

Names: Harsohail Brar (30041921)

Gary Wu (30038110)

Ryan Holt (30038609)

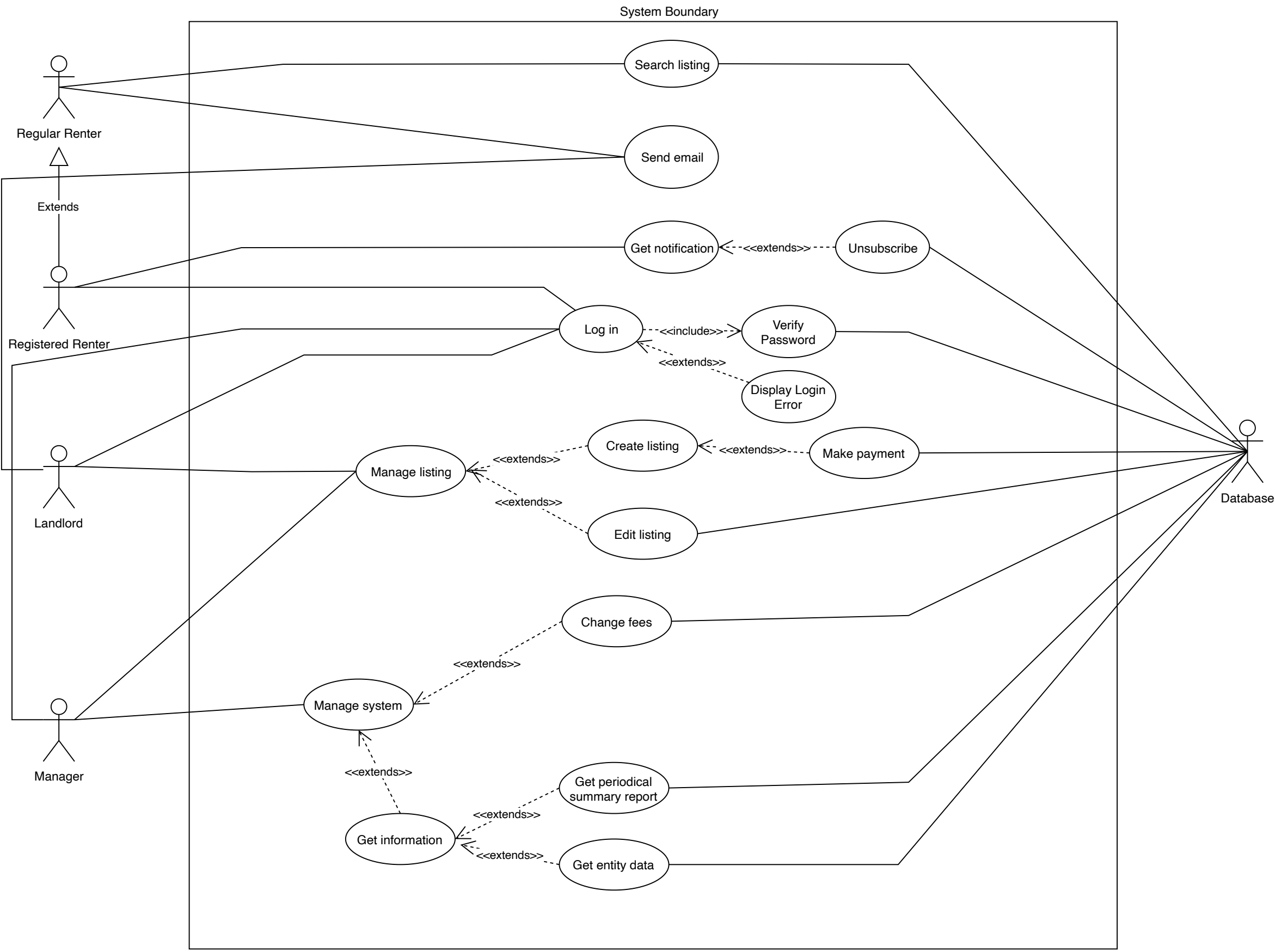
Course Name: Principles of Software Design

Course Code: ENSF 480

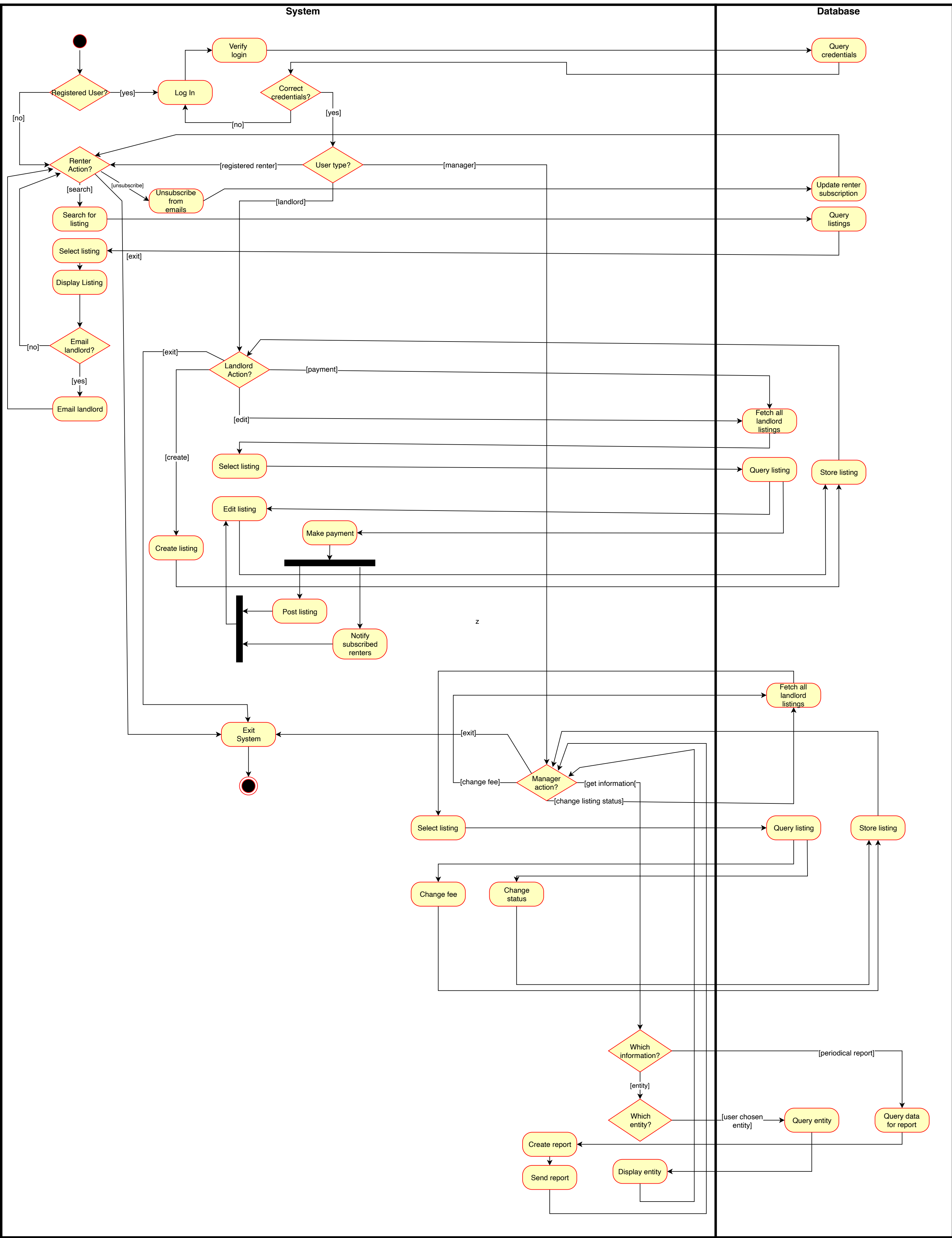
Assignment Number: Term Project Design Document

Submission Date and Time: 09/11/2019

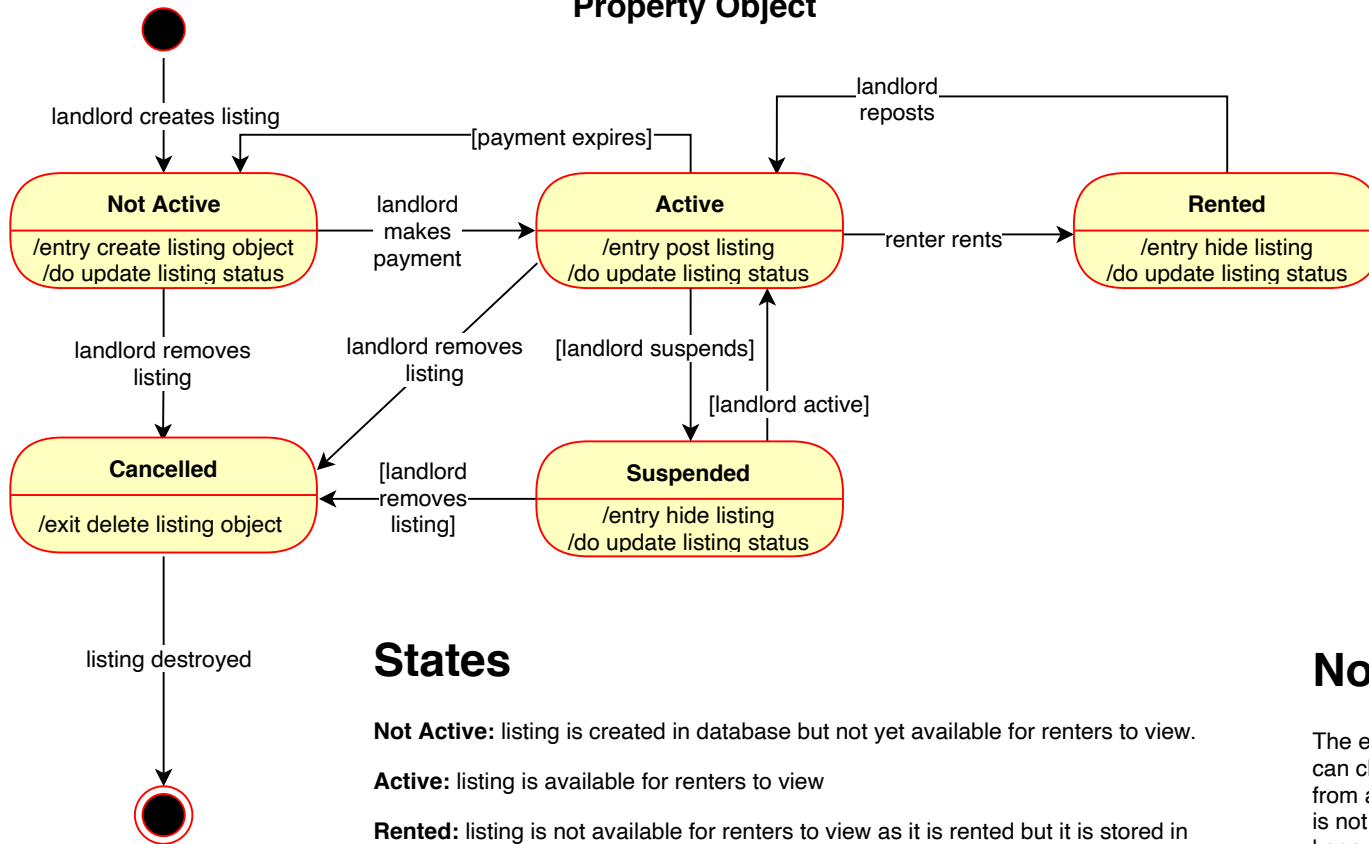
System's Use Case Diagram



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

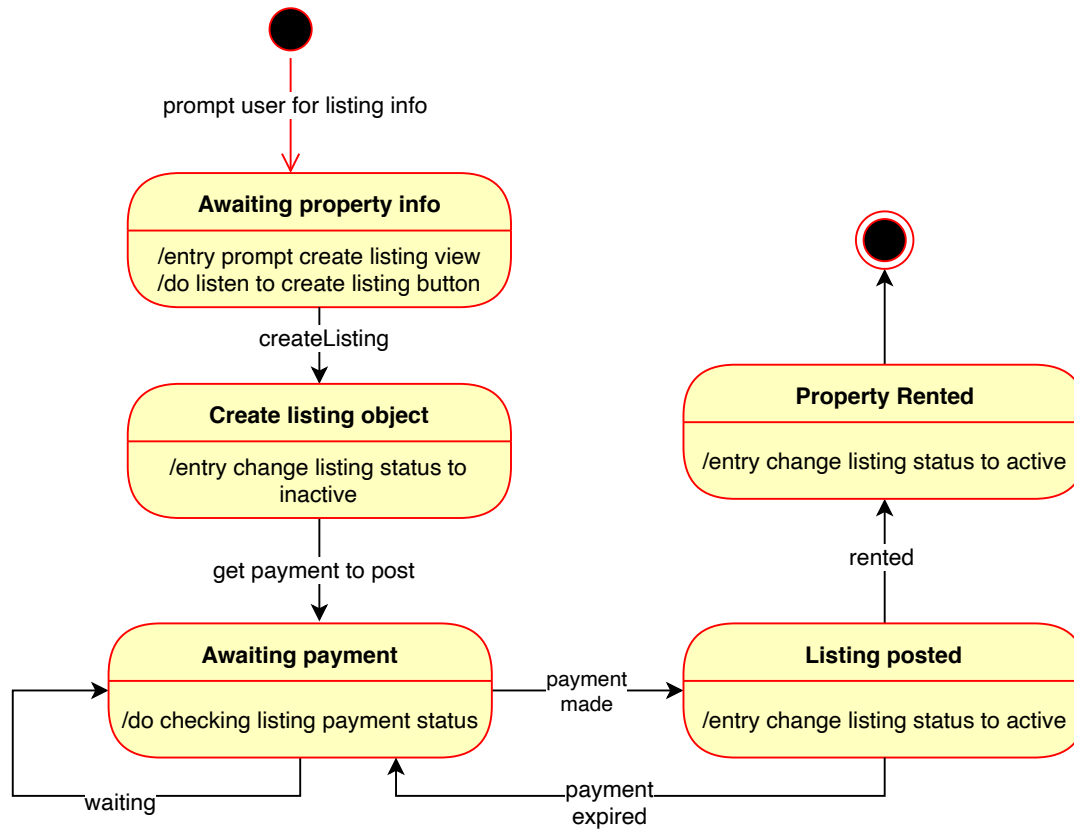
Cancelled: listing is deleted from the system

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

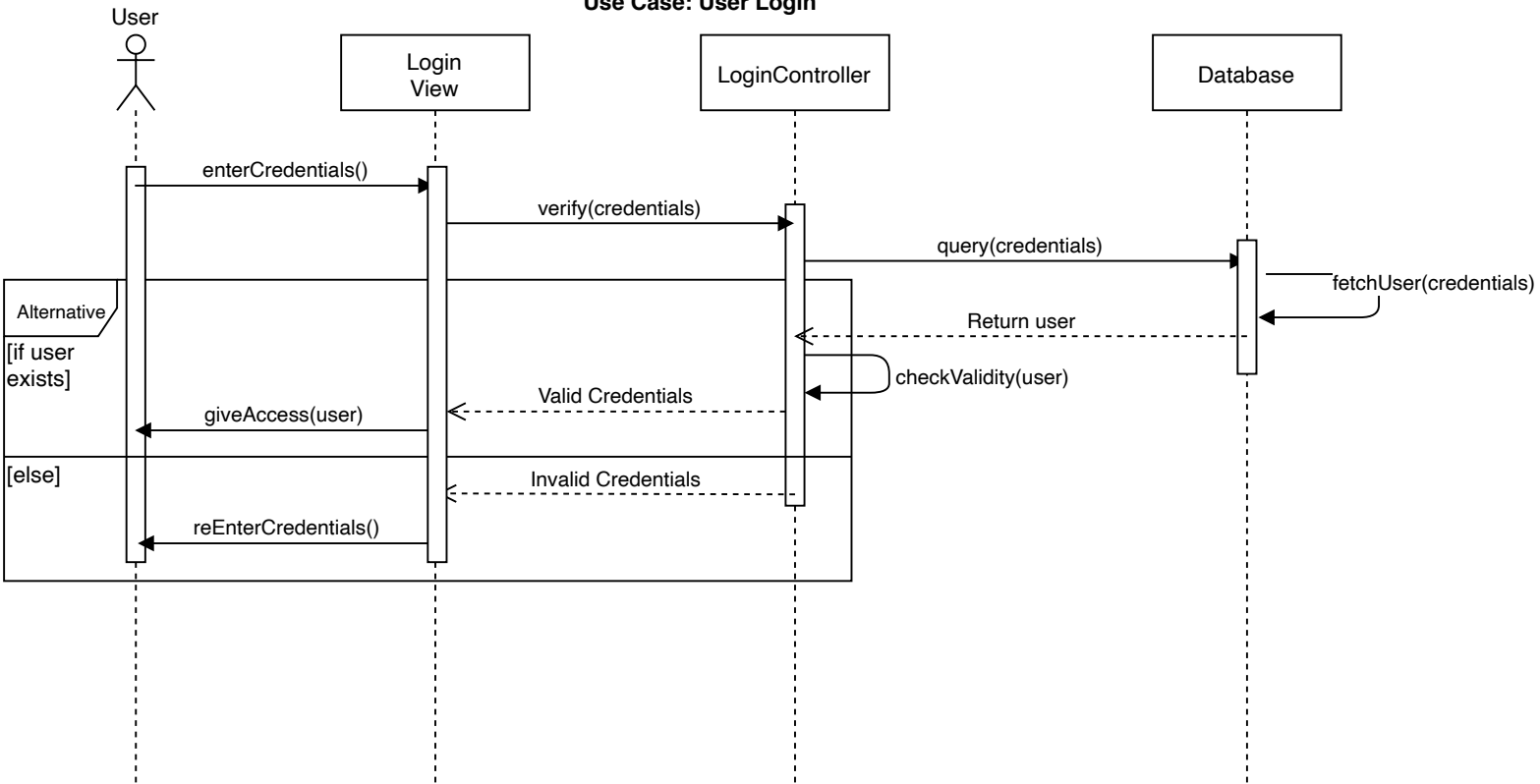
Note

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

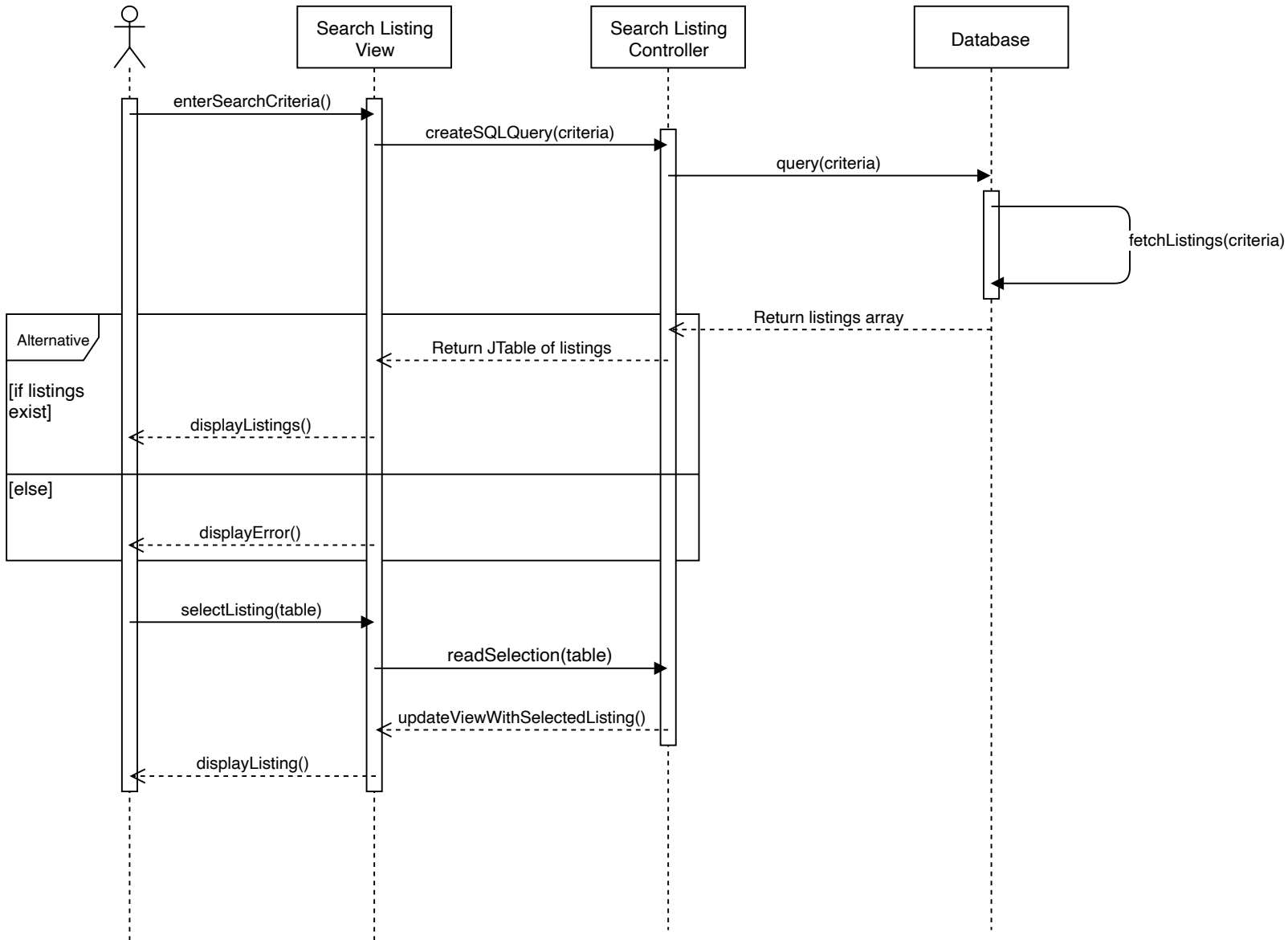
Process that landlord posts his/her property



Use Case: User Login

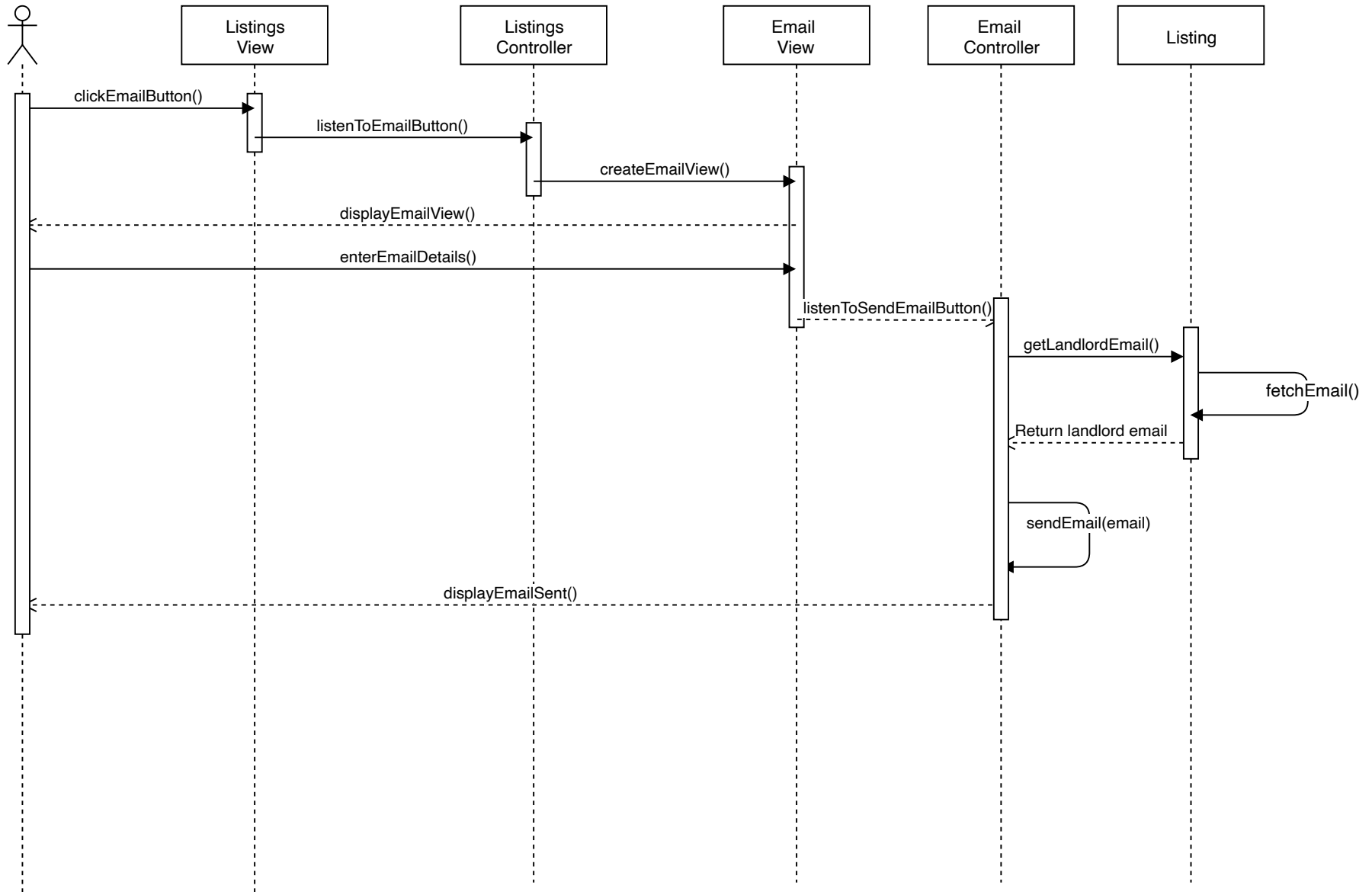


Use Case: Search Listing



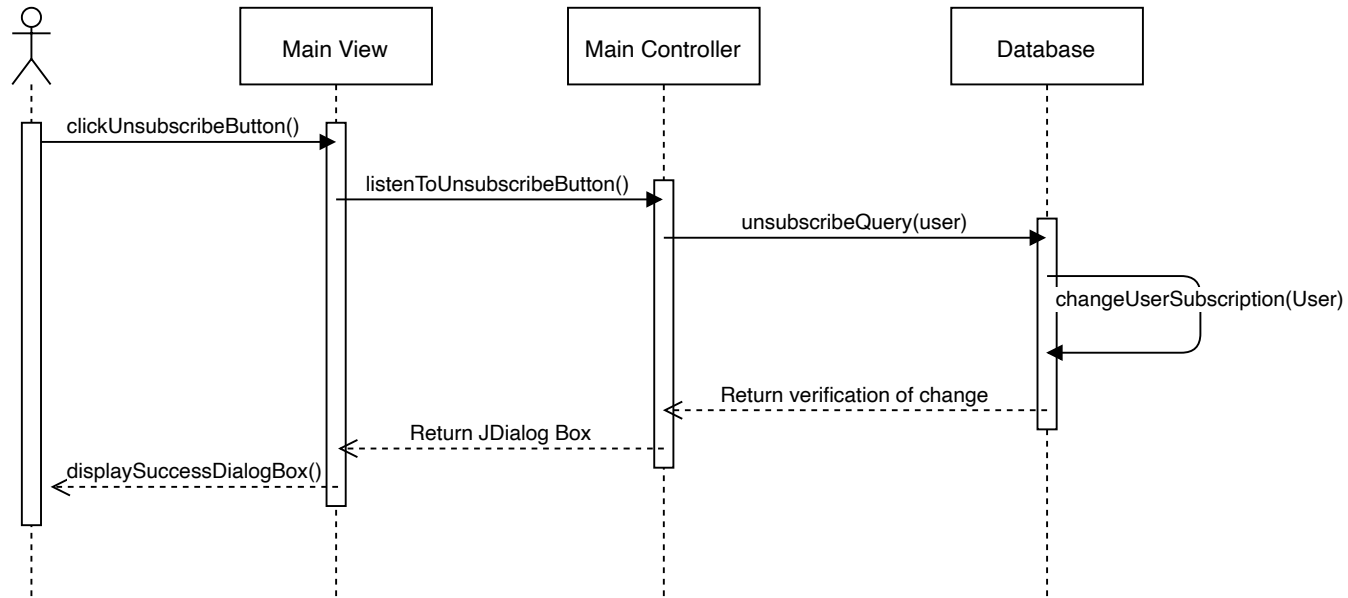
Renter

Use Case: Send Email



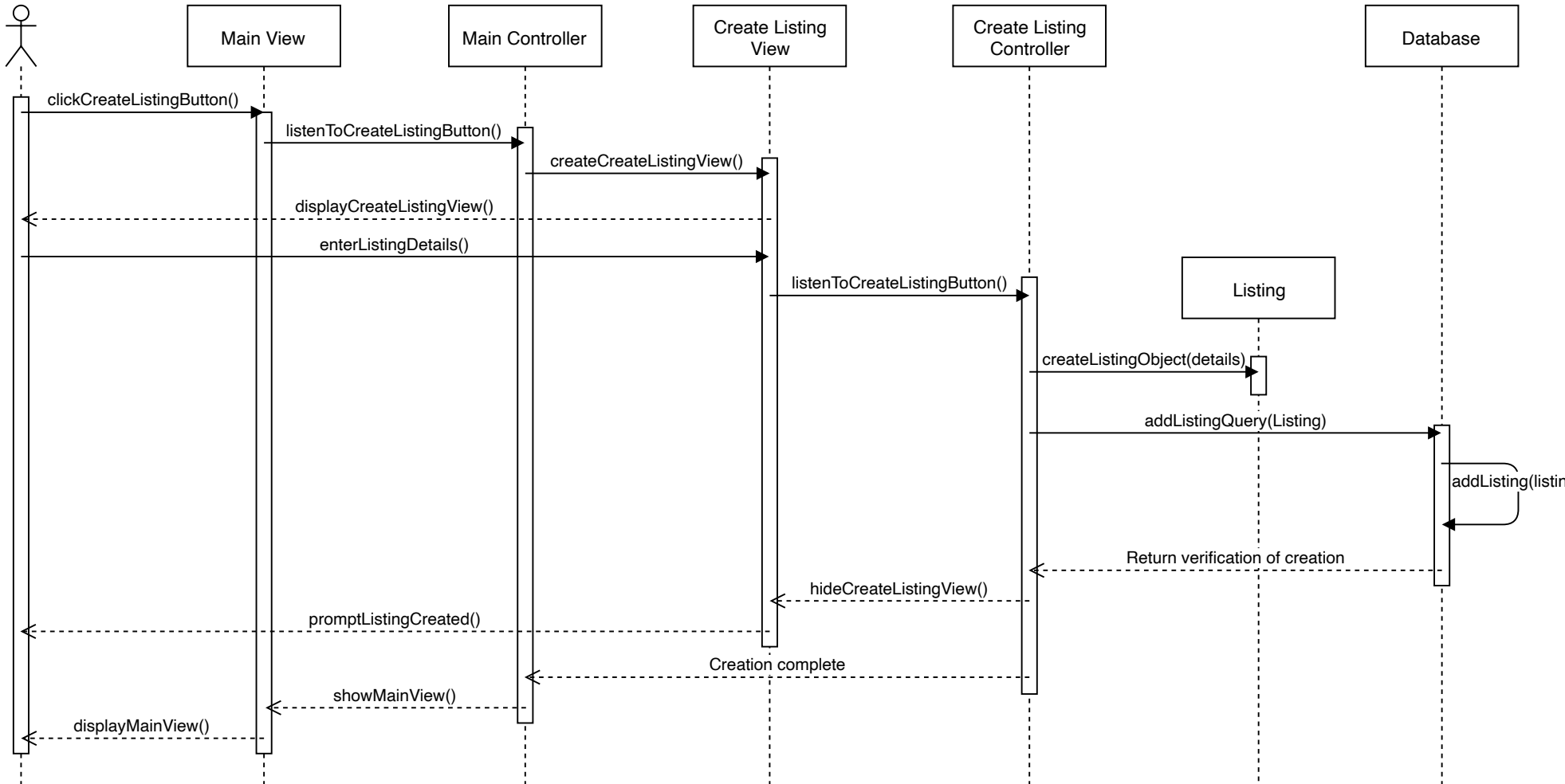
Registered
Renter

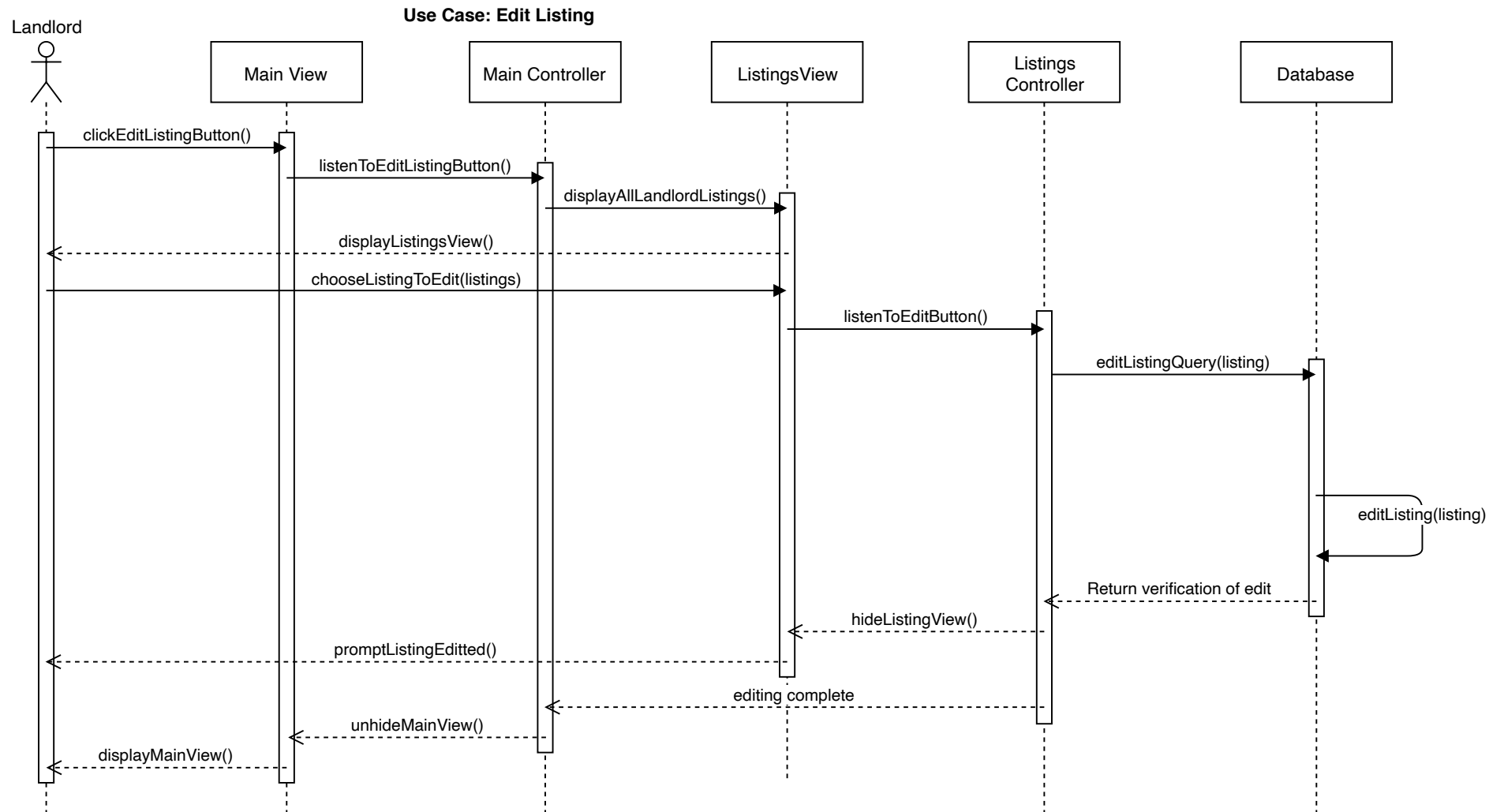
Use Case: Unsubscribe



Landlord

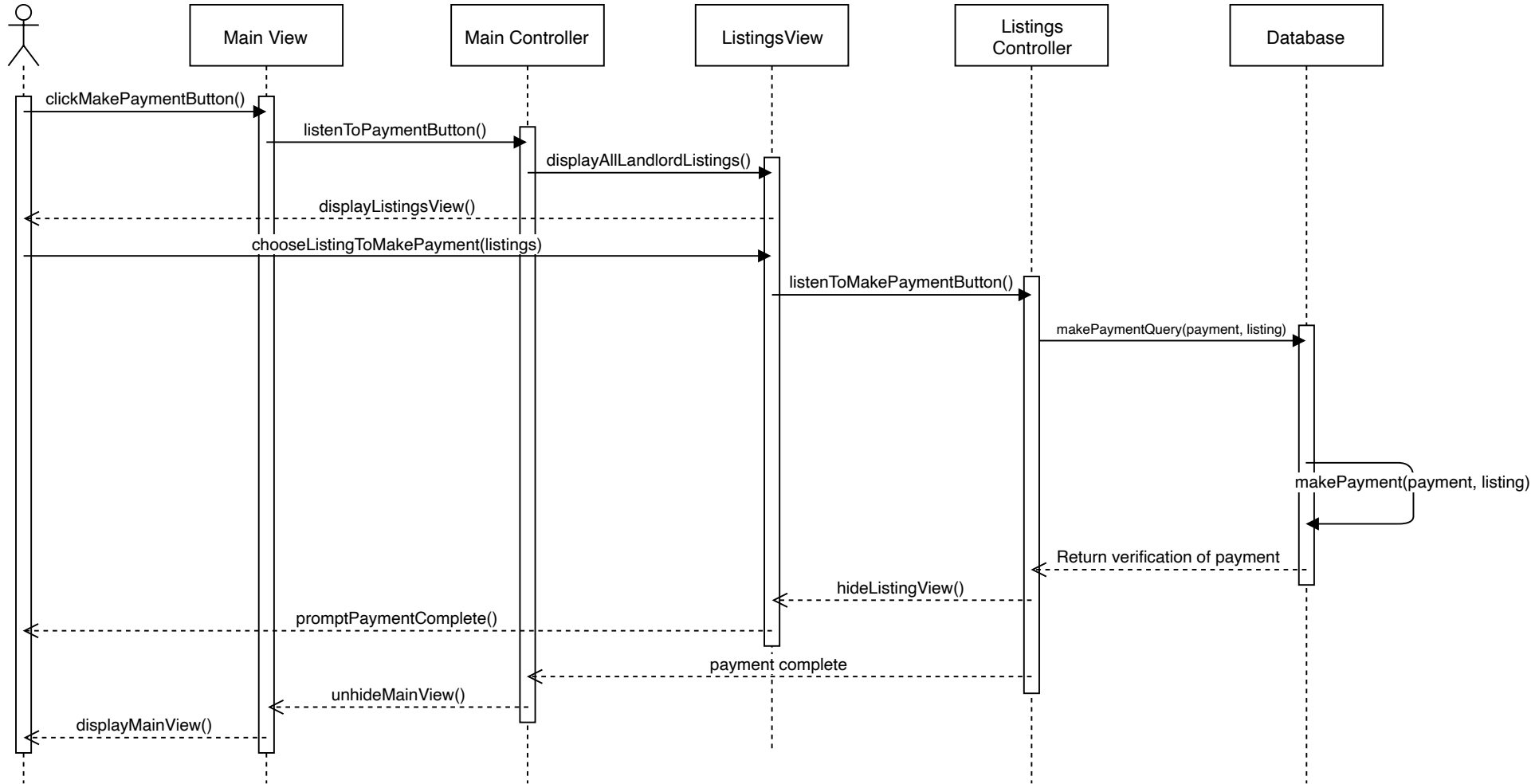
Use Case: Create Listing





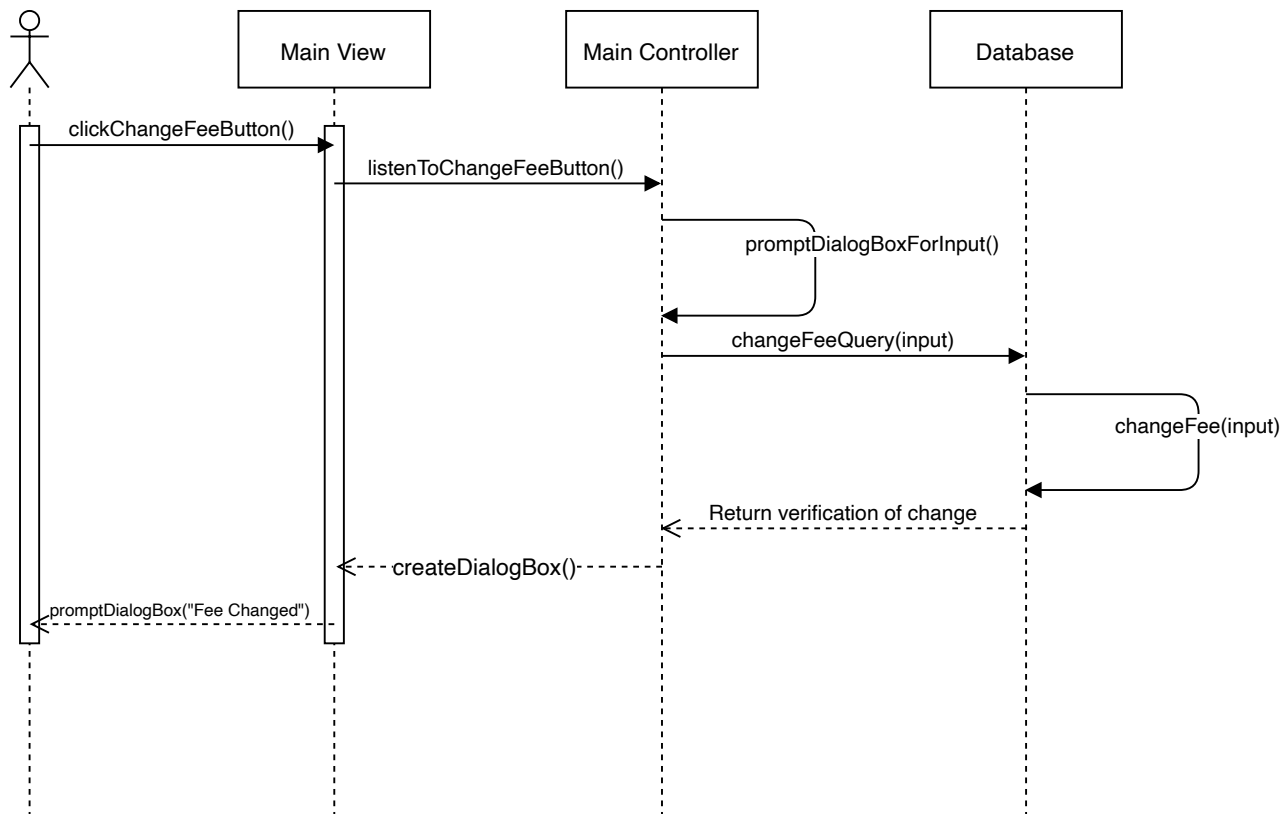
Use Case: Make Payment

Landlord



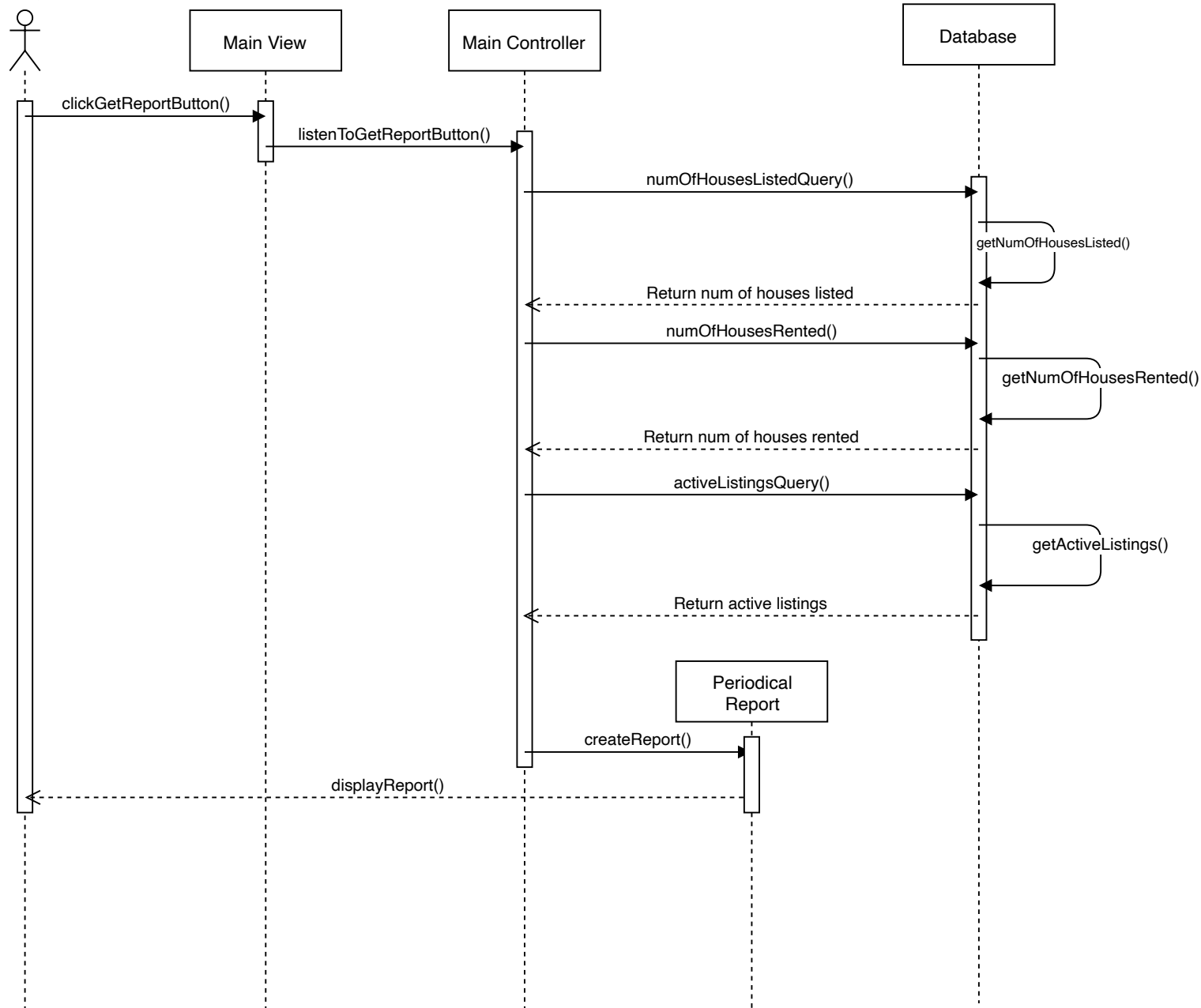
Manager

Use Case: Change fees



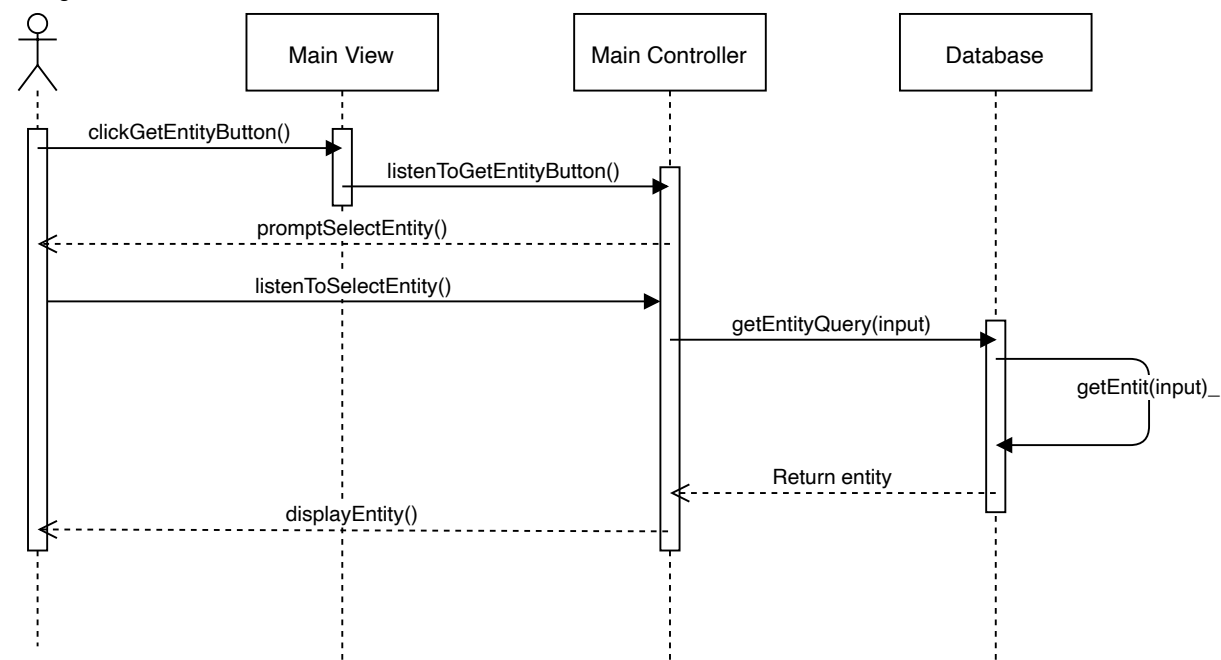
Manager

Use Case: Get periodical report

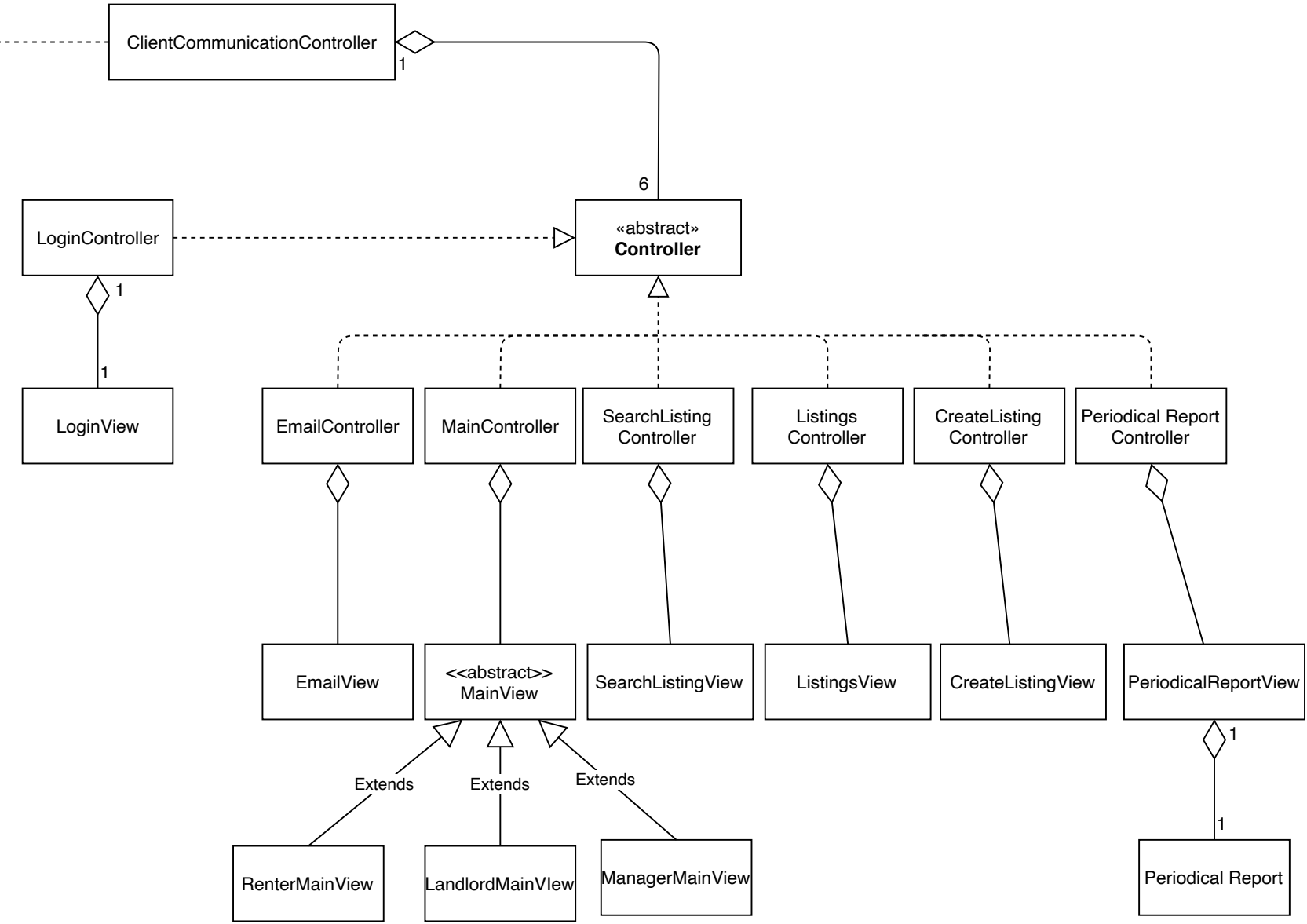
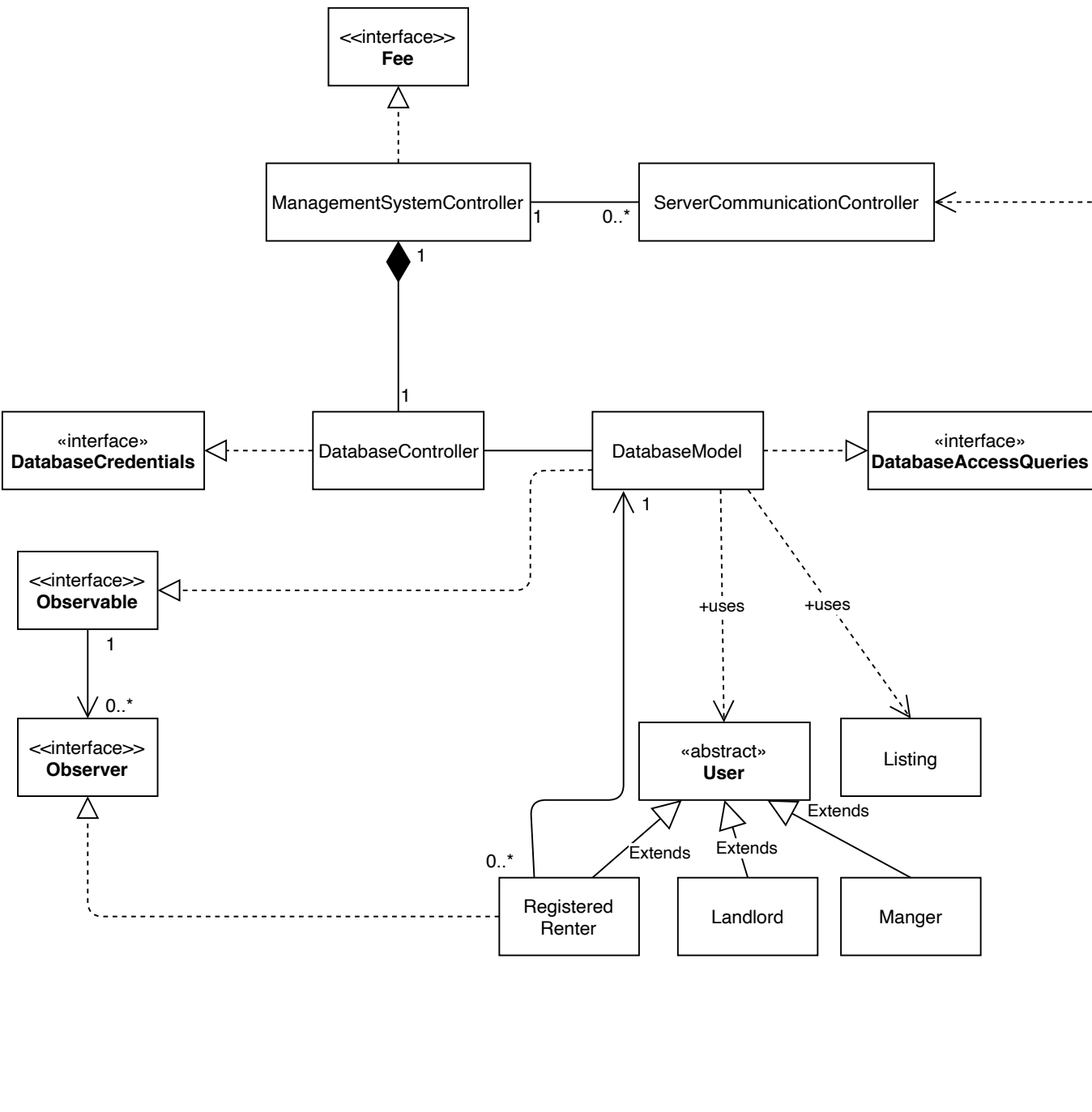


Manager

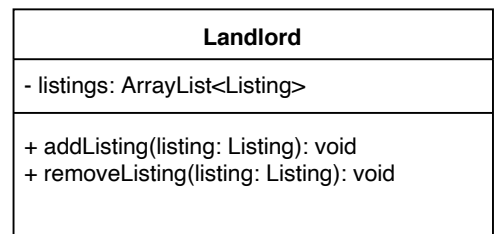
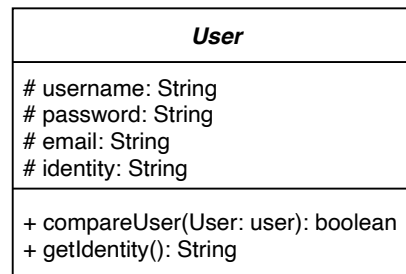
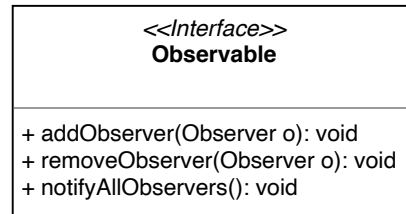
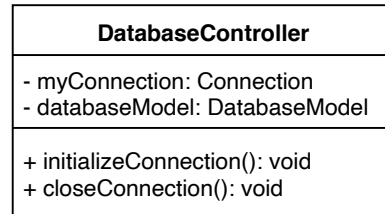
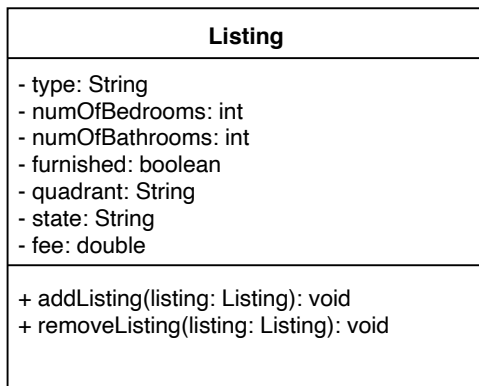
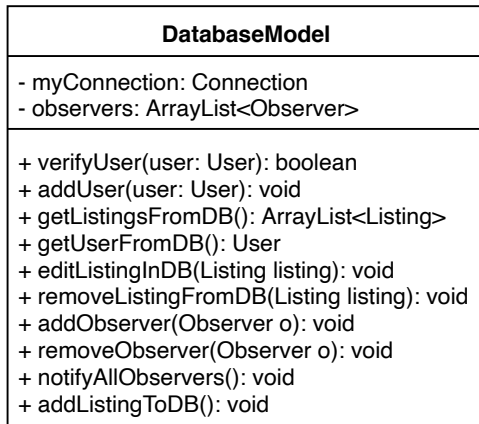
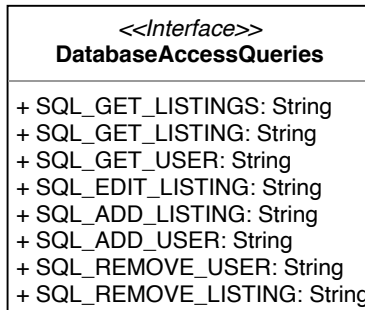
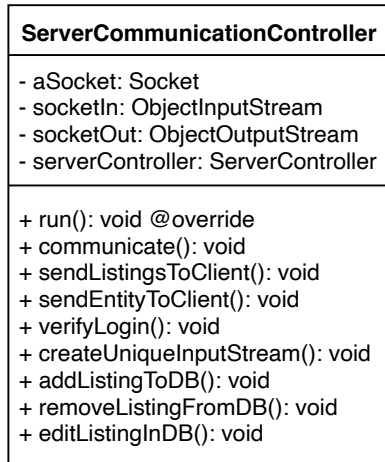
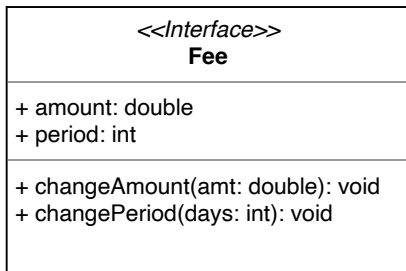
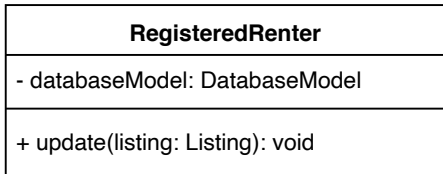
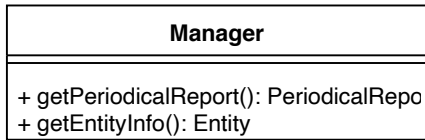
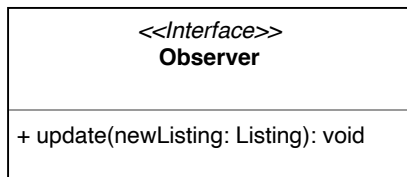
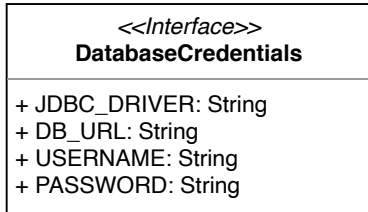
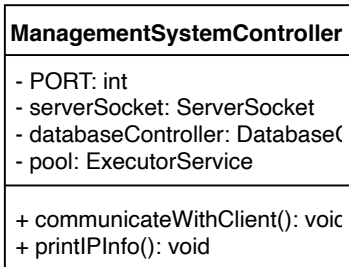
Use Case: Get entity info (entity can be renter, landlord, or property)



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(): void + viewEntityListen(): void

LoginController
- loginView: LoginView - verified: boolean - user: User
+ loginListen(): void + isVerified(): boolean

SearchListingController
- searchListingView: SearchListingView
+ searchListingListen();

ListingsController
- listingsView: ListingsView
+ selectListingListen(): void

PeriodicalReportView
- components: Components
+ display(): void + hide(): void

PeriodicalReportController
- periodicalReportView: PeriodicalReportView
+ createReportListen(): void

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

LoginView
- components: Components
+ display(): void + hide(): void

CreateListingController
- createListingView: CreateListingView
+ createListingListen(): void

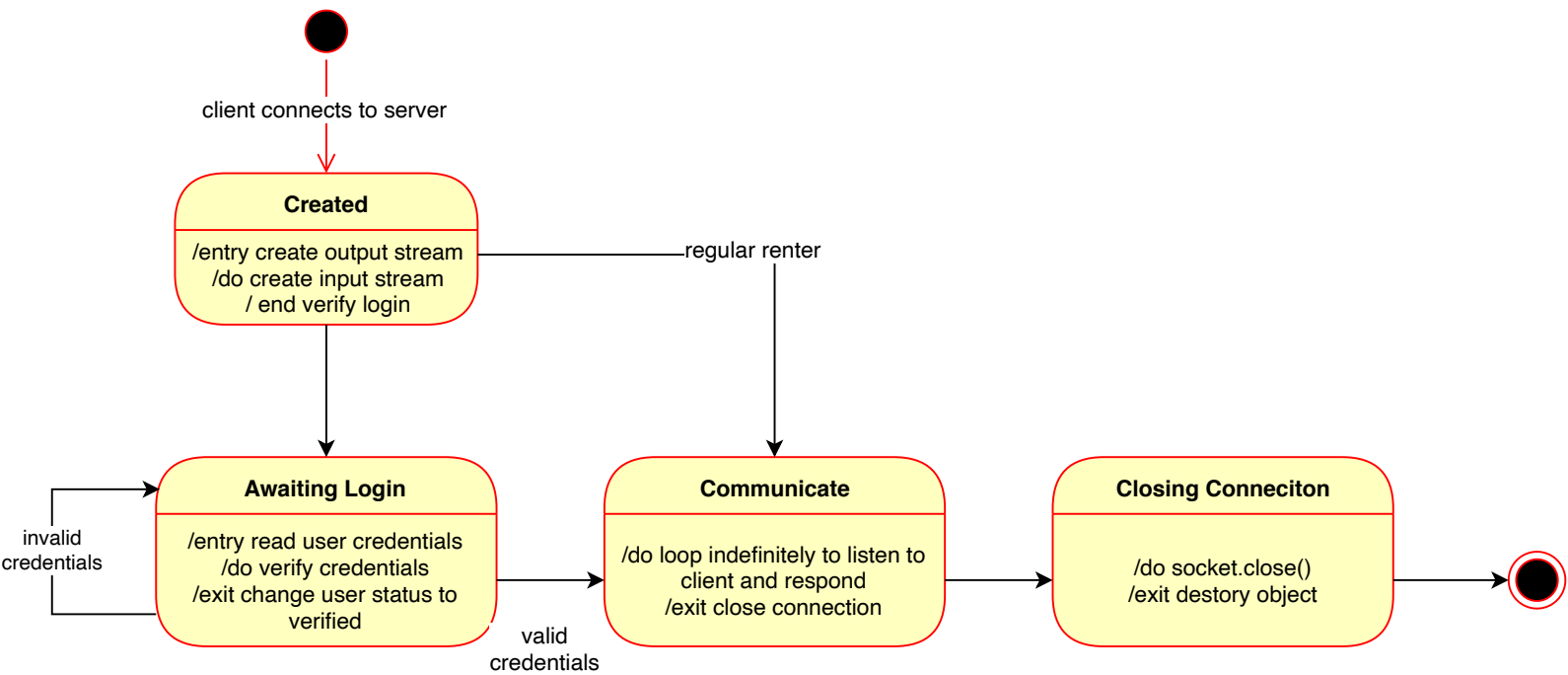
RenterMainView
- components: Components
+ display(): void + hide(): void

LandlordMainView
- components: Components
+ display(): void + hide(): void

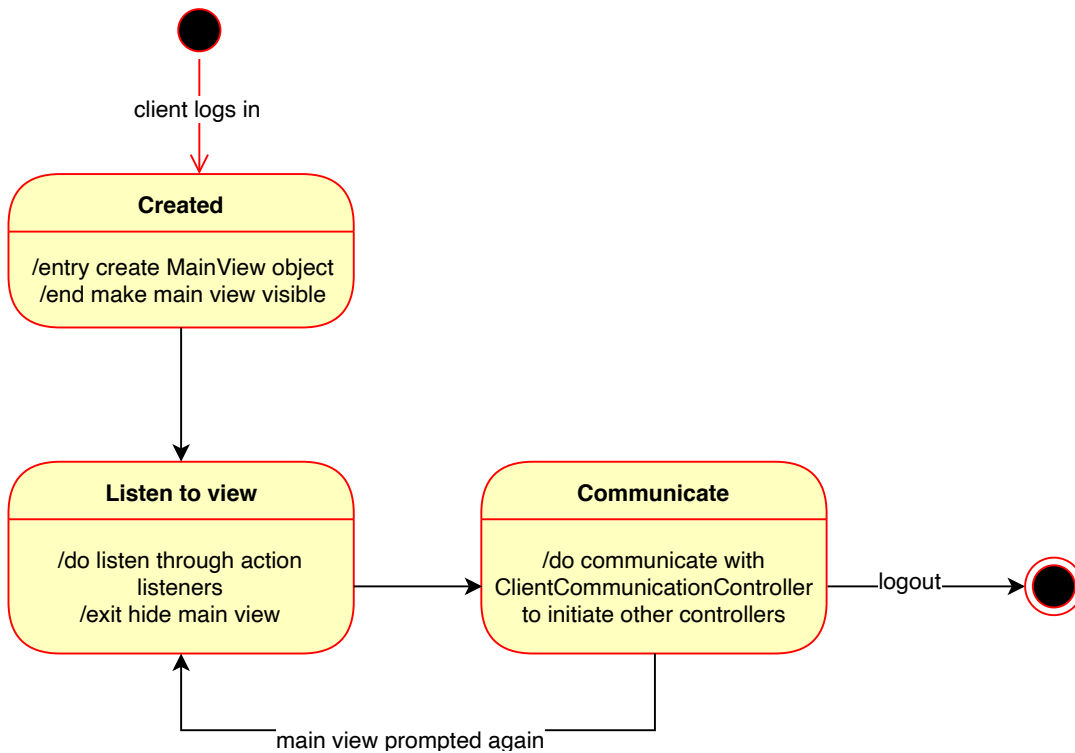
ManagerMainView
- components: Components
+ display(): void + hide(): void

MainView
- components: Components
+ display(): void + hide(): void

ServerCommunicationController State Diagram



MainController State Diagram



Package Diagram

Server

DataBase

Domain

Controllers

Observe

Utils

Client

Domain

Controllers

Utils

Presentation

Views

+uses

+uses

+request

+respond

+uses

+uses

Package Diagram

Server

DataBase

Domain

Controllers

Observe

Utils

Client

Domain

Controllers

Utils

Presentation

Views

+uses

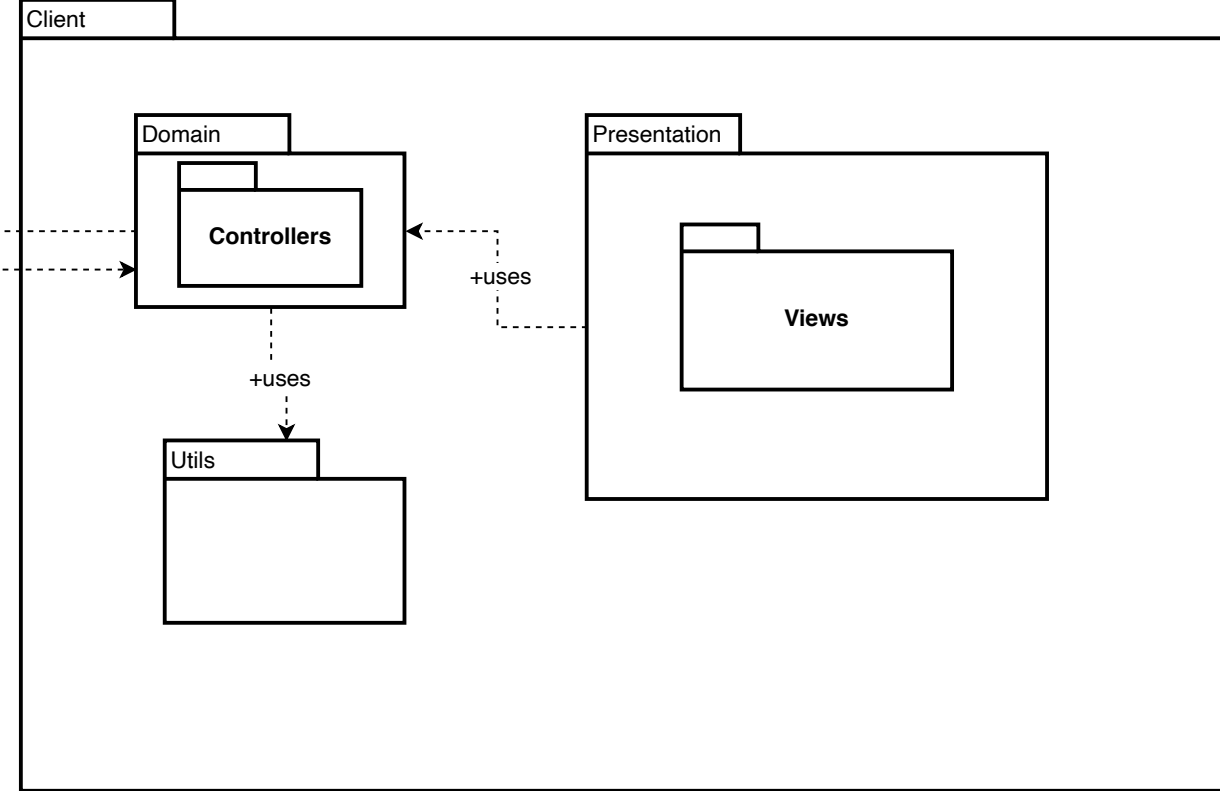
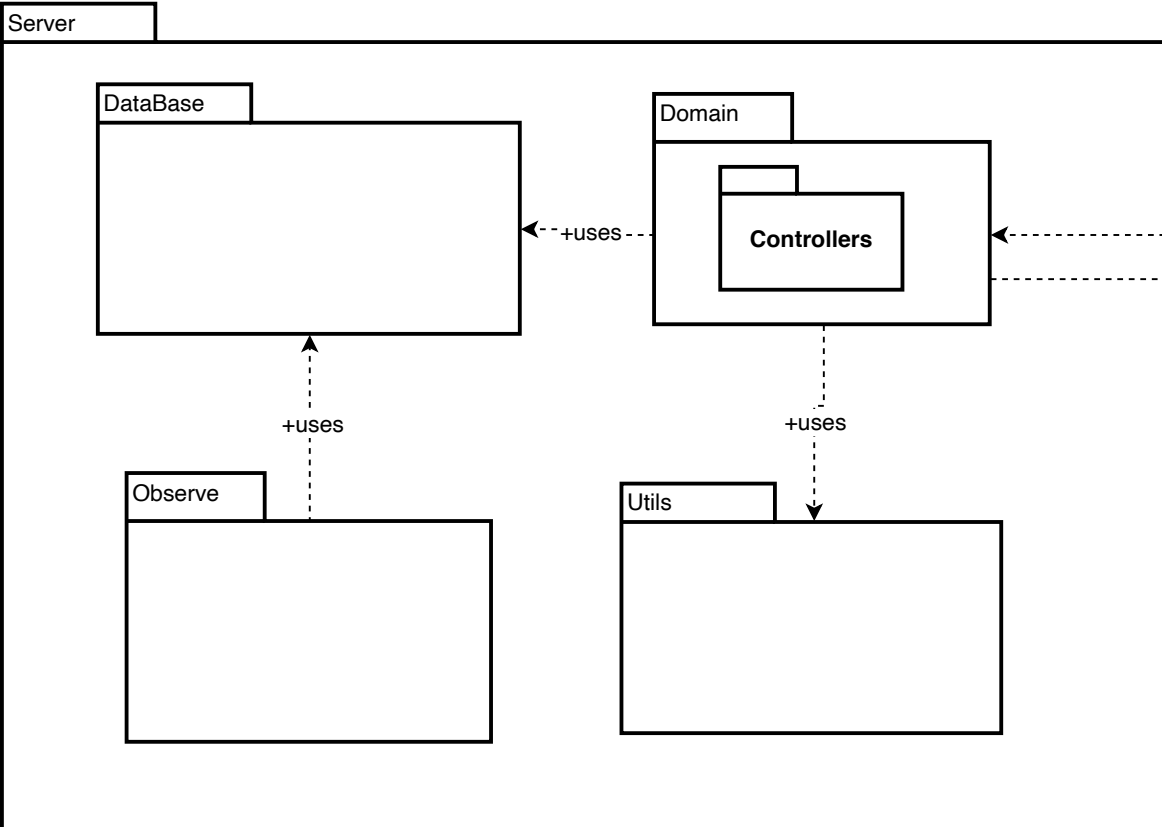
+uses

+request

+respond

+uses

+uses



Deployment Diagram

