

**Title:** Online Property Management System Design Phase

**Names:** Dat Lam, Zhifan Li, Evan Lester, Jacob Lansang

**Group Number:** 11

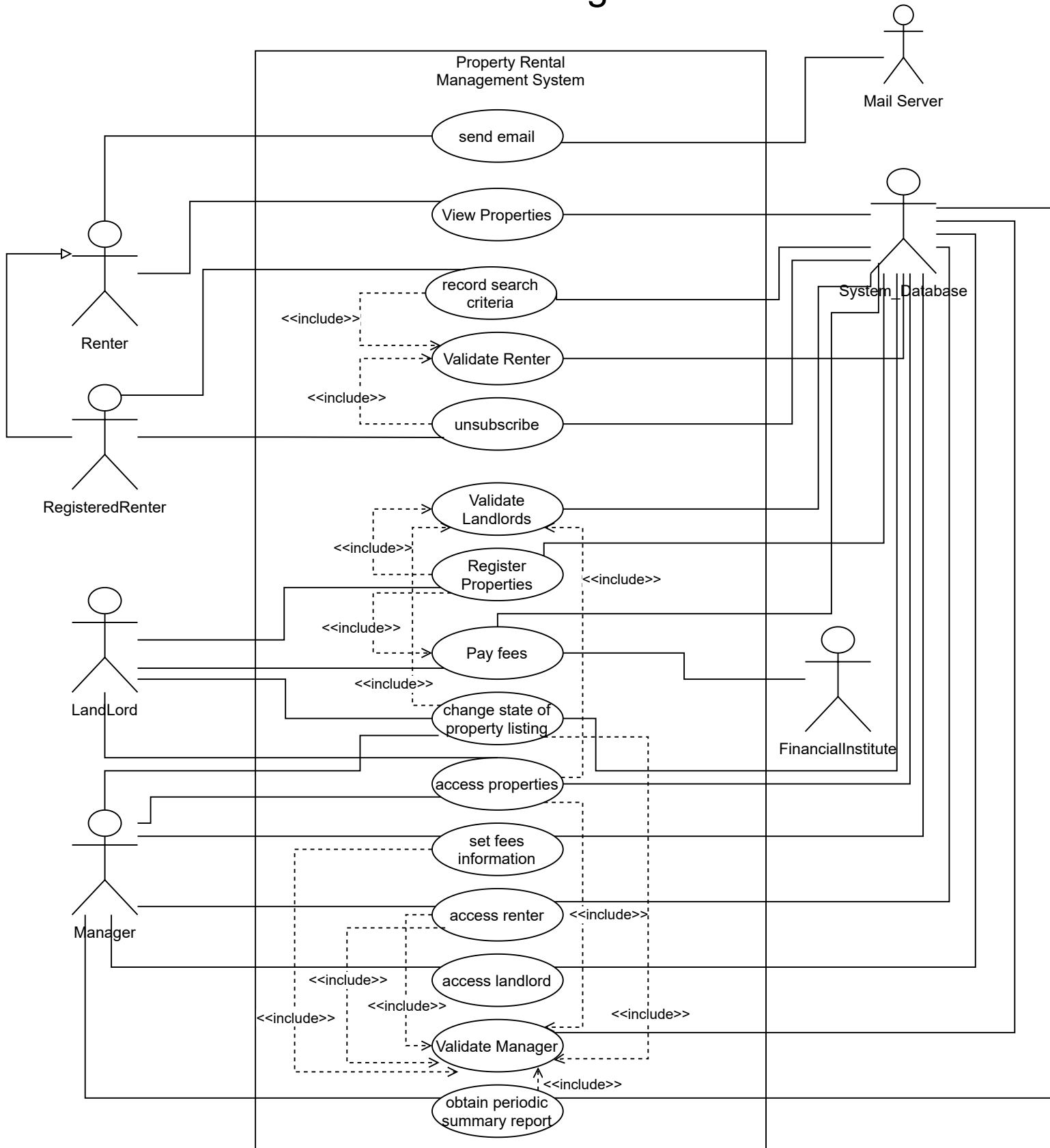
**Course Name:** Principles of Software Design

**Lab Section:** B01

**Course Code:** ENSF 480

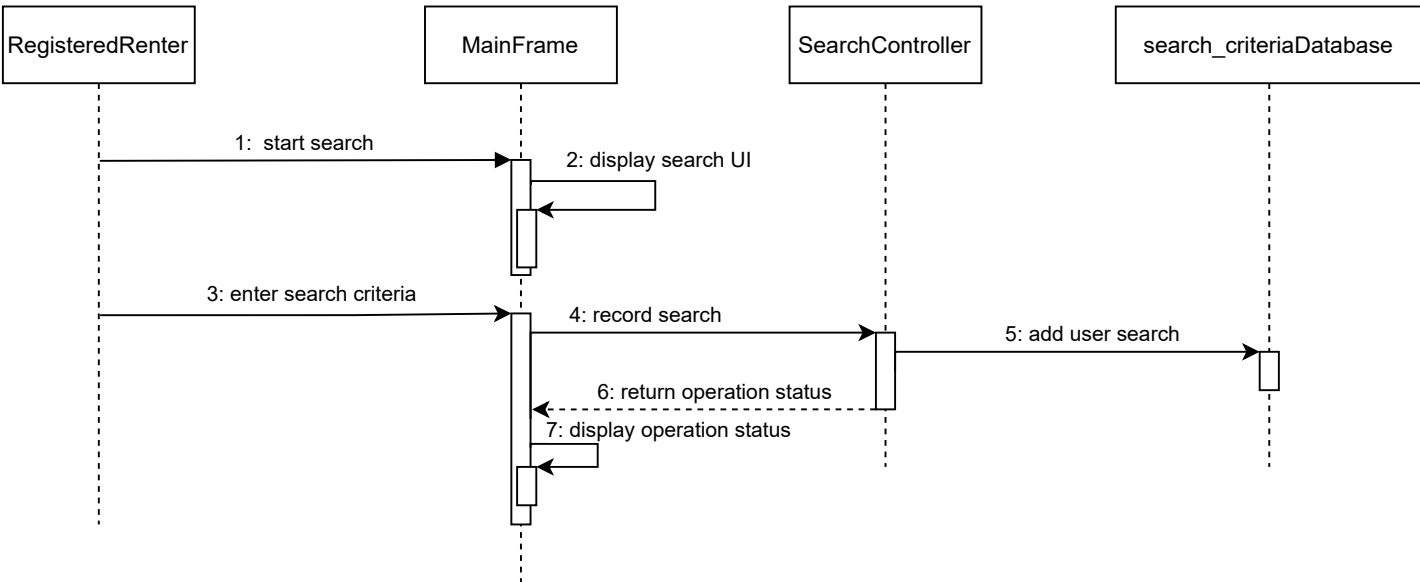
**Submission Date:** 29/11/2021

# Use Case Diagram

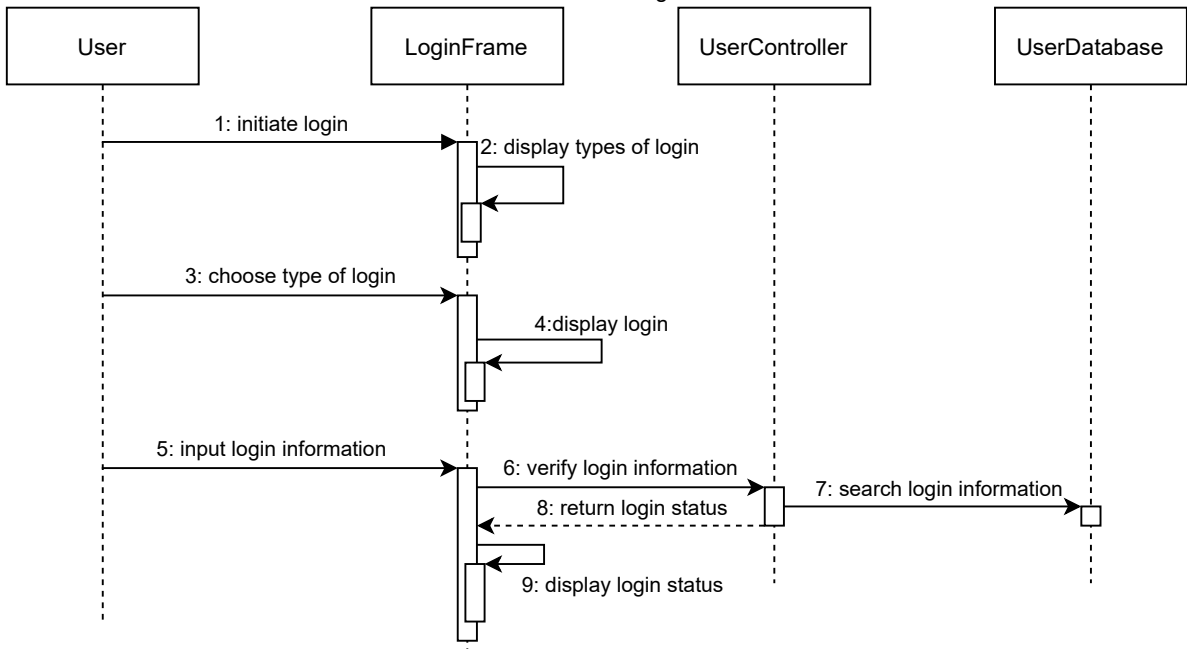


# Interaction Diagram

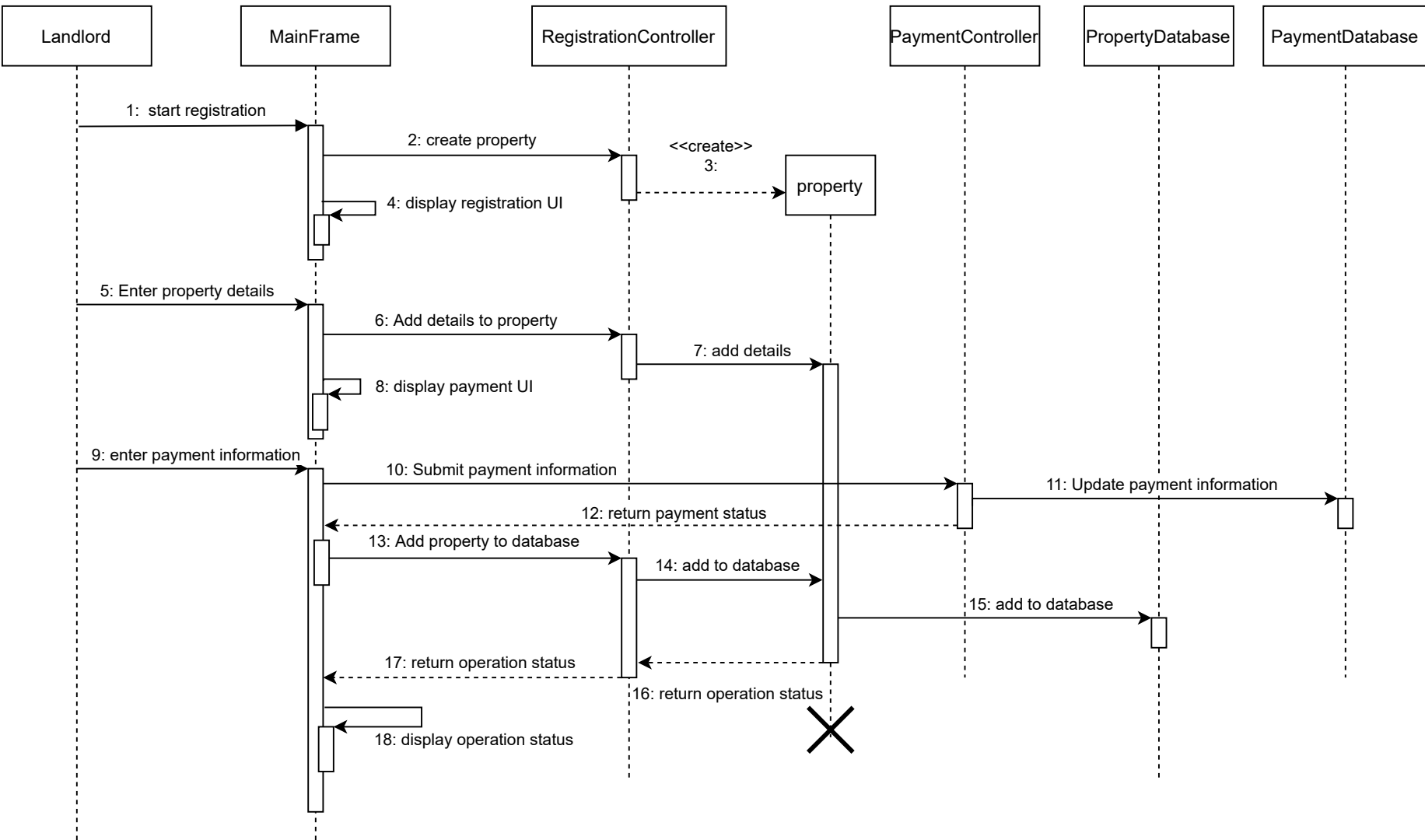
Registered Renter record and subscribe



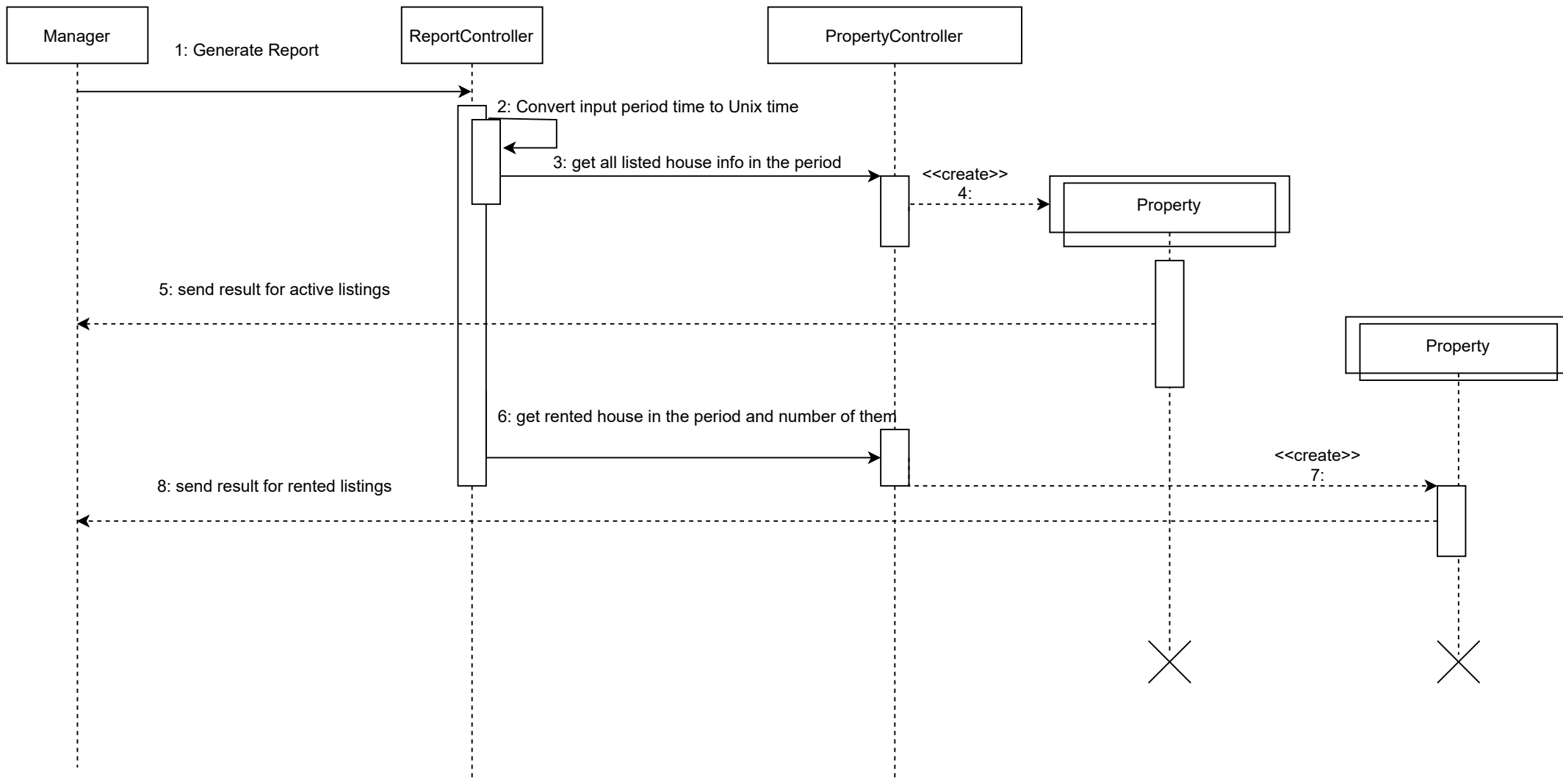
User Login



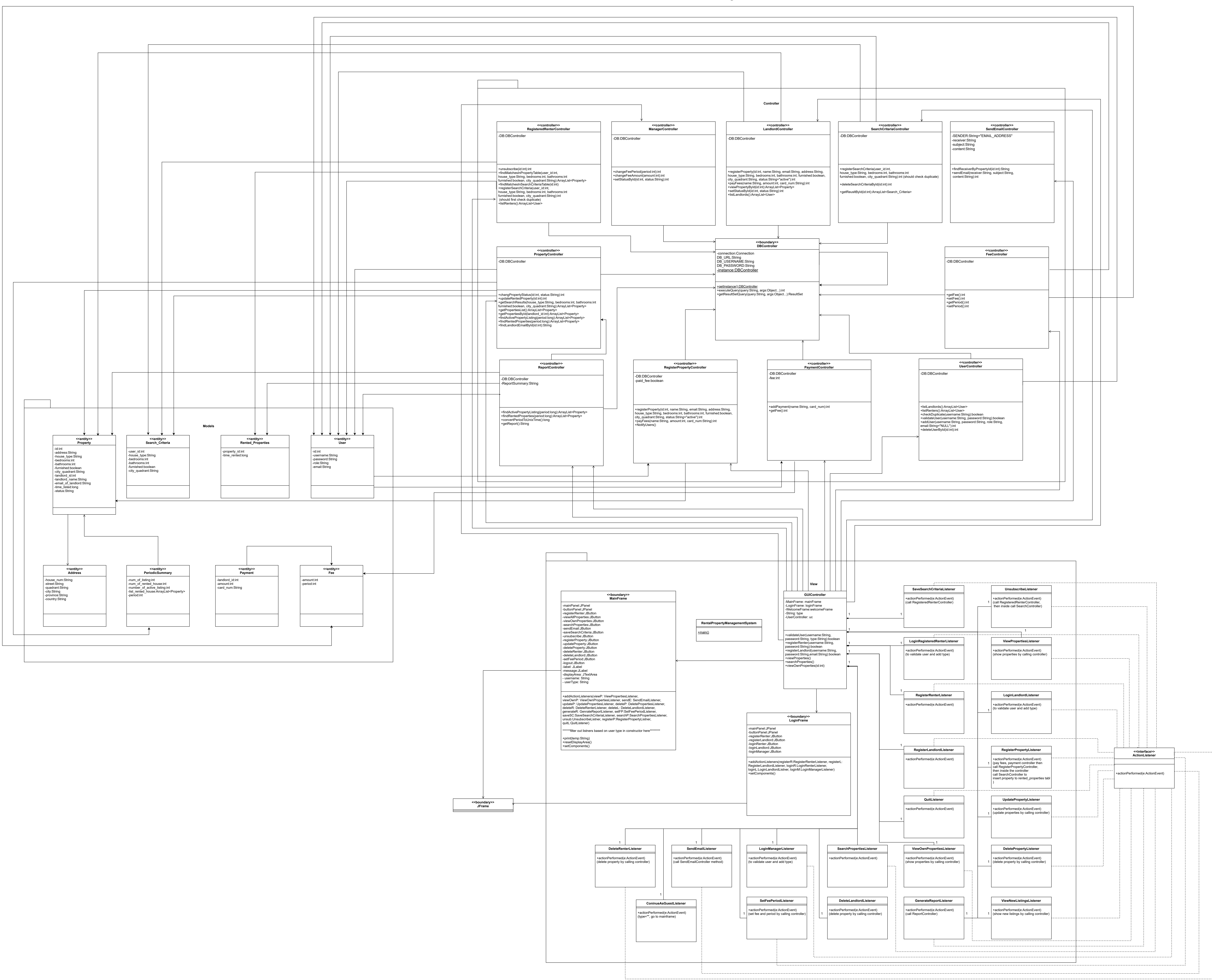
Landlord register property



Manager Request Periodic Report



Class Diagram



# Class Specification Controller Package

<<controller>> RegisteredRenterController
-DB:DBController
+unsubscribe(id:int):int +findMatchesInPropertyTable(user_id:int, house_type:String, bedrooms:int, bathrooms:int, furnished:boolean, city_quadrant:String):ArrayList<Property> +findMatchesInSearchCriteriaTable(id:int) +registerSearchCriteria(user_id:int, house_type:String, bedrooms:int, bathrooms:int, furnished:boolean, city_quadrant:String):int (should first check duplicate) +listRenters():ArrayList<User>

<<boundary>> DBController
-connection:Connection DB_URL:String DB_USERNAME:String DB_PASSWORD:String <u>-instance:DBController</u>
+getInstance():DBController +executeQuery(query:String, args:Object...):int +getResultSetQuery(query:String, args:Object...):ResultSet

<<controller>> LandlordController
-DB:DBController
+registerProperty(id:int, name:String, email:String, address:String, house_type:String, bedrooms:int, bathrooms:int, furnished:boolean, city_quadrant:String, status:String="active"):int +payFees(name:String, amount:int, card_num:String):int +viewPropertyById(id:int):ArrayList<Property> +setStatusById(id:int, status:String):int +listLandlords():ArrayList<User>

<<controller>> SendEmailController
-SENDER:String="EMAIL_ADDRESS" -receiver:String -subject:String -content:String
+findReceiverByPropertyId(id:int):String +sendEmail(receiver:String, subject:String, content:String):int

<<controller>> RegisterPropertyController
-DB:DBController -paid_fee:boolean
+registerProperty(id:int, name:String, email:String, address:String, house_type:String, bedrooms:int, bathrooms:int, furnished:boolean, city_quadrant:String, status:String="active"):int +payFees(name:String, amount:int, card_num:String):int +NotifyUsers()

<<controller>> UserController
-DB:DBController
+listLandlords():ArrayList<User> +listRenters():ArrayList<User> +checkDuplicate(username:String):boolean +validateUser(username:String, password:String):boolean +addUser(username:String, password:String, role:String, email:String="NULL"):int +deleteUserById(id:int):int

<<controller>> PropertyController
-DB:DBController
+changPropertyStatus(id:int, status:String):int +updateRentedProperty(id:int):int +getSearchResults(house_type:String, bedrooms:int, bathrooms:int, furnished:boolean, city_quadrant:String):ArrayList<Property> +getPropertiesList():ArrayList<Property> +getPropertiesById(landlord_id:int):ArrayList<Property> +findActivePropertyListing(period:long):ArrayList<Property> +findRentedProperties(period:long):ArrayList<Property> +findLandlordEmailById(id:int):String

<<controller>> ManagerController
-DB:DBController
+changeFeePeriod(period:int):int +changeFeeAmount(amount:int):int +setStatusById(id:int, status:String):int

<<controller>> FeeController
-DB:DBController
+getFee():int +setFee():int +getPeriod():int +setPeriod():int

<<controller>> SearchCriteriaController
-DB:DBController
+registerSearchCriteria(user_id:int, house_type:String, bedrooms:int, bathrooms:int, furnished:boolean, city_quadrant:String):int (should check duplicate)  +deleteSearchCriteriaById(id:int):int  +getReusltById(id:int):ArrayList<Search_Criteria>

<<controller>> PaymentController
-DB:DBController -fee:int
+addPayment(name:String, card_num):int +getFee():int

<<controller>> ReportController
-DB:DBController -ReportSummary:String
+findActivePropertyListing(period:long):ArrayList<Property> +findRentedProperties(period:long):ArrayList<Property> +convertPeriodToUnixTime():long +getReport():String

# Class Specification Models Package

<<entity>> Payment
-landlord_id:int -amount:int -card_num:String

<<entity>> Address
-house_num:String -street:String -quadrant:String -city:String -province:String -country:String

<<entity>> Property
-id:int -address:String -house_type:String -bedrooms:int -bathrooms:int -furnished:boolean -city_quadrant:String -landlord_id:int -landlord_name:String -email_of_landlord:String -time_listed:long -status:String

<<entity>> PeriodicSummary
-num_of_listing:int -num_of_rented_house:int -number_of_active_listing:int -list_rented_house:ArrayList<Property> -period:int

<<entity>> Fee
-amount:int -period:int

<<entity>> Search_Criteria
-user_id:int -house_type:String -bedrooms:int -bathrooms:int -furnished:boolean -city_quadrant:String

<<entity>> User
-id:int -username:String -password:String -role:String -email:String

<<entity>> Rented_Properties
-property_id:int -time_rented:long

RentalPropertyManagementSystem
<div>+main()</div>

<<boundary>> MainFrame
-mainPanel: JPanel -buttonPanel: JPanel -registerRenter: JButton -viewAllProperties: JButton -viewOwnProperties: JButton -searchProperties: JButton -sendEmail: JButton -saveSearchCriteria: JButton -unsubscribe: JButton -registerProperty: JButton -updateProperty: JButton -deleteProperty: JButton -deleteRenter: JButton -deleteLandlord: JButton -setFeePeriod: JButton -logout: JButton -label: JLabel -message: JLabel -displayArea: JTextArea - username: String - userType: String
+addActionListeners(viewP: ViewPropertiesListener, viewOwnP: ViewOwnPropertiesListener, sendE: SendEmailListener, updateP: UpdatePropertiesListener, deleteP: DeletePropertiesListener, deleteR: DeleteRenterListener, deleteL: DeleteLandlordListener, generateR: GenrateReportListener, setFP: SetFeePeriodListener, saveSC: SaveSearchCriteriaListener, searchP: SearchPropertiesListener, unsub: UnsubscribeListner, registerP: RegisterPropertyListner, quitL: QuitListener)  *****filter out listners based on user type in constructor here*****  +print(temp:String) +resetDisplayArea() +setComponents()

DeleteRenterListener
+actionPerformed(e:ActionEvent) (delete property by calling controller)

GenerateReportListener
+actionPerformed(e:ActionEvent) (call ReportController)

GUIController
-MainFrame: mainFrame -LoginFrame: loginFrame -WelcomeFrame: welcomeFrame -String: type -UserController: uc
+validateUser(username:String, password:String, type:String):boolean +registerRenter(username:String, password:String):boolean +registerLandlord(username:String, password:String,email:String):boolean +viewProperties() +searchProperties() +viewOwnProperties(id:int)

JFrame

SetFeePeriodListener
+actionPerformed(e:ActionEvent) (set fee and period by calling controller)

DeleteLandlordListener
+actionPerformed(e:ActionEvent) (delete property by calling controller)

<<interface>> ActionListener
+actionPerformed(e:ActionEvent)

# Class Specification View Package

RegisterRenterListener
+actionPerformed(e:ActionEvent)

RegisterLandlordListener
+actionPerformed(e:ActionEvent)

QuitListener
+actionPerformed(e:ActionEvent)

UnsubscribeListener
+actionPerformed(e:ActionEvent) (call RegisteredRenterController, then inside call SearchController)

RegisterPropertyListener
+actionPerformed(e:ActionEvent) (pay fees, payment controller then call RegisterPropertyController, then inside the controller call SearchController to insert property to rented_properties tabl )

LoginManagerListener
+actionPerformed(e:ActionEvent) (to validate user and add type)

LoginRegisteredRenterListener
+actionPerformed(e:ActionEvent) (to validate user and add type)

ConinueAsGuestListener
+actionPerformed(e:ActionEvent) (type="", go to mainframe)

UpdatePropertyListener
+actionPerformed(e:ActionEvent) (update properties by calling controller)

DeletePropertyListener
+actionPerformed(e:ActionEvent) (delete property by calling controller)

SearchPropertiesListener
+actionPerformed(e:ActionEvent)

ViewPropertiesListener
+actionPerformed(e:ActionEvent) (show properties by calling controller)

ViewOwnPropertiesListener
+actionPerformed(e:ActionEvent) (show properties by calling controller)

SendEmailListener
+actionPerformed(e:ActionEvent) (call SendEmailController method)

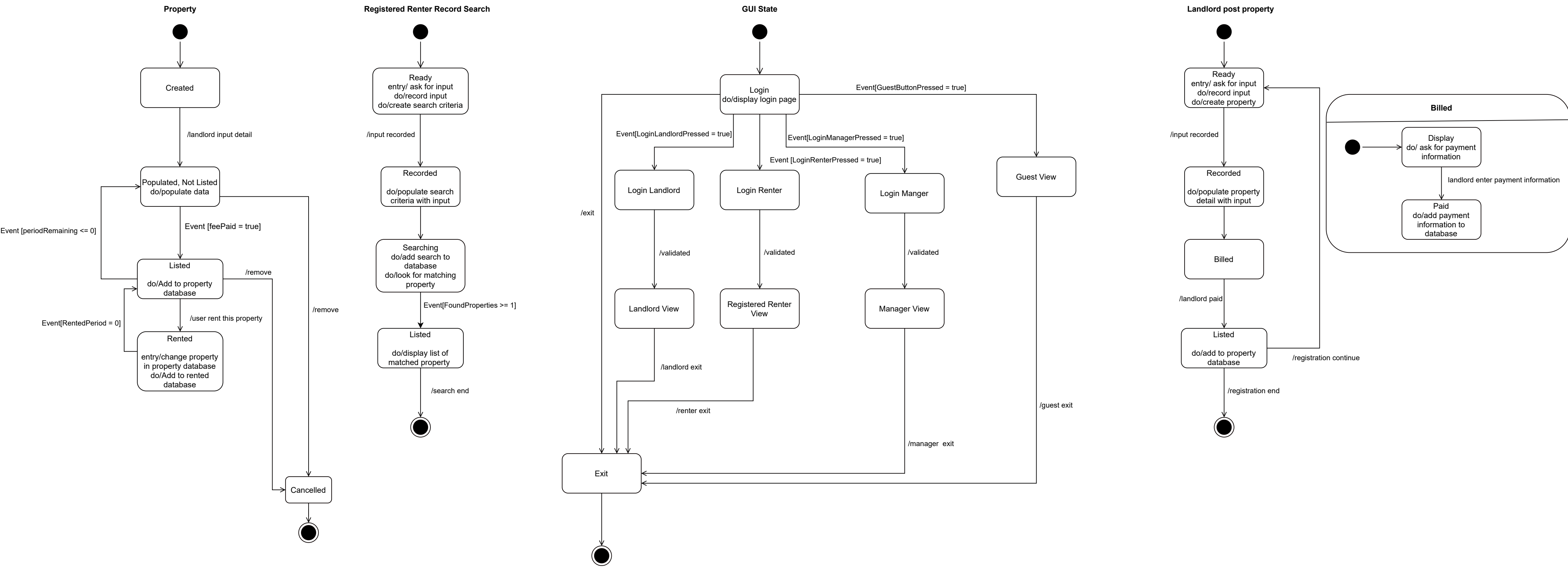
ViewNewListingsListener
+actionPerformed(e:ActionEvent) (show new listings by calling controller)

<<boundary>> LoginFrame
-mainPanel: JPanel -buttonPanel: JPanel -registerRenter: JButton -registerLandlord: JButton -loginRenter: JButton -loginLandlord: JButton -loginManager: JButton -loginGuest: JButton
+addActionListeners(registerR:RegisterRenterListener, registerL: RegisterLandlordListener, loginR: LoginRenterListener, loginL: LoginLandlordListner, loginM: LoginManagerListener, loginGuest: ConinueAsGuestListener) +setComponents()

LoginLandlordListener
+actionPerformed(e:ActionEvent) (to validate user and add type)

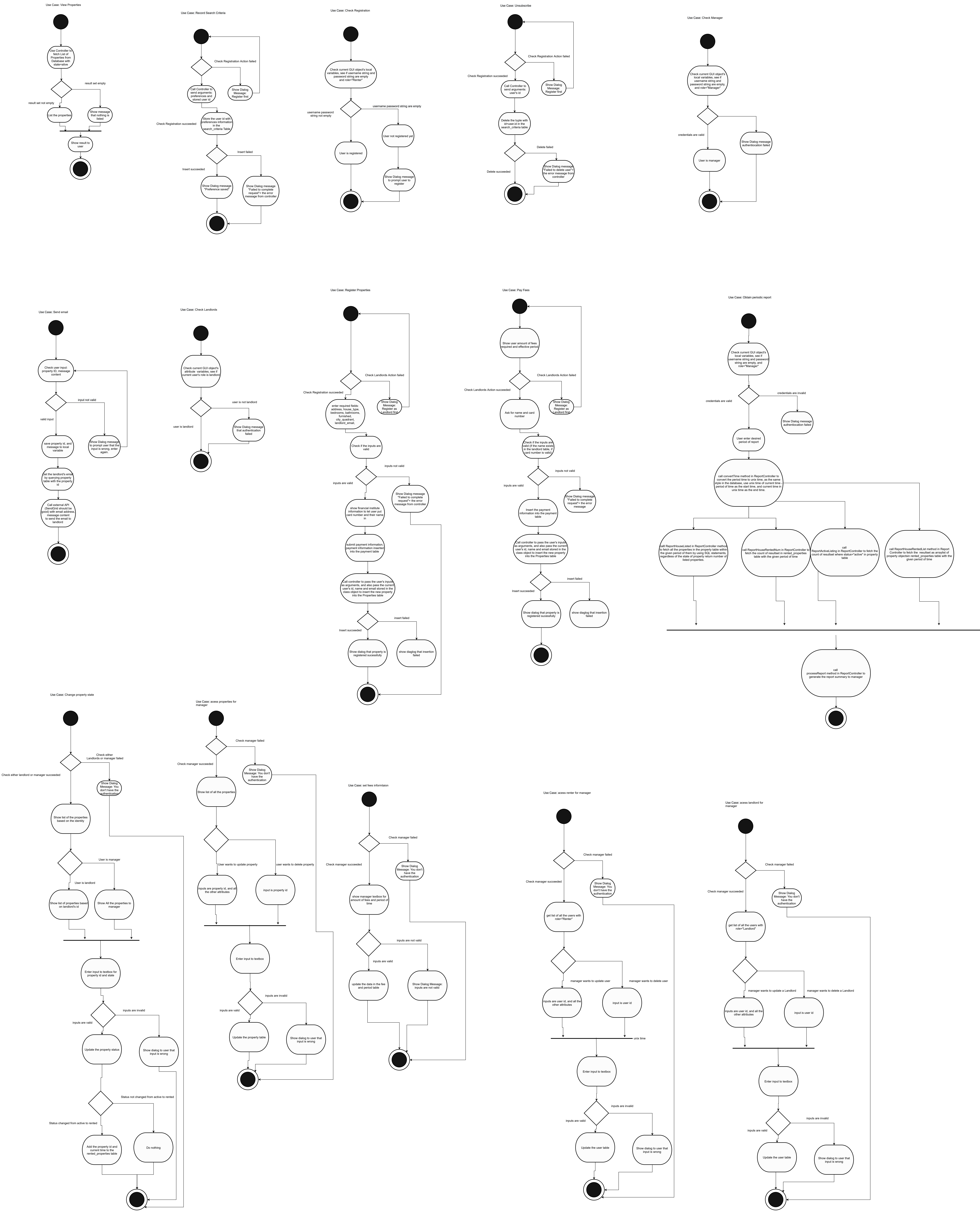
SaveSearchCriteriaListener
+actionPerformed(e:ActionEvent) (call RegisteredRenterController)

# State Transition Diagram

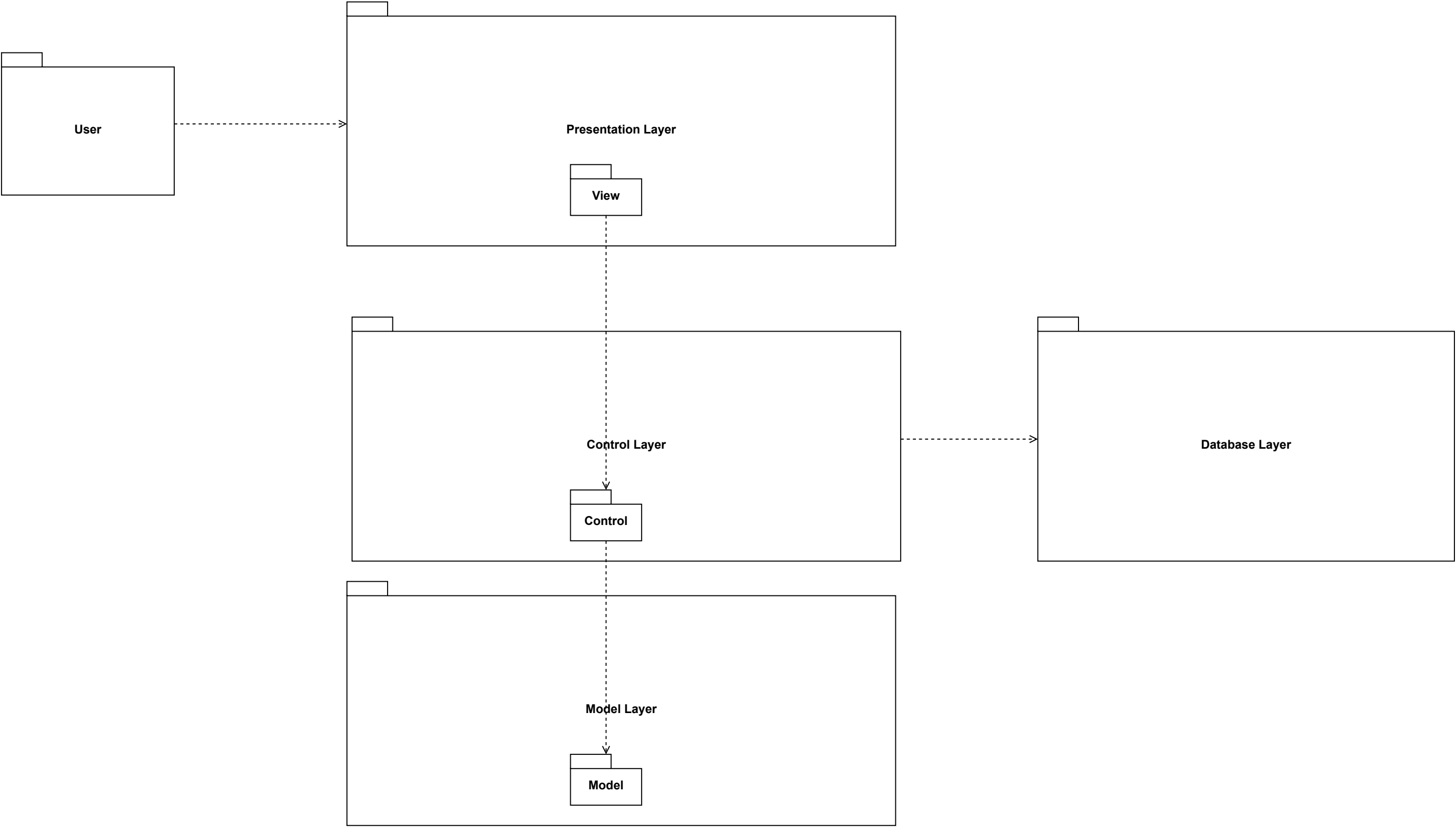




Activity diagram



# Package Diagram



# Deployment Diagram

