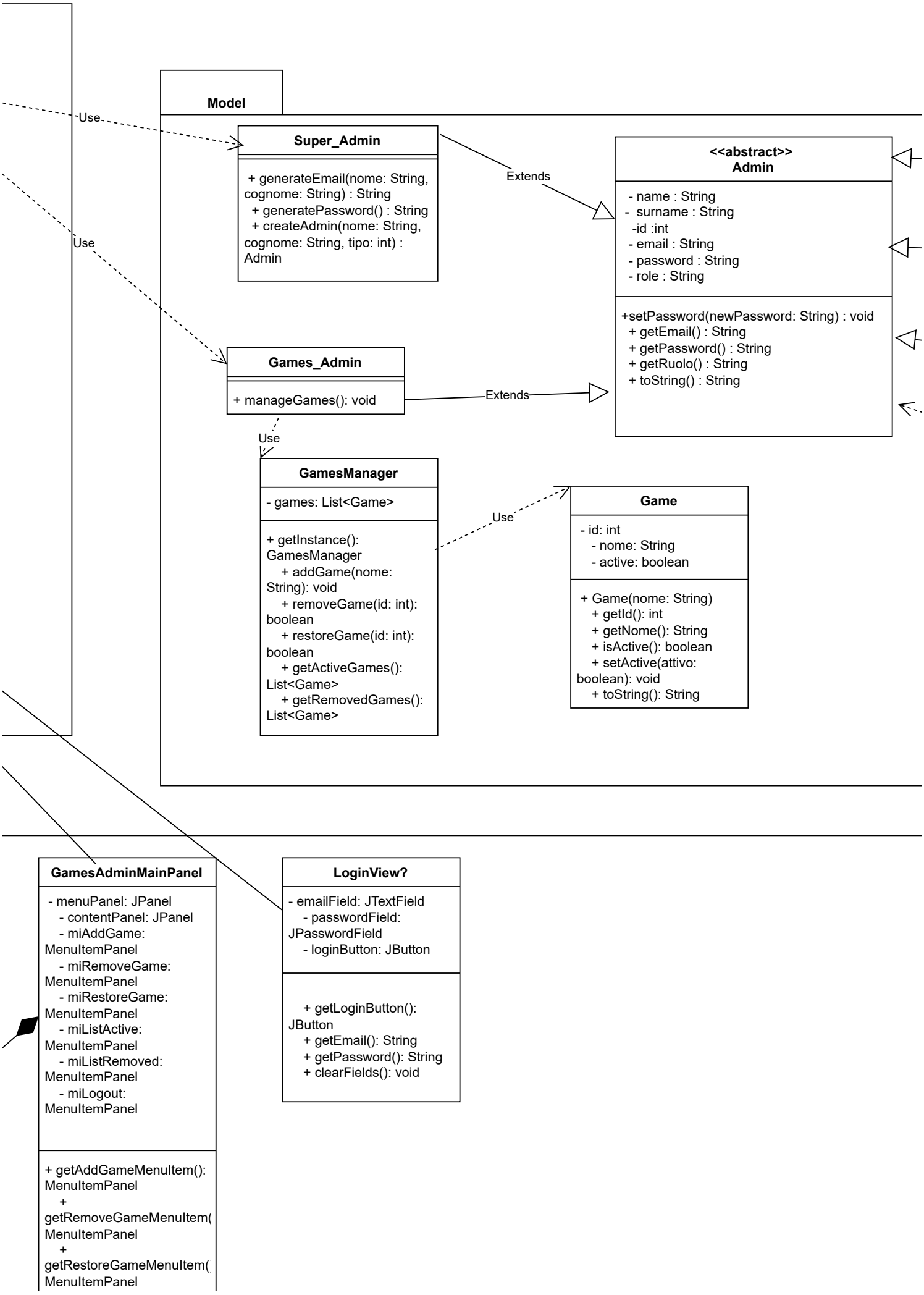
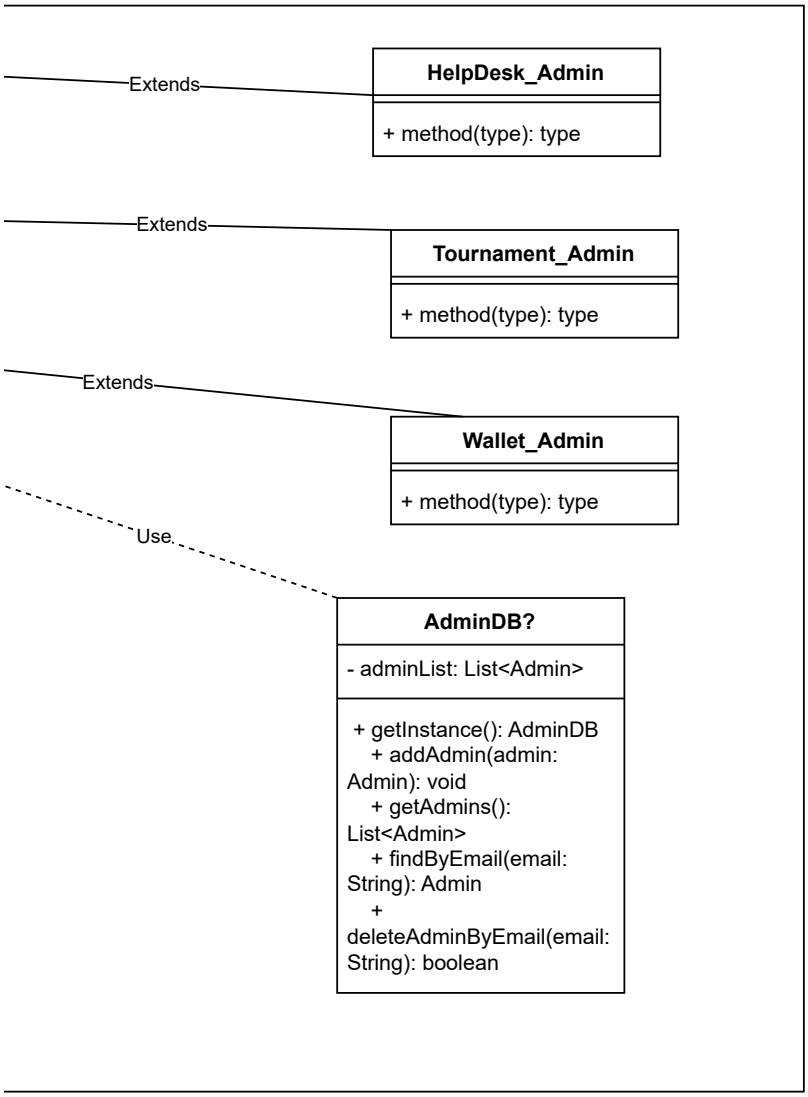
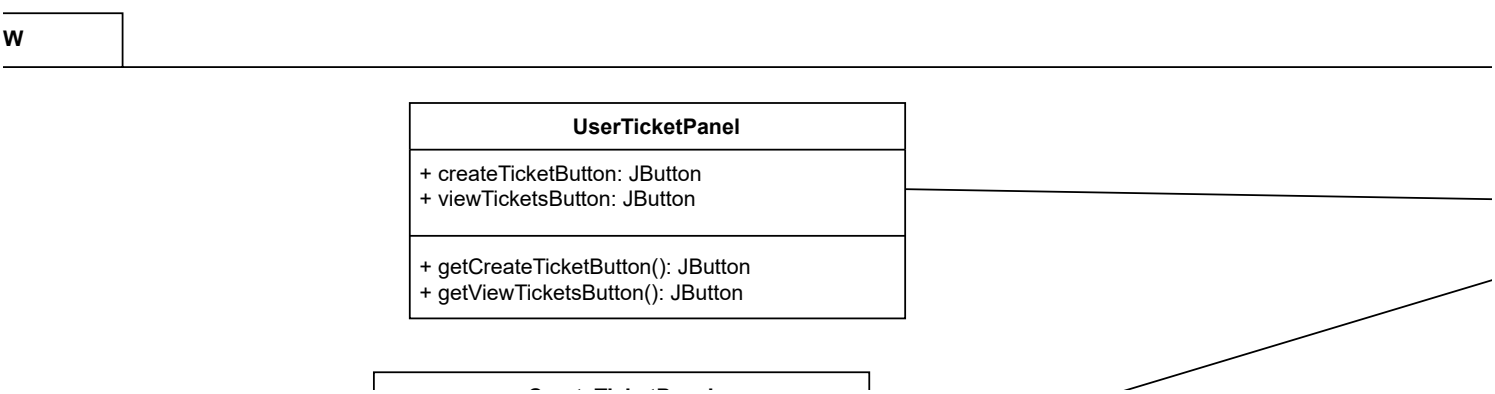

controller

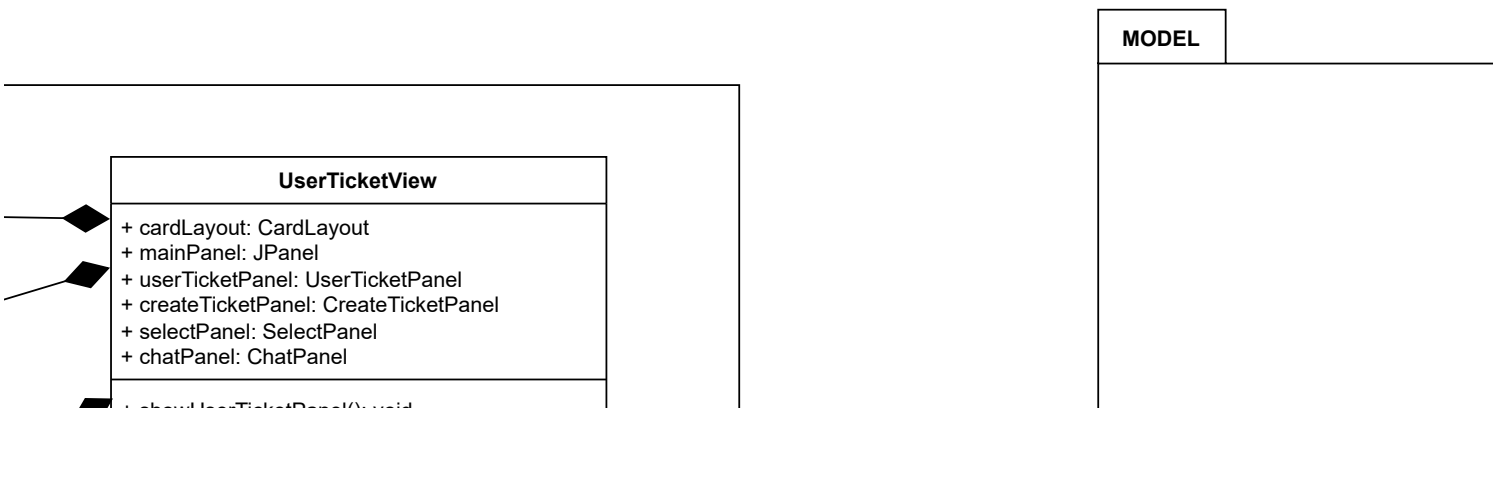

AdminManagementController
- frame: JFrame
- mainPanel: JPanel
- cardLayout: CardLayout
- currentAdmin: Admin
- loginView: LoginView
- superAdminView: SuperAdminMainPanelView
- gamesAdminView: GamesAdminMainPanelView
+ registerLoginView(lv: LoginView): void
+ registerSuperAdminView(sav: SuperAdminMainPanelView): void
+ registerGamesAdminView(gav: GamesAdminMainPanelView): void
+ initController(): void
+ login(email: String, password: String): void
+ showAdminPanel(): void
+ logout(): void
+ showCreateAdminForm(): void
+ showAdminsListWithDeleteButtons(): void
+ showAddGameForm(): void
+ showRemoveGameForm(): void
+ showRestoreGameForm(): void
+ showActiveGames(): void
+ showRemovedGames(): void
view
SuperAdminMainView
- menuPanel: JPanel
- contentPanel: JPanel
- miCreateAdmin: MenuItemPanel
- miViewAdmins: MenuItemPanel
- miLogout: MenuItemPanel
+ getCreateAdminMenuItem(): MenuItemPanel
+ getViewAdminsMenuItem(): MenuItemPanel
+ getLogoutMenuItem(): MenuItemPanel
+ getContentPanel(): JPanel
+ setContent(comp: Component): void
MenuItemPanel


Model
Use
Use
Super_Admin
+ generateEmail(nome: String, cognome: String) : String
+ generatePassword() : String
+ createAdmin(nome: String, cognome: String, tipo: int) : Admin
Extends
<<abstract>>
Admin
- name : String
- surname : String
-id :int
- email : String
- password : String
- role : String
+setPassword(newPassword: String) : void
+ getEmail() : String
+ getPassword() : String
+ getRuolo() : String
+ toString() : String
Games_Admin
+ manageGames(): void
Extends
Use
GamesManager
- games: List<Game>
+ getInstance(): GamesManager
+ addGame(nome: String): void
+ removeGame(id: int): boolean
+ restoreGame(id: int): boolean
+ getActiveGames(): List<Game>
+ getRemovedGames(): List<Game>
Use
Game
- id: int
- nome: String
- active: boolean
+ Game(nome: String)
+ getId(): int
+ getNome(): String
+ isActive(): boolean
+ setActive(attivo: boolean): void
+ toString(): String
GamesAdminMainPanel
- menuPanel: JPanel
- contentPanel: JPanel
- miAddGame: MenuItemPanel
- miRemoveGame: MenuItemPanel
- miRestoreGame: MenuItemPanel
- miListActive: MenuItemPanel
- miListRemoved: MenuItemPanel
- miLogout: MenuItemPanel
+ getAddGameMenuItem(): MenuItemPanel
+ getRemoveGameMenuItem(MenuItemPanel
+ getRestoreGameMenuItem(MenuItemPanel
LoginView?
- emailField: JTextField
- passwordField: JPasswordField
- loginButton: JButton
+ getLoginButton(): JButton
+ getEmail(): String
+ getPassword(): String
+ clearFields(): void

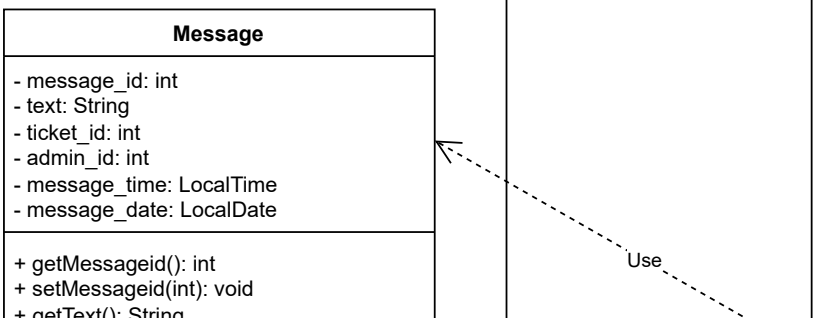

Extends
HelpDesk_Admin
+ method(type): type
Extends
Tournament_Admin
+ method(type): type
Extends
Wallet_Admin
+ method(type): type
Use
AdminDB?
- adminList: List<Admin>
+ getInstance(): AdminDB
+ addAdmin(admin: Admin): void
+ getAdmins(): List<Admin>
+ findByEmail(email: String): Admin
+ deleteAdminByEmail(email: String): boolean

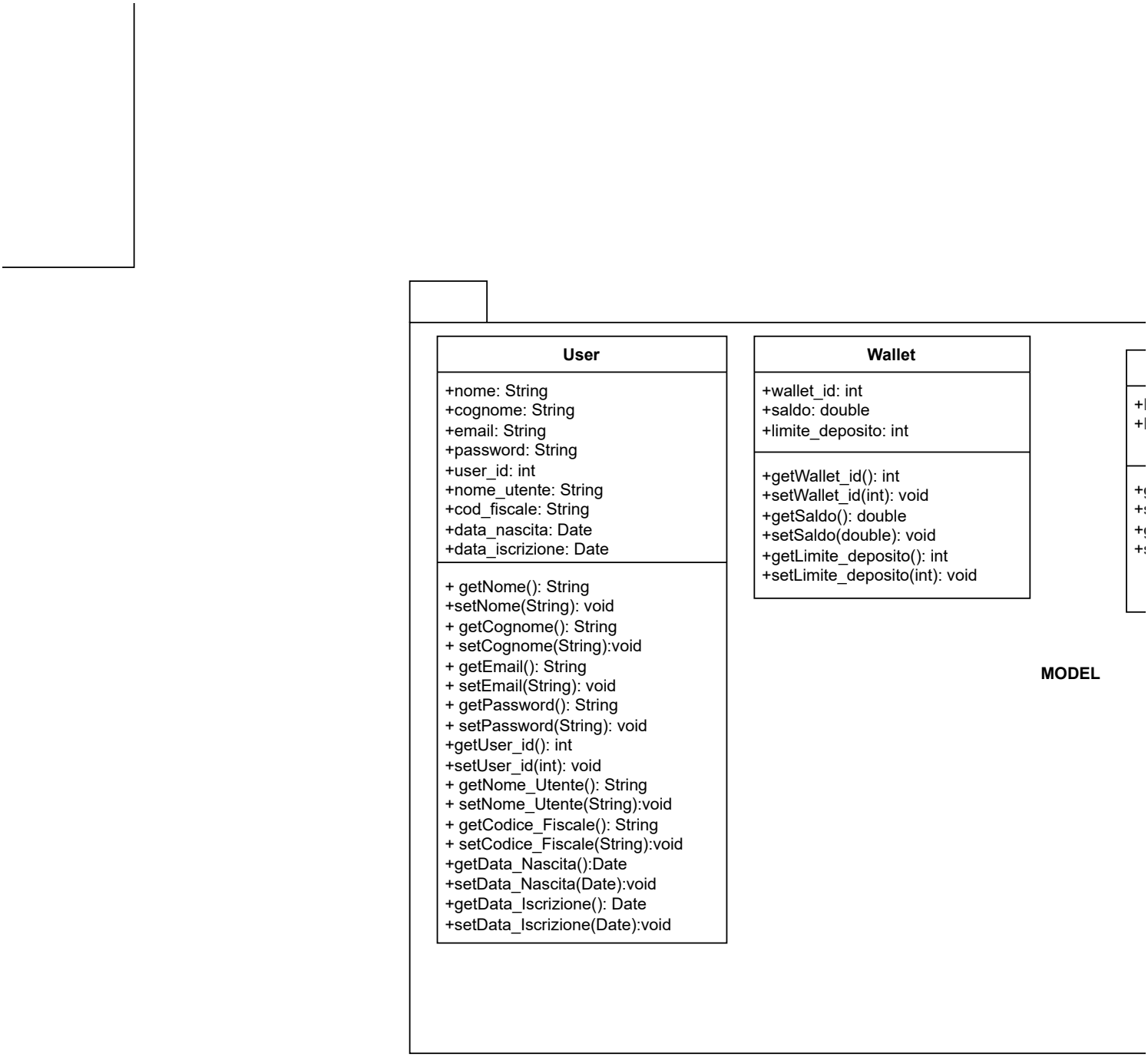
VIE

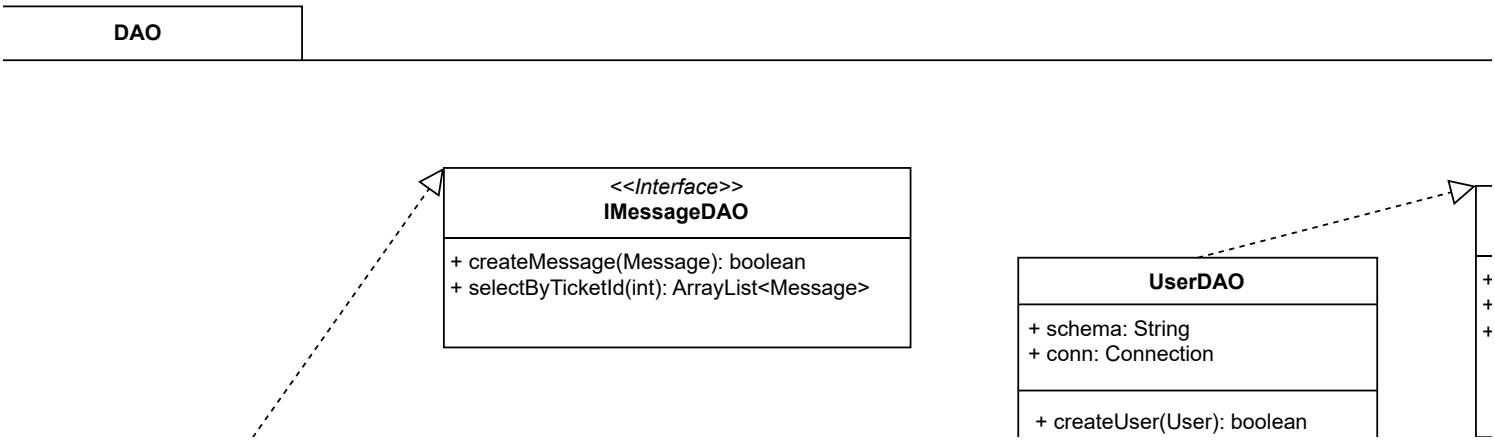

W
UserTicketPanel
+ createTicketButton: JButton
+ viewTicketsButton: JButton
+ getCreateTicketButton(): JButton
+ getViewTicketsButton(): JButton

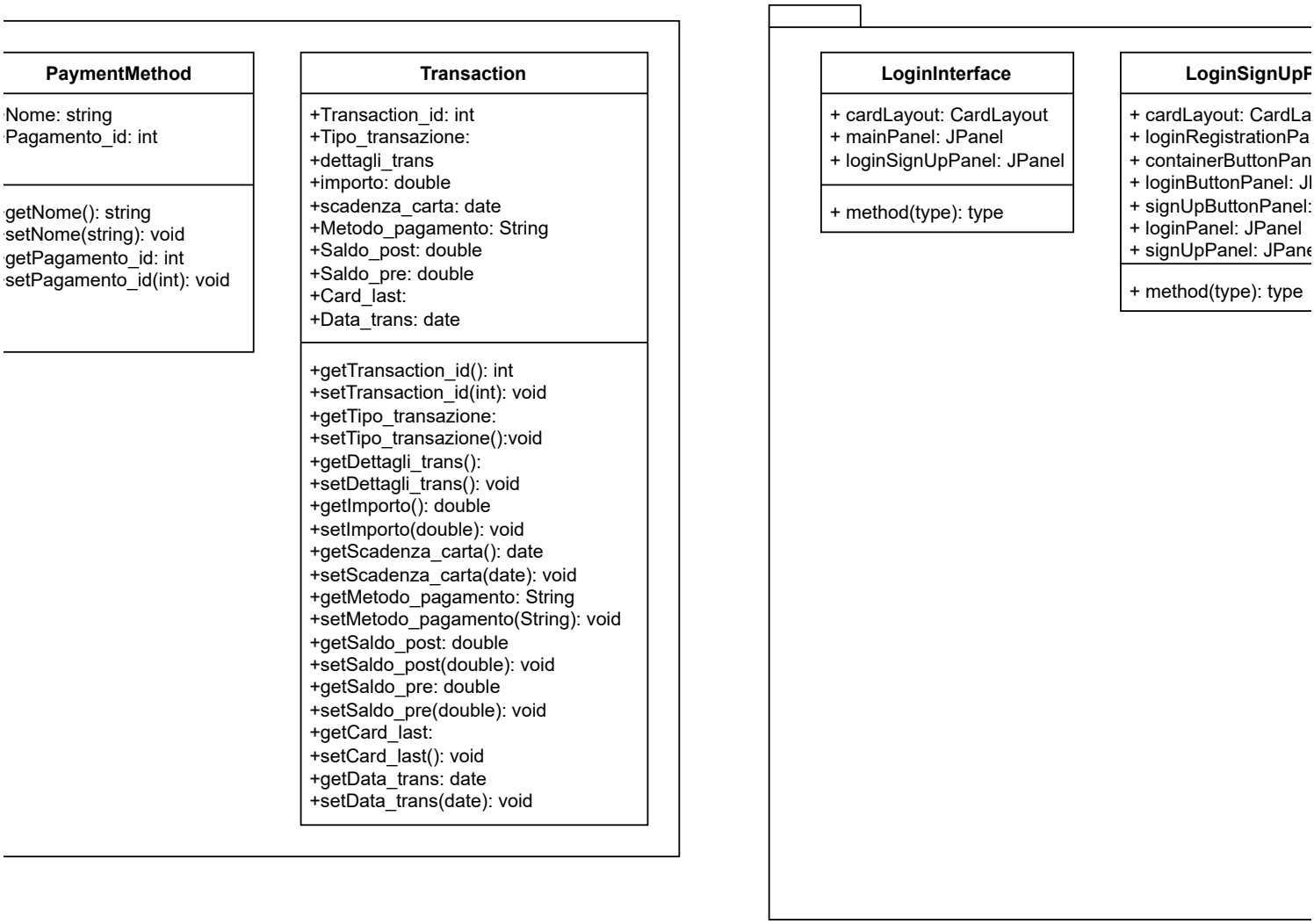

MODEL
UserTicketView
+ cardLayout: CardLayout
+ mainPanel: JPanel
+ userTicketPanel: UserTicketPanel
+ createTicketPanel: CreateTicketPanel
+ selectPanel: SelectPanel
+ chatPanel: ChatPanel

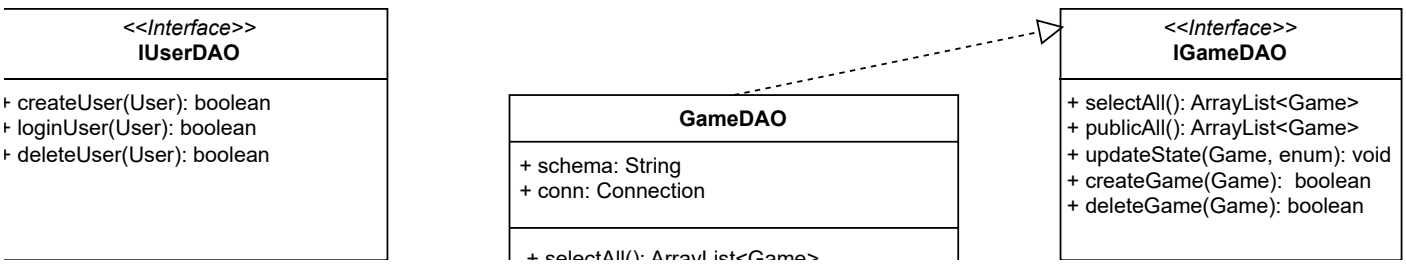
+ getListActiveMenuItem(
MenuItemPanel
+
getListRemovedMenuItem():
MenuItemPanel
+ getLogoutMenuItem():
MenuItemPanel
+ getContentPanel():
JPanel
+ setContent(comp:
Component): void

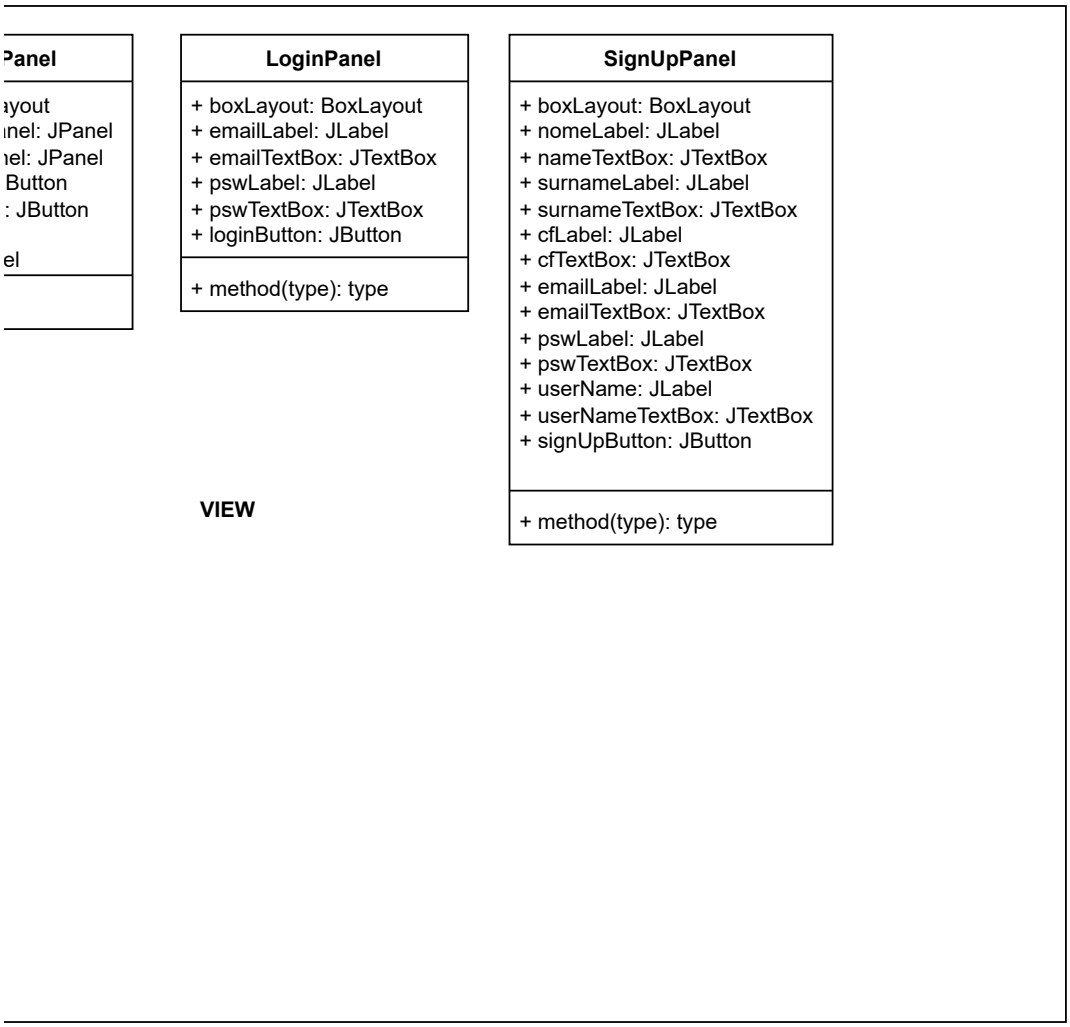


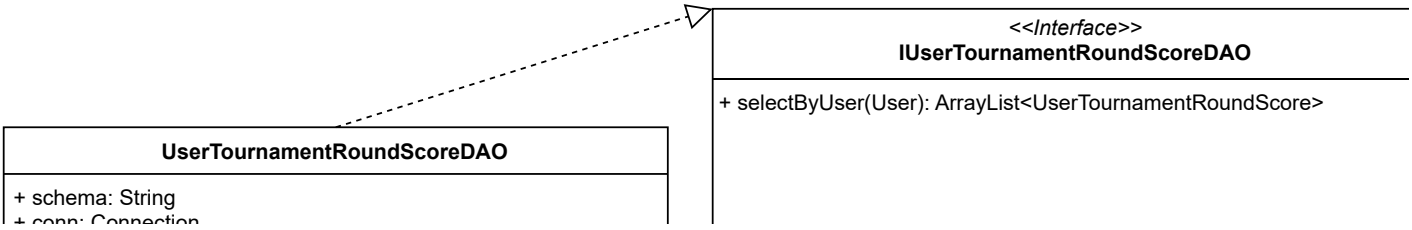

User
+nome: String
+cognome: String
+email: String
+password: String
+user_id: int
+nome_utente: String
+cod_fiscale: String
+data_nascita: Date
+data_iscrizione: Date
+ getNome(): String
+setNome(String): void
+ getCognome(): String
+ setCognome(String):void
+ getEmail(): String
+ setEmail(String): void
+ getPassword(): String
+ setPassword(String): void
+getUser_id(): int
+setUser_id(int): void
+ getNome_Utente(): String
+ setNome_Utente(String):void
+ getCodice_Fiscale(): String
+ setCodice_Fiscale(String):void
+getData_Nascita():Date
+setData_Nascita(Date):void
+getData_Iscrizione(): Date
+setData_Iscrizione(Date):void
Wallet
+wallet_id: int
+saldo: double
+limite_deposito: int
+getWallet_id(): int
+setWallet_id(int): void
+getSaldo(): double
+setSaldo(double): void
+getLimite_deposito(): int
+setLimite_deposito(int): void
MODEL

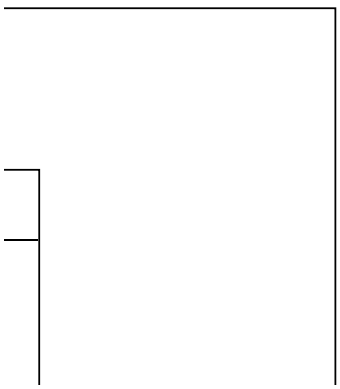

DAO
<<Interface>>
IMessageDAO
+ createMessage(Message): boolean
+ selectByTicketId(int): ArrayList<Message>
UserDAO
+ schema: String
+ conn: Connection
+ createUser(User): boolean

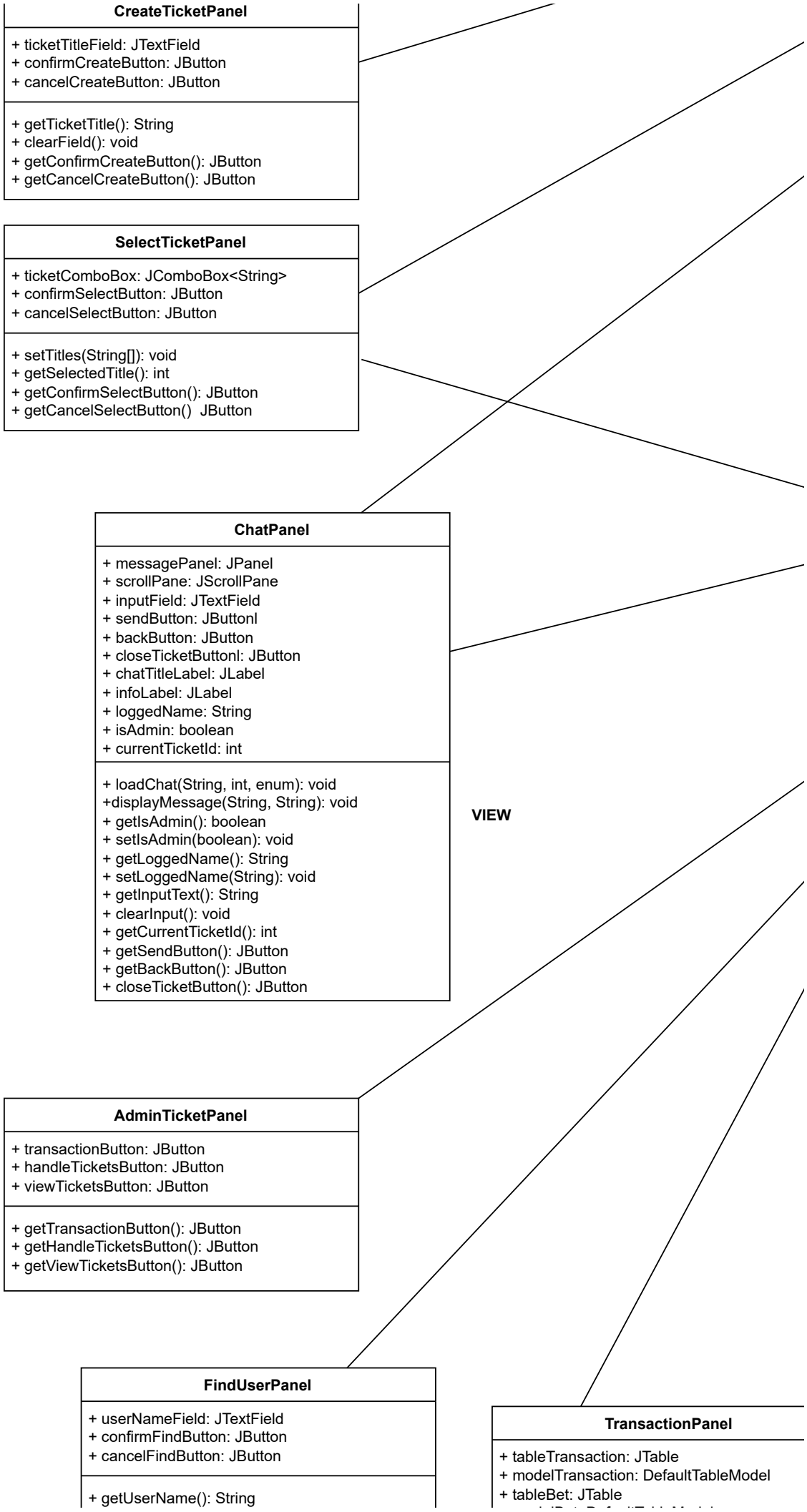

PaymentMethod
Nome: string
Pagamento_id: int
getNome(): string
setNome(string): void
getPagamento_id: int
setPagamento_id(int): void
Transaction
+Transaction_id: int
+Tipo_transazione:
+dettagli_trans
+importo: double
+scadenza_carta: date
+Metodo_pagamento: String
+Saldo_post: double
+Saldo_pre: double
+Card_last:
+Data_trans: date
+getTransaction_id(): int
+setTransaction_id(int): void
+getTipo_transazione:
+setTipo_transazione():void
+getDettagli_trans():
+setDettagli_trans(): void
+getImporto(): double
+setImporto(double): void
+getScadenza_carta(): date
+setScadenza_carta(date): void
+getMetodo_pagamento: String
+setMetodo_pagamento(String): void
+getSaldo_post: double
+setSaldo_post(double): void
+getSaldo_pre: double
+setSaldo_pre(double): void
+getCard_last:
+setCard_last(): void
+getData_trans: date
+setData_trans(date): void
LoginInterface
+ cardLayout: CardLayout
+ mainPanel: JPanel
+ loginSignUpPanel: JPanel
+ method(type): type
LoginSignUpP
+ cardLayout: CardLa
+ loginRegistrationPa
+ containerButtonPan
+ loginButtonPanel: Jl
+ signUpButtonPanel:
+ loginPanel: JPanel
+ signUpPanel: JPane
+ method(type): type

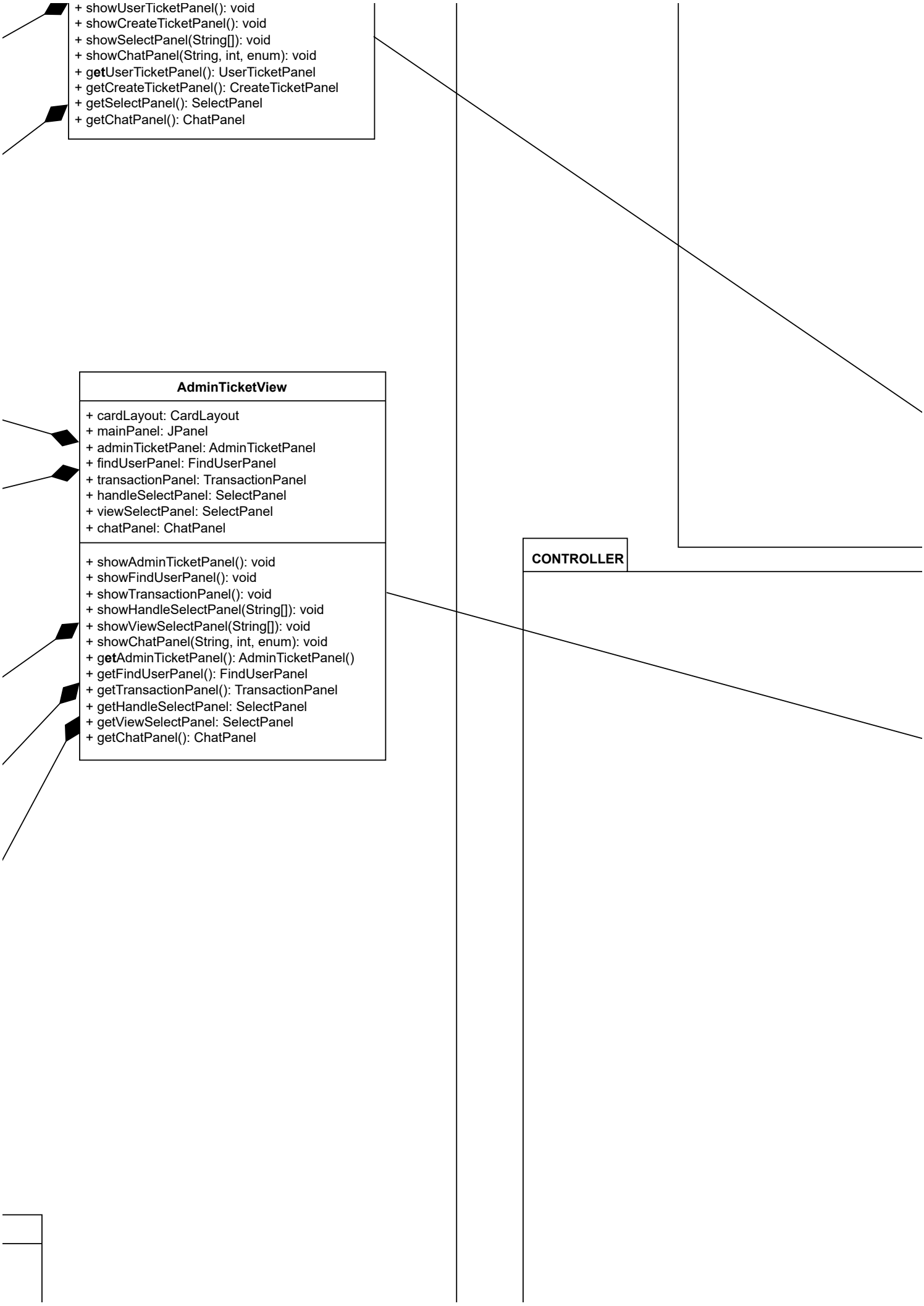

<<Interface>>
IUserDAO
+ createUser(User): boolean
+ loginUser(User): boolean
+ deleteUser(User): boolean
GameDAO
+ schema: String
+ conn: Connection
<<Interface>>
IGameDAO
+ selectAll(): ArrayList<Game>
+ publicAll(): ArrayList<Game>
+ updateState(Game, enum): void
+ createGame(Game): boolean
+ deleteGame(Game): boolean

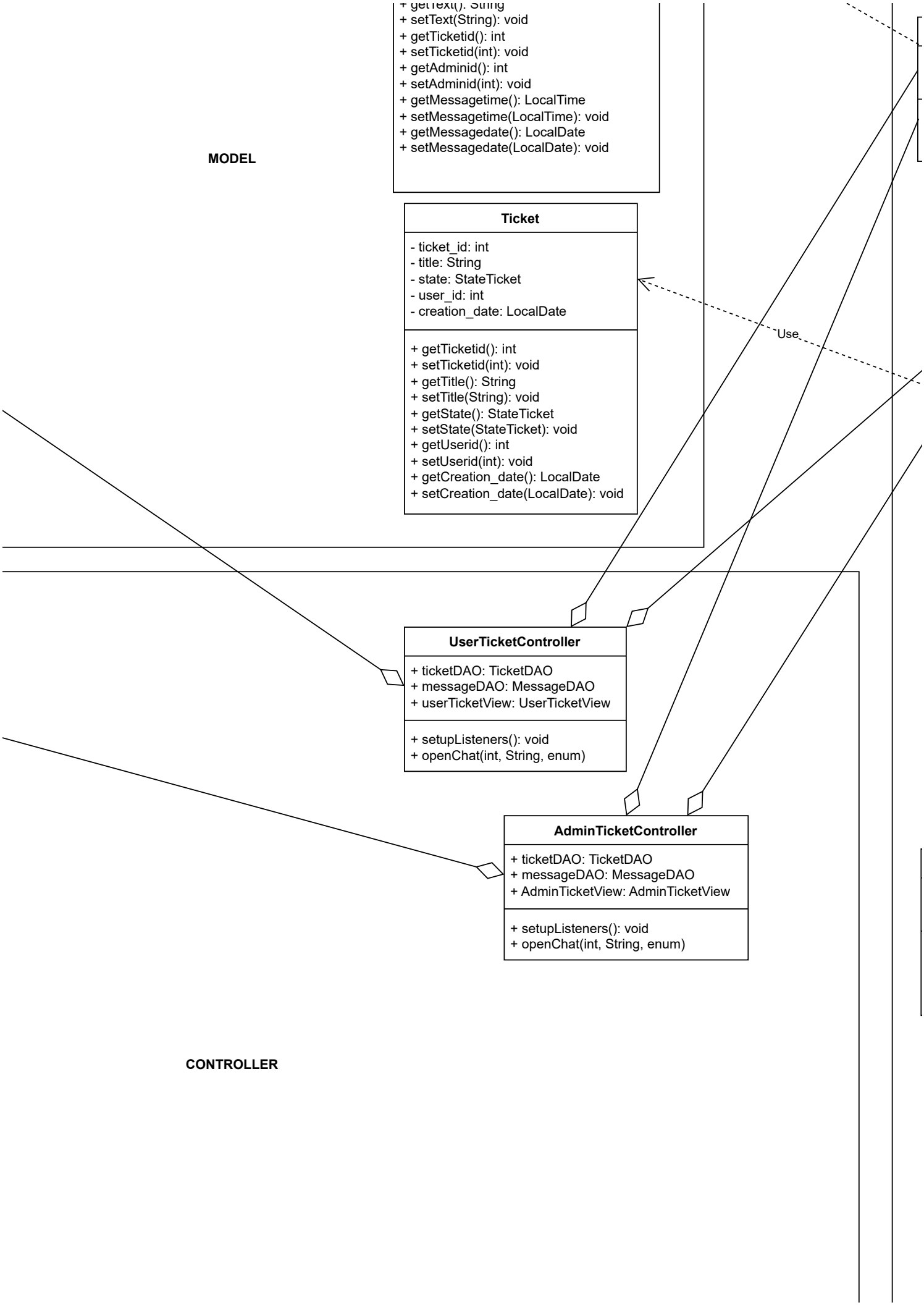

Panel
ayout
nel: JPanel
el: JPanel
Button
: JButton
el
LoginPanel
+ boxLayout: BoxLayout
+ emailLabel: JLabel
+ emailTextBox: JTextBox
+ pswLabel: JLabel
+ pswTextBox: JTextBox
+ loginButton: JButton
+ method(type): type
SignUpPanel
+ boxLayout: BoxLayout
+ nomeLabel: JLabel
+ nameTextBox: JTextBox
+ surnameLabel: JLabel
+ surnameTextBox: JTextBox
+ cfLabel: JLabel
+ cfTextBox: JTextBox
+ emailLabel: JLabel
+ emailTextBox: JTextBox
+ pswLabel: JLabel
+ pswTextBox: JTextBox
+ userName: JLabel
+ userNameTextBox: JTextBox
+ signUpButton: JButton
+ method(type): type
VIEW

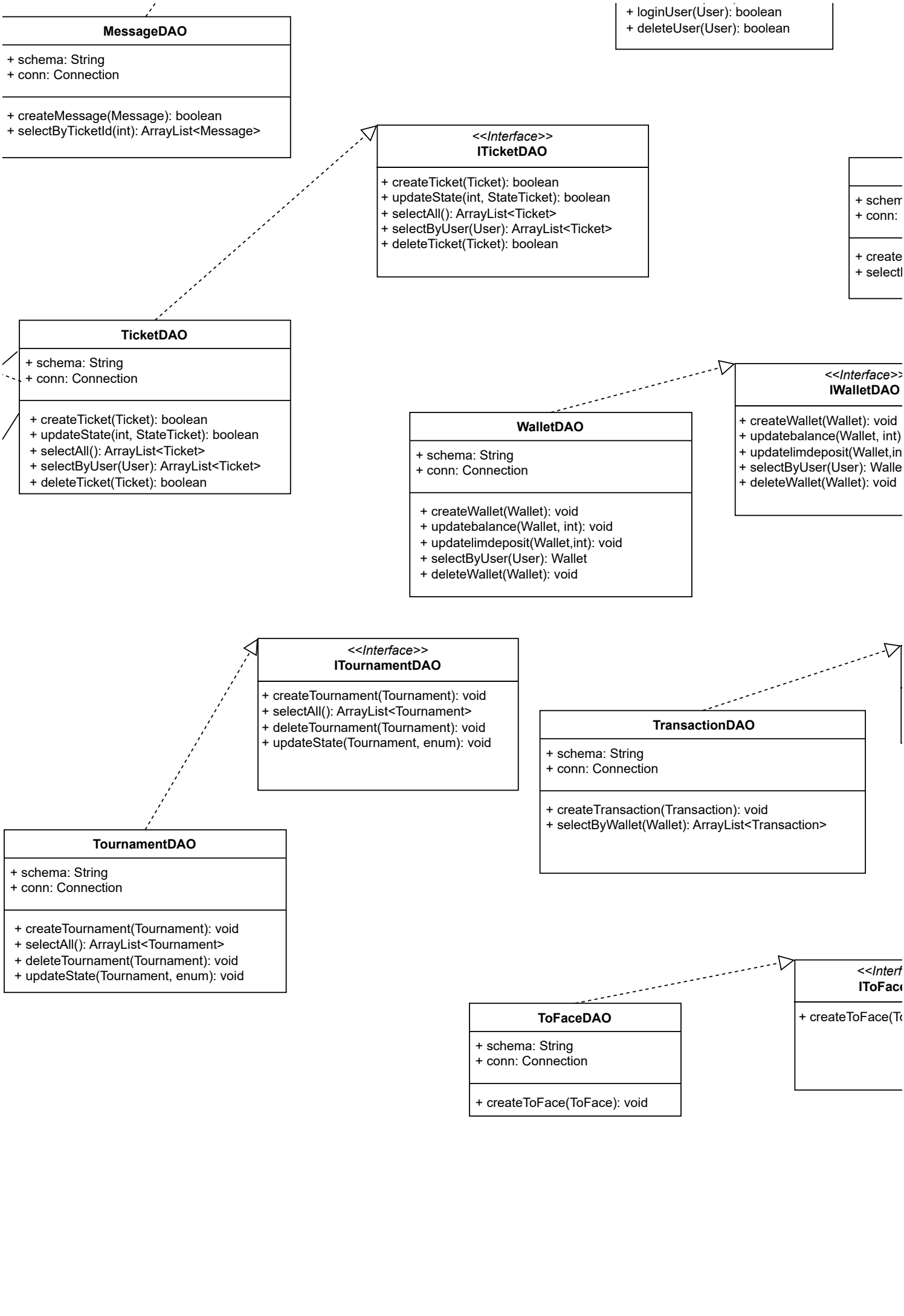

<<Interface>>
IUserTournamentRoundScoreDAO
+ selectByUser(User): ArrayList<UserTournamentRoundScore>
UserTournamentRoundScoreDAO
+ schema: String

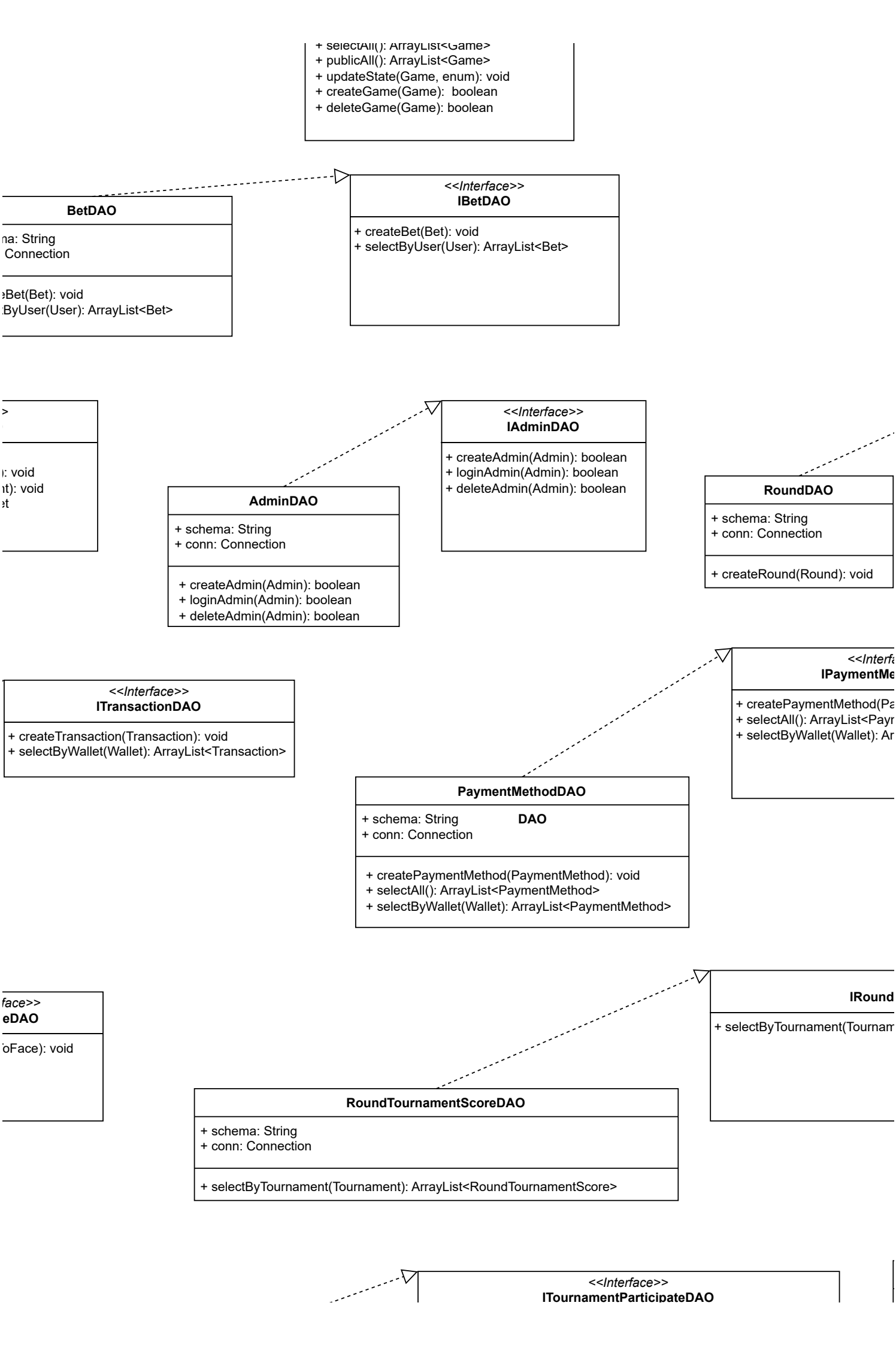


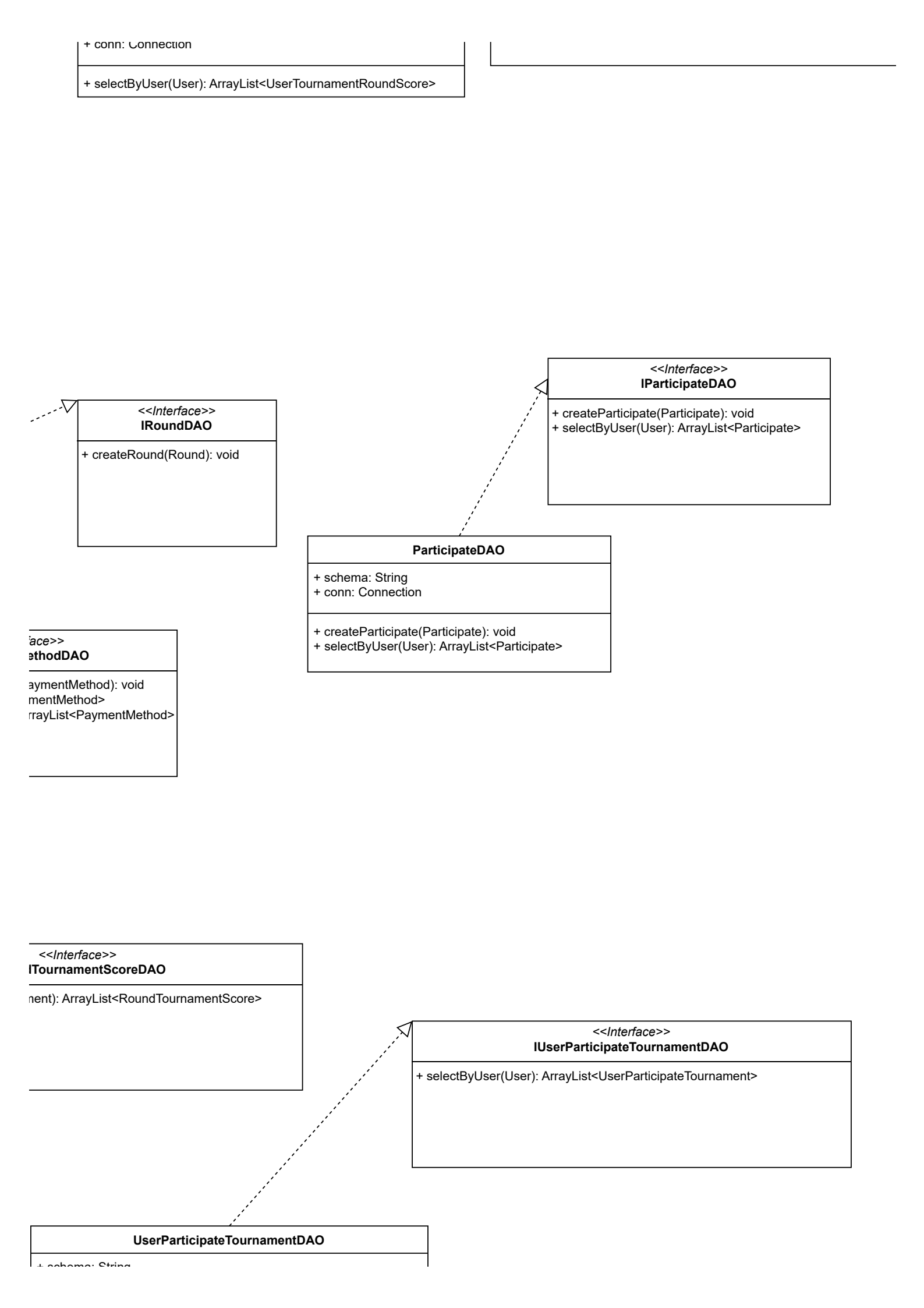

CreateTicketPanel
+ ticketTitleField: JTextField
+ confirmCreateButton: JButton
+ cancelCreateButton: JButton
+ getTicketTitle(): String
+ clearField(): void
+ getConfirmCreateButton(): JButton
+ getCancelCreateButton(): JButton
SelectTicketPanel
+ ticketComboBox: JComboBox<String>
+ confirmSelectButton: JButton
+ cancelSelectButton: JButton
+ setTitles(String[]): void
+ getSelectedTitle(): int
+ getConfirmSelectButton(): JButton
+ getCancelSelectButton() JButton
ChatPanel
+ messagePanel: JPanel
+ scrollPane: JScrollPane
+ inputField: JTextField
+ sendButton: JButtonl
+ backButton: JButton
+ closeTicketButtonl: JButton
+ chatTitleLabel: JLabel
+ infoLabel: JLabel
+ loggedName: String
+ isAdmin: boolean
+ currentTicketId: int
+ loadChat(String, int, enum): void
+displayMessage(String, String): void
+ getIsAdmin(): boolean
+ setIsAdmin(boolean): void
+ getLoggedName(): String
+ setLoggedName(String): void
+ getInputText(): String
+ clearInput(): void
+ getCurrentTicketId(): int
+ getSendButton(): JButton
+ getBackButton(): JButton
+ closeTicketButton(): JButton
VIEW
AdminTicketPanel
+ transactionButton: JButton
+ handleTicketsButton: JButton
+ viewTicketsButton: JButton
+ getTransactionButton(): JButton
+ getHandleTicketsButton(): JButton
+ getViewTicketsButton(): JButton
FindUserPanel
+ userNameField: JTextField
+ confirmFindButton: JButton
+ cancelFindButton: JButton
+ getUserName(): String
TransactionPanel
+ tableTransaction: JTable
+ modelTransaction: DefaultTableModel
+ tableBet: JTable

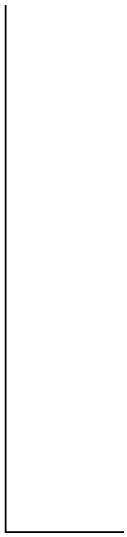

+ showUserTicketPanel(): void
+ showCreateTicketPanel(): void
+ showSelectPanel(String[]): void
+ showChatPanel(String, int, enum): void
+ getUserTicketPanel(): UserTicketPanel
+ getCreateTicketPanel(): CreateTicketPanel
+ getSelectPanel(): SelectPanel
+ getChatPanel(): ChatPanel
AdminTicketView
+ cardLayout: CardLayout
+ mainPanel: JPanel
+ adminTicketPanel: AdminTicketPanel
+ findUserPanel: FindUserPanel
+ transactionPanel: TransactionPanel
+ handleSelectPanel: SelectPanel
+ viewSelectPanel: SelectPanel
+ chatPanel: ChatPanel
+ showAdminTicketPanel(): void
+ showFindUserPanel(): void
+ showTransactionPanel(): void
+ showHandleSelectPanel(String[]): void
+ showViewSelectPanel(String[]): void
+ showChatPanel(String, int, enum): void
+ getAdminTicketPanel(): AdminTicketPanel()
+ getFindUserPanel(): FindUserPanel
+ getTransactionPanel(): TransactionPanel
+ getHandleSelectPanel: SelectPanel
+ getViewSelectPanel: SelectPanel
+ getChatPanel(): ChatPanel
CONTROLLER

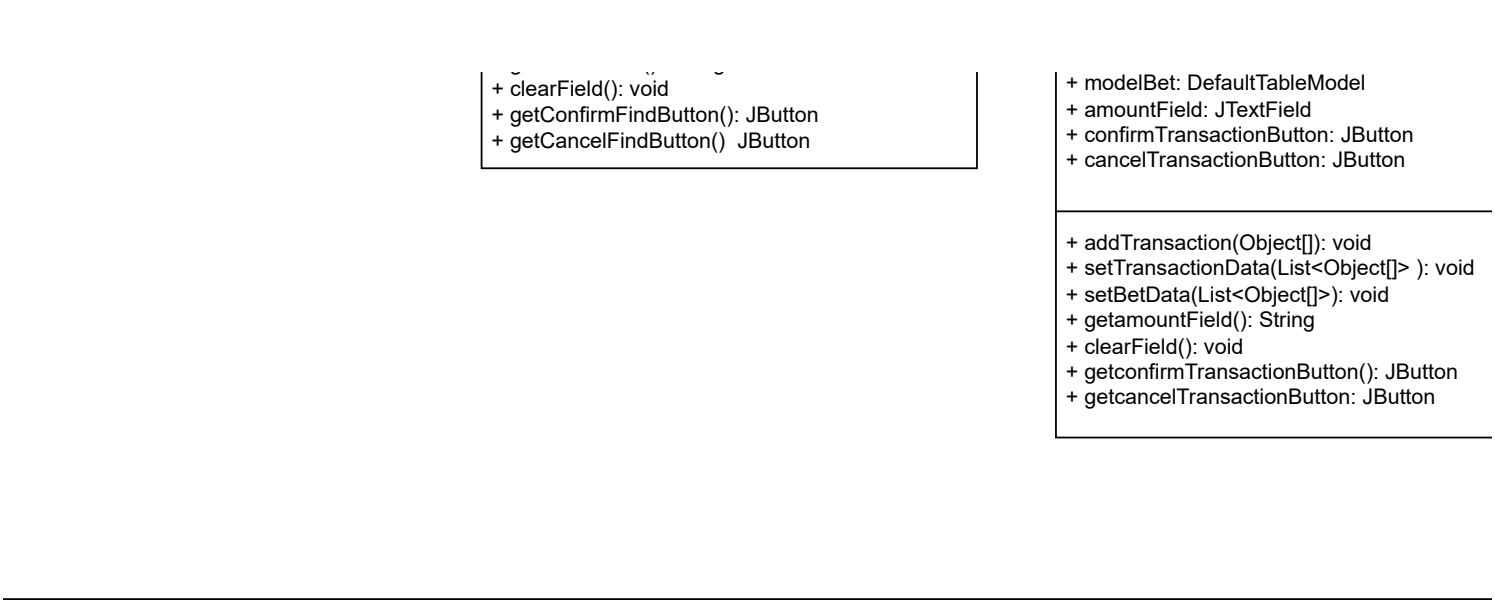

+ getText(): String
+ setText(String): void
+ getTicketid(): int
+ setTicketid(int): void
+ getAdminid(): int
+ setAdminid(int): void
+ getMessagetime(): LocalTime
+ setMessagetime(LocalTime): void
+ getMessagedate(): LocalDate
+ setMessagedate(LocalDate): void
MODEL
Ticket
- ticket_id: int
- title: String
- state: StateTicket
- user_id: int
- creation_date: LocalDate
+ getTicketid(): int
+ setTicketid(int): void
+ getTitle(): String
+ setTitle(String): void
+ getState(): StateTicket
+ setState(StateTicket): void
+ getUserid(): int
+ setUserid(int): void
+ getCreation_date(): LocalDate
+ setCreation_date(LocalDate): void
Use
UserTicketController
+ ticketDAO: TicketDAO
+ messageDAO: MessageDAO
+ userTicketView: UserTicketView
+ setupListeners(): void
+ openChat(int, String, enum)
AdminTicketController
+ ticketDAO: TicketDAO
+ messageDAO: MessageDAO
+ AdminTicketView: AdminTicketView
+ setupListeners(): void
+ openChat(int, String, enum)
CONTROLLER

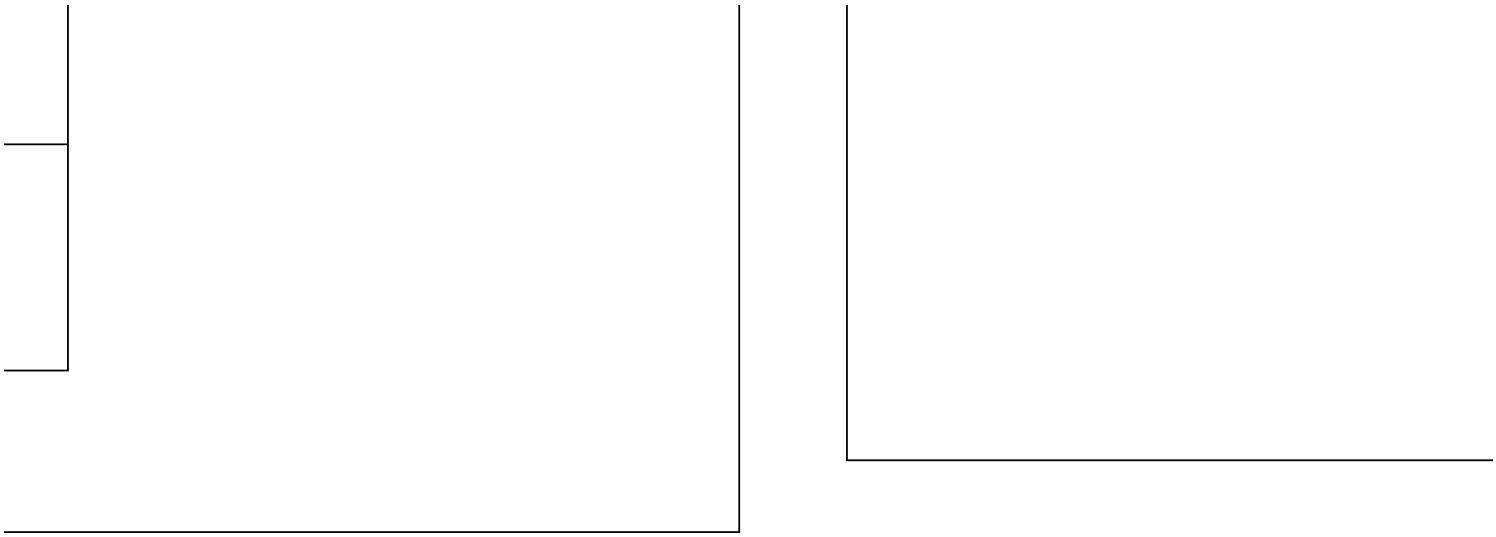

+ loginUser(User): boolean
+ deleteUser(User): boolean
MessageDAO
+ schema: String
+ conn: Connection
+ createMessage(Message): boolean
+ selectByTicketId(int): ArrayList<Message>
<<Interface>>
ITicketDAO
+ createTicket(Ticket): boolean
+ updateState(int, StateTicket): boolean
+ selectAll(): ArrayList<Ticket>
+ selectByUser(User): ArrayList<Ticket>
+ deleteTicket(Ticket): boolean
+ schem
+ conn:
+ create
+ selectl
TicketDAO
+ schema: String
+ conn: Connection
+ createTicket(Ticket): boolean
+ updateState(int, StateTicket): boolean
+ selectAll(): ArrayList<Ticket>
+ selectByUser(User): ArrayList<Ticket>
+ deleteTicket(Ticket): boolean
<<Interface>>
IWalletDAO
+ createWallet(Wallet): void
+ updatebalance(Wallet, int)
+ updatelimdeposit(Wallet,in
+ selectByUser(User): Walle
+ deleteWallet(Wallet): void
WalletDAO
+ schema: String
+ conn: Connection
+ createWallet(Wallet): void
+ updatebalance(Wallet, int): void
+ updatelimdeposit(Wallet,int): void
+ selectByUser(User): Wallet
+ deleteWallet(Wallet): void
<<Interface>>
ITournamentDAO
+ createTournament(Tournament): void
+ selectAll(): ArrayList<Tournament>
+ deleteTournament(Tournament): void
+ updateState(Tournament, enum): void
TransactionDAO
+ schema: String
+ conn: Connection
+ createTransaction(Transaction): void
+ selectByWallet(Wallet): ArrayList<Transaction>
TournamentDAO
+ schema: String
+ conn: Connection
+ createTournament(Tournament): void
+ selectAll(): ArrayList<Tournament>
+ deleteTournament(Tournament): void
+ updateState(Tournament, enum): void
<<Interf
IToFac
+ createToFace(T
ToFaceDAO
+ schema: String
+ conn: Connection
+ createToFace(ToFace): void

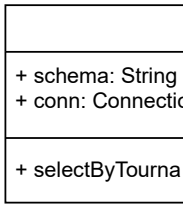

+ selectAll(): ArrayList<Game>
+ publicAll(): ArrayList<Game>
+ updateState(Game, enum): void
+ createGame(Game): boolean
+ deleteGame(Game): boolean
BetDAO
na: String
Connection
eBet(Bet): void
ByUser(User): ArrayList<Bet>
<<Interface>>
IBetDAO
+ createBet(Bet): void
+ selectByUser(User): ArrayList<Bet>
: void
t): void
et
AdminDAO
+ schema: String
+ conn: Connection
+ createAdmin(Admin): boolean
+ loginAdmin(Admin): boolean
+ deleteAdmin(Admin): boolean
<<Interface>>
IAdminDAO
+ createAdmin(Admin): boolean
+ loginAdmin(Admin): boolean
+ deleteAdmin(Admin): boolean
RoundDAO
+ schema: String
+ conn: Connection
+ createRound(Round): void
<<Interface>>
ITransactionDAO
+ createTransaction(Transaction): void
+ selectByWallet(Wallet): ArrayList<Transaction>
<<Interfa
IPaymentMe
+ createPaymentMethod(Pa
+ selectAll(): ArrayList<Payr
+ selectByWallet(Wallet): Ar
PaymentMethodDAO
+ schema: String
+ conn: Connection
DAO
+ createPaymentMethod(PaymentMethod): void
+ selectAll(): ArrayList<PaymentMethod>
+ selectByWallet(Wallet): ArrayList<PaymentMethod>
face>>
eDAO
oFace): void
IRound
+ selectByTournament(Tournam
RoundTournamentScoreDAO
+ schema: String
+ conn: Connection
+ selectByTournament(Tournament): ArrayList<RoundTournamentScore>
<<Interface>>
ITournamentParticipateDAO

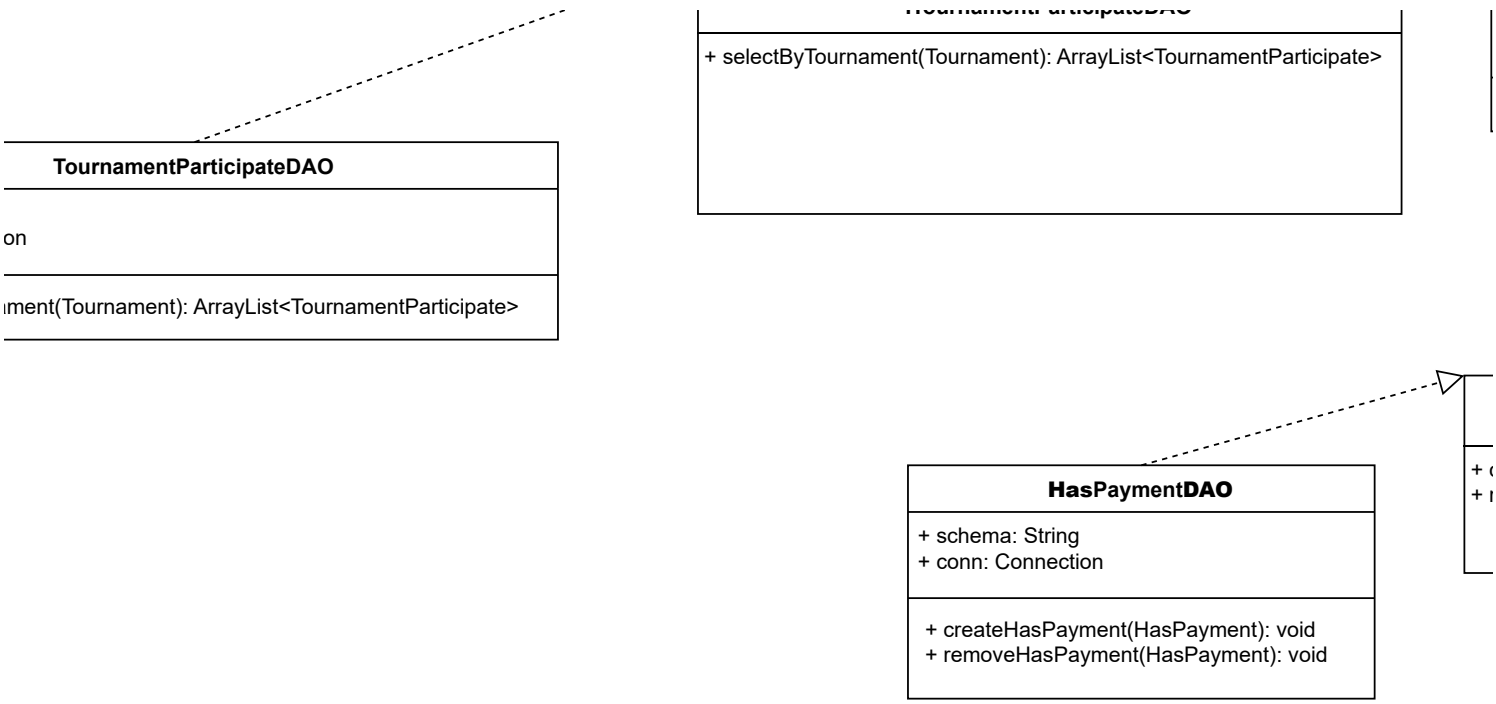

+ conn: Connection
+ selectByUser(User): ArrayList<UserTournamentRoundScore>
<<Interface>>
IRoundDAO
+ createRound(Round): void
<<Interface>>
IParticipateDAO
+ createParticipate(Participate): void
+ selectByUser(User): ArrayList<Participate>
ParticipateDAO
+ schema: String
+ conn: Connection
+ createParticipate(Participate): void
+ selectByUser(User): ArrayList<Participate>
<<Interface>>
IUserParticipateTournamentDAO
+ selectByUser(User): ArrayList<UserParticipateTournament>
UserParticipateTournamentDAO




+ clearField(): void
+ getConfirmFindButton(): JButton
+ getCancelFindButton() JButton
+ modelBet: DefaultTableModel
+ amountField: JTextField
+ confirmTransactionButton: JButton
+ cancelTransactionButton: JButton
+ addTransaction(Object[]): void
+ setTransactionData(List<Object[]>): void
+ setBetData(List<Object[]>): void
+ getamountField(): String
+ clearField(): void
+ getconfirmTransactionButton(): JButton
+ getcancelTransactionButton: JButton




+ schema: String
+ conn: Connectio
+ selectByTourna


+ selectByTournament(Tournament): ArrayList<TournamentParticipate>
TournamentParticipateDAO
on
ıment(Tournament): ArrayList<TournamentParticipate>
HasPaymentDAO
+ schema: String
+ conn: Connection
+ createHasPayment(HasPayment): void
+ removeHasPayment(HasPayment): void

+ schema: String
+ conn: Connection

+ selectByUser(User): ArrayList<UserParticipateTournament>

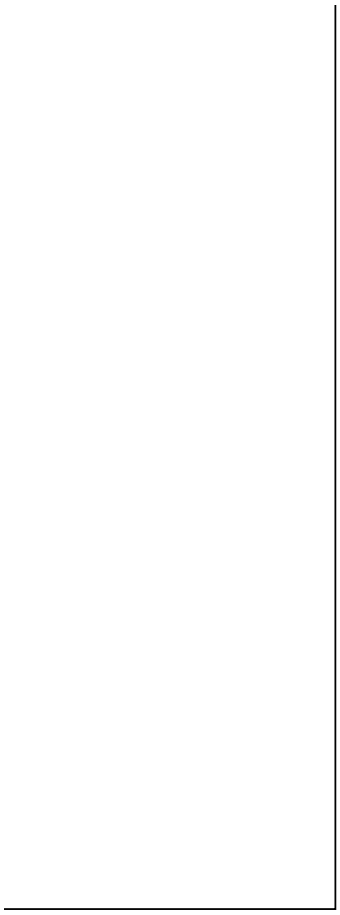
<<Interface>>
IHasPaymentDAO

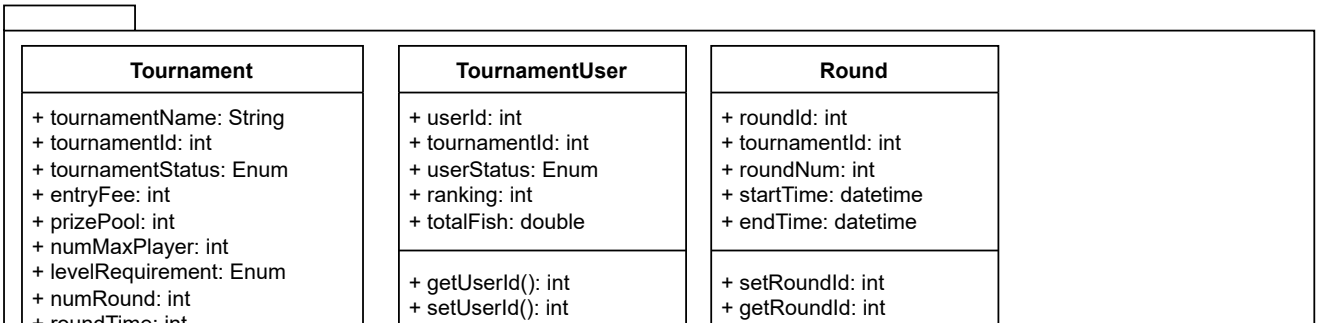
createHasPayment(HasPayment): void
removeHasPayment(HasPayment): void

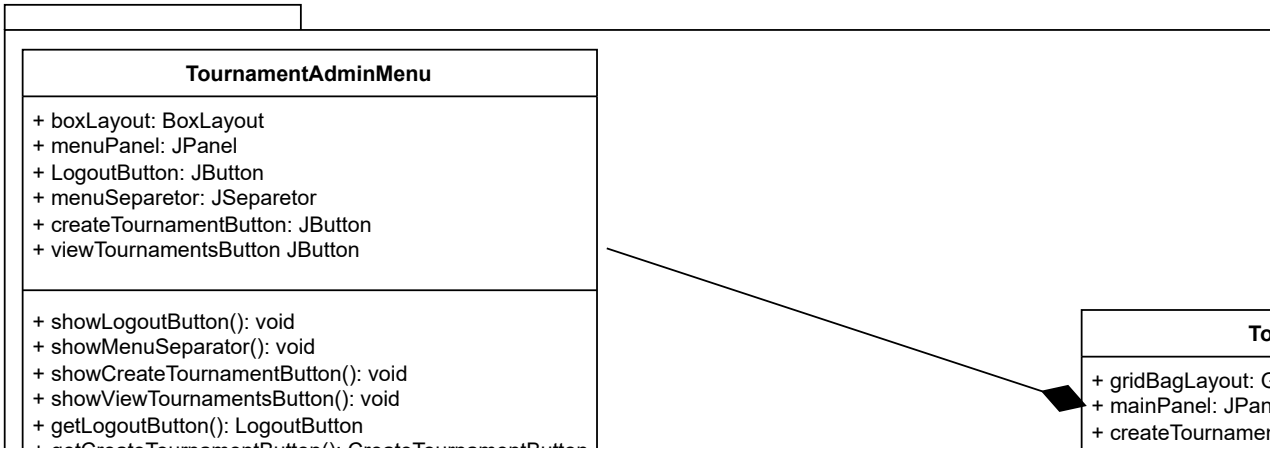
DBConnection

+ PROPERTYDBDRIVER: String
+ PROPERTYDBURL: String
+ PROPERTYNAME: String
+ PROPERTYPSW: String
+ username: String
+ password: String
+ dbDriver: String
+ dbURL: String
+ conn: DBConnection: String

+ init(): void
+ startConnection(Connection, String): Connection
+ isOpen(Connection): boolean
+ closeConnection(Connection): Connection

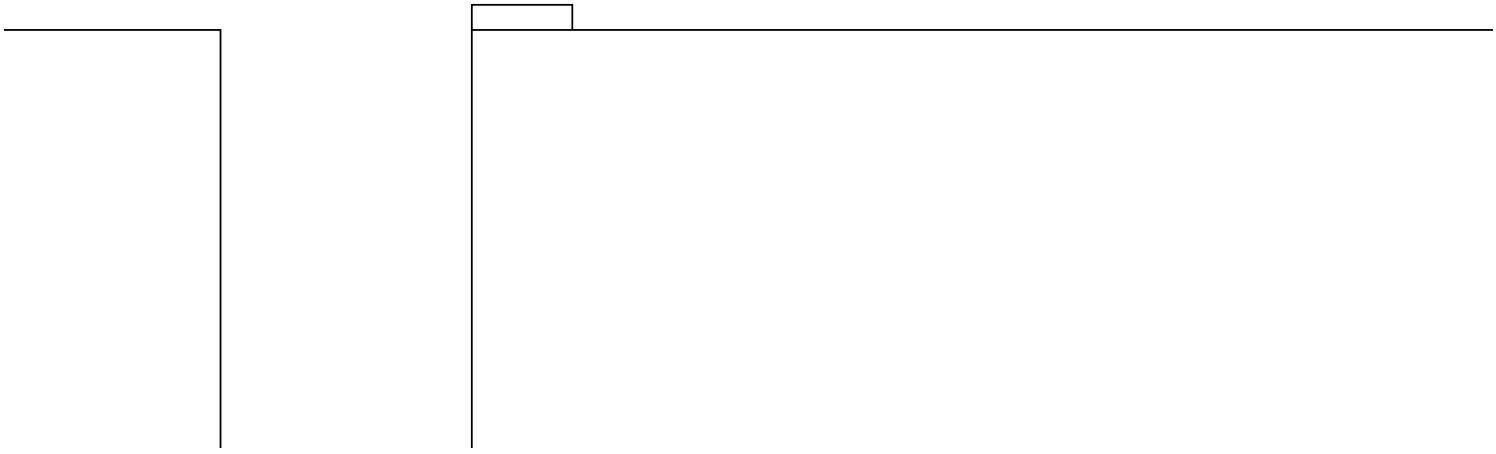


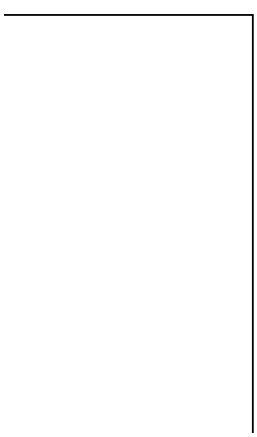

Tournament
+ tournamentName: String
+ tournamentId: int
+ tournamentStatus: Enum
+ entryFee: int
+ prizePool: int
+ numMaxPlayer: int
+ levelRequirement: Enum
+ numRound: int
TournamentUser
+ userId: int
+ tournamentId: int
+ userStatus: Enum
+ ranking: int
+ totalFish: double
+ getUserId(): int
+ setUserId(): int
Round
+ roundId: int
+ tournamentId: int
+ roundNum: int
+ startTime: datetime
+ endTime: datetime
+ setRoundId: int
+ getRoundId: int

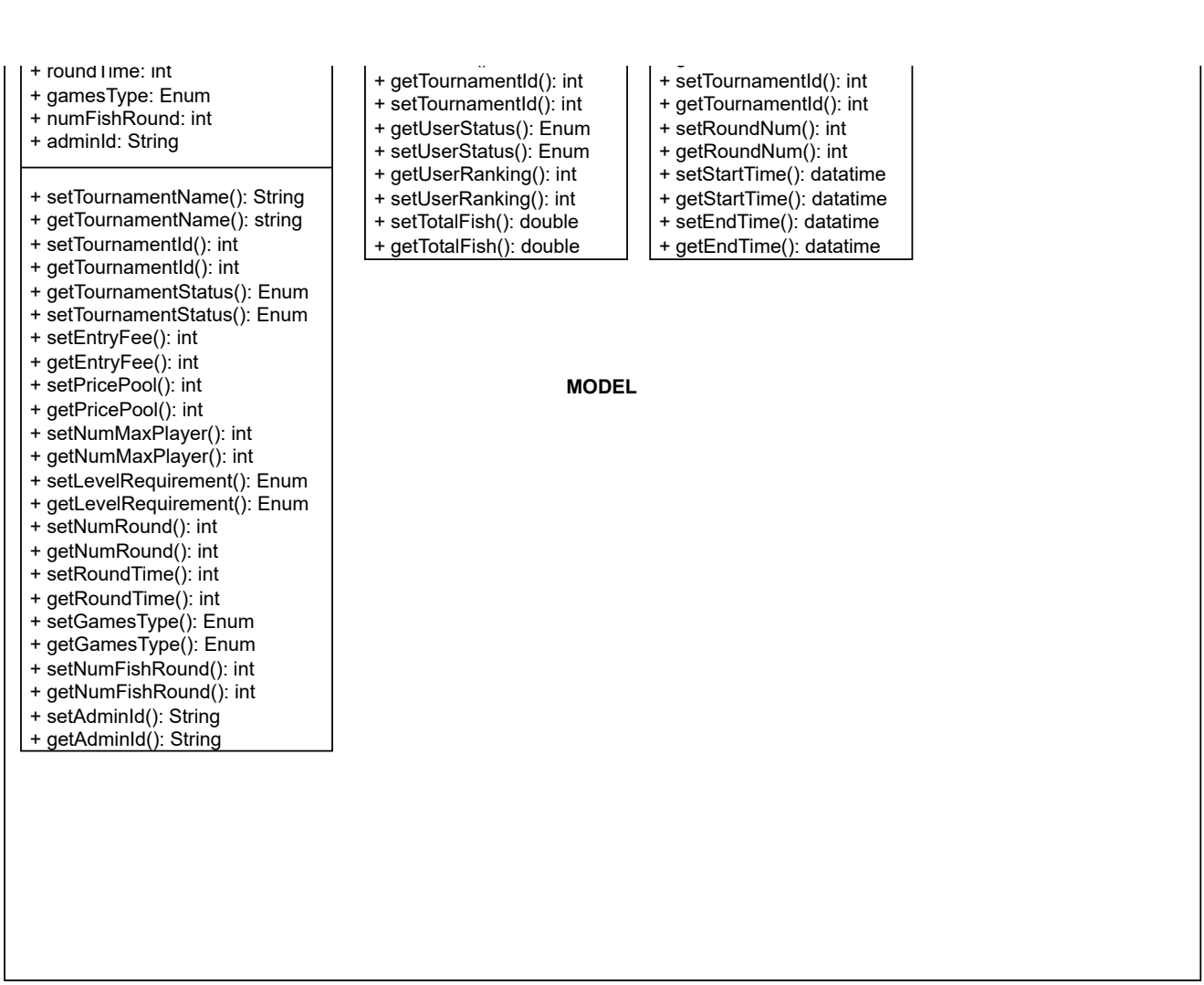

TournamentAdminMenu
+ boxLayout: BoxLayout
+ menuPanel: JPanel
+ LogoutButton: JButton
+ menuSeparetor: JSeparetor
+ createTournamentButton: JButton
+ viewTournamentsButton JButton
+ showLogoutButton(): void
+ showMenuSeparator(): void
+ showCreateTournamentButton(): void
+ showViewTournamentsButton(): void
+ getLogoutButton(): LogoutButton
To
+ gridBagLayout: G
+ mainPanel: JPan
+ createTournamer

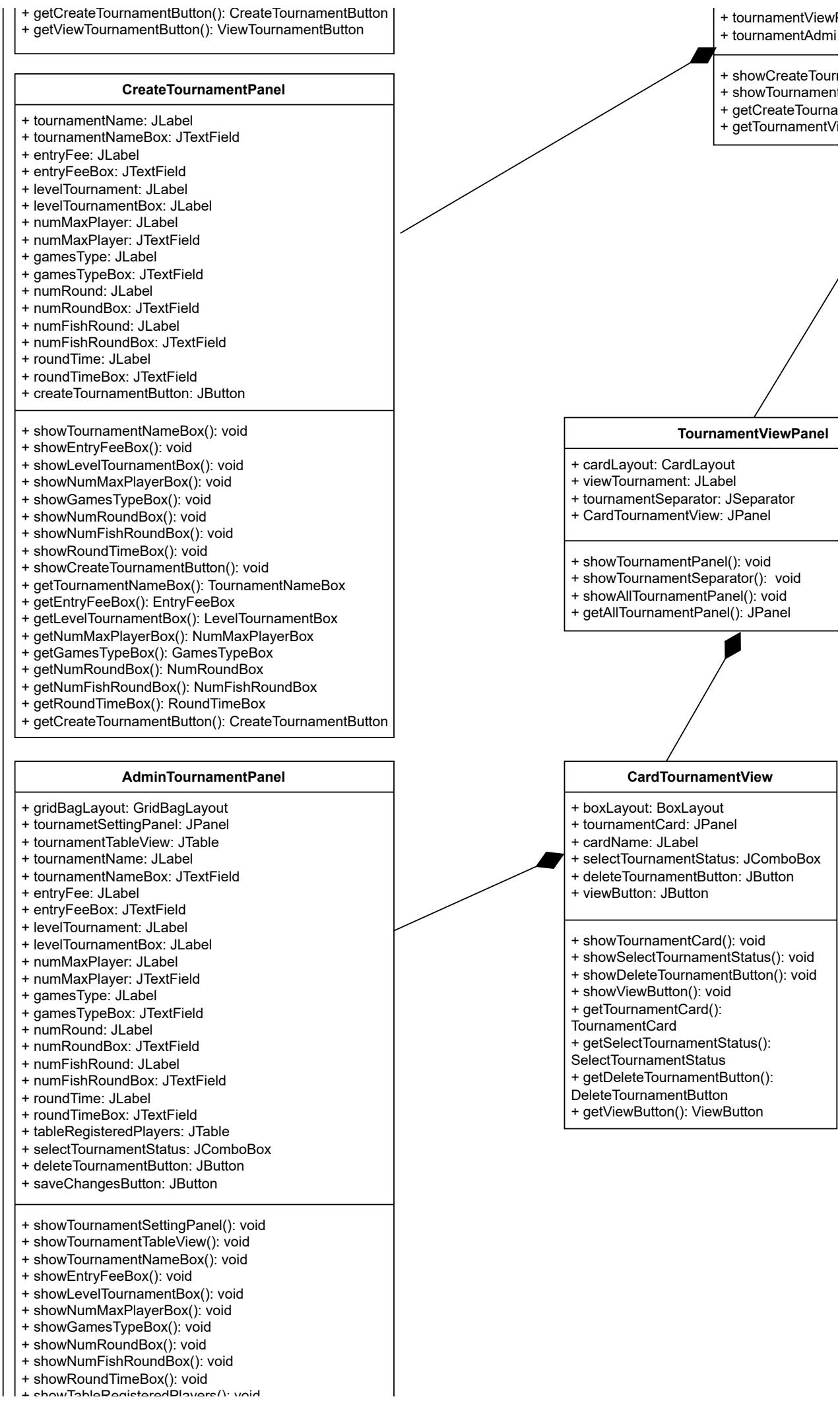
ournamentAdminView

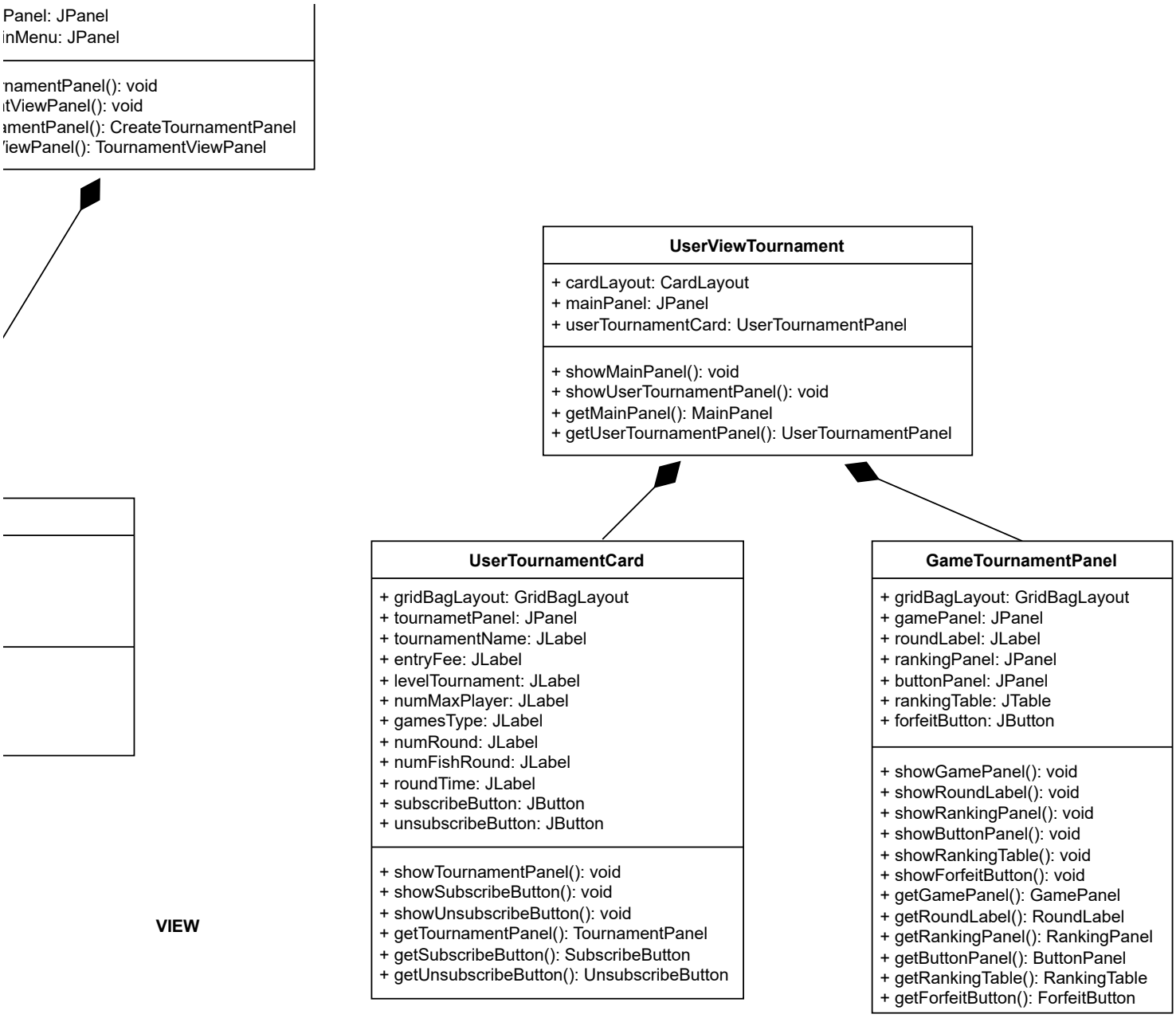
GridBagLayout
nel
ntPanel: JPanel



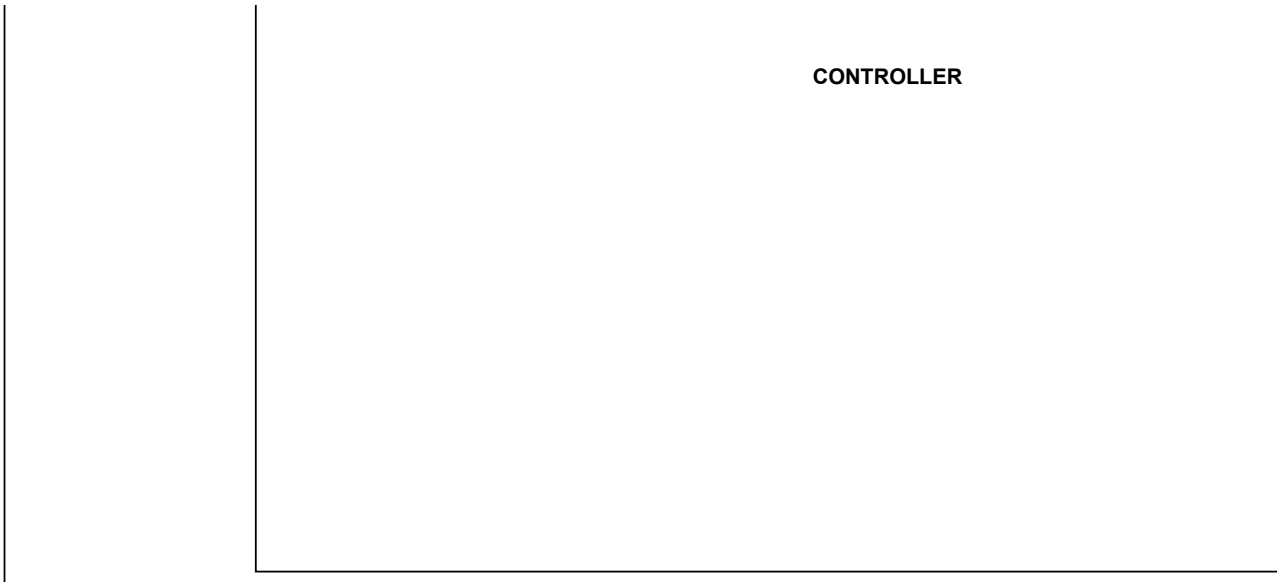


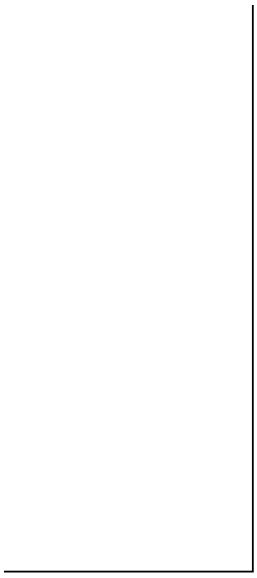

+ roundTime: int
+ gamesType: Enum
+ numFishRound: int
+ adminId: String
+ setTournamentName(): String
+ getTournamentName(): string
+ setTournamentId(): int
+ getTournamentId(): int
+ getTournamentStatus(): Enum
+ setTournamentStatus(): Enum
+ setEntryFee(): int
+ getEntryFee(): int
+ setPricePool(): int
+ getPricePool(): int
+ setNumMaxPlayer(): int
+ getNumMaxPlayer(): int
+ setLevelRequirement(): Enum
+ getLevelRequirement(): Enum
+ setNumRound(): int
+ getNumRound(): int
+ setRoundTime(): int
+ getRoundTime(): int
+ setGamesType(): Enum
+ getGamesType(): Enum
+ setNumFishRound(): int
+ getNumFishRound(): int
+ setAdminId(): String
+ getAdminId(): String
+ getTournamentId(): int
+ setTournamentId(): int
+ getUserStatus(): Enum
+ setUserStatus(): Enum
+ getUserRanking(): int
+ setUserRanking(): int
+ setTotalFish(): double
+ getTotalFish(): double
+ setTournamentId(): int
+ getTournamentId(): int
+ setRoundNum(): int
+ getRoundNum(): int
+ setStartTime(): datatime
+ getStartTime(): datatime
+ setEndTime(): datatime
+ getEndTime(): datatime
MODEL

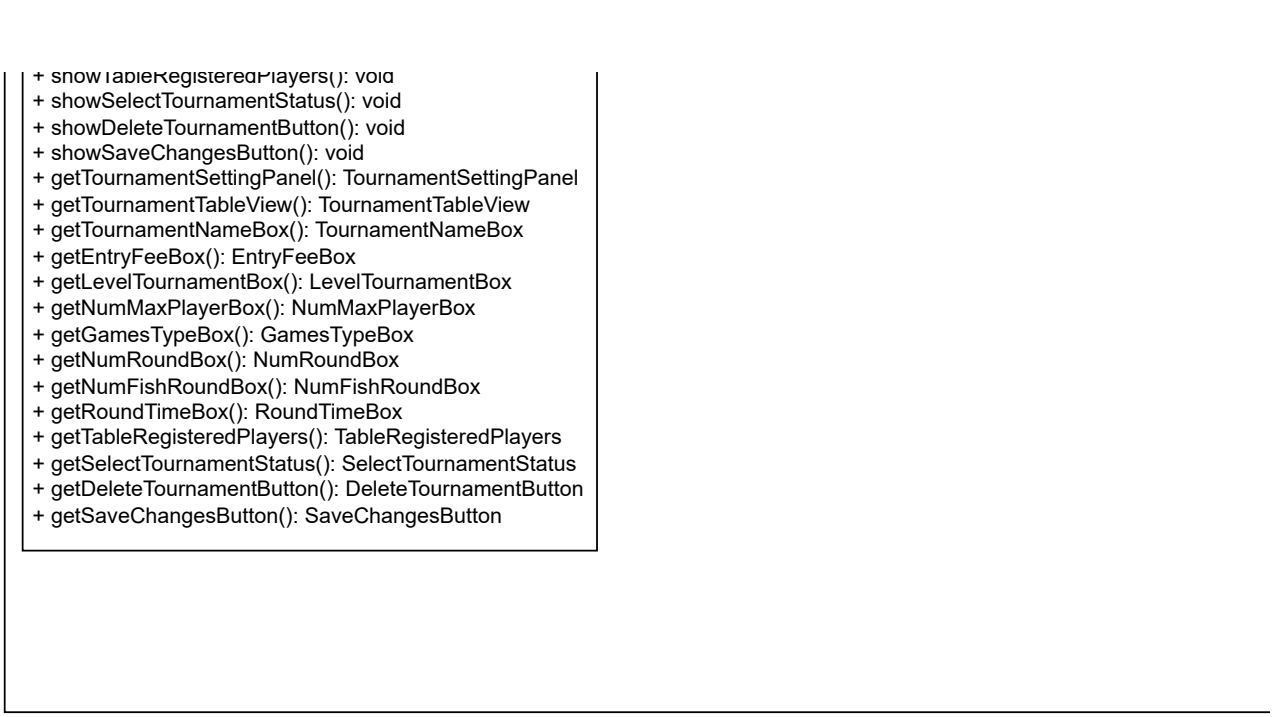
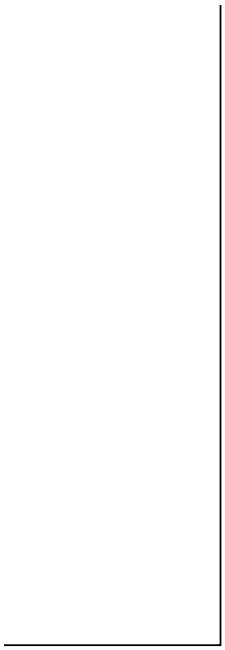

+ getCreateTournamentButton(): CreateTournamentButton
+ getViewTournamentButton(): ViewTournamentButton
+ tournamentViewP
+ tournamentAdmi
+ showCreateTourn
+ showTournament
+ getCreateTourna
+ getTournamentVi
CreateTournamentPanel
+ tournamentName: JLabel
+ tournamentNameBox: JTextField
+ entryFee: JLabel
+ entryFeeBox: JTextField
+ levelTournament: JLabel
+ levelTournamentBox: JLabel
+ numMaxPlayer: JLabel
+ numMaxPlayer: JTextField
+ gamesType: JLabel
+ gamesTypeBox: JTextField
+ numRound: JLabel
+ numRoundBox: JTextField
+ numFishRound: JLabel
+ numFishRoundBox: JTextField
+ roundTime: JLabel
+ roundTimeBox: JTextField
+ createTournamentButton: JButton
+ showTournamentNameBox(): void
+ showEntryFeeBox(): void
+ showLevelTournamentBox(): void
+ showNumMaxPlayerBox(): void
+ showGamesTypeBox(): void
+ showNumRoundBox(): void
+ showNumFishRoundBox(): void
+ showRoundTimeBox(): void
+ showCreateTournamentButton(): void
+ getTournamentNameBox(): TournamentNameBox
+ getEntryFeeBox(): EntryFeeBox
+ getLevelTournamentBox(): LevelTournamentBox
+ getNumMaxPlayerBox(): NumMaxPlayerBox
+ getGamesTypeBox(): GamesTypeBox
+ getNumRoundBox(): NumRoundBox
+ getNumFishRoundBox(): NumFishRoundBox
+ getRoundTimeBox(): RoundTimeBox
+ getCreateTournamentButton(): CreateTournamentButton
TournamentViewPanel
+ cardLayout: CardLayout
+ viewTournament: JLabel
+ tournamentSeparator: JSeparator
+ CardTournamentView: JPanel
+ showTournamentPanel(): void
+ showTournamentSeparator(): void
+ showAllTournamentPanel(): void
+ getAllTournamentPanel(): JPanel
AdminTournamentPanel
+ gridBagLayout: GridBagLayout
+ tournametSettingPanel: JPanel
+ tournamentTableView: JTable
+ tournamentName: JLabel
+ tournamentNameBox: JTextField
+ entryFee: JLabel
+ entryFeeBox: JTextField
+ levelTournament: JLabel
+ levelTournamentBox: JLabel
+ numMaxPlayer: JLabel
+ numMaxPlayer: JTextField
+ gamesType: JLabel
+ gamesTypeBox: JTextField
+ numRound: JLabel
+ numRoundBox: JTextField
+ numFishRound: JLabel
+ numFishRoundBox: JTextField
+ roundTime: JLabel
+ roundTimeBox: JTextField
+ tableRegisteredPlayers: JTable
+ selectTournamentStatus: JComboBox
+ deleteTournamentButton: JButton
+ saveChangesButton: JButton
+ showTournamentSettingPanel(): void
+ showTournamentTableView(): void
+ showTournamentNameBox(): void
+ showEntryFeeBox(): void
+ showLevelTournamentBox(): void
+ showNumMaxPlayerBox(): void
+ showGamesTypeBox(): void
+ showNumRoundBox(): void
+ showNumFishRoundBox(): void
+ showRoundTimeBox(): void
CardTournamentView
+ boxLayout: BoxLayout
+ tournamentCard: JPanel
+ cardName: JLabel
+ selectTournamentStatus: JComboBox
+ deleteTournamentButton: JButton
+ viewButton: JButton
+ showTournamentCard(): void
+ showSelectTournamentStatus(): void
+ showDeleteTournamentButton(): void
+ showViewButton(): void
+ getTournamentCard(): TournamentCard
+ getSelectTournamentStatus(): SelectTournamentStatus
+ getDeleteTournamentButton(): DeleteTournamentButton
+ getViewButton(): ViewButton


Panel: JPanel
inMenu: JPanel
namentPanel(): void
tViewPanel(): void
amentPanel(): CreateTournamentPanel
iewPanel(): TournamentViewPanel
UserViewTournament
+ cardLayout: CardLayout
+ mainPanel: JPanel
+ userTournamentCard: UserTournamentPanel
+ showMainPanel(): void
+ showUserTournamentPanel(): void
+ getMainPanel(): MainPanel
+ getUserTournamentPanel(): UserTournamentPanel
UserTournamentCard
+ gridBagLayout: GridBagLayout
+ tournametPanel: JPanel
+ tournamentName: JLabel
+ entryFee: JLabel
+ levelTournament: JLabel
+ numMaxPlayer: JLabel
+ gamesType: JLabel
+ numRound: JLabel
+ numFishRound: JLabel
+ roundTime: JLabel
+ subscribeButton: JButton
+ unsubscribeButton: JButton
+ showTournamentPanel(): void
+ showSubscribeButton(): void
+ showUnsubscribeButton(): void
+ getTournamentPanel(): TournamentPanel
+ getSubscribeButton(): SubscribeButton
+ getUnsubscribeButton(): UnsubscribeButton
GameTournamentPanel
+ gridBagLayout: GridBagLayout
+ gamePanel: JPanel
+ roundLabel: JLabel
+ rankingPanel: JPanel
+ buttonPanel: JPanel
+ rankingTable: JTable
+ forfeitButton: JButton
+ showGamePanel(): void
+ showRoundLabel(): void
+ showRankingPanel(): void
+ showButtonPanel(): void
+ showRankingTable(): void
+ showForfeitButton(): void
+ getGamePanel(): GamePanel
+ getRoundLabel(): RoundLabel
+ getRankingPanel(): RankingPanel
+ getButtonPanel(): ButtonPanel
+ getRankingTable(): RankingTable
+ getForfeitButton(): ForfeitButton

VIEW


CONTROLLER




+ showTableRegisteredPlayers(): void
+ showSelectTournamentStatus(): void
+ showDeleteTournamentButton(): void
+ showSaveChangesButton(): void
+ getTournamentSettingPanel(): TournamentSettingPanel
+ getTournamentTableView(): TournamentTableView
+ getTournamentNameBox(): TournamentNameBox
+ getEntryFeeBox(): EntryFeeBox
+ getLevelTournamentBox(): LevelTournamentBox
+ getNumMaxPlayerBox(): NumMaxPlayerBox
+ getGamesTypeBox(): GamesTypeBox
+ getNumRoundBox(): NumRoundBox
+ getNumFishRoundBox(): NumFishRoundBox
+ getRoundTimeBox(): RoundTimeBox
+ getTableRegisteredPlayers(): TableRegisteredPlayers
+ getSelectTournamentStatus(): SelectTournamentStatus
+ getDeleteTournamentButton(): DeleteTournamentButton
+ getSaveChangesButton(): SaveChangesButton