Names: Harsohail Brar (30041921)

Gary Wu (30038110) Ryan Holt (30038609)

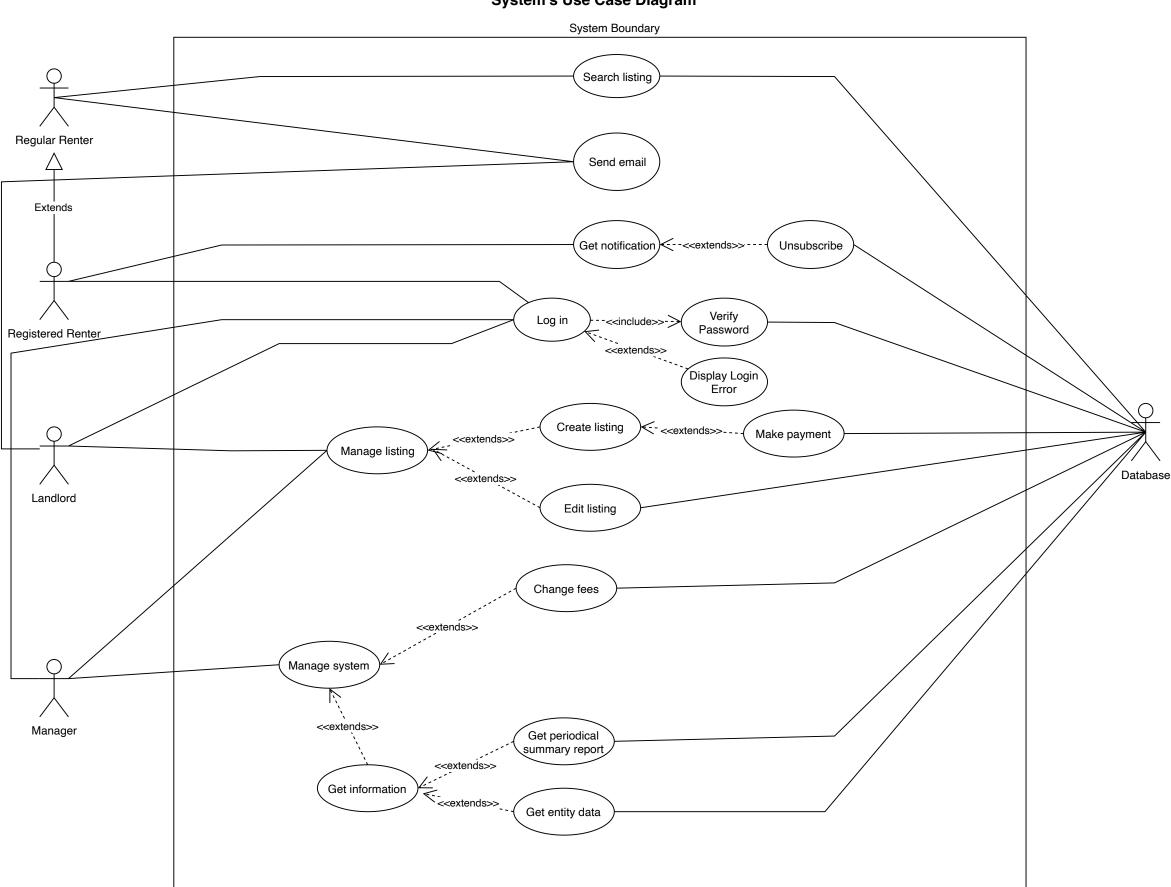
Course Name: Principles of Software Design

Course Code: ENSF 480

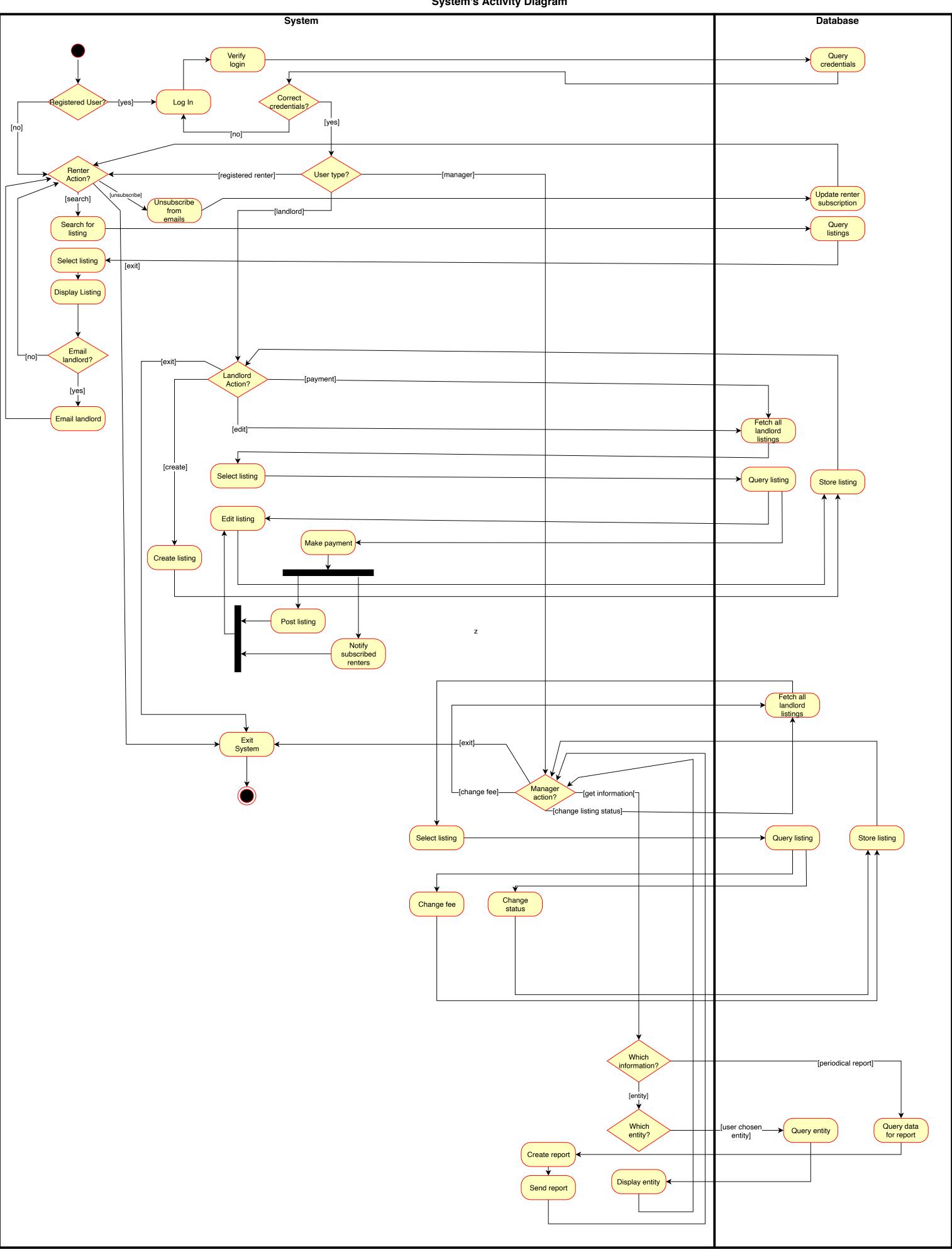
Assignment Number: Term Project Design Document

Submission Date and Time: 09/11/2019

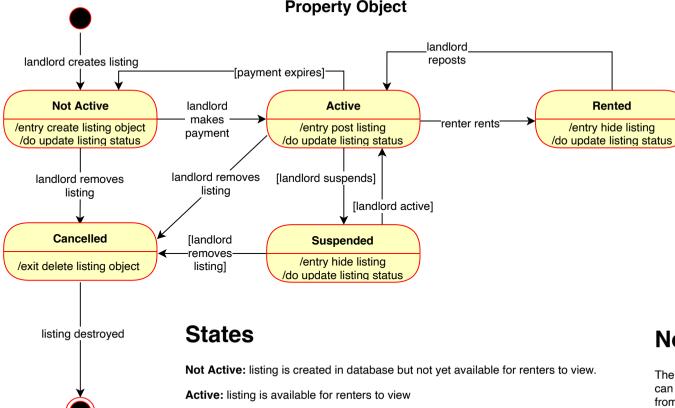
System's Use Case Diagram



System's Activity Diagram



State Transition Diagrams Property Object



Rented: listing is not available for renters to view as it is rented but it is stored in

the database with the changed status

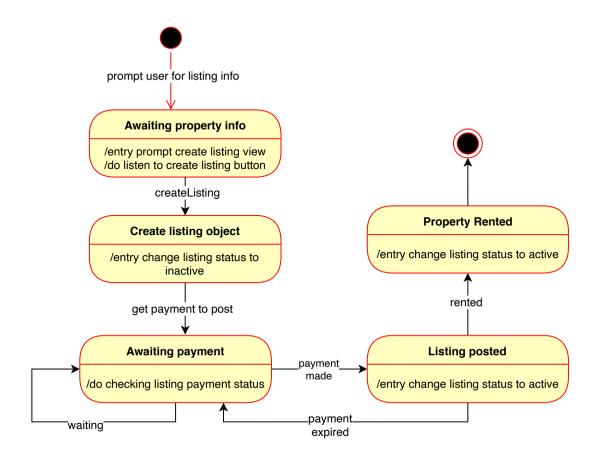
Cancelled: listing is deleted from the system

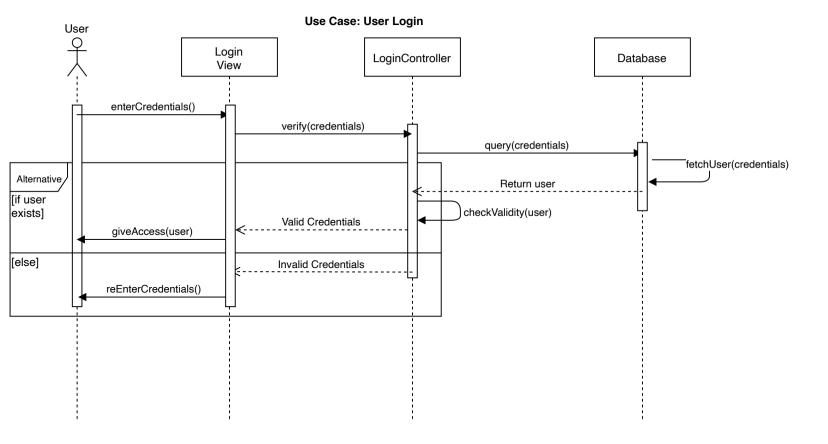
Suspended: landlord suspends the listing temporarly to hide it from renters viewing it. (ex. if landlord is the middle of signing a contract with a potential renter)

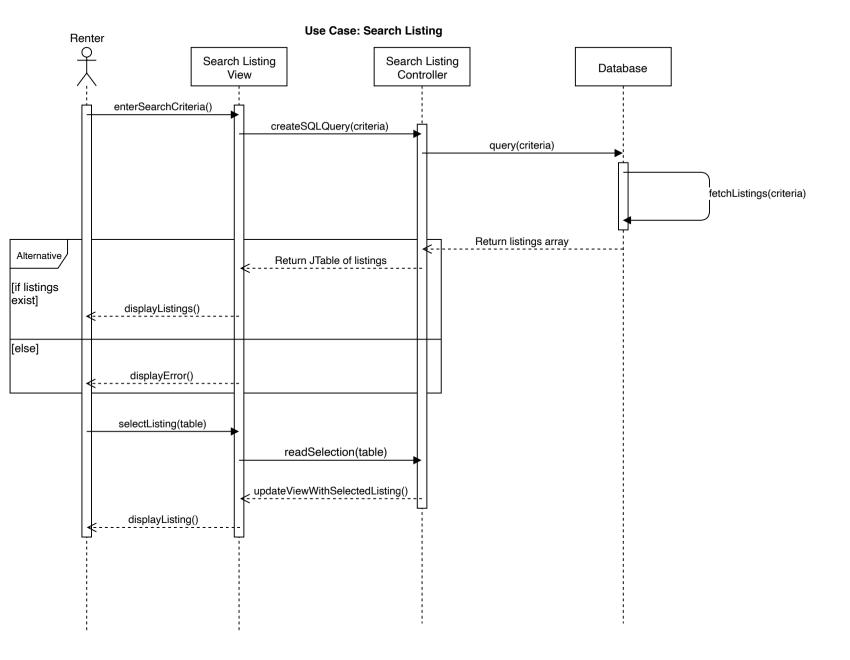
Note

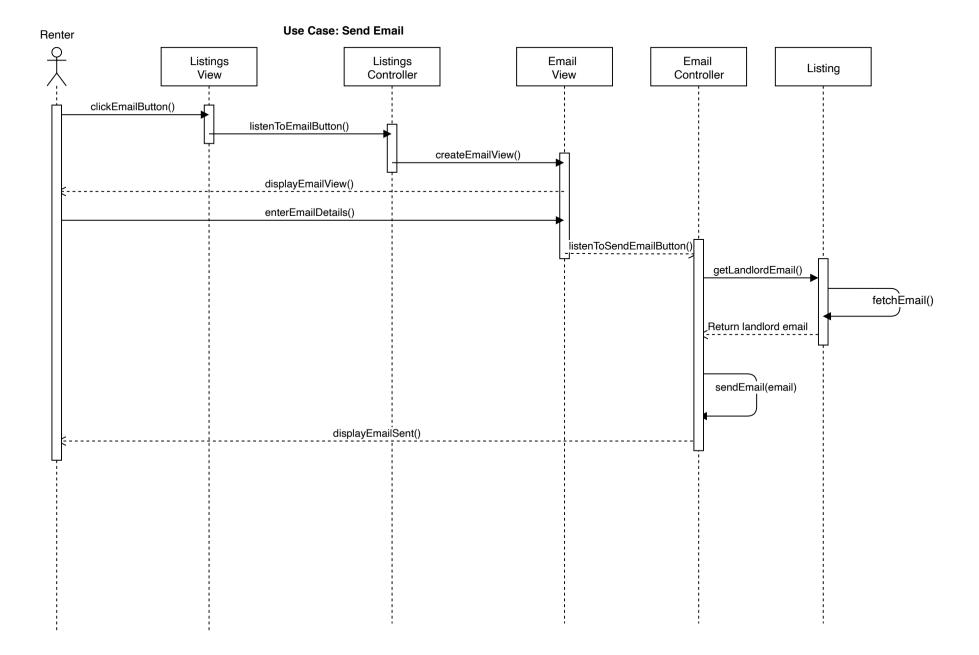
The external stimuli of a manager can change a property obejct from any state to any state which is not shown in the diagram to keep simplicity and clarity

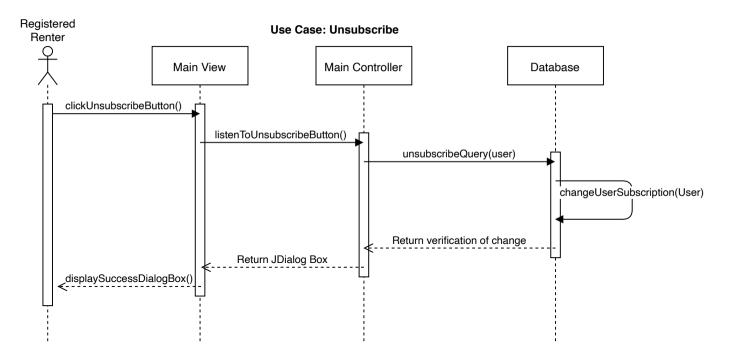
Process that landlord posts his/her property

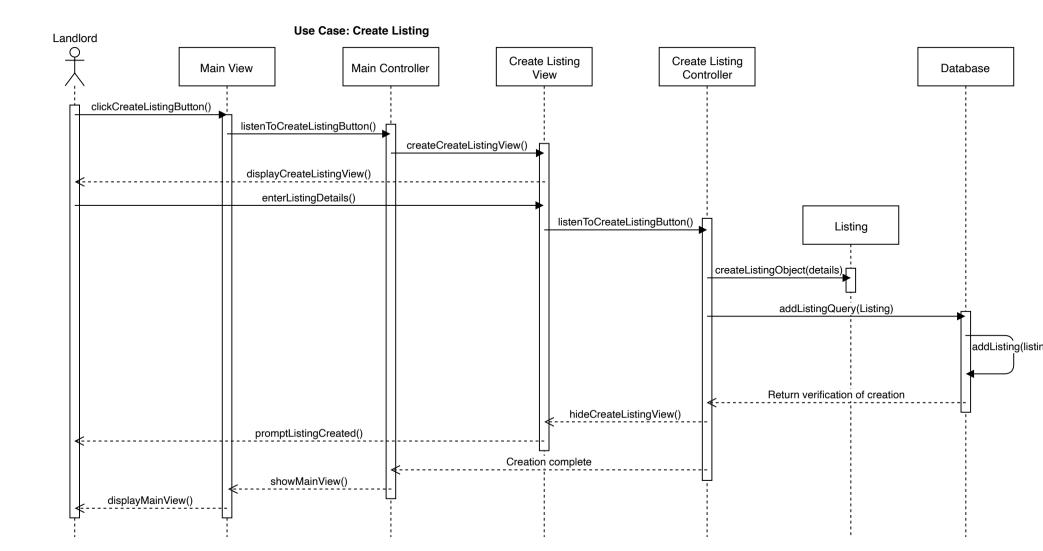


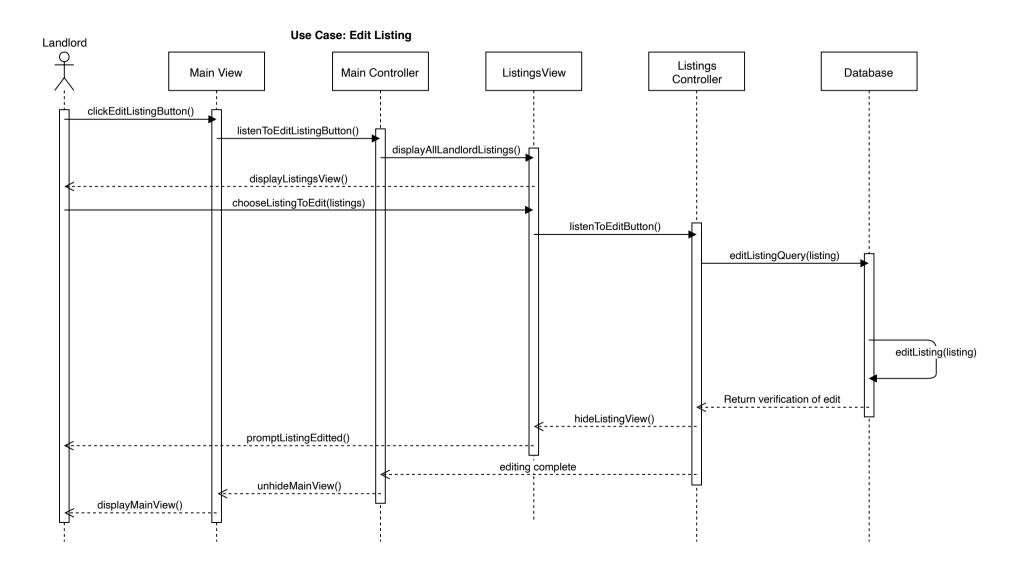


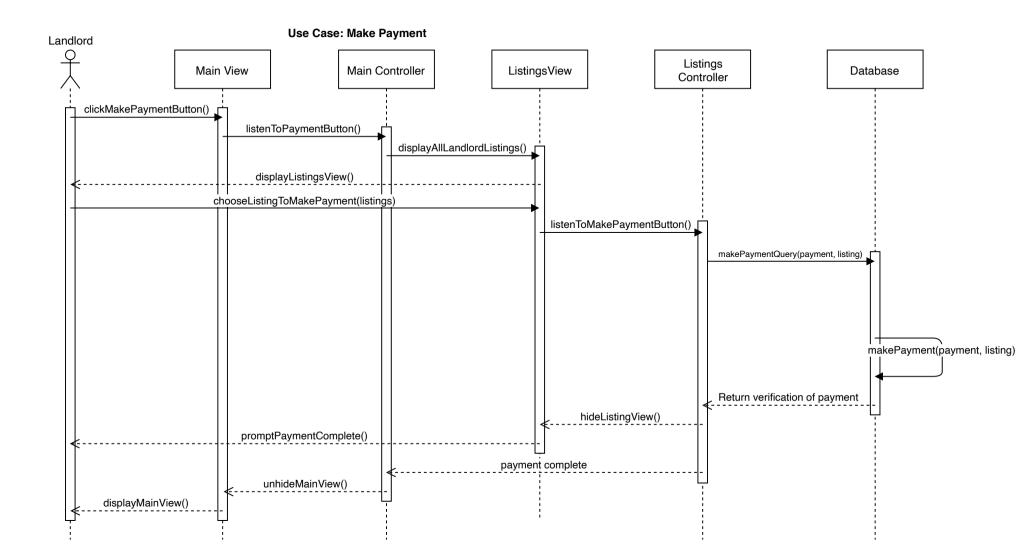


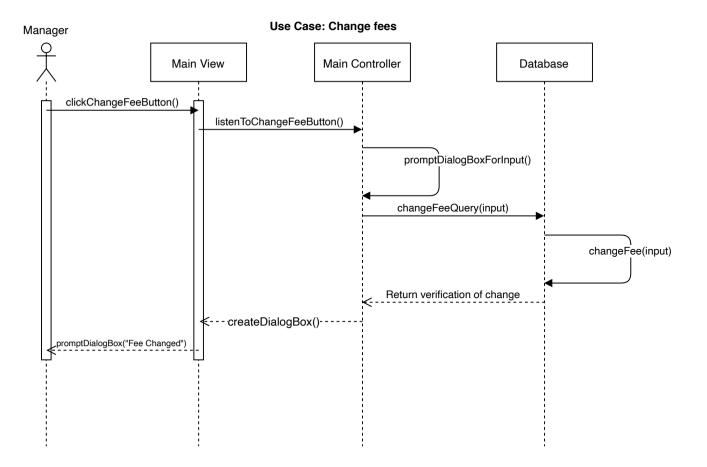


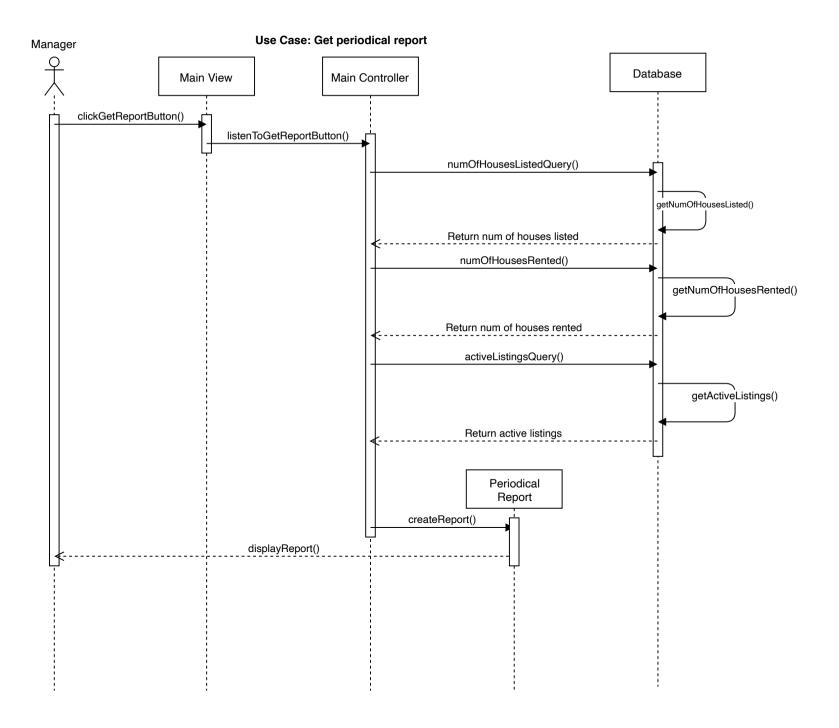


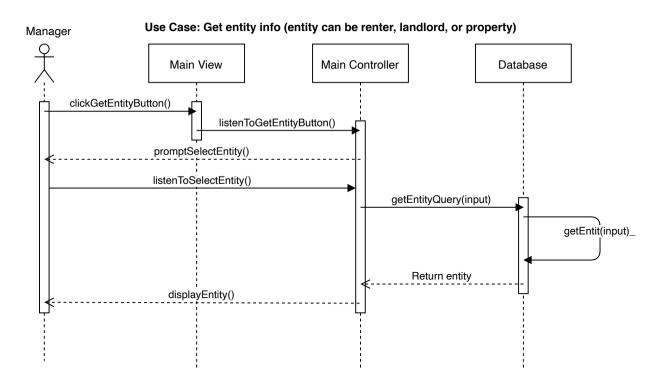


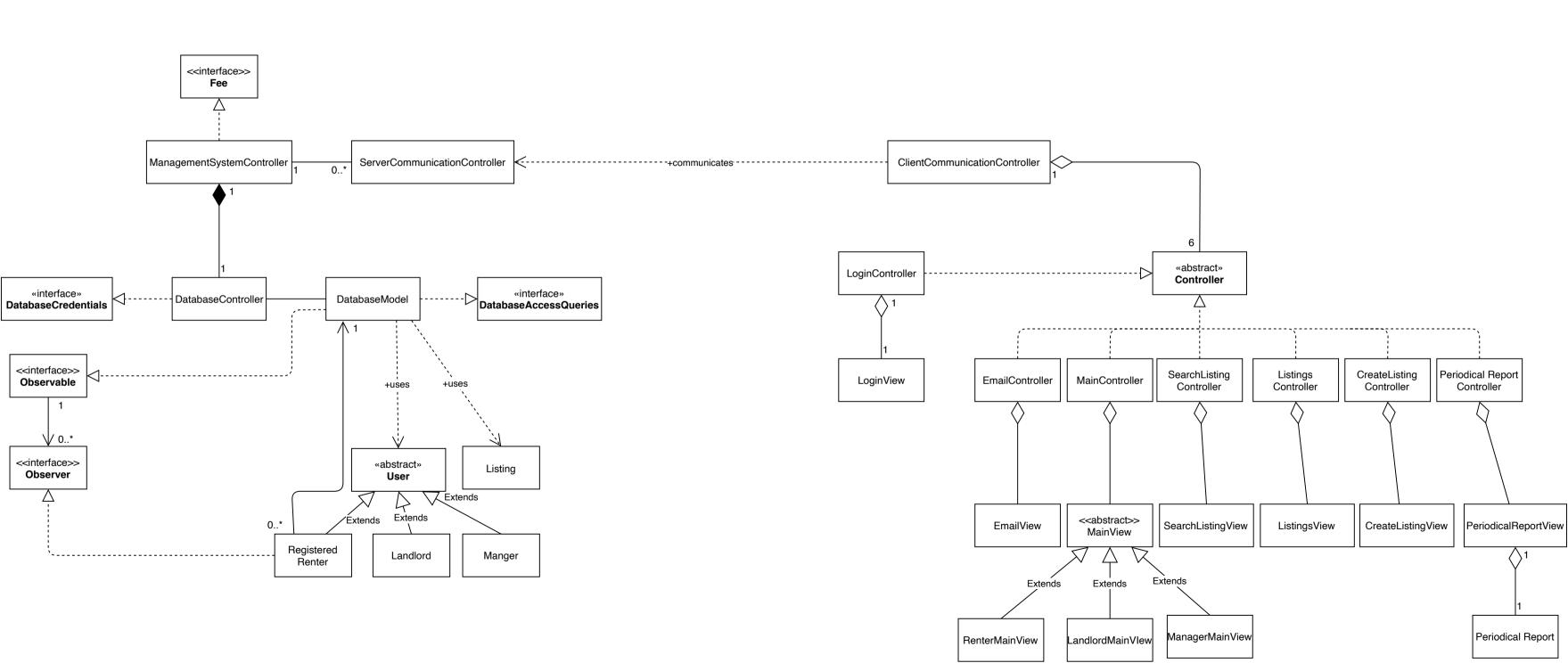












Attributes UML

ManagementSystemController

- PORT: int
- serverSocket: ServerSocket
- databaseController: Database(
- pool: ExecutorService
- + communicateWithClient(): voic
- + printlPInfo(): void

<<interface>> DatabaseCredentials

- + JDBC_DRIVER: String + DB_URL: String
- + USERNAME: String
- + PASSWORD: String

<<!nterface>> Observer

+ update(newListing: Listing): void

Manager

- + getPeriodicalReport(): PeriodicalRepo
- + getEntityInfo(): Entity

RegisteredRenter

- databaseModel: DatabaseModel
- + update(listing: Listing): void

ServerCommunicationController

- aSocket: Socket
- socketIn: ObjectInputStream
- socketOut: ObjectOutputStream
- serverController: ServerController
- + run(): void @override
- + communicate(): void
- + sendListingsToClient(): void
- + sendEntityToClient(): void
- + verifyLogin(): void
- + createUniqueInputStream(): void
- + addListingToDB(): void
- + removeListingFromDB(): void
- + editListingInDB(): void

<<Interface>>

DatabaseAccessQueries

- + SQL_GET_LISTINGS: String
- + SQL_GET_LISTING: String
- + SQL_GET_USER: String
- + SQL_EDIT_LISTING: String
- + SQL_ADD_LISTING: String
- + SQL_ADD_USER: String
- + SQL REMOVE USER: String
- + SQL_REMOVE_LISTING: String

DatabaseController

- myConnection: Connection
- databaseModel: DatabaseModel
- + initializeConnection(): void
- + closeConnection(): void

<<Interface>> Observable

- + addObserver(Observer o): void
- + removeObserver(Observer o): void
- + notifyAllObservers(): void

User

- # username: String
- # password: String # email: String
- # identity: String
- + compareUser(User: user): boolean
- + getIdentity(): String

DatabaseModel

- myConnection: Connection
- observers: ArrayList<Observer>
- + verifyUser(user: User): boolean
- + addUser(user: User): void
- + getListingsFromDB(): ArrayList<Listing>
- + getUserFromDB(): Üser
- + editListingInDB(Listing listing): void
- + removeListingFromDB(Listing listing): void
- + addObserver(Observer o): void
- + removeObserver(Observer o): void
- + notifyAllObservers(): void
- + addListingToDB(): void

Landlord

- listings: ArrayList<Listing>
- + addListing(listing: Listing): void
- + removeListing(listing: Listing): void

ClientCommunicationController

- socketOut: ObjectOutputStream
- aSocket: Socket
- socketIn: ObjectInputStream - loginController: Controller - emailController: Controller
- mainController: Controller
- $\hbox{-} search Listing Controller: Controller \\$
- listingsController: Controller
- createListingController: Controller
- + main(args: String[]): void
- + showMainWindow(): void
- + showEmailView(): void
- + showSearchListingView(): void
- + showListingsView(): void
- + showCreateListingView(): void

LoginController

- loginView: LoginViewverified: booleanuser: User
- + loginListen(): void + isVerified(): boolean

Controller

clientCommunicationController: ClientCommunicationController

EmailController

- emailView: EmailView
- + sendEmailListen();

SearchListingController

- searchListingView: SearchListingView
- + searchListingListen();

ListingsController

- listingsView: ListingsView
- + selectListingListen();

MainController

- mainView: MainView
- + searchListingListen(): void
- + editListingListen(): void
- + unsubscribeListen(): void
- + createListingListen(): void
- + getPeriodicalReportListen(): void
- + changeFeeListen();
- + viewEntityListen();

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

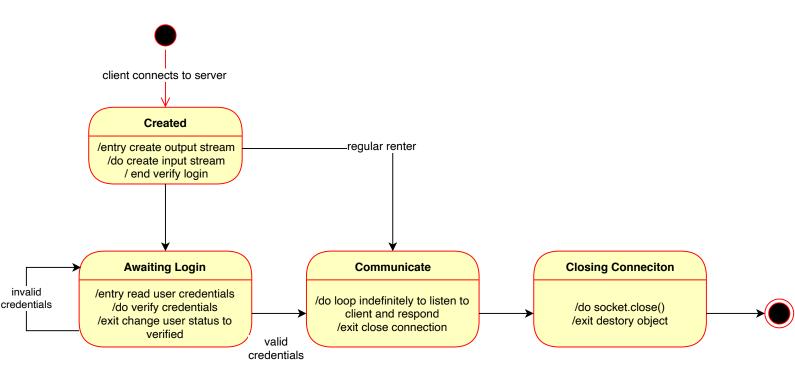
SearchListingView

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

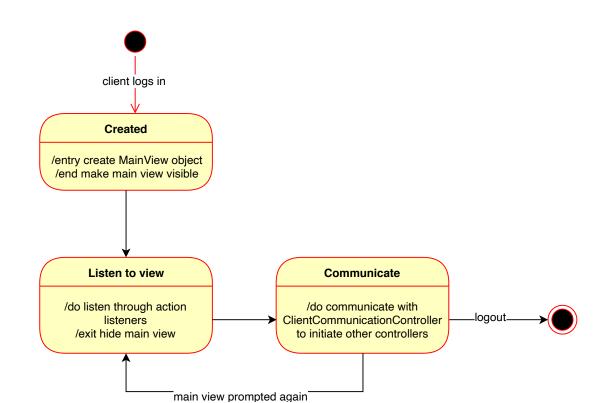
CreateListingView

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

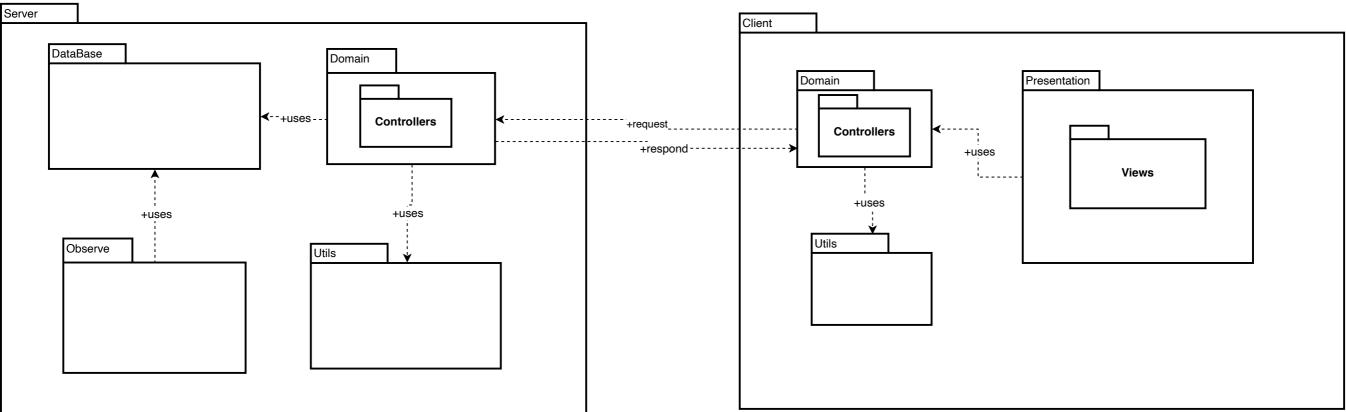
ServerCommunicationController State Diagram



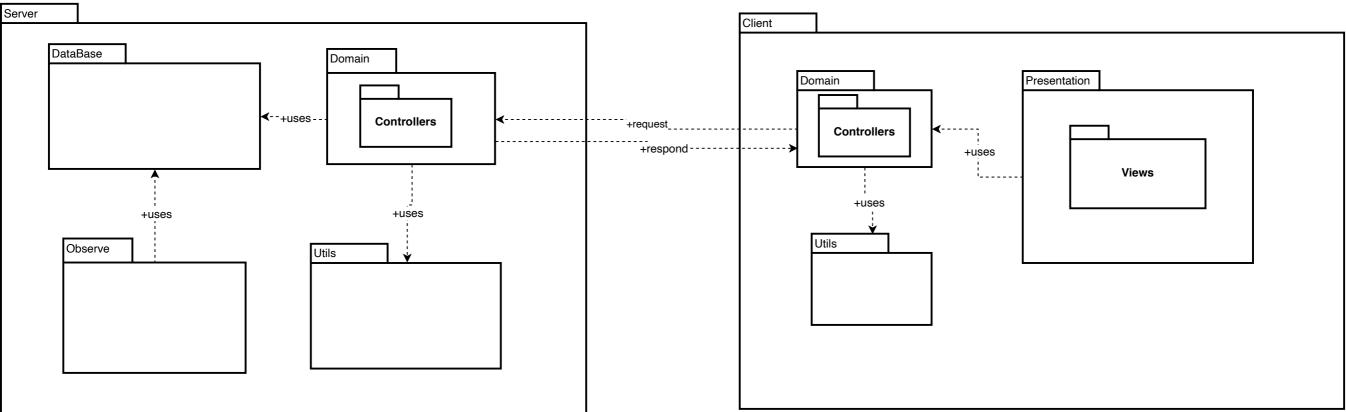
MainController State Diagram



Package Diagram



Package Diagram



Deployement Diagram

