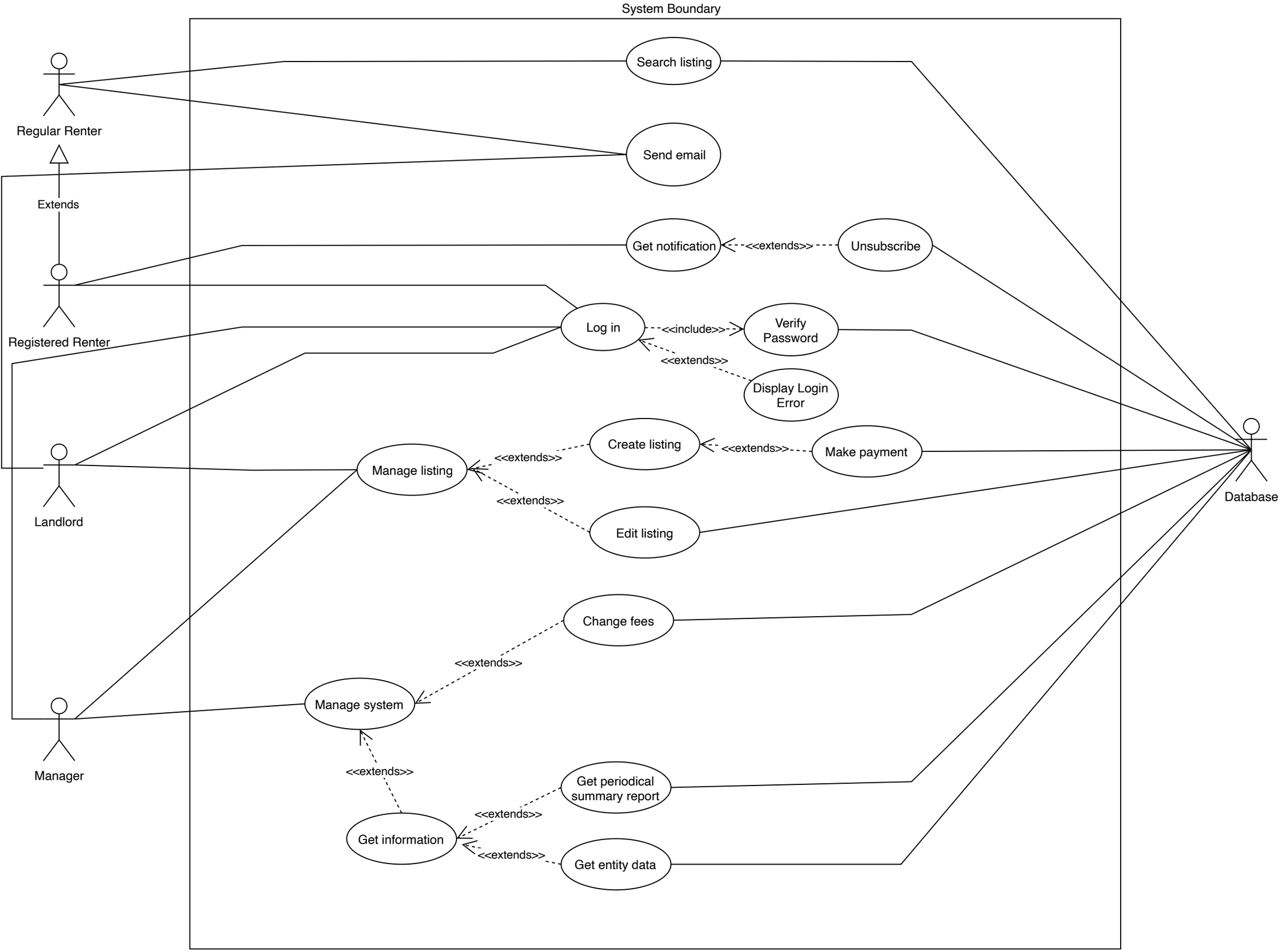
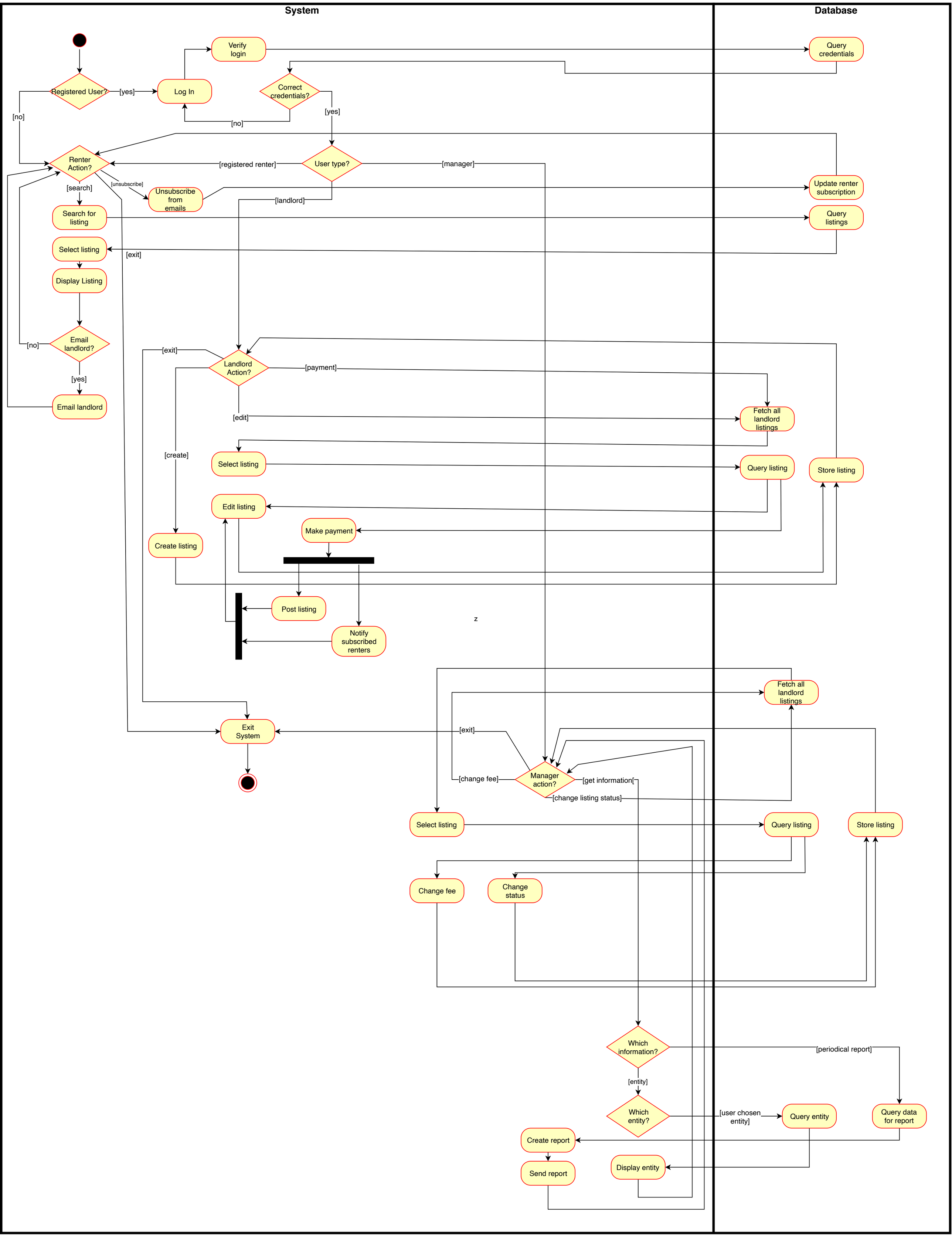
Names: Harsohail Brar (30041921)
Gary Wu (30038110)
Ryan Holt (30038609)
Course Name: Principles of Software Design
Course Code: ENSF 480
Assignment Number: Term Project Design Document
Submission Date and Time: 09/11/2019

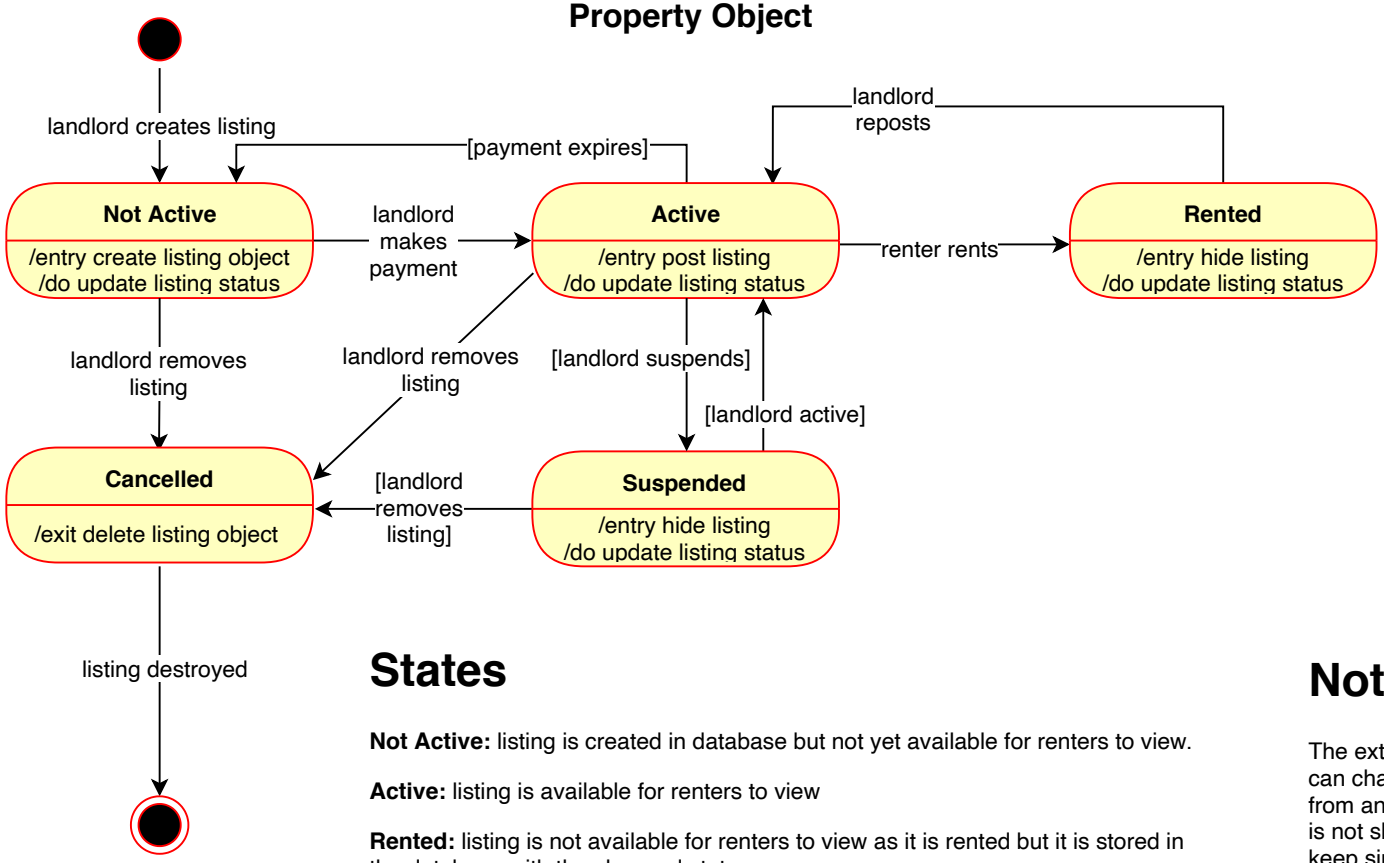
System's Use Case Diagram



System's Activity Diagram



State Transition Diagrams
Property Object



States

Not Active: listing is created in database but not yet available for renters to view.

Active: listing is available for renters to view

Rented: listing is not available for renters to view as it is rented but it is stored in the database with the changed status

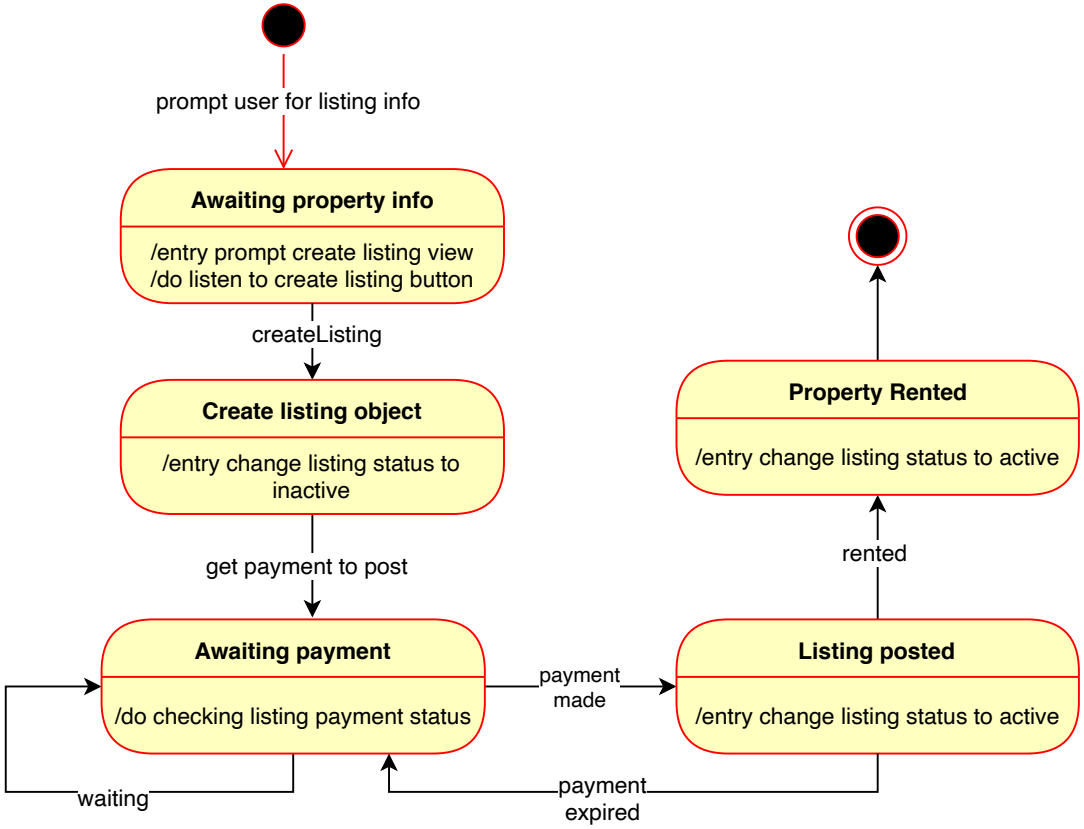
Cancelled: listing is deleted from the system

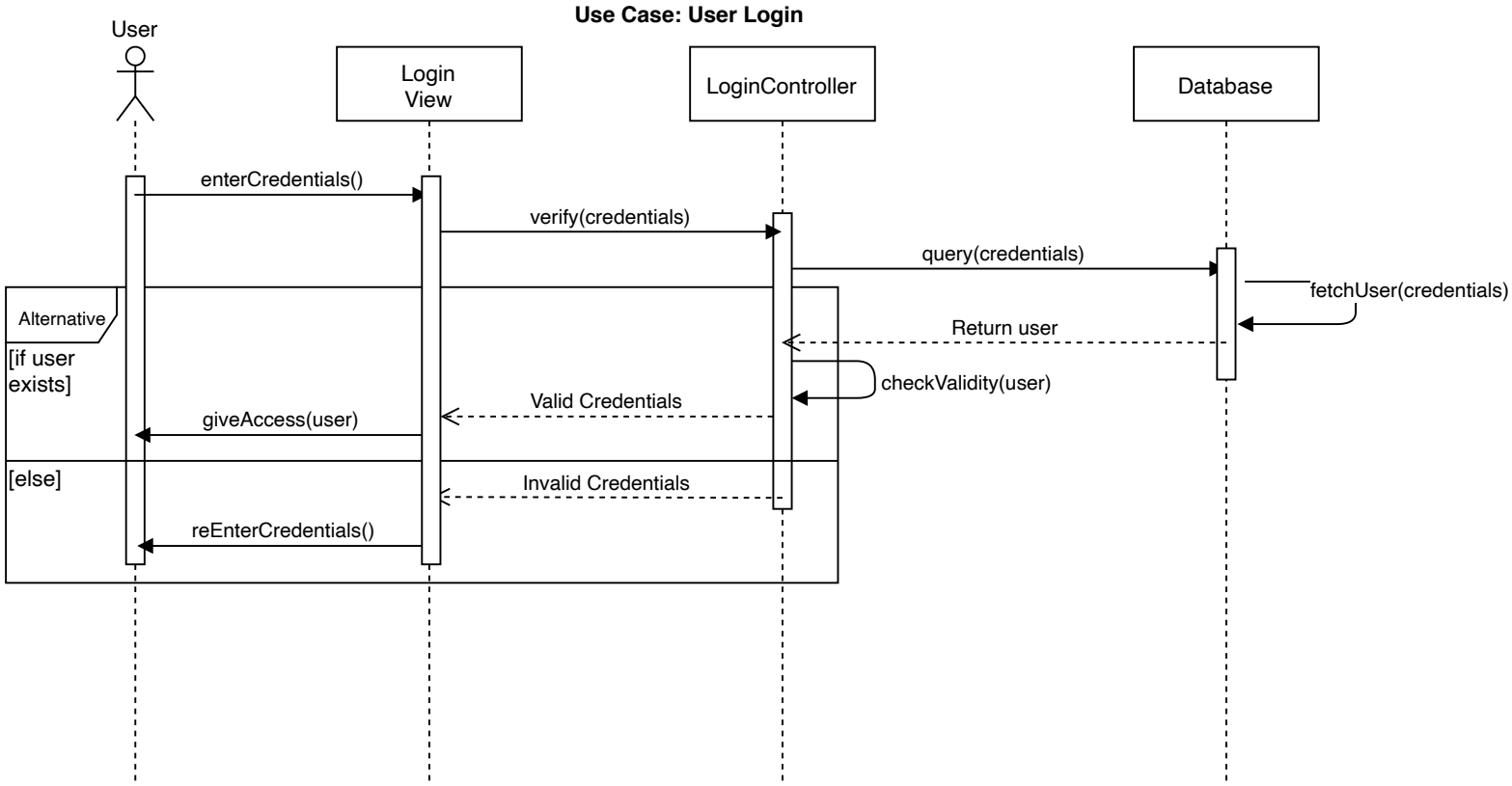
Suspended: landlord suspends the listing temporarly to hide it from renters viewing it. (ex. if landlord is the middle of signing a contract with a potential renter)

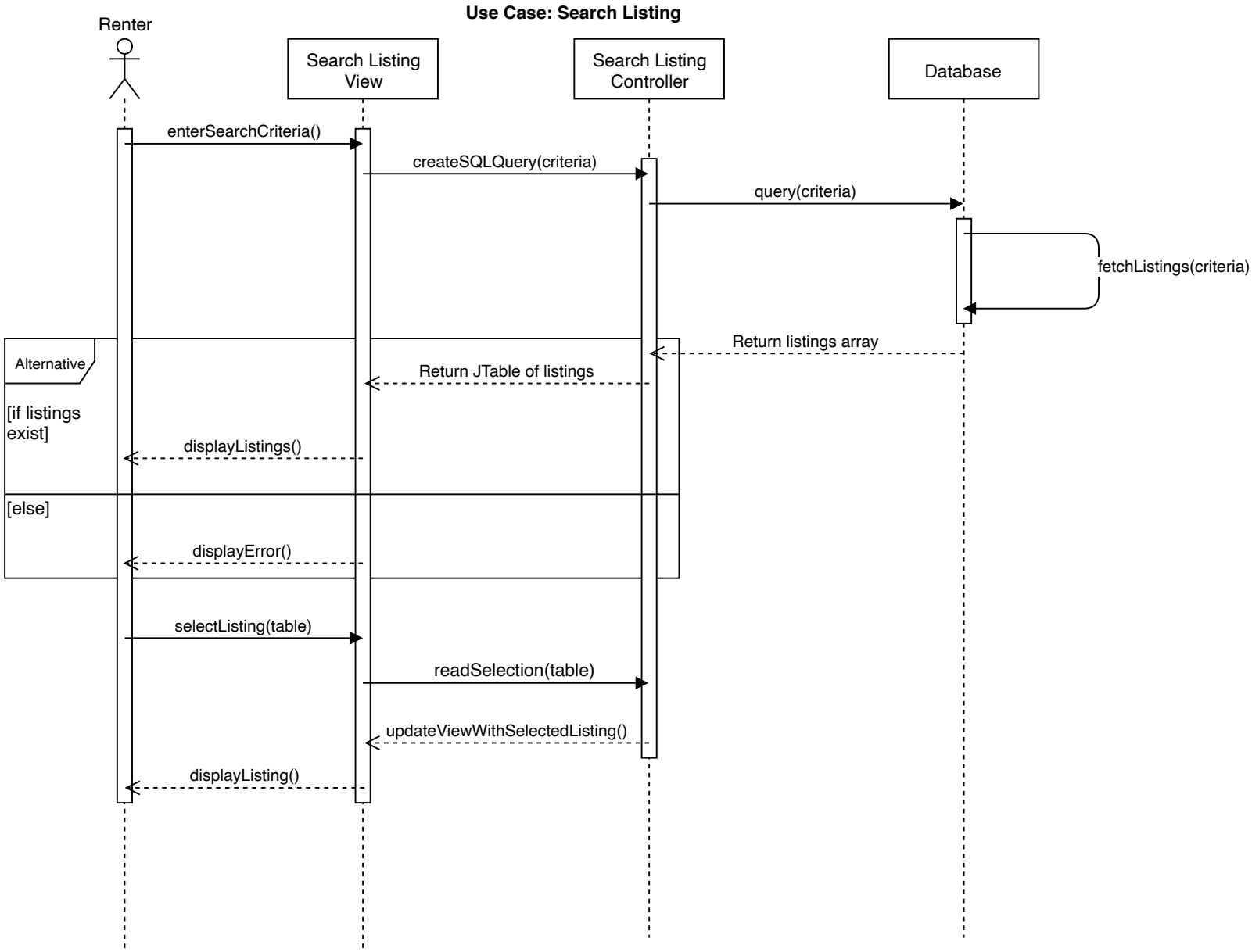
Note

The external stimuli of a manager can change a property obejct from any state to any state which is not shown in the diagram to keep simplicity and clarity

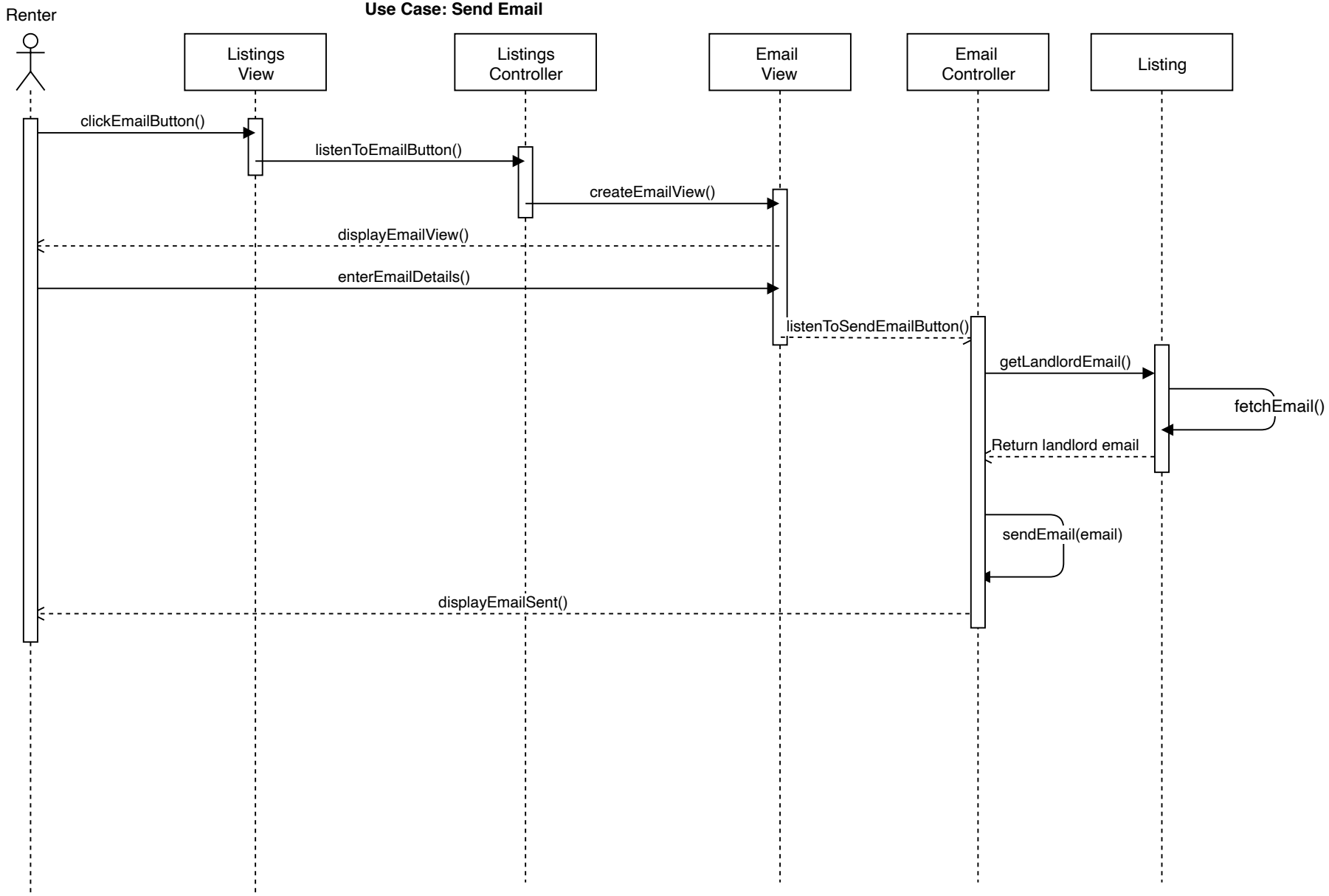
Process that landlord posts his/her property

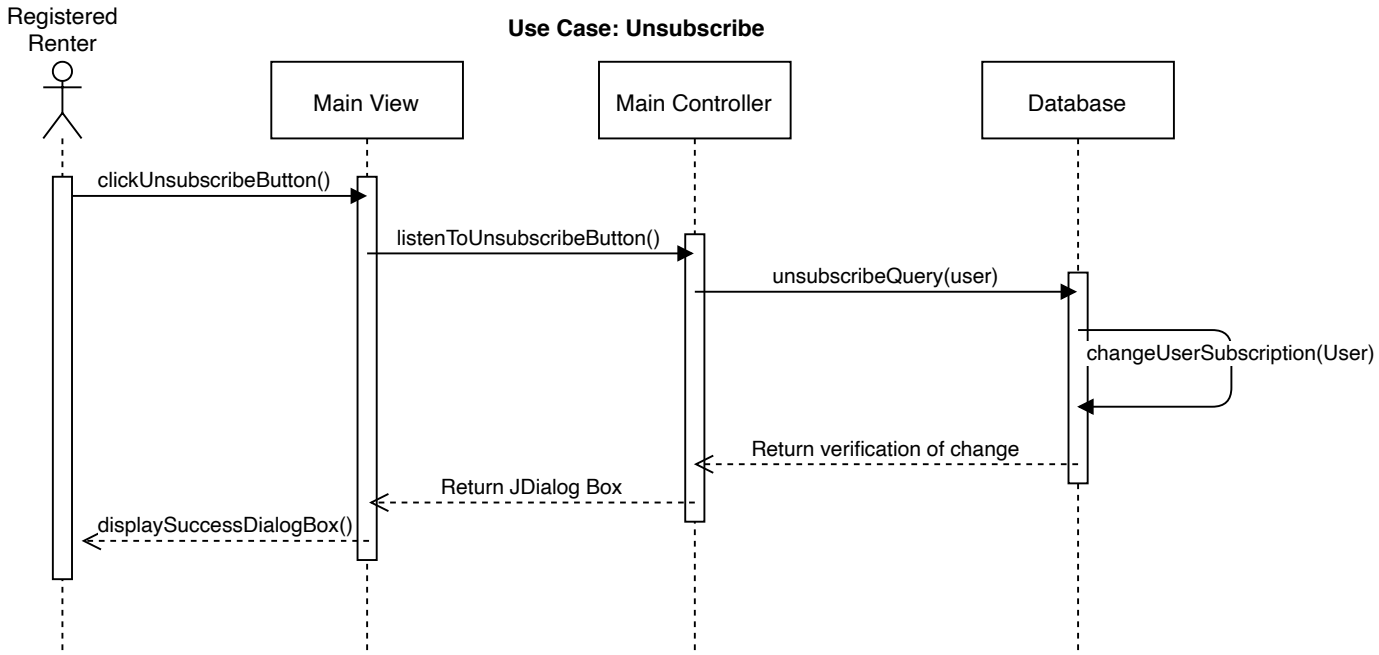


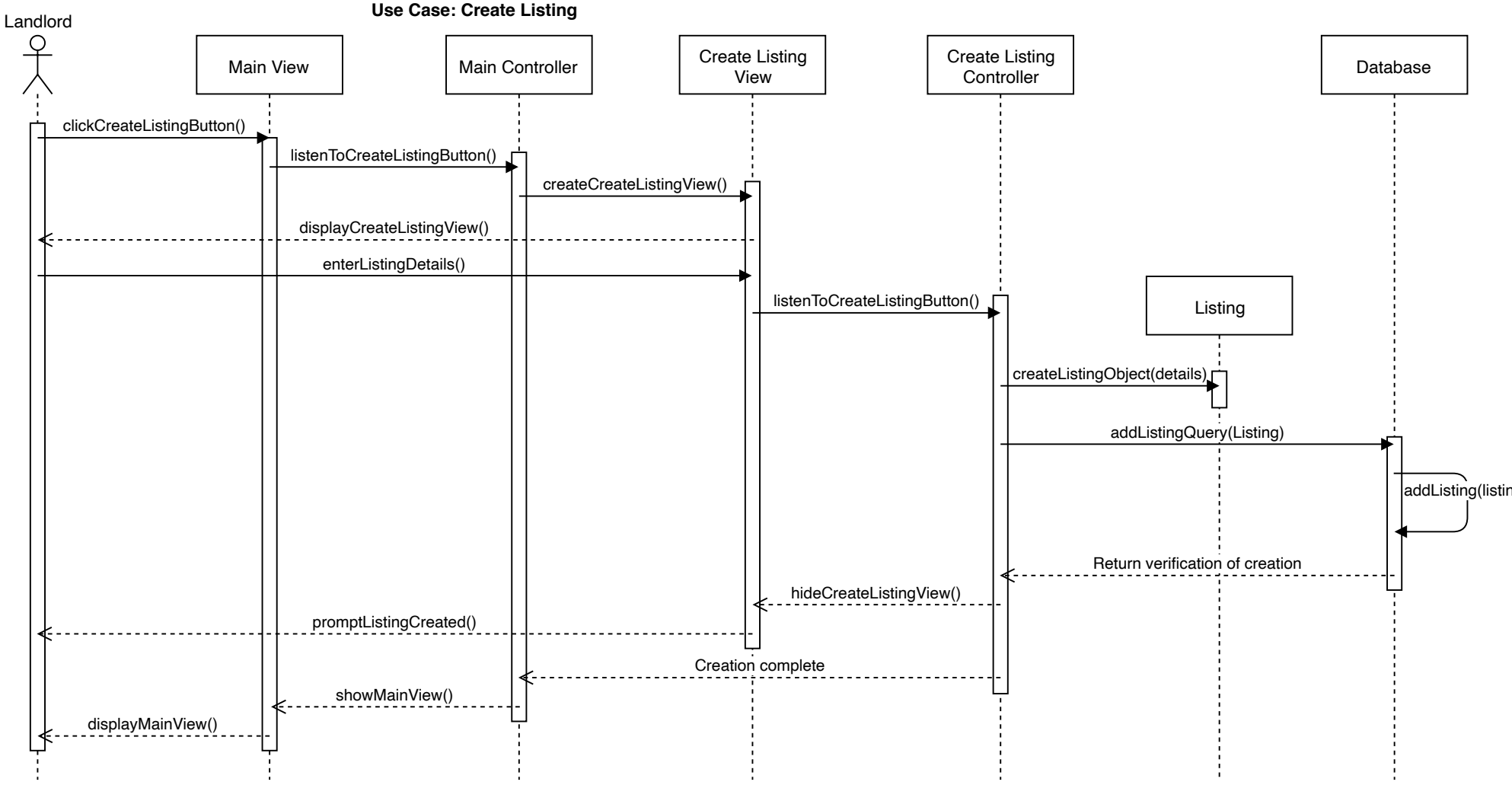

Use Case: User Login
User
Login View
LoginController
Database
enterCredentials()
verify(credentials)
query(credentials)
fetchUser(credentials)
Alternative
[if user exists]
Return user
checkValidity(user)
Valid Credentials
giveAccess(user)
[else]
Invalid Credentials
reEnterCredentials()


Use Case: Search Listing
Renter
Search Listing View
Search Listing Controller
Database
enterSearchCriteria()
createSQLQuery(criteria)
query(criteria)
fetchListings(criteria)
Return listings array
Alternative
Return JTable of listings
[if listings exist]
displayListings()
[else]
displayError()
selectListing(table)
readSelection(table)
updateViewWithSelectedListing()
displayListing()

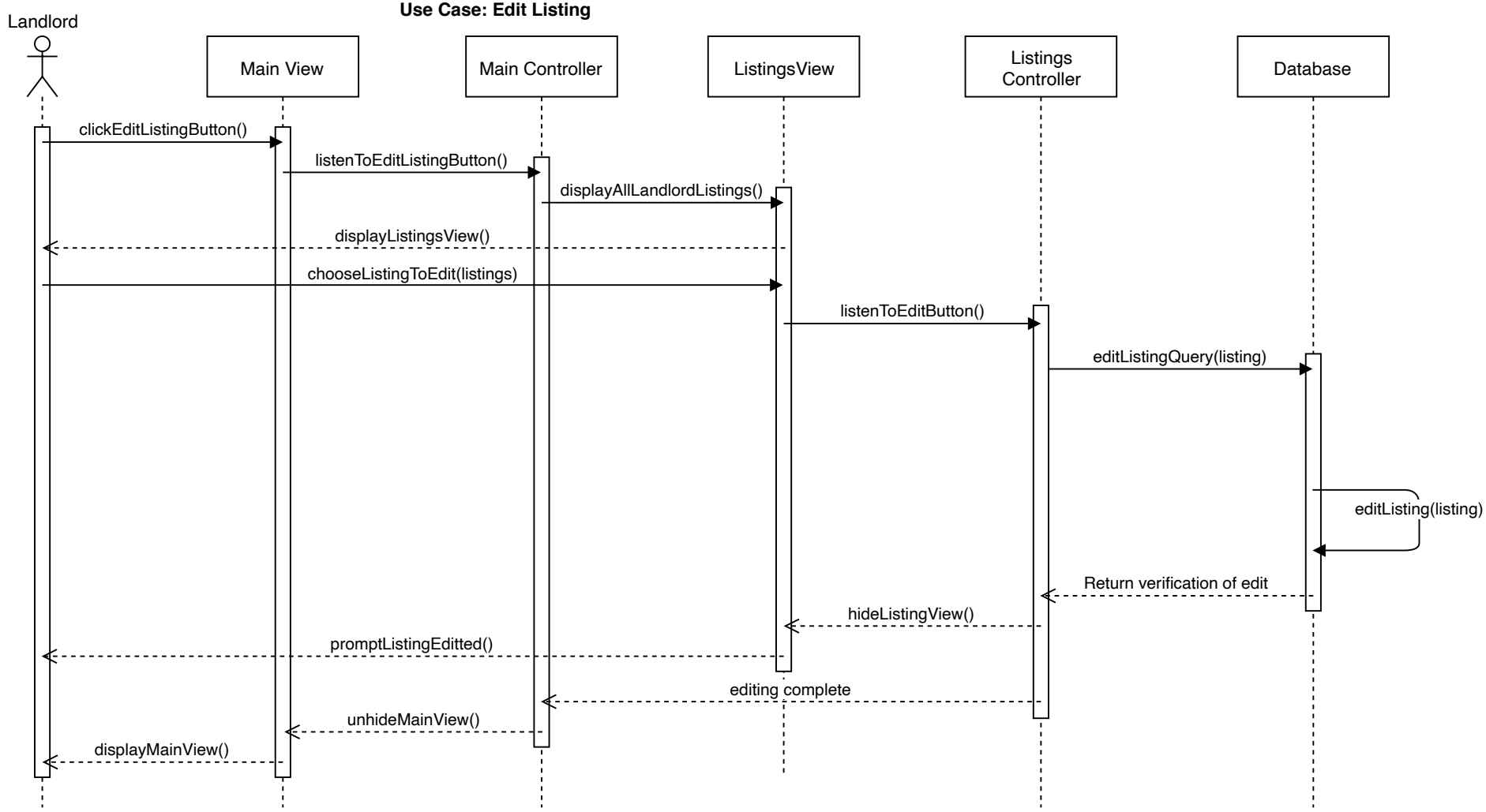
Use Case: Send Email



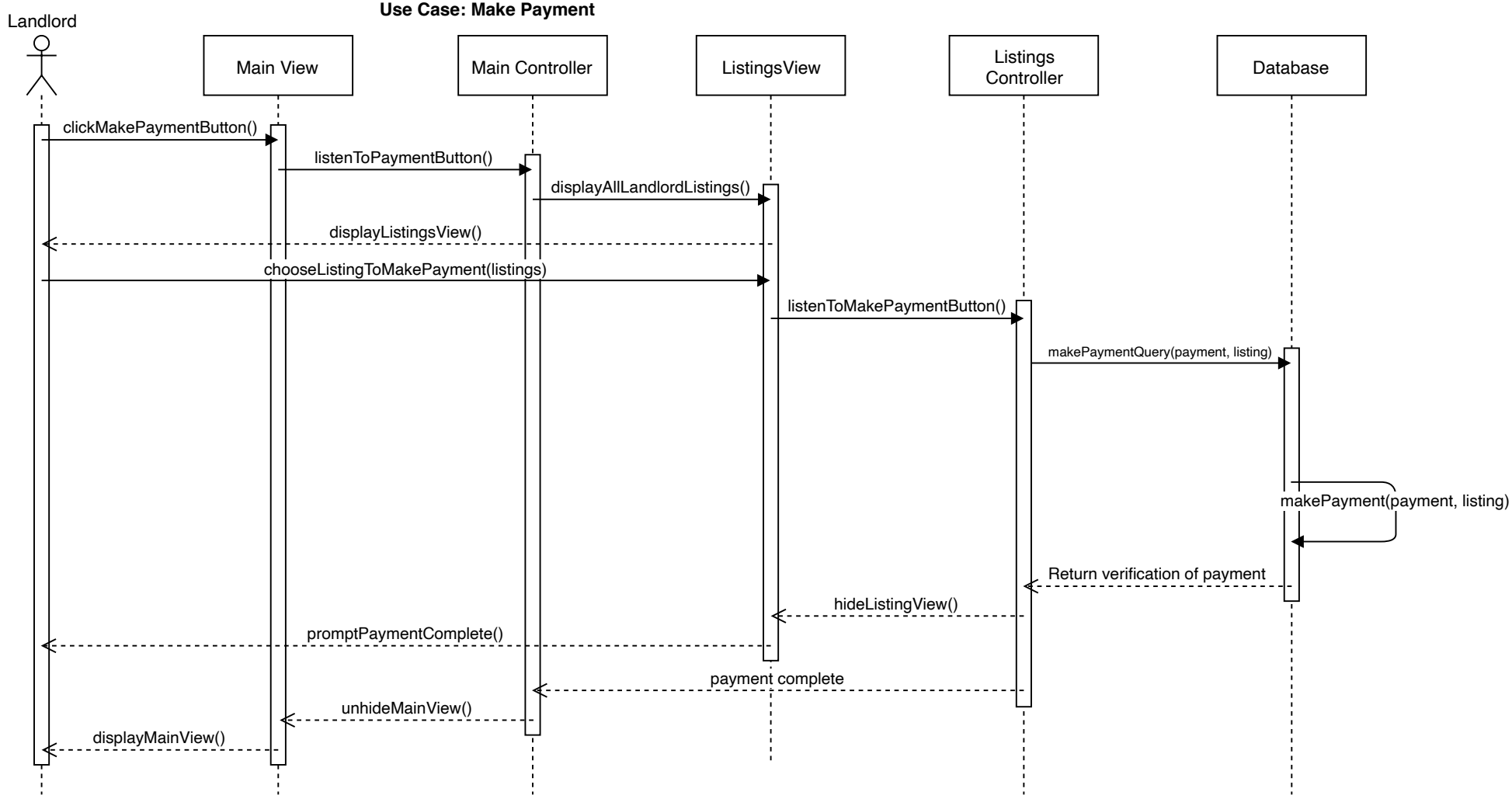

Use Case: Unsubscribe
Registered Renter
Main View
Main Controller
Database
clickUnsubscribeButton()
listenToUnsubscribeButton()
unsubscribeQuery(user)
changeUserSubscription(User)
Return verification of change
Return JDialog Box
displaySuccessDialogBox()


Use Case: Create Listing
Landlord
Main View
Main Controller
Create Listing View
Create Listing Controller
Database
Listing
clickCreateListingButton()
listenToCreateListingButton()
createCreateListingView()
displayCreateListingView()
enterListingDetails()
listenToCreateListingButton()
createListingObject(details)
addListingQuery(Listing)
addListing(listin
Return verification of creation
hideCreateListingView()
promptListingCreated()
Creation complete
showMainView()
displayMainView()

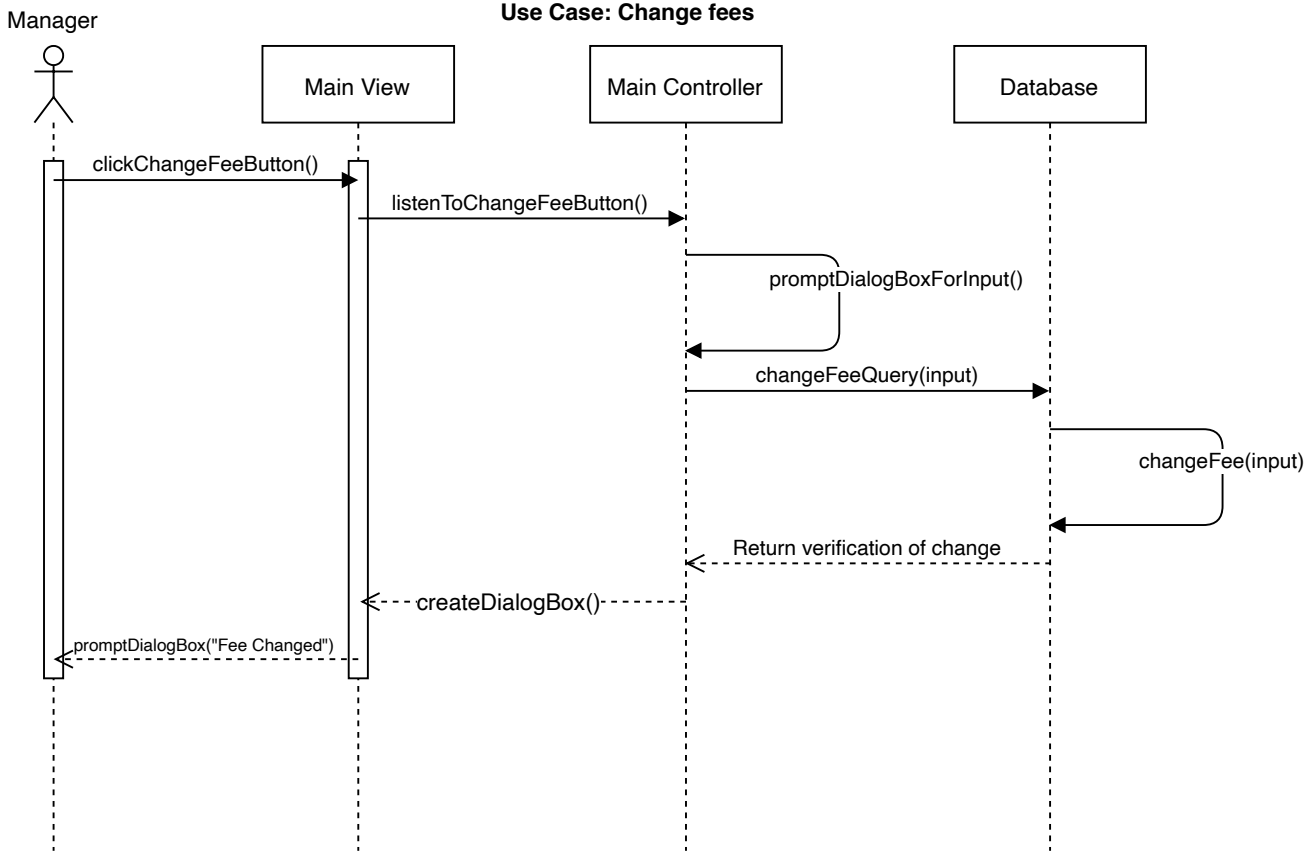
Use Case: Edit Listing

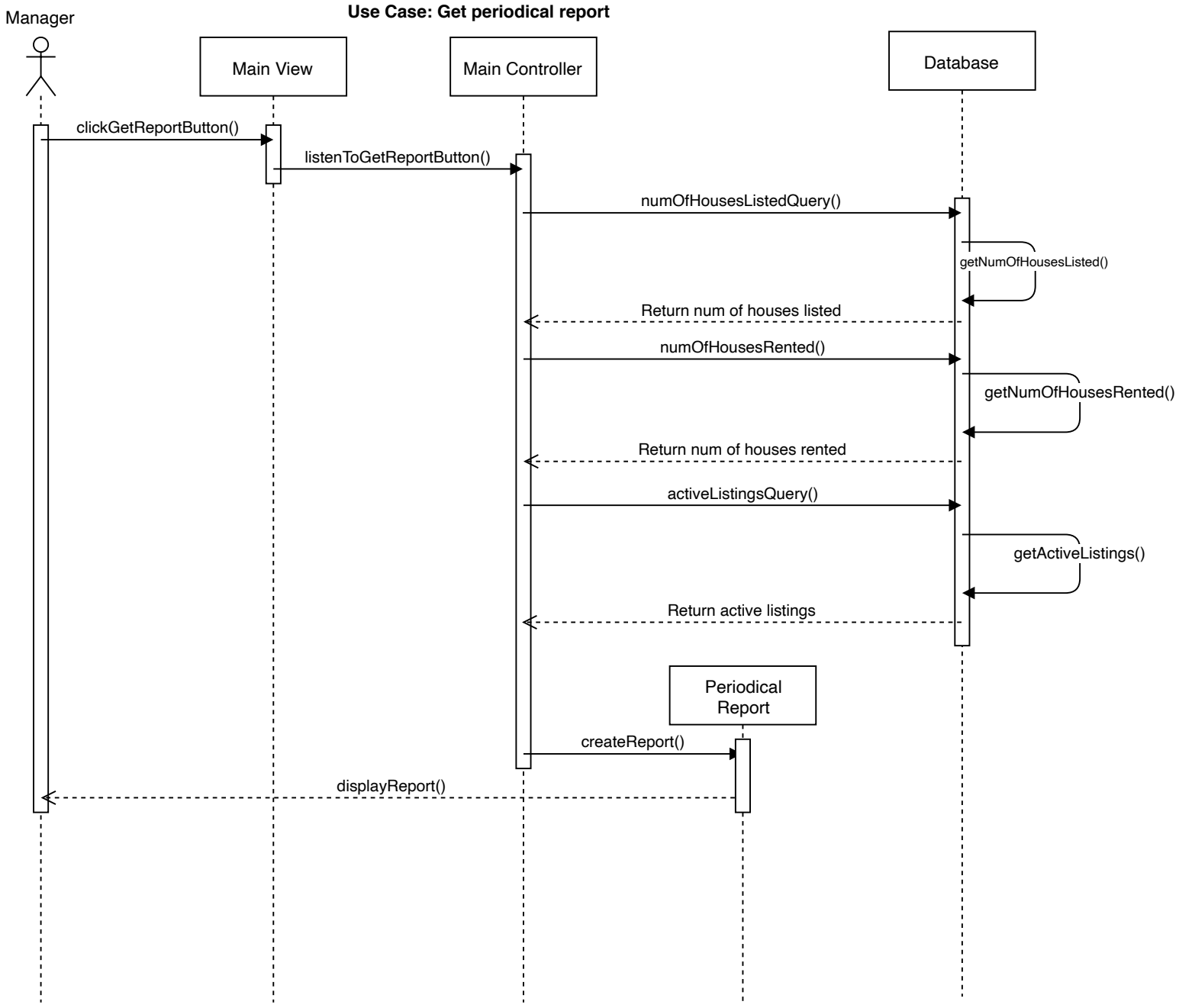


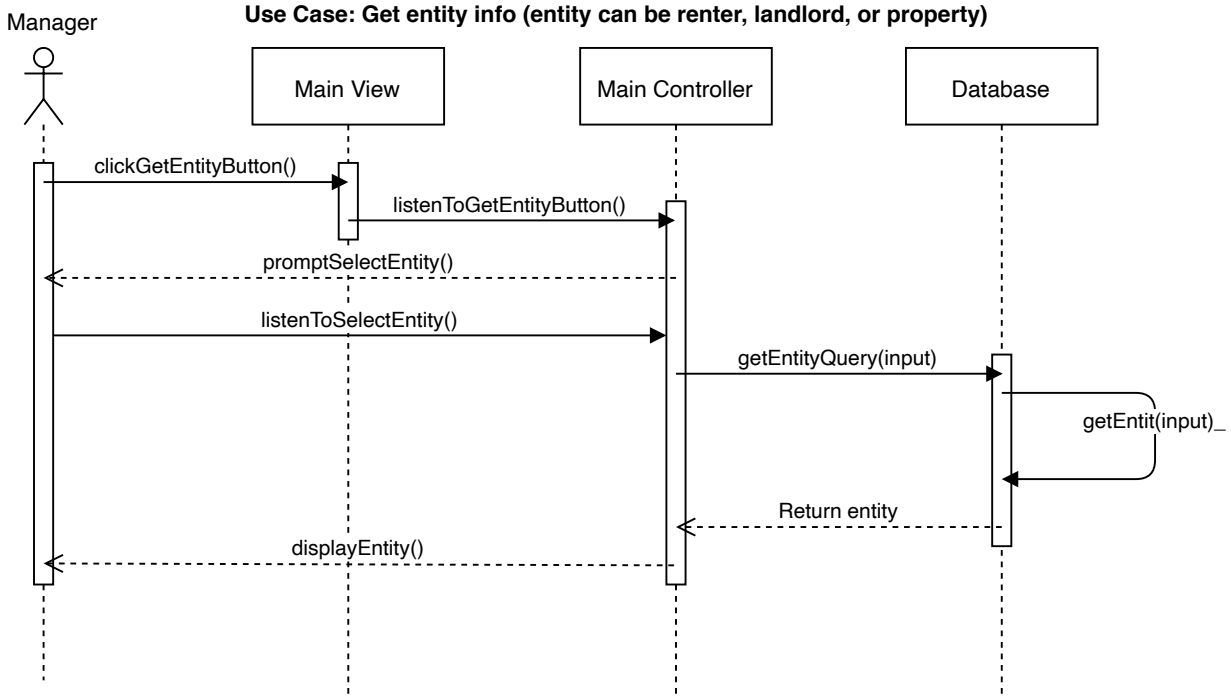
Use Case: Make Payment



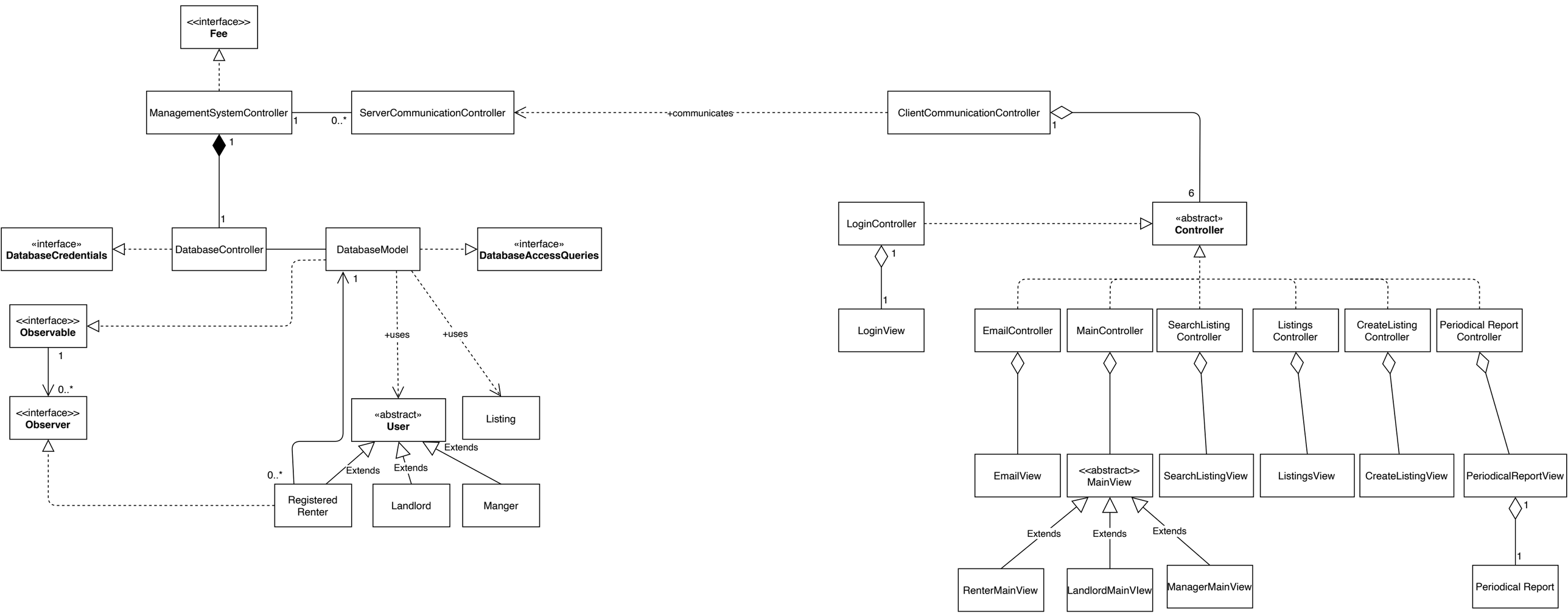
Use Case: Change fees



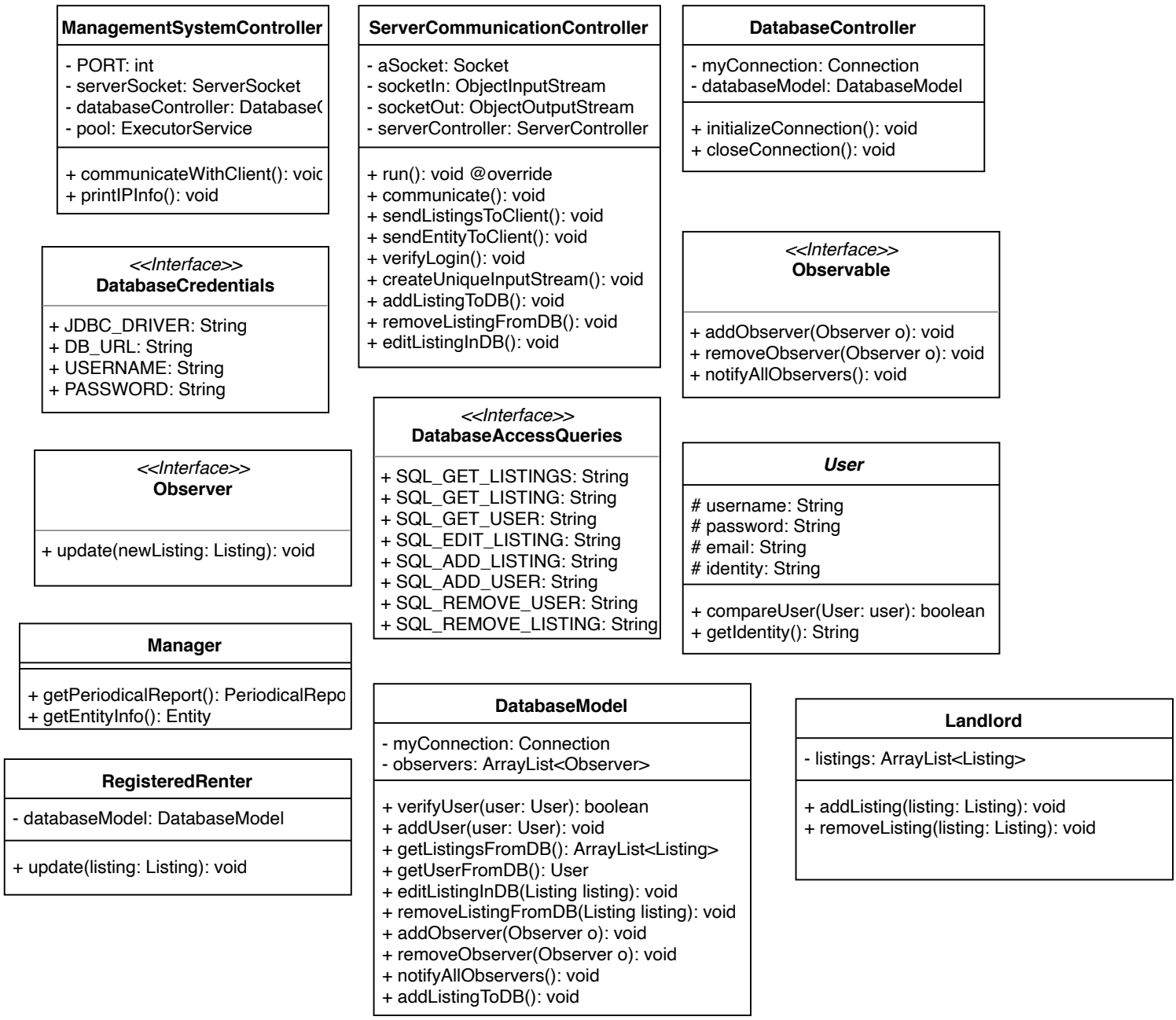

Use Case: Get periodical report
Manager
Main View
Main Controller
Database
clickGetReportButton()
listenToGetReportButton()
numOfHousesListedQuery()
getNumOfHousesListed()
Return num of houses listed
numOfHousesRented()
getNumOfHousesRented()
Return num of houses rented
activeListingsQuery()
getActiveListings()
Return active listings
Periodical Report
createReport()
displayReport()

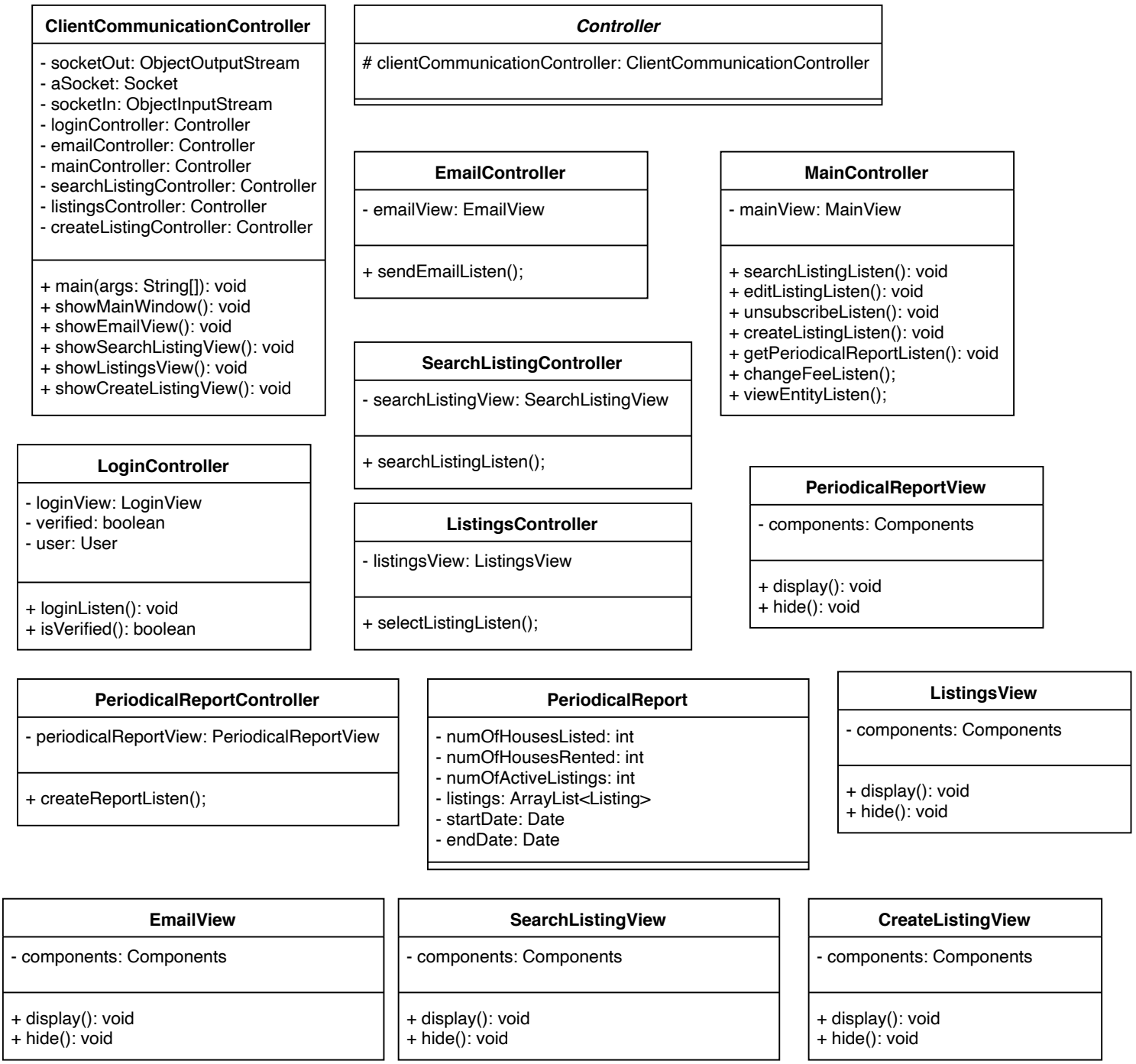

Use Case: Get entity info (entity can be renter, landlord, or property)
Manager
Main View
Main Controller
Database
clickGetEntityButton()
listenToGetEntityButton()
promptSelectEntity()
listenToSelectEntity()
getEntityQuery(input)
getEntit(input)_
Return entity
displayEntity()

Relationships UML

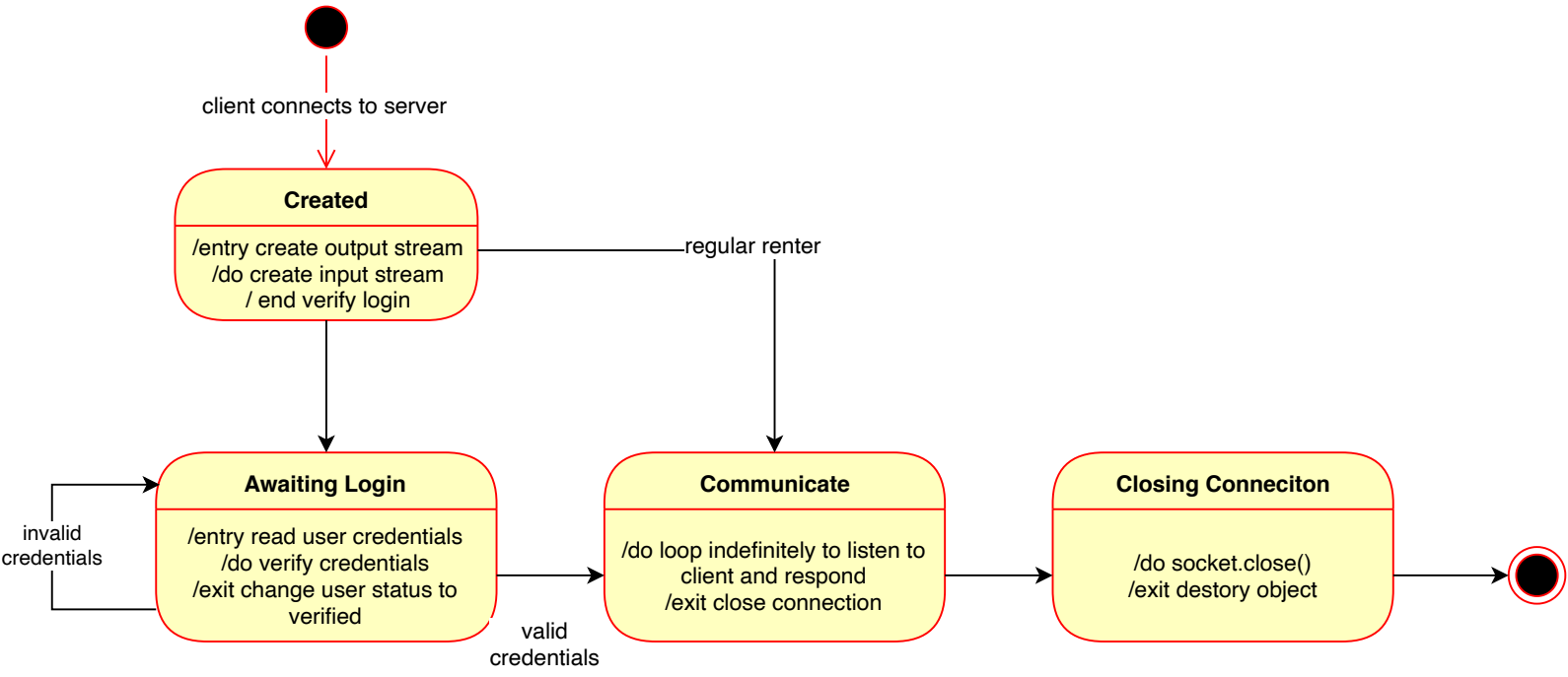


Attributes UML

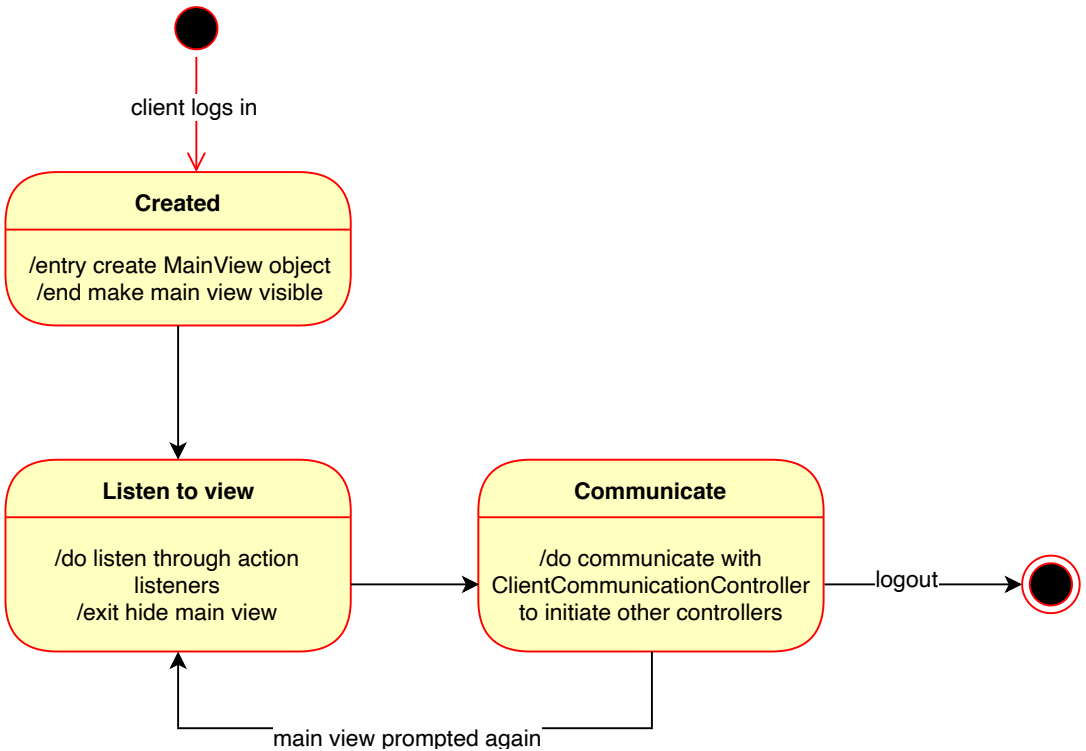



ClientCommunicationController
- socketOut: ObjectOutputStream
- aSocket: Socket
- socketIn: ObjectInputStream
- loginController: Controller
- emailController: Controller
- mainController: Controller
- searchListingController: Controller
- listingsController: Controller
- createListingController: Controller
+ main(args: String[]): void
+ showMainWindow(): void
+ showEmailView(): void
+ showSearchListingView(): void
+ showListingsView(): void
+ showCreateListingView(): void
Controller
clientCommunicationController: ClientCommunicationController
EmailController
- emailView: EmailView
+ sendEmailListen();
MainController
- mainView: MainView
+ searchListingListen(): void
+ editListingListen(): void
+ unsubscribeListen(): void
+ createListingListen(): void
+ getPeriodicalReportListen(): void
+ changeFeeListen();
+ viewEntityListen();
SearchListingController
- searchListingView: SearchListingView
+ searchListingListen();
LoginController
- loginView: LoginView
- verified: boolean
- user: User
+ loginListen(): void
+ isVerified(): boolean
ListingsController
- listingsView: ListingsView
+ selectListingListen();
PeriodicalReportView
- components: Components
+ display(): void
+ hide(): void
PeriodicalReportController
- periodicalReportView: PeriodicalReportView
+ createReportListen();
PeriodicalReport
- numOfHousesListed: int
- numOfHousesRented: int
- numOfActiveListings: int
- listings: ArrayList<Listing>
- startDate: Date
- endDate: Date
ListingsView
- components: Components
+ display(): void
+ hide(): void
EmailView
- components: Components
+ display(): void
+ hide(): void
SearchListingView
- components: Components
+ display(): void
+ hide(): void
CreateListingView
- components: Components
+ display(): void
+ hide(): void

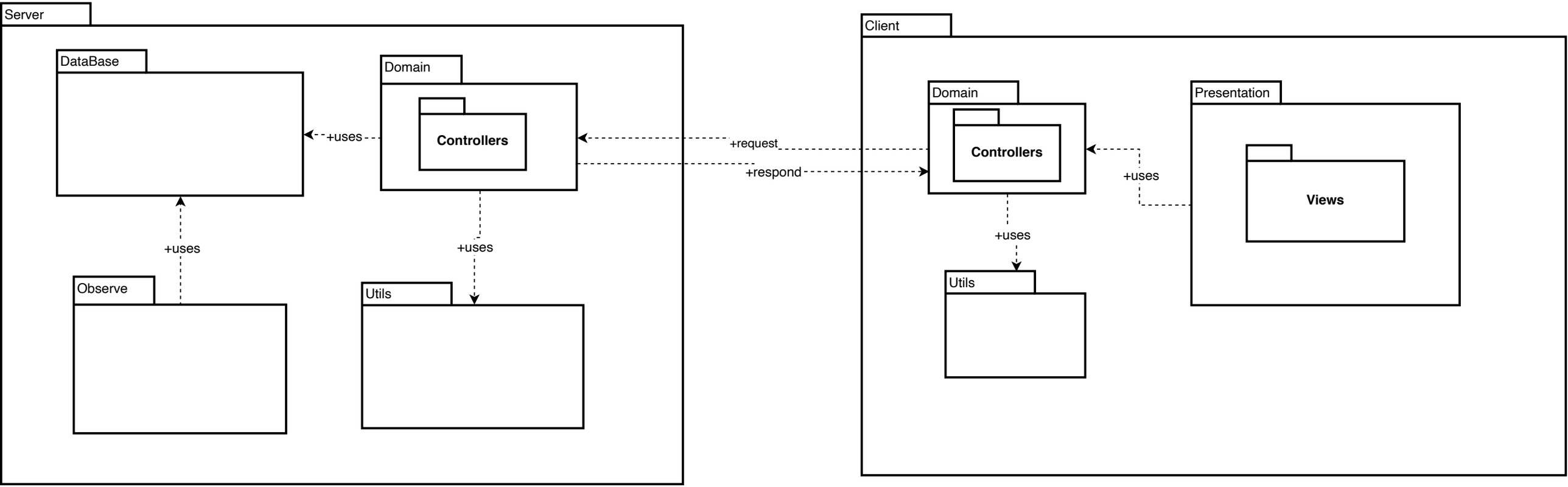
ServerCommunicationController State Diagram



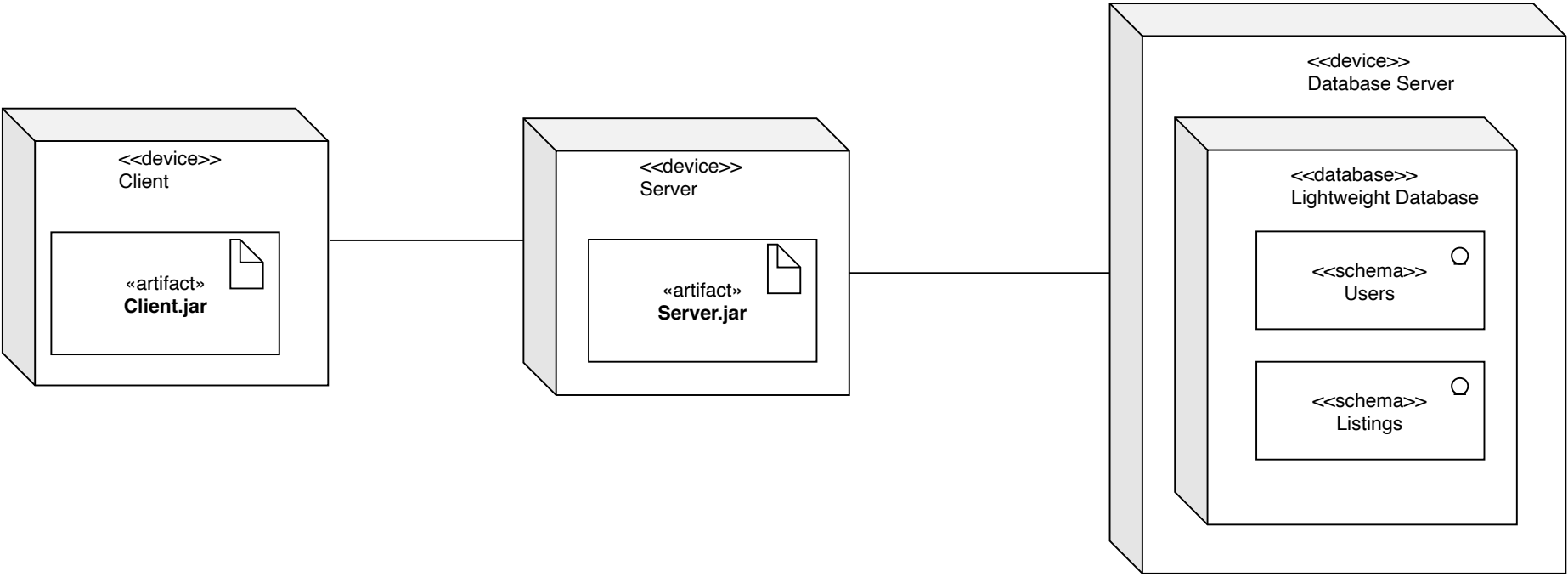
MainController State Diagram



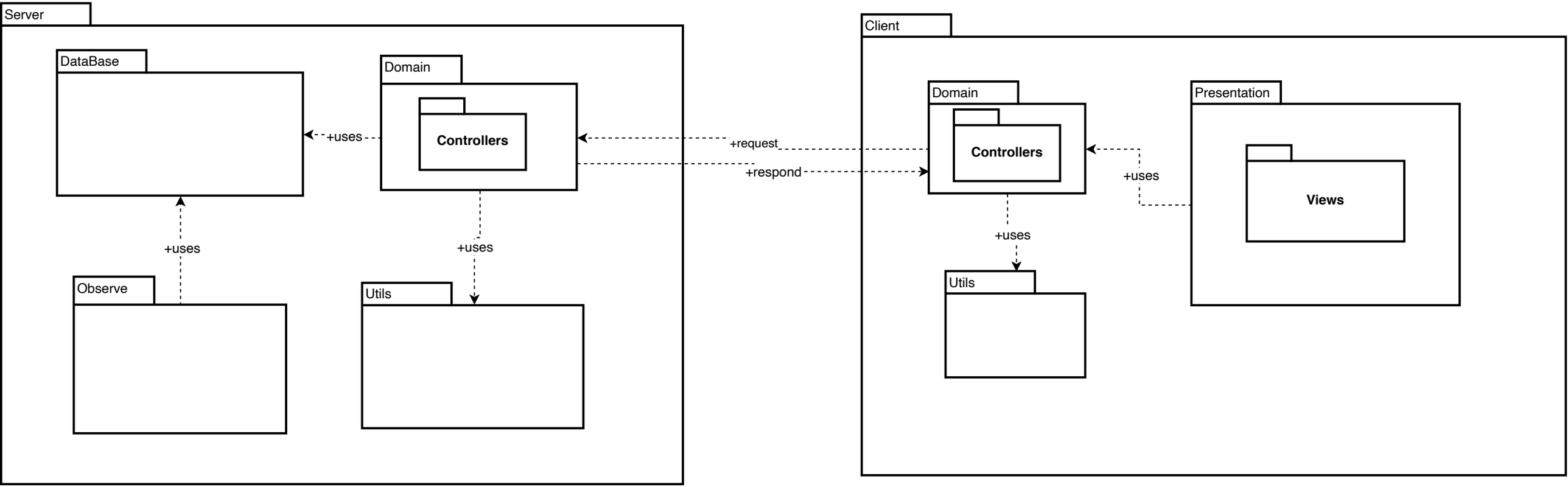
Package Diagram



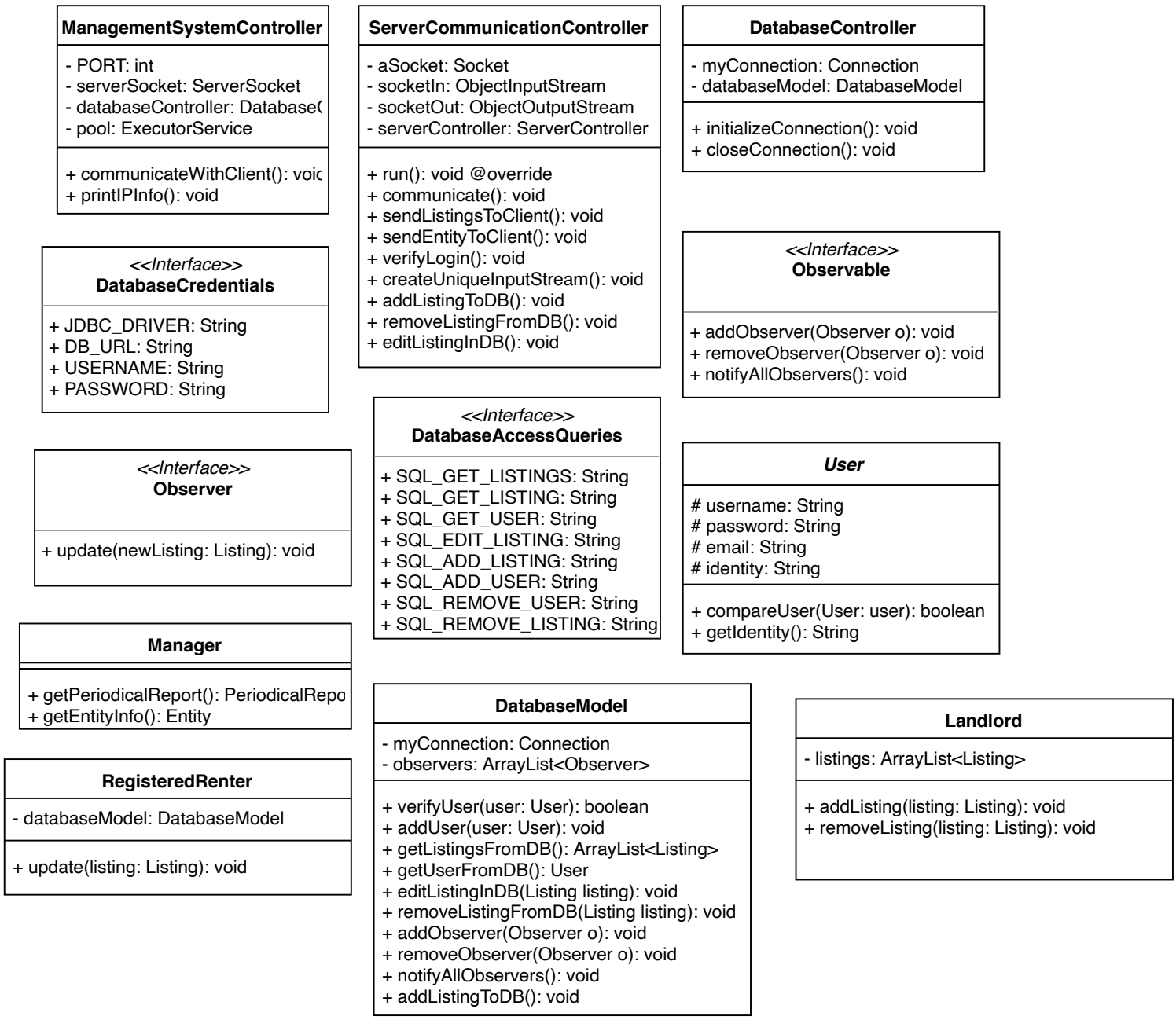
Deployement Diagram

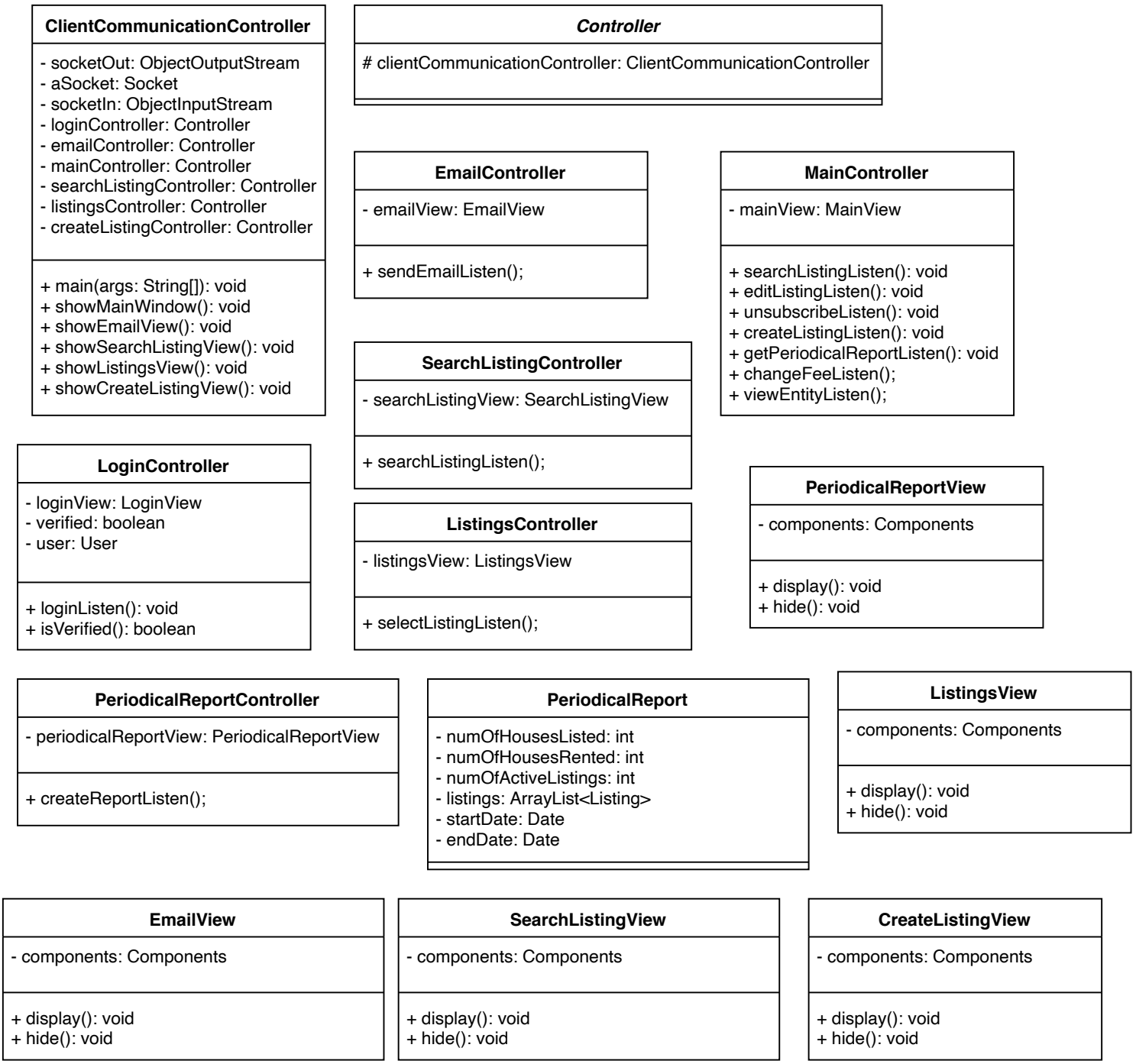


Package Diagram



Attributes UML




ClientCommunicationController
- socketOut: ObjectOutputStream
- aSocket: Socket
- socketIn: ObjectInputStream
- loginController: Controller
- emailController: Controller
- mainController: Controller
- searchListingController: Controller
- listingsController: Controller
- createListingController: Controller
+ main(args: String[]): void
+ showMainWindow(): void
+ showEmailView(): void
+ showSearchListingView(): void
+ showListingsView(): void
+ showCreateListingView(): void
Controller
clientCommunicationController: ClientCommunicationController
EmailController
- emailView: EmailView
+ sendEmailListen();
MainController
- mainView: MainView
+ searchListingListen(): void
+ editListingListen(): void
+ unsubscribeListen(): void
+ createListingListen(): void
+ getPeriodicalReportListen(): void
+ changeFeeListen();
+ viewEntityListen();
SearchListingController
- searchListingView: SearchListingView
+ searchListingListen();
LoginController
- loginView: LoginView
- verified: boolean
- user: User
+ loginListen(): void
+ isVerified(): boolean
ListingsController
- listingsView: ListingsView
+ selectListingListen();
PeriodicalReportView
- components: Components
+ display(): void
+ hide(): void
PeriodicalReportController
- periodicalReportView: PeriodicalReportView
+ createReportListen();
PeriodicalReport
- numOfHousesListed: int
- numOfHousesRented: int
- numOfActiveListings: int
- listings: ArrayList<Listing>
- startDate: Date
- endDate: Date
ListingsView
- components: Components
+ display(): void
+ hide(): void
EmailView
- components: Components
+ display(): void
+ hide(): void
SearchListingView
- components: Components
+ display(): void
+ hide(): void
CreateListingView
- components: Components
+ display(): void
+ hide(): void

Relationships UML