Names: Alliana Alyssa Dela Pena, Kunal Dhawan, Hesham Elkaliouby,

Maram Elsayed

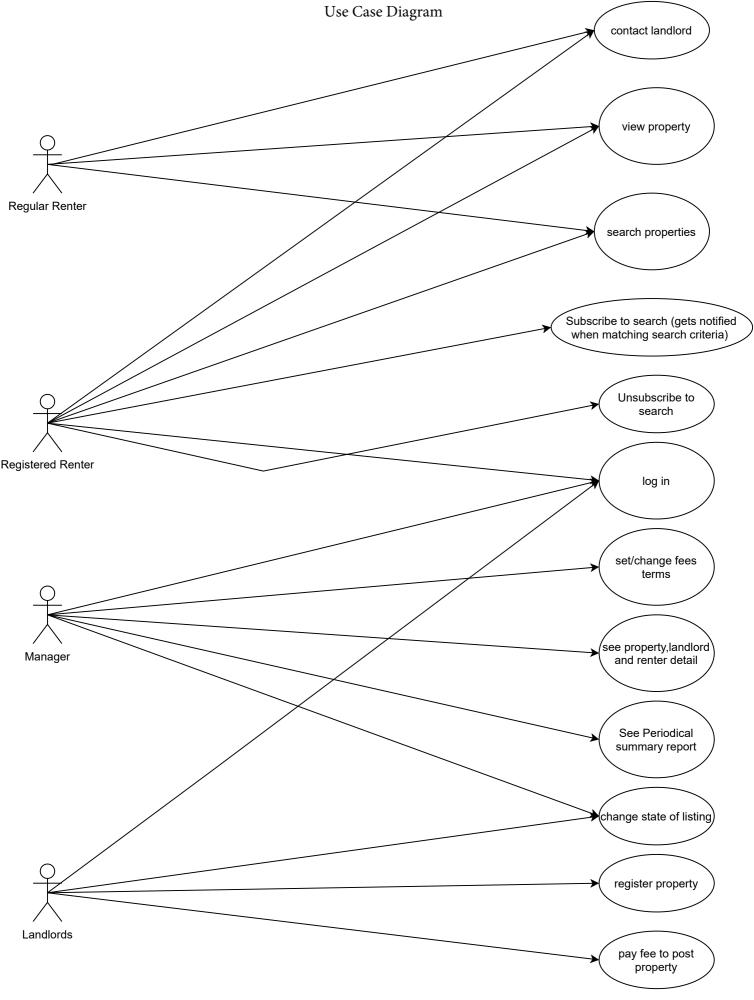
Course Name: Principles of Software Design

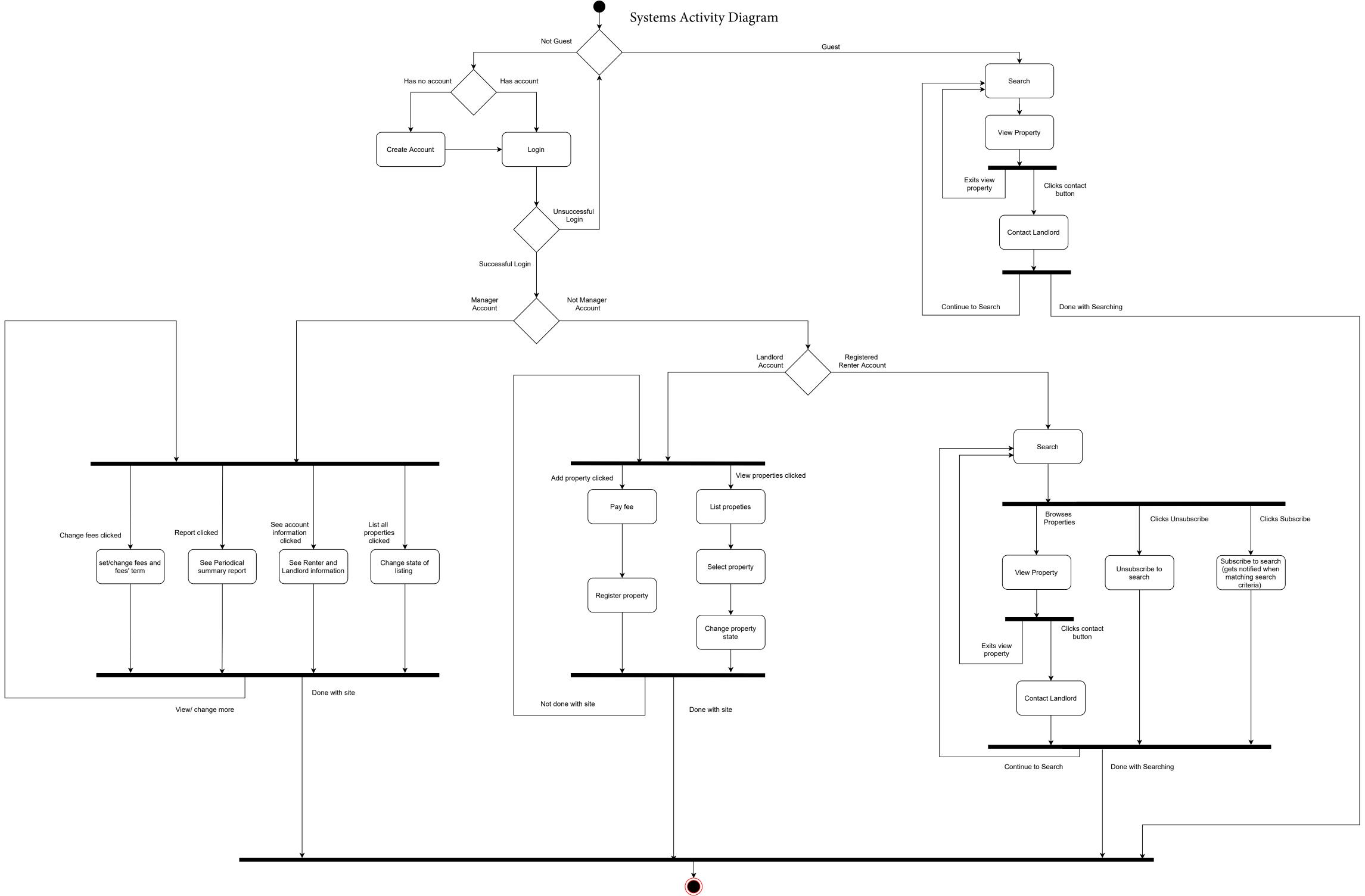
Lecture Section: L02

Course Code: ENSF 480

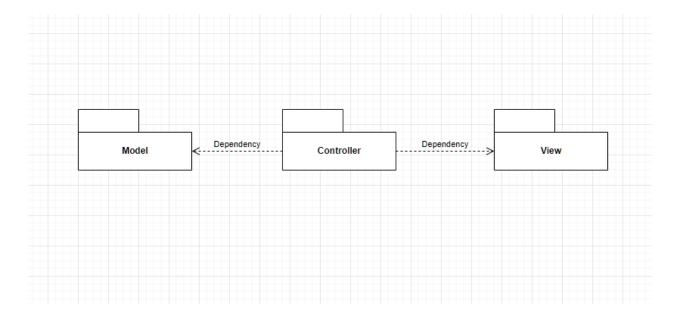
Assignment: Term Project Design Document

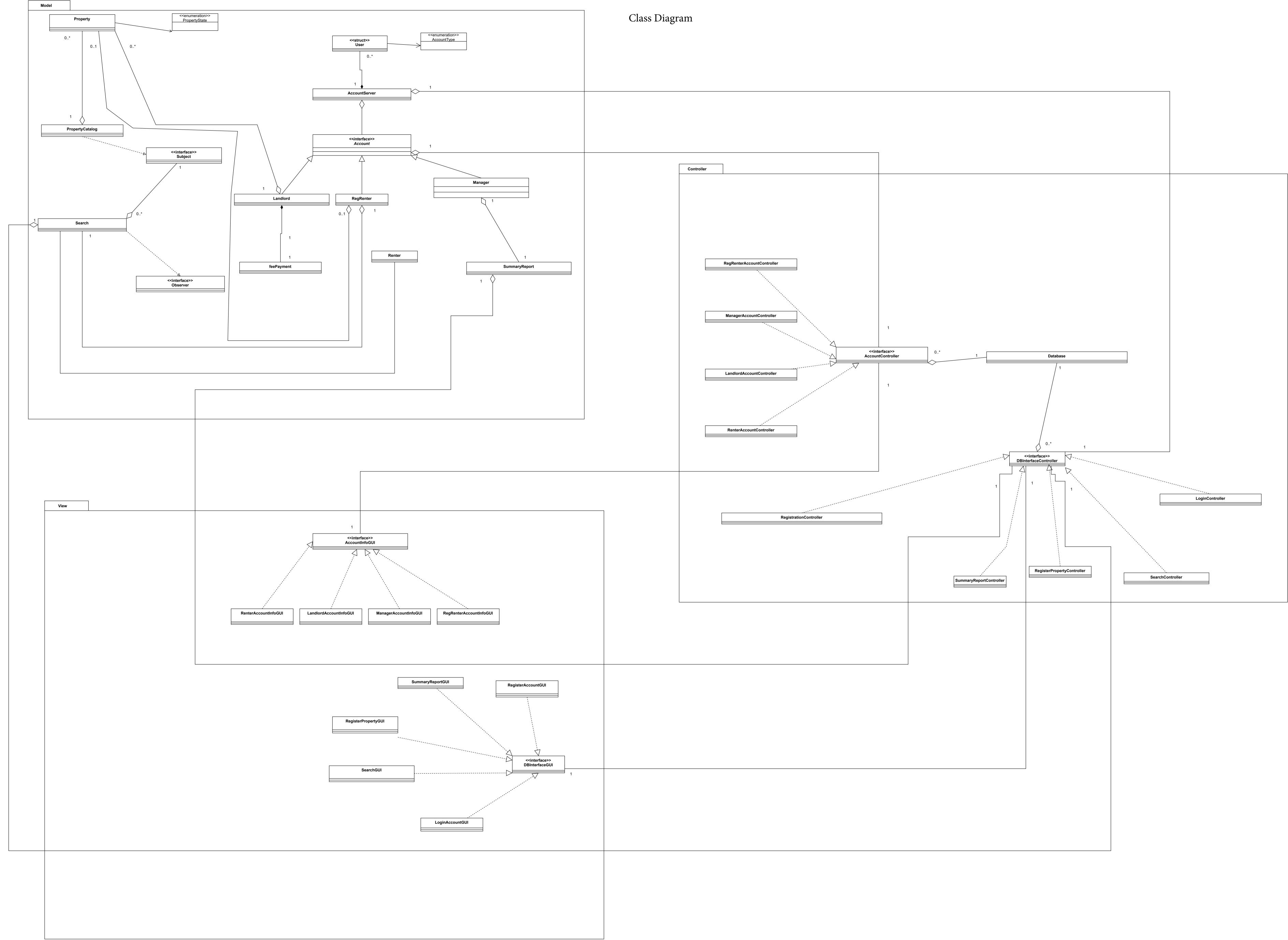
Submission Date: 29/11/2021





Package Diagram





Model

Property

- houseid: int
- status: PropertyState
- address: String
- type: String
- bedroomNo: int
- bathroomNo: int - furnished: String
- citvQuad: String
- cost: double landlord: Landlord
- dateListed: String - renter: Renter
- dateRented: String

<<interface>> Subject

- + registerObserver(Observer o): void + removeObserver(Observer o): void
- + notifyAllObserver(): void

PropertyCatalog

- catalog: ArrayList<Property> - observers: ArrayList<Observers>
- + registerObserver(Observer o): void
- + removeObserver(Observer o): void
- + notifyAllObserver(): void
- + addProperty(p: Property): void - removeProperty(address: string): void

<<interface>> Observer

+ update(catalog: ArrayList<Property>): void

Search

- subject: Subject
- type: ArrayList<String> - bedroomNo: ArrayList<int>
- bathroomNo: ArrayList<int>
- furnished: ArrayList<String>
- cityQuad: ArrayList<String>
- searchController: DBInterfaceController
- + addType(type: String): void
- + addBedroomNo(num: int): void + addBathroomNo(num: int): void
- + addFurnished(f: String): void
- + addCityQuad(quad: String): void
- + removeType(type: String): void + removeBedroomNo(num: int): void
- + removeBathroomNo(num: int): void
- + removeFurnished(f: String): void + removeCityQuad(quad: String): void
- + becomeSubscribed(): void + getNotifiedByEmail(): void

<<enumeration>> <u>PropertyState</u>

RENTED

ACTIVE

CANCELLED

SUSPENDED

- + username: string + password: string

<<struct>>

User

- + name: string + email: string
- + accountType: AccountType

<<enumeration>> AccountType LANDLORD

MANAGER

AccountServer

users: vector<User>

- <<static>> ~ instance: AccountServer
- · loginController: DBInterfaceController
- REGISTEREDRENTER

<<static>> + getInstance(): AccountServer +add(username: string, password: string): void +validate(username: string, password: string): User

<<interface>> Account

accountConroller: AccountController

Landlord

- fee: feePayment
- registraionController: DBInterfaceController - properties: ArrayList<Property>
- + registerProperty(): void

+ payFee(): void

RegRenter subscription: Search

performSearch(): void

rental: Property

Manager

- report: SummaryReport
- + generateSummaryReport(): void

feePayment

- paid: boolean
- + timePeriod: Date + amount: double
- pay(paid: boolean, amount: double): void
- + periodCountdown(timePeriod: Date): void
- getNotified(): Notifications modifyAmount(): void

SummaryReport

- catalog: ArrayList<Property>
- totalNo_of_Properties:int
- noOfActiveProperties:int noOfRentedProperties: int
- summaryController: DBInterfaceController
- + update(catalog: ArrayList<Property>): void
- + displayReport(): void
- + countProperties(): void
- + countActiveProperties(): void + countRentedProperties(): void

Renter

AccountInfoGUI

<<interface>>

Detailed Class Specification

+ showGUI(): void

showGUI(): void

LandlordAccountInfoGUI

RenterAccountInfoGUI

showGUI(): void

ManagerAccountInfoGUI

showGUI(): void

RegRenterAccountInfoGUI

+ showGUI(): void

<<interface>> **DBInterfaceGUI**

+ showGUI(): void

RegisterPropertyGUI

+ showGUI():void

SearchGUI

- + update(catalog: ArrayList<Property>): void
- + showAllPropertiesGUI(): void
- + showOnePropertyGUI(): void emailLandLord(): void
- searchCatalog(): void

LoginAccountGUI

showGUI():void

SummaryReportGUI

+ showGUI():void

RegisterAccountGUI

+ showGUI():void

Database

- + DBURL: String
- + USERNAME: String
- + PASSWORD: String - connect: Connection
- + addAccount(account: Account): void
- + getAccountInfo(username: String): Account
- + addProperty(property: Property): void
- + getProperty(id: int): Property
- + updatePropertyState(property: Property, state: String): void
- + addSubscription(renter: regRenter, subscription: Subscription): void + changeFeePeriod(periodDays: int): void
- + changeFeeAmount(amount: int): void + updateProperty(property: Property): void
- + getPropertyList(): ArrayList<Property>
- + updateMessage(message: String, property: Property): void + initializeConnection(): void

<<interface>> AccountController

#guiStyle: AccountInfoGUI

#database: Database

+ performGUIStrategy(): void

+ updateDatabase(): void

+ performGUIStrategy(): void

+ updateDatabase(): void

RegRenterAccountController

ManagerAccountController

+ performGUIStrategy(): void

+ updateDatabase(): void

LandlordAccountController

- performGUIStrategy(): void + updateDatabase(): void

RenterAccountController

+ performGUIStrategy(): void

+ updateDatabase(): void

<<interface>>

DBInterfaceController # database: Database

RegistrationController

+ createAccount(username: String, password: String, name: String, email: String): void

SummaryReportController

RegisterPropertyController

- regProperty: Property

+ registerProperty(): void

SearchController

+ fetchPropertyCatalog(): PropertyCatalog +emailLandLord(property: Property): void

LoginController

instance: AccountServer

+ validate(username: String, password: String):

ManagerAccountController

3: Login Successsful

5: Forward property state

ManagerAccountInfoGUI

2: Validate information

4: Enter Property State

6: Save property state

Database

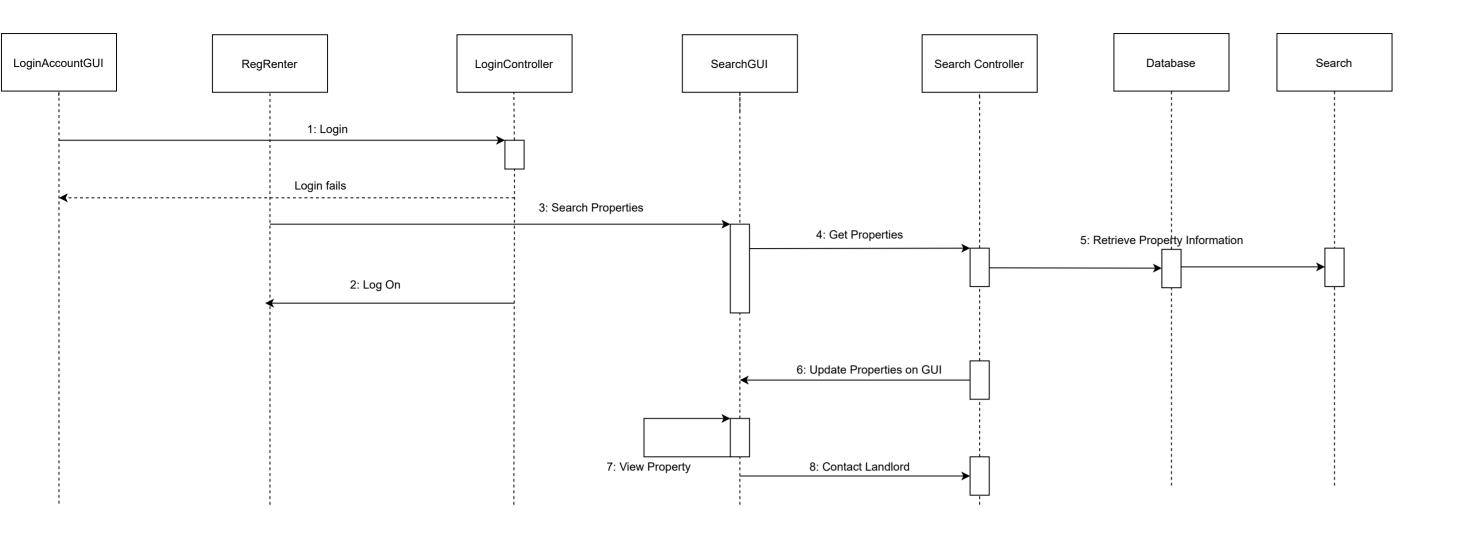
LoginController

Manager

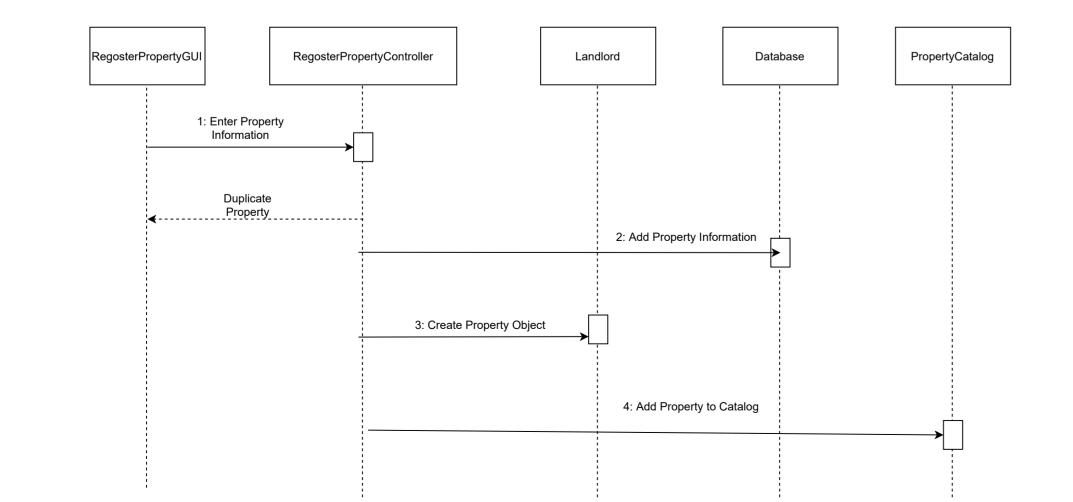
Login fails

1: Login

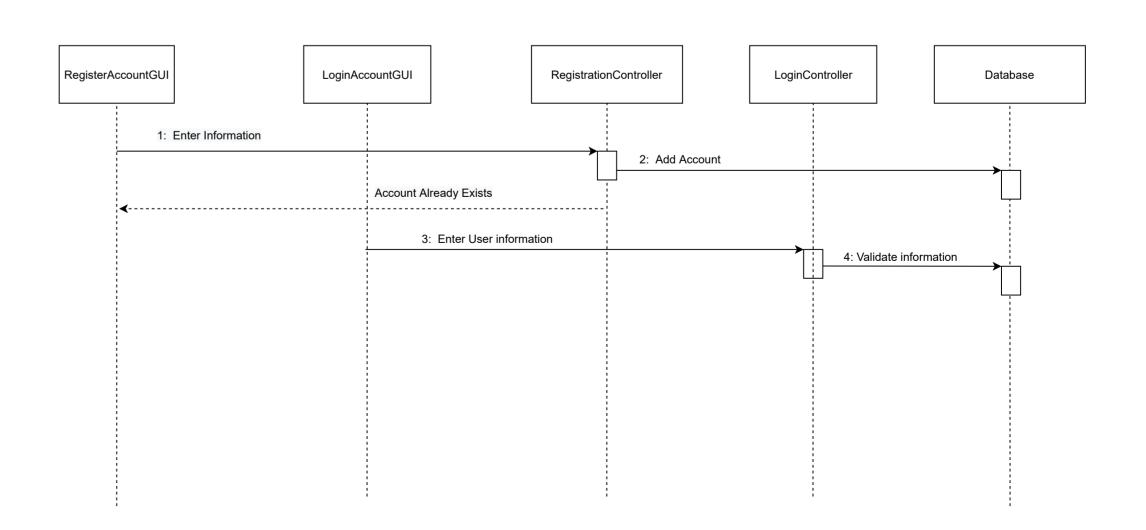
LoginAccountGUI



Landlord Posts Property

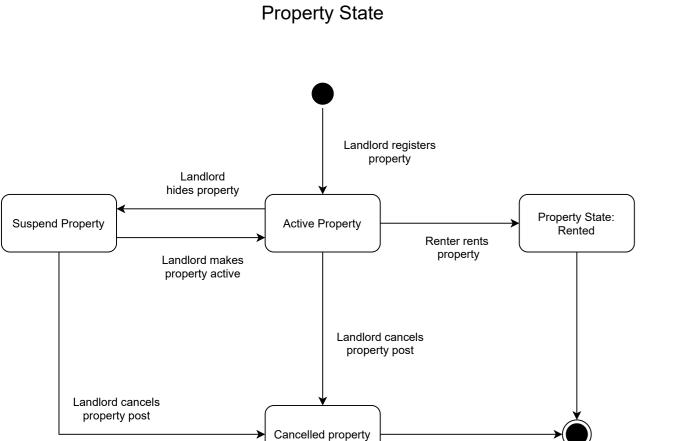


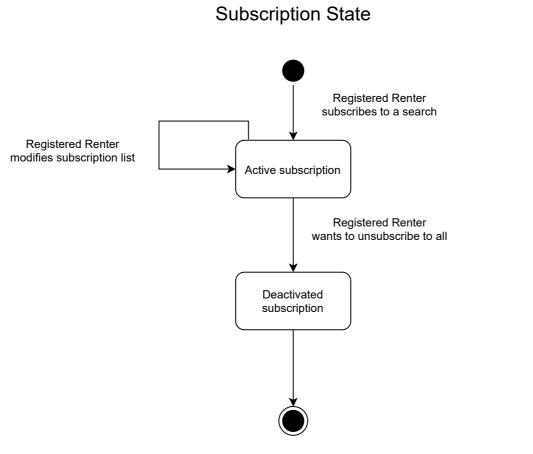
Registered Renter Registering and Logging to an Account

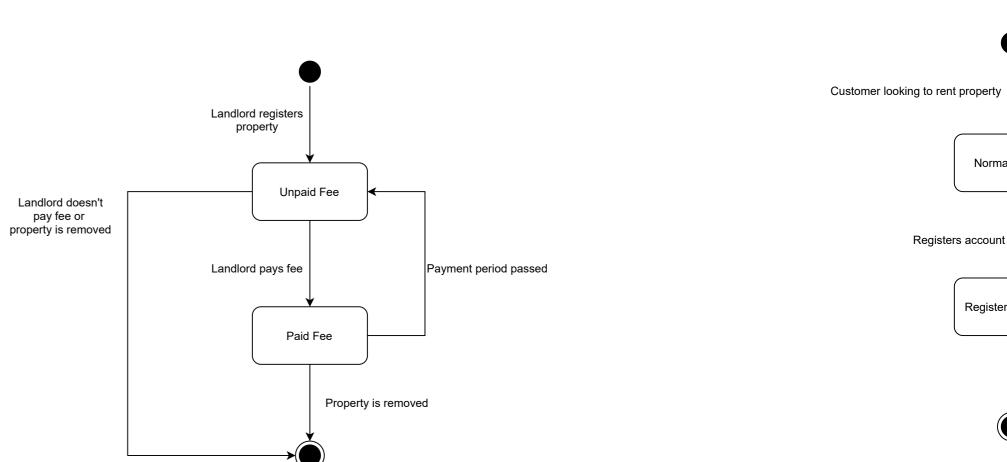


Normal Renter

Registered Renter







Fee Payment State

