Rental Property Management System

ENSF 480 - Principles of Software Design
Design Documentation
Lecture 01 - Group #2

By: Gibran Akmal, Rohan Amjad, Azlan Amjad, Ibrahim Asad

Table of Contents

Summary

This design documentation outlines the design specifications, diagrams, and models for the Rental Property Management System. The Rental Property Management System allows Landlords to register their properties and pay certain fees to activate their listings for Renters to view. Unregistered Renters can search and view properties, and send emails to Landlords. Registered Renters—in addition to the Unregistered Renter functionalities—can subscribe or unsubscribe to/from a search criteria and be notified when properties matching their criteria are activated.

The Rental Property Management System allows Managers to modify fee amounts and period, modify Property listing state, and view all Property listings and Landlords. The managers can request periodical summary reports that show the total number of houses listed in a period, total number of houses rented in a period, total number of active listings, and a list of rented houses in a period.

The Rental Property Management System allows users to register as Landlords, Managers, Registered Renters. Each user must login to access the user specific functionalities listed above.

1. Design Diagrams

1.1 System Use Case Diagram

Rental Property Management System

- Signup
- <<Extend>>
- Login
- Subscribe Search Criteria
- Unsubscribe
- Search Properties
- Send Landlord Email
- Register Property
- <<Include>>
- Pay Fee
- Delete Property
- Change Fees
- View Company DBS
- <<Extend>>
- <<Include>>
- Generate Periodical Summary Report
- Change Listing State
- Change Listing Details
- View My Properties

Registered Renter

Company DataBase System

User

Renter

Landlord

Manager

Figure 1.1 System Use Case Diagram

1.2 System Activity Diagram
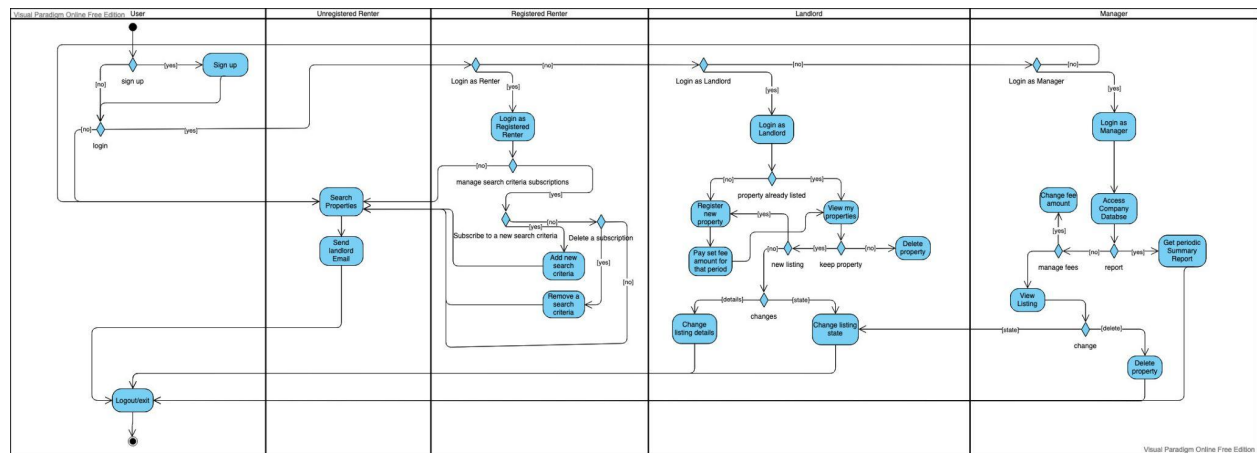
Figure 1.2 System Activity Diagram

1.3 System Sequence Diagrams

1.3.1 Sign-up Sequence Diagram

User

<<boundary>>
SignupGUI

<<control>>
SignupController

<<entity>>
CompanyDatabaseModel

1 : display view

2 : click signup button

3 : display choice of user

4 : choose user type

alt

[Renter or Landlord]

5 : display email, password fields

6 : enter email, password

[Manager]

5 : display email, password, manager key fields

6 : enter email, password, manager key

7 : create user

8 : validate email and password

9 : get emails and passwords

10 : return result set

opt

[Manager]

11 : validate manager key

12 : get valid manager key

13 : return result set

alt

[Invalid Input]

14 : error in creating user

15 : display error message

[Valid Input]

14 : create user

<<create>>
15 :

<<entity>>
UserModel

16 : user successfully created

17 : success in creating user

18 : display success message

Figure 1.3.1 Sign-up Sequence Diagram

1.3.2 Log-in Sequence Diagram

Figure 1.3.2 Log-in Sequence Diagram

1.3.3 Register Property Sequence Diagram
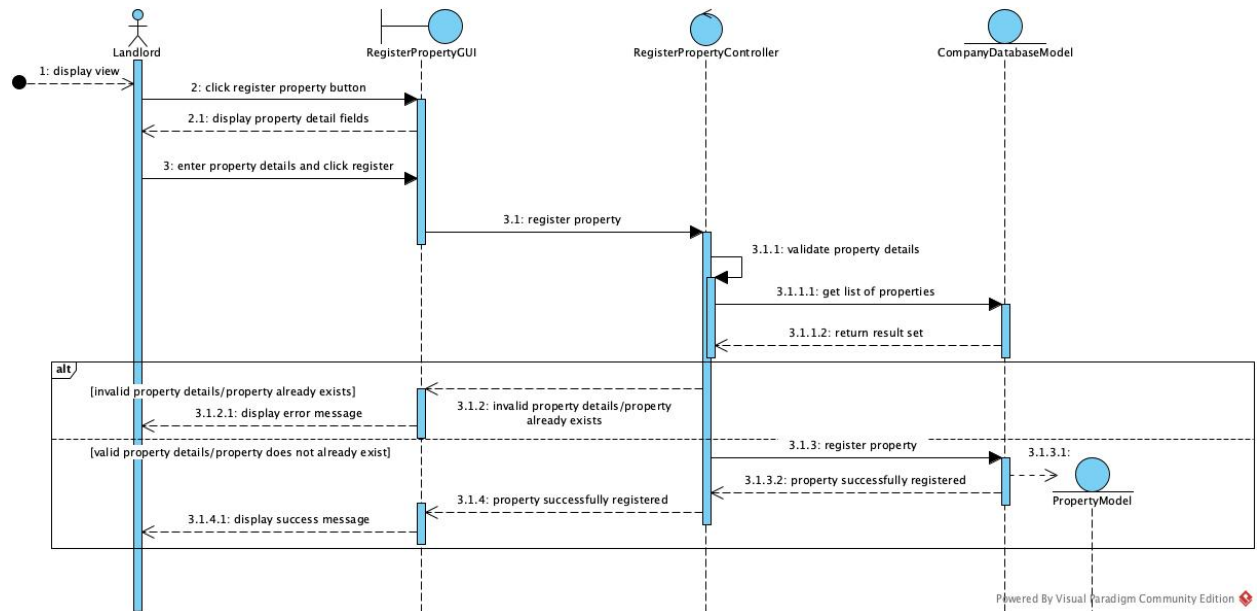
Figure 1.3.3 Register Property Sequence Diagram

1.3.4 Subscribe and Unsubscribe Sequence Diagram
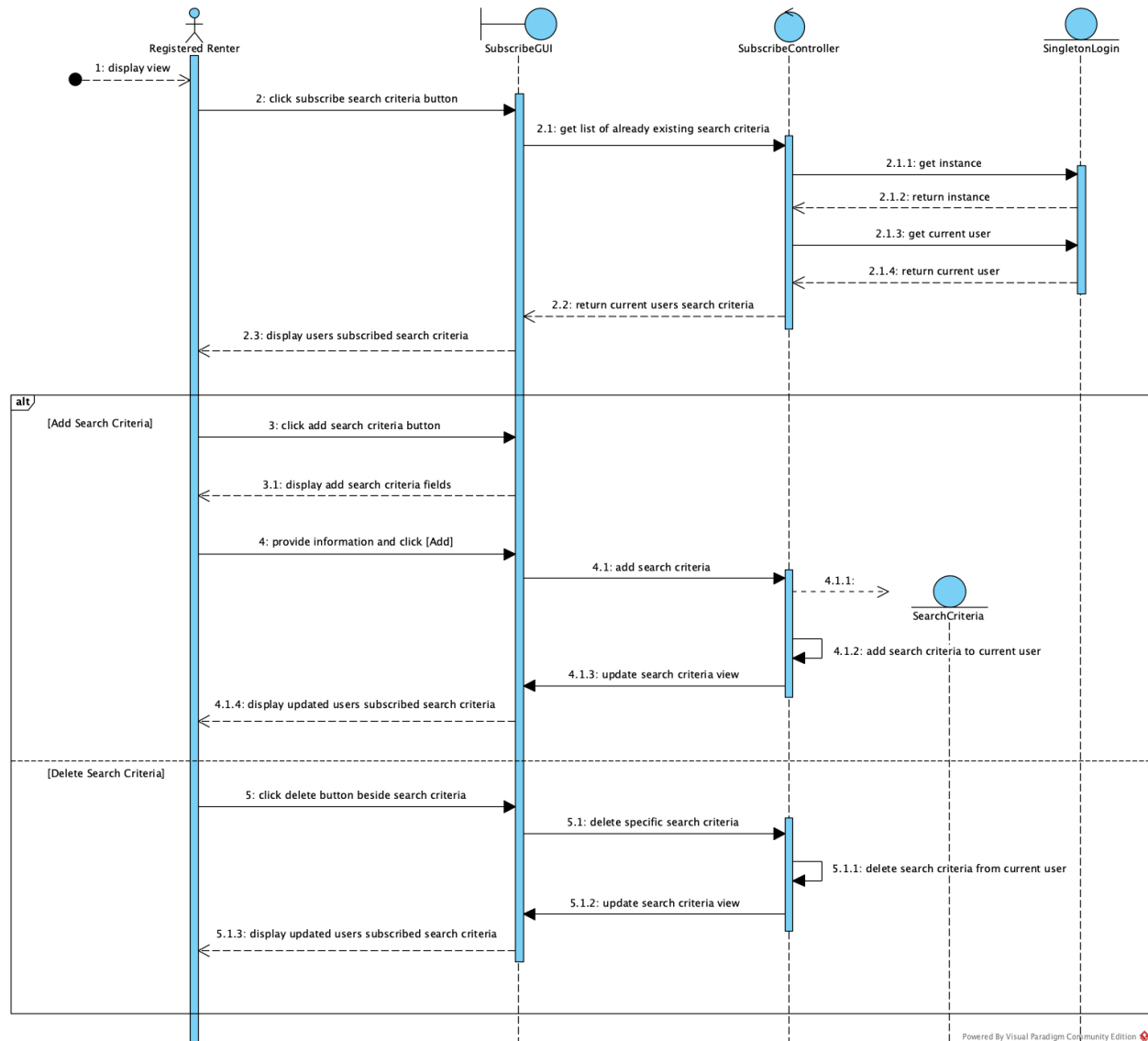
Figure 1.3.4 Subscribe and Unsubscribe Sequence Diagram

1.4 Class Relationship Diagram

Figure 1.4 Class Relationship Diagram (Partial)

1.5 Class Specification Diagram

1.5.1 GUIComponents Class Specification Diagram

GUIComponents

RegisterPropertyGUI
-titleLabel : JLabel
-title : JTextField
-descriptionLabel : JLabel
-description : JTextField
-propertyTypeLabel : JLabel
-propertyType : JTextField
-rentLabel : JLabel
-rent : JTextField
-bedroomsLabel : JLabel
-bedrooms : JTextField
-bathroomsLabel : JLabel
-bathrooms : JTextField
-squareFeetLabel : JLabel
-squareFeet : JTextField
-streetNameLabel : JLabel
-streetName : JTextField
-postalCodeLabel : JLabel
-postalCode : JTextField
-cityQuadrantLabel : JLabel
-cityQuadrant : JTextField
-cityLabel : JLabel
-city : JTextField
-stateProvinceLabel : JLabel
-stateProvince : JTextField
-countryLabel : JLabel
-country : JTextField
-furnishedLabel : JLabel
-furnished : JTextField
-registerPropertyButton : JButton
-doneButton : JButton
-cancelButton : JButton
-registerPropertyListener : Controller
-view : Frontend
+RegisterPropertyGUI(view : Frontend)
+setRegistrationPropertyListener(registerPropertyListener : Controller) : void
+registrationForm() : void
+errorPage() : void
+successPage() : void

Frontend
-signup : SignupGUI
-login : LoginGUI
-registerProperty : RegisterPropertyGUI
-manageSubscriptions : ManageSubscriptionsGUI
-addSearchCriteriaForm : AddSearchCriteriaFormGUI
-manageLandlordProperties : ManageLandlordPropertiesGUI
-payFeeForm : PayFeeFormGUI
-manageManagersProperties : ManageManagersPropertiesGUI
-changeFees : ChangeFeesGUI
-viewCompanyDatabase : ViewCompanyDatabaseGUI
-periodicalReportForm : PeriodicalReportFormGUI
-dashboard : DashboardGUI
+Frontend()
+getSignupGUI() : SignupGUI
+getLoginGUI() : LoginGUI
+getRegisterPropertyGUI() : RegisterPropertyGUI
+getManageSubscriptionsGUI() : ManageSubscriptionsGUI
+getAddSearchCriteriaFormGUI() : AddSearchCriteriaFormGUI
+getManageLandlordPropertiesGUI() : ManageLandlordPropertiesGUI
+getPayFeeFormGUI() : PayFeeFormGUI
+getManageManagersPropertiesGUI() : ManageManagersPropertiesGUI
+getChangeFeesGUI() : ChangeFeesGUI
+getViewCompanyDatabaseGUI() : ViewCompanyDatabaseGUI
+getPeriodicalReportFormGUI() : PeriodicalReportFormGUI
+getDashboardGUI() : DashboardGUI

SignupGUI
-renterLabel : JLabel
-renterButton : JButton
-landlordLabel : JLabel
-landlordButton : JButton
-managerLabel : JLabel
-managerButton : JButton
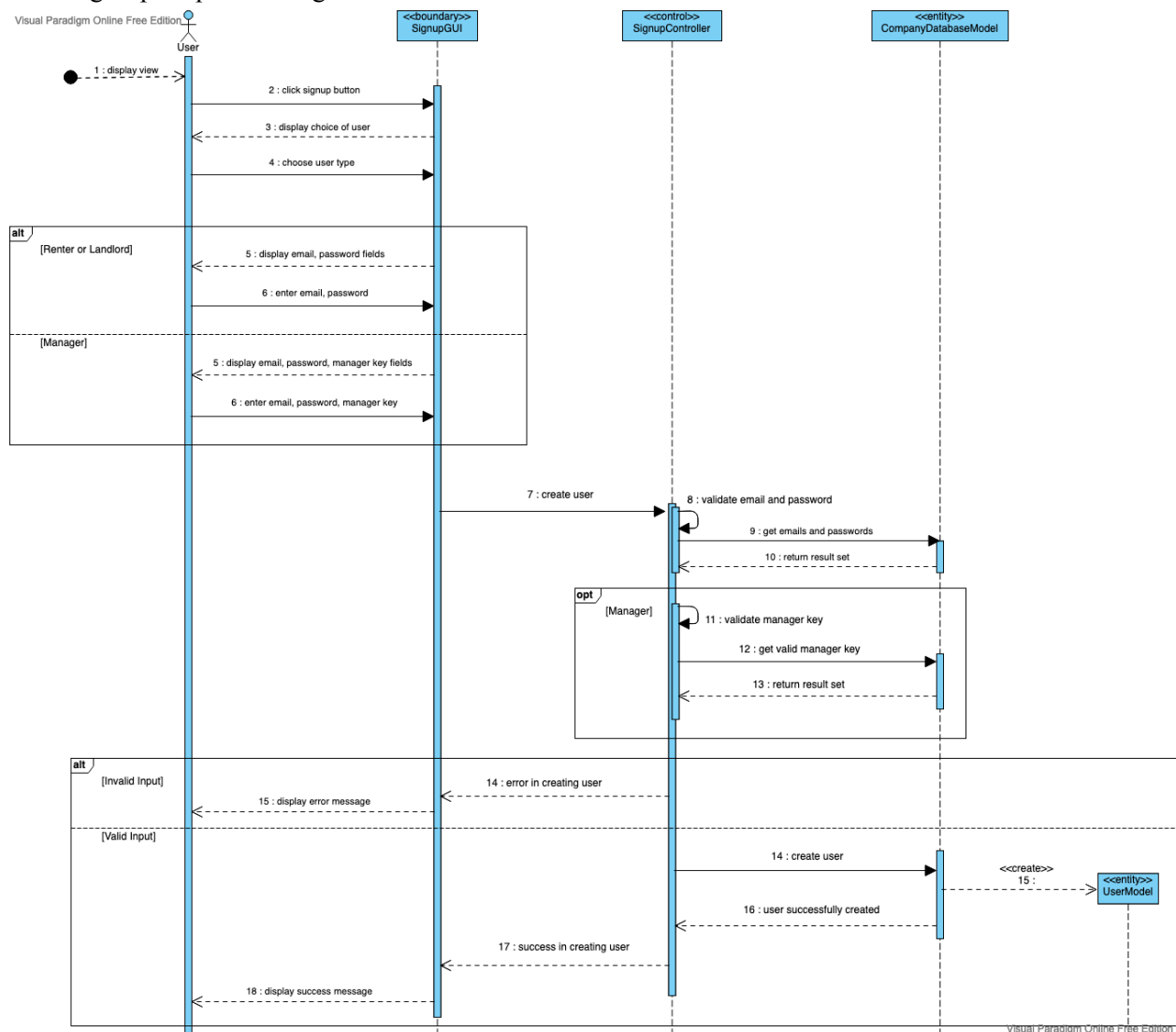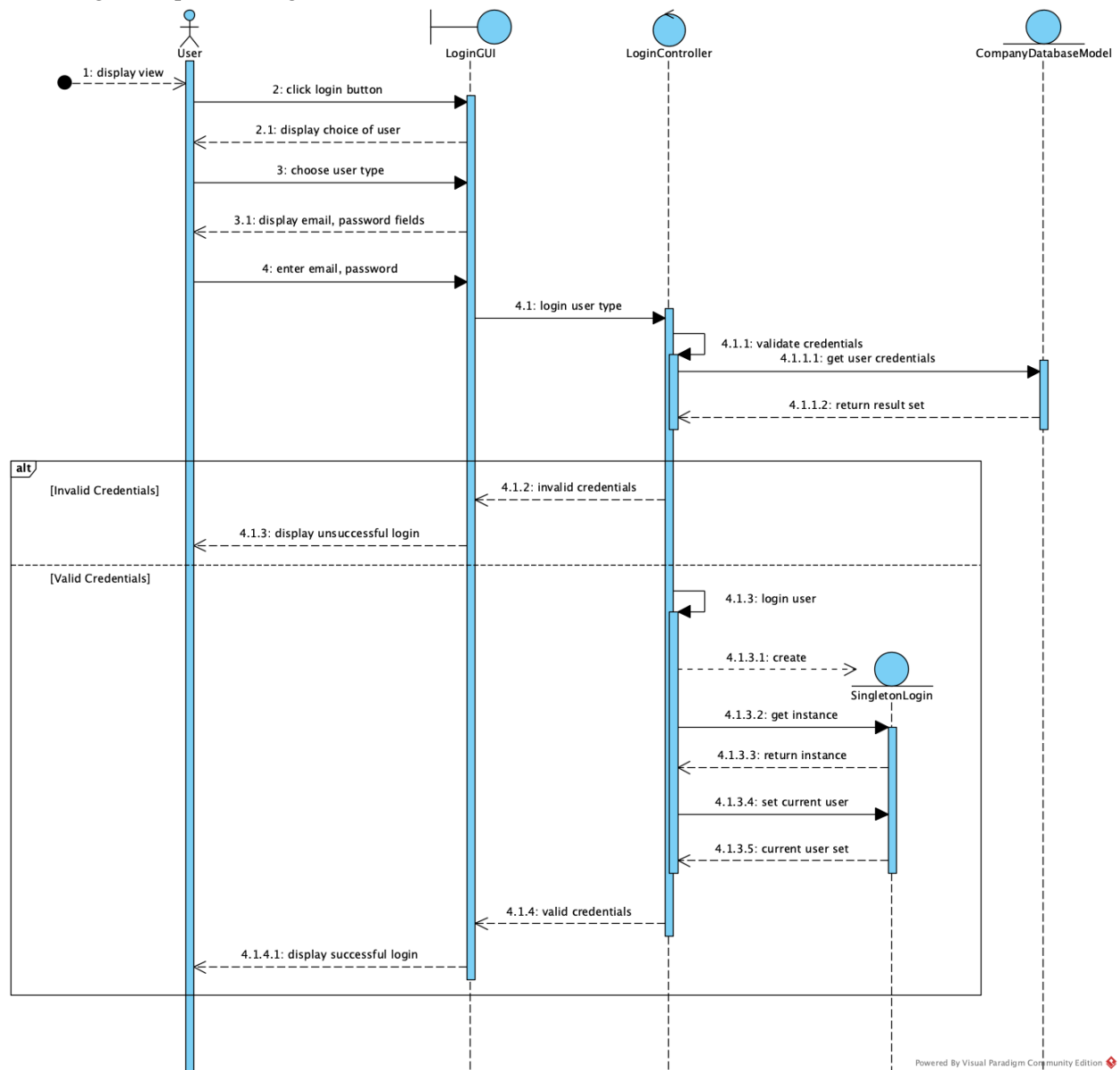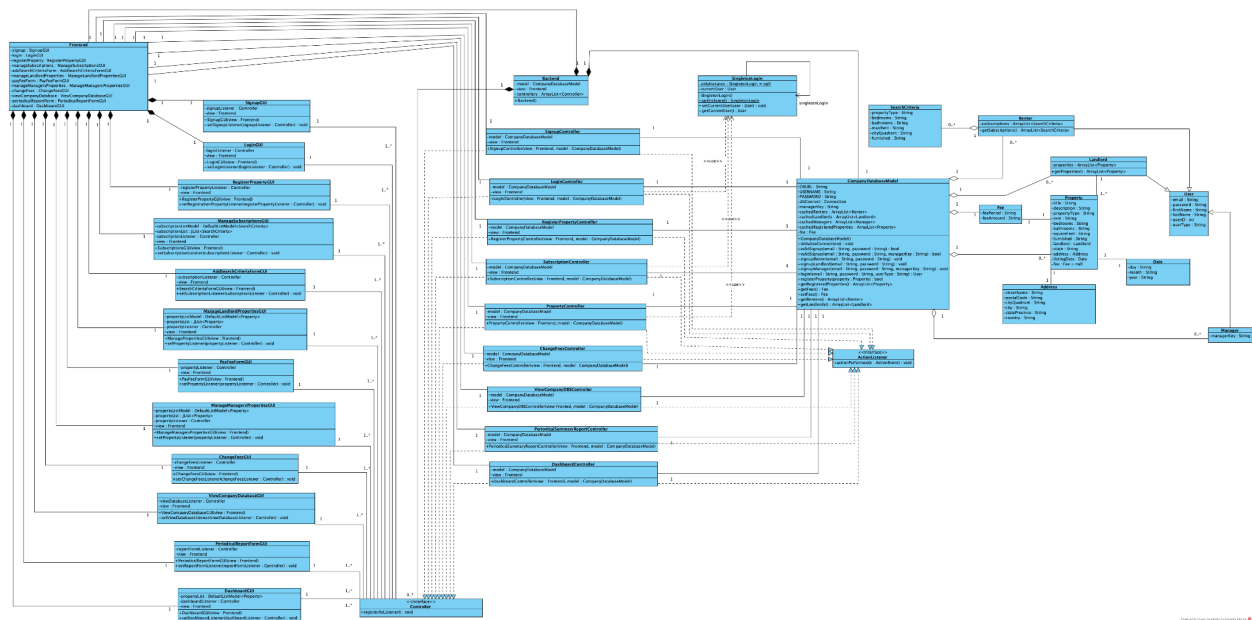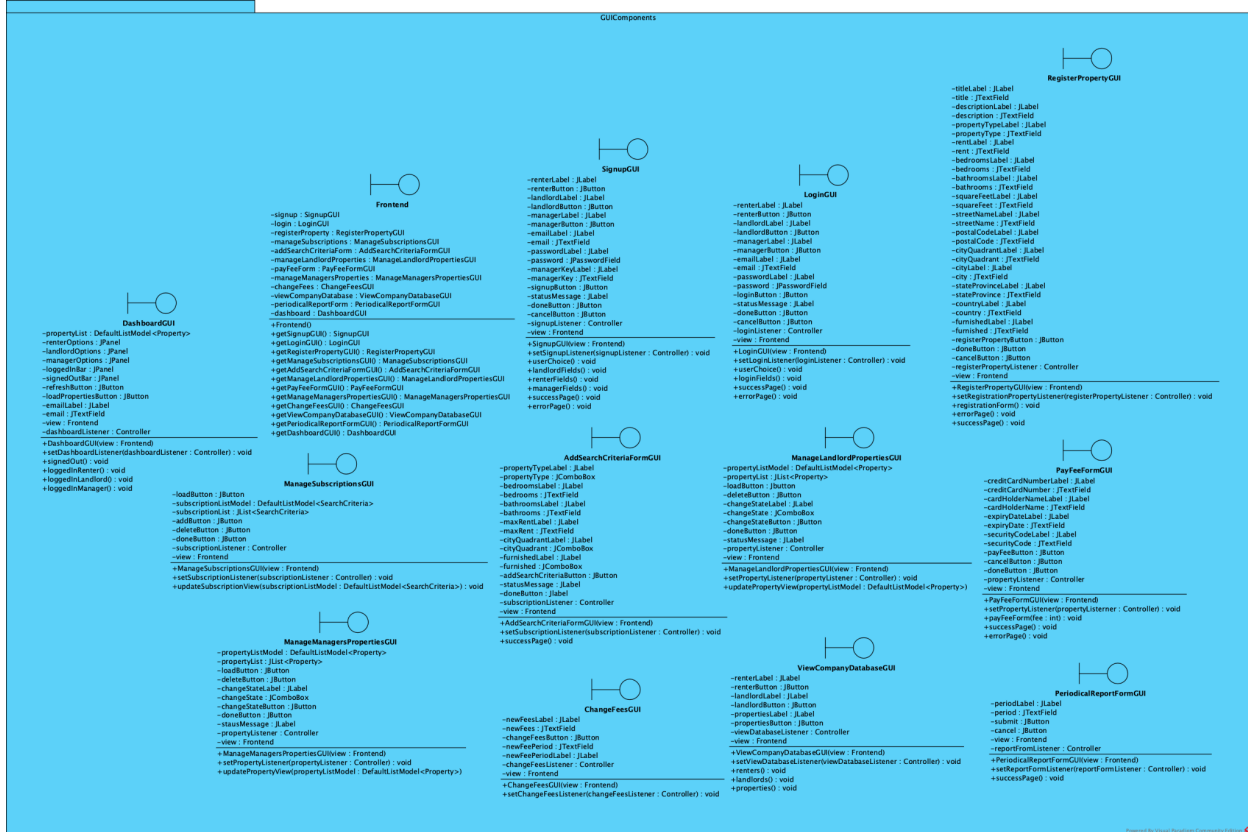-emailLabel : JLabel
-email : JTextField
-passwordLabel : JLabel
-password : JPasswordField
-managerKeyLabel : JLabel
-managerKey : JTextField
-signupButton : JButton
-statusMessage : JLabel
-doneButton : JButton
-cancelButton : JButton
-signupListener : Controller
-view : Frontend
+SignupGUI(view : Frontend)
+setSignupListener(signupListener : Controller) : void
+userChoice() : void
+renterFields() : void
+landlordFields() : void
+managerFields() : void
+successPage() : void
+errorPage() : void

LoginGUI
-renterLabel : JLabel
-renterButton : JButton
-landlordLabel : JLabel
-landlordButton : JButton
-managerLabel : JLabel
-managerButton : JButton
-emailLabel : JLabel
-email : JTextField
-passwordLabel : JLabel
-password : JPasswordField
-loginButton : JButton
-statusMessage : JLabel
-doneButton : JButton
-cancelButton : JButton
-loginListener : Controller
-view : Frontend
+LoginGUI(view : Frontend)
+setLoginListener(loginListener : Controller) : void
+userChoice() : void
+loginFields() : void
+successPage() : void
+errorPage() : void

DashboardGUI
-propertyList : DefaultListModel<Property>
-renterOptions : JPanel
-landlordOptions : JPanel
-managerOptions : JPanel
-loggedInBar : JPanel
-signedOutBar : JPanel
-refreshButton : JButton
-loadPropertiesButton : JButton
-emailLabel : JLabel
-email : JTextField
-view : Frontend
-dashboardListener : Controller
+DashboardGUI(view : Frontend)
+setDashboardListener(dashboardListener : Controller) : void
+signedOut() : void
+loggedInRenter() : void
+loggedInLandlord() : void
+loggedInManager() : void

ManageSubscriptionsGUI
-loadButton : JButton
-subscriptionListModel : DefaultListModel<SearchCriteria>
-subscriptionList : JList<SearchCriteria>
-addButton : JButton
-deleteButton : JButton
-doneButton : JButton
-subscriptionListener : Controller
-view : Frontend
+ManageSubscriptionsGUI(view : Frontend)
+setSubscriptionListener(subscriptionListener : Controller) : void
+updateSubscriptionView(subscriptionListModel : DefaultListModel<SearchCriteria>) : void

AddSearchCriteriaFormGUI
-propertyTypeLabel : JLabel
-propertyType : JComboBox
-bedroomsLabel : JLabel
-bedrooms : JTextField
-bathroomsLabel : JLabel
-bathrooms : JTextField
-maxRentLabel : JLabel
-maxRent : JTextField
-cityQuadrantLabel : JLabel
-cityQuadrant : JComboBox
-furnishedLabel : JLabel
-furnished : JComboBox
-addSearchCriteriaButton : JButton
-statusMessage : JLabel
-doneButton : JButton
-subscriptionListener : Controller
-view : Frontend
+AddSearchCriteriaFormGUI(view : Frontend)
+setSubscriptionListener(subscriptionListener : Controller) : void
+successPage() : void

ManageLandlordPropertiesGUI
-propertyListModel : DefaultListModel<Property>
-propertyList : JList<Property>
-loadButton : JButton
-deleteButton : JButton
-changeStateLabel : JLabel
-changeState : JComboBox
-changeStateButton : JButton
-doneButton : JButton
-statusMessage : JLabel
-propertyListener : Controller
-view : Frontend
+ManageLandlordPropertiesGUI(view : Frontend)
+setPropertyListener(propertyListener : Controller) : void
+updatePropertyView(propertyListModel : DefaultListModel<Property>)

PayFeeFormGUI
-creditCardNumberLabel : JLabel
-creditCardNumber : JTextField
-cardHolderNameLabel : JLabel
-cardHolderName : JTextField
-expiryDateLabel : JLabel
-expiryDate : JTextField
-securityCodeLabel : JLabel
-securityCode : JTextField
-payFeeButton : JButton
-cancelButton : JButton
-doneButton : JButton
-propertyListener : Controller
-view : Frontend
+PayFeeFormGUI(view : Frontend)
+setPropertyListener(propertyListener : Controller) : void
+payFeeForm(fee : int) : void
+successPage() : void
+errorPage() : void

ManageManagersPropertiesGUI
-propertyListModel : DefaultListModel<Property>
-propertyList : JList<Property>
-loadButton : JButton
-deleteButton : JButton
-changeStateLabel : JLabel
-changeState : JComboBox
-changeStateButton : JButton
-doneButton : JButton
-statusMessage : JLabel
-propertyListener : Controller
-view : Frontend
+ManageManagersPropertiesGUI(view : Frontend)
+setPropertyListener(propertyListener : Controller) : void
+updatePropertyView(propertyListModel : DefaultListModel<Property>)

ChangeFeesGUI
-newFeesLabel : JLabel
-newFees : JTextField
-changeFeesButton : JButton
-newFeePeriod : JTextField
-newFeePeriodLabel : JLabel
-changeFeesListener : Controller
-view : Frontend
+ChangeFeesGUI(view : Frontend)
+setChangeFeesListener(changeFeesListener : Controller) : void

ViewCompanyDatabaseGUI
-renterLabel : JLabel
-renterButton : JButton
-landlordLabel : JLabel
-landlordButton : JButton
-propertiesLabel : JLabel
-propertiesButton : JButton
-viewDatabaseListener : Controller
-view : Frontend
+ViewCompanyDatabaseGUI(view : Frontend)
+setViewDatabaseListener(viewDatabaseListener : Controller) : void
+renters() : void
+landlords() : void
+properties() : void

PeriodicalReportFormGUI
-periodLabel : JLabel
-period : JTextField
-submit : JButton
-cancel : JButton
-view : Frontend
-reportFormListener : Controller
+PeriodicalReportFormGUI(view : Frontend)
+setReportFormListener(reportFormListener : Controller) : void
+successPage() : void

Figure 1.5.1 GUIComponents Class Specification Diagram

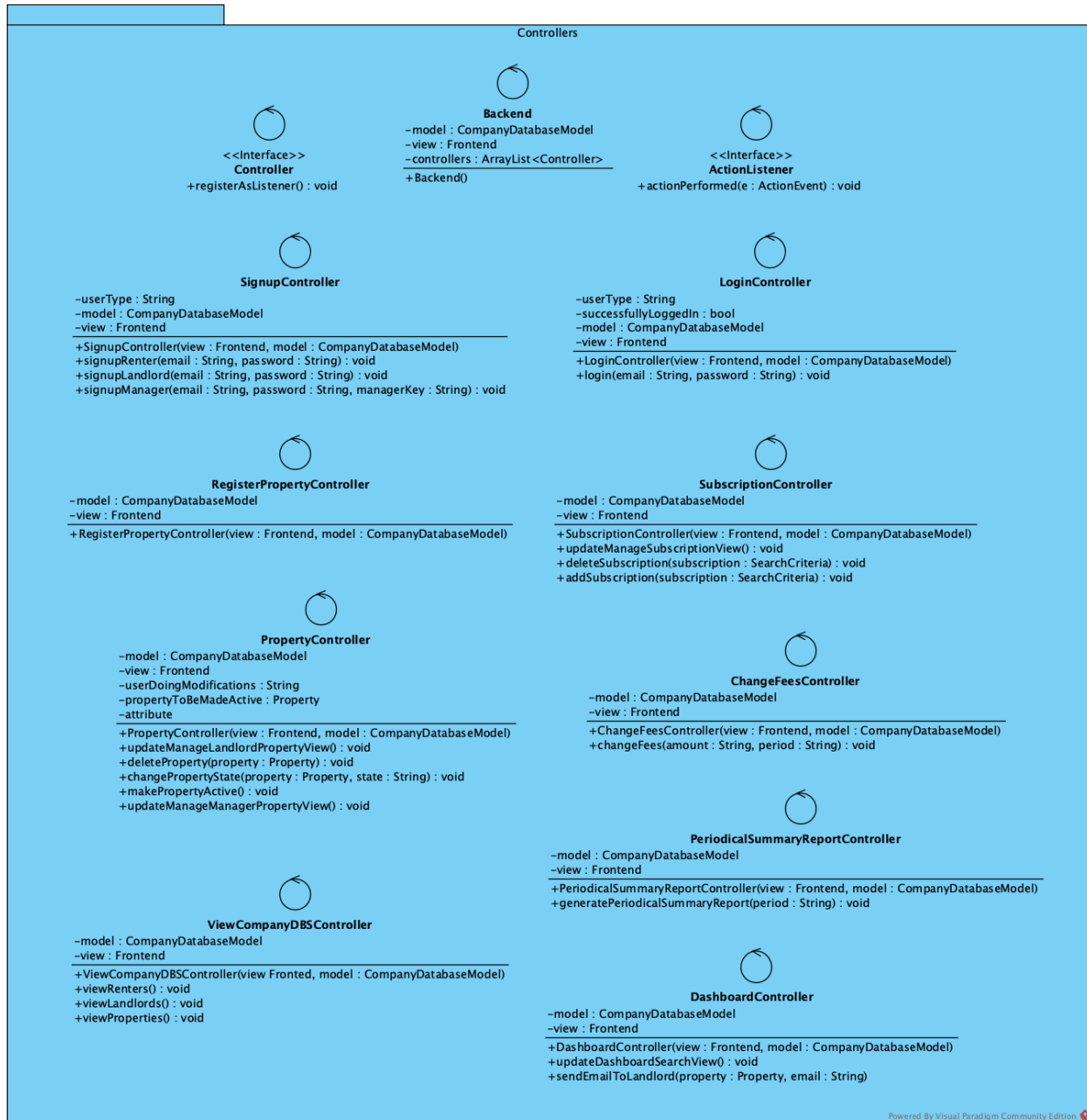## 1.5.2 Controllers Class Specification Diagram

Figure 1.5.2 Controllers Class Specification Diagram
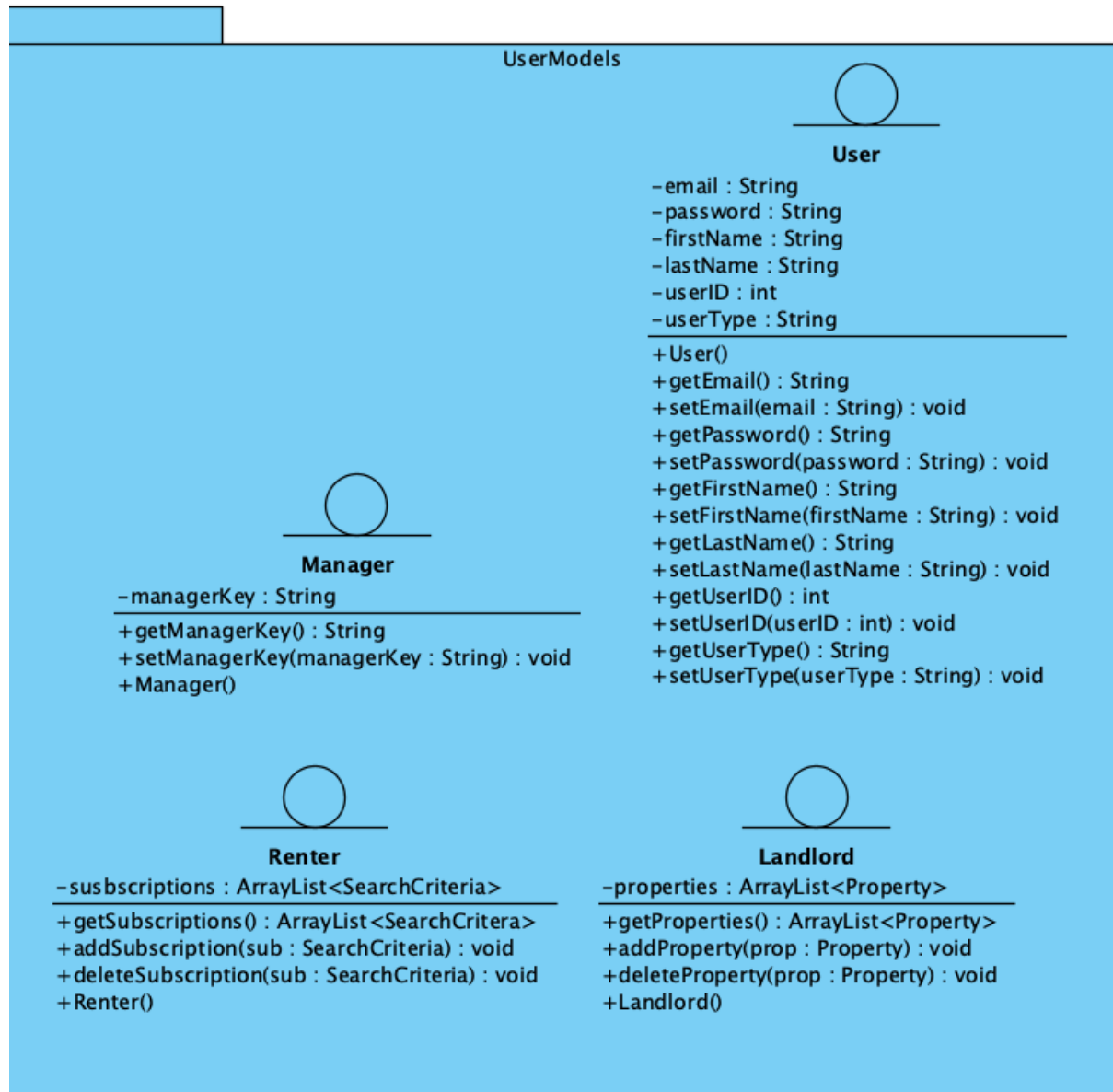
1.5.3 UserModels Class Specification Diagram

Figure 1.5.3 UserModels Class Specification Diagram

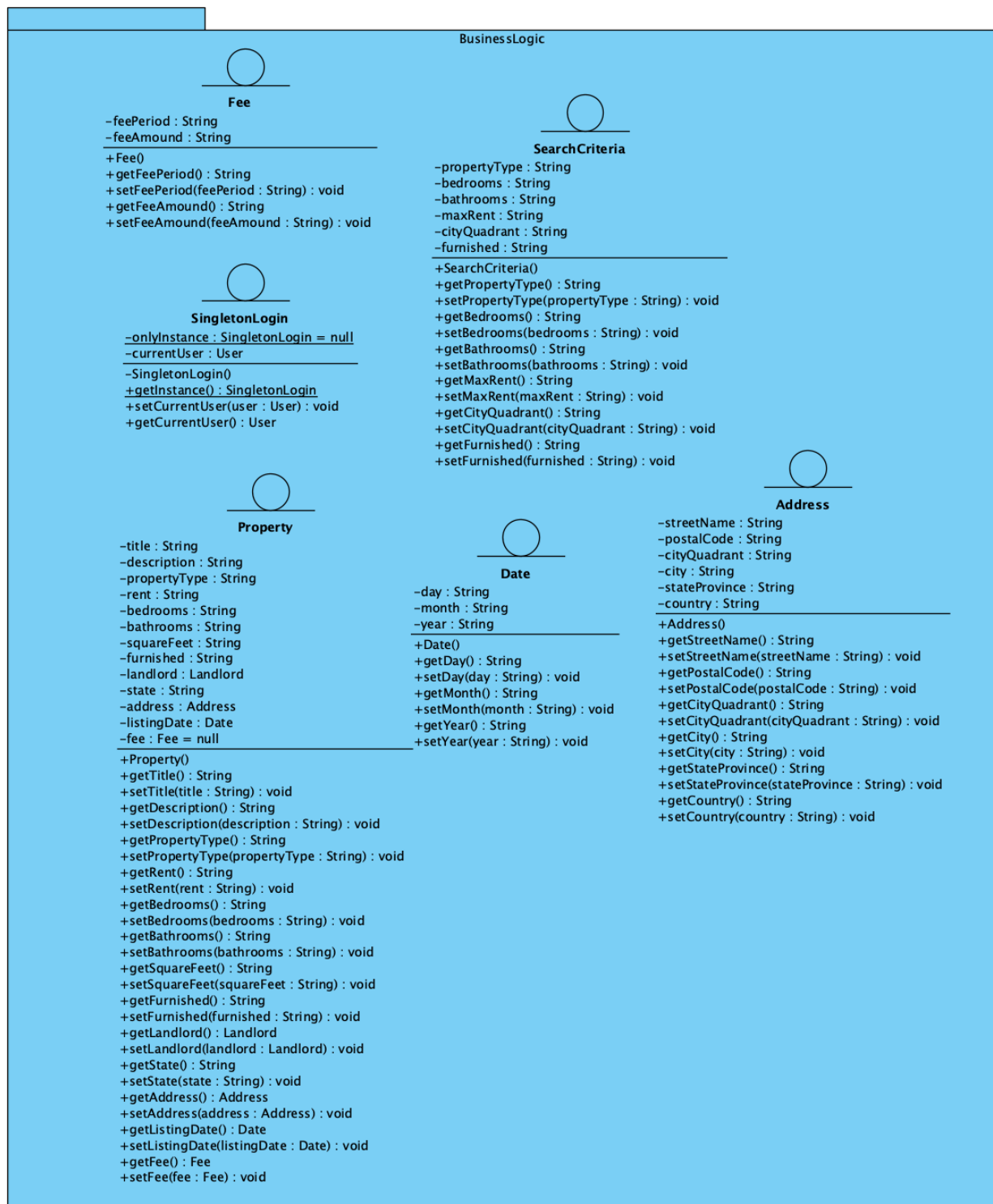1.5.4 BusinessLogic Class Specification Diagram

BusinessLogic

Fee
-feePeriod : String
-feeAmound : String

+Fee()
+getFeePeriod() : String
+setFeePeriod(feePeriod : String) : void
+getFeeAmound() : String
+setFeeAmound(feeAmound : String) : void

SingletonLogin
-onlyInstance : SingletonLogin = null
-currentUser : User

-SingletonLogin()
+getInstance() : SingletonLogin
+setCurrentUser(user : User) : void
+getCurrentUser() : User

SearchCriteria
-propertyType : String
-bedrooms : String
-bathrooms : String
-maxRent : String
-cityQuadrant : String
-furnished : String

+SearchCriteria()
+getPropertyType() : String
+setPropertyType(propertyType : String) : void
+getBedrooms() : String
+setBedrooms(bedrooms : String) : void
+getBathrooms() : String
+setBathrooms(bathrooms : String) : void
+getMaxRent() : String
+setMaxRent(maxRent : String) : void
+getCityQuadrant() : String
+setCityQuadrant(cityQuadrant : String) : void
+getFurnished() : String
+setFurnished(furnished : String) : void

Property
-title : String
-description : String
-propertyType : String
-rent : String
-bedrooms : String
-bathrooms : String
-squareFeet : String
-furnished : String
-landlord : Landlord
-state : String
-address : Address
-listingDate : Date
-fee : Fee = null

+Property()
+getTitle() : String
+setTitle(title : String) : void
+getDescription() : String
+setDescription(description : String) : void
+getPropertyType() : String
+setPropertyType(propertyType : String) : void
+getRent() : String
+setRent(rent : String) : void
+getBedrooms() : String
+setBedrooms(bedrooms : String) : void
+getBathrooms() : String
+setBathrooms(bathrooms : String) : void
+getSquareFeet() : String
+setSquareFeet(squareFeet : String) : void
+getFurnished() : String
+setFurnished(furnished : String) : void
+getLandlord() : Landlord
+setLandlord(landlord : Landlord) : void
+getState() : String
+setState(state : String) : void
+getAddress() : Address
+setAddress(address : Address) : void
+getListingDate() : Date
+setListingDate(listingDate : Date) : void
+getFee() : Fee
+setFee(fee : Fee) : void

Date
-day : String
-month : String
-year : String

+Date()
+getDay() : String
+setDay(day : String) : void
+getMonth() : String
+setMonth(month : String) : void
+getYear() : String
+setYear(year : String) : void

Address
-streetName : String
-postalCode : String
-cityQuadrant : String
-city : String
-stateProvince : String
-country : String

+Address()
+getStreetName() : String
+setStreetName(streetName : String) : void
+getPostalCode() : String
+setPostalCode(postalCode : String) : void
+getCityQuadrant() : String
+setCityQuadrant(cityQuadrant : String) : void
+getCity() : String
+setCity(city : String) : void
+getStateProvince() : String
+setStateProvince(stateProvince : String) : void
+getCountry() : String
+setCountry(country : String) : void

Figure 1.5.4 BusinessLogic Class Specification Diagram

1.5.5 CompanyDatabase Class Specification Diagram

CompanyDatabase

CompanyDatabaseModel

−DBURL : String
−USERNAME : String
−PASSWORD : String
−dbConnect : Connection
−managerKey : String
−cachedRenters : ArrayList<Renter>
−cachedLandlords : ArrayList<Landlord>
−cachedManagers : ArrayList<Manager>
−cachedRegisteredProperties : ArrayList<Property>
−fee : Fee

+CompanyDatabaseModel()
+initializeConnection() : void
+validSignup(email : String, password : String) : bool
+validSignup(email : String, password : String, managerKey : String) : bool
+signupRenter(email : String, password : String) : void
+signupLandlord(email : String, password : String) : void
+signupManager(email : String, password : String, managerKey : String) : void
+login(email : String, password : String, userType : String) : User
+registerProperty(property : Property) : bool
+getRegisteredProperties() : ArrayList<Property>
+getFees() : Fee
+setFees() : Fee
+getRenters() : ArrayList<Renter>
+getLandlords() : ArrayList<Landlord>

Figure 1.5.5 CompanyDatabase Class Specification Diagram

1.6 State Transition Diagram

1.6.1 Sign-up State Transition Diagram

Figure 1.6.1 Sign-up State Transition Diagram

1.6.2 Register Property State Transition Diagram

Figure 1.6.2 Register Property State Transition Diagram

1.6.3 Subscribe and Unsubscribe State Transition Diagram
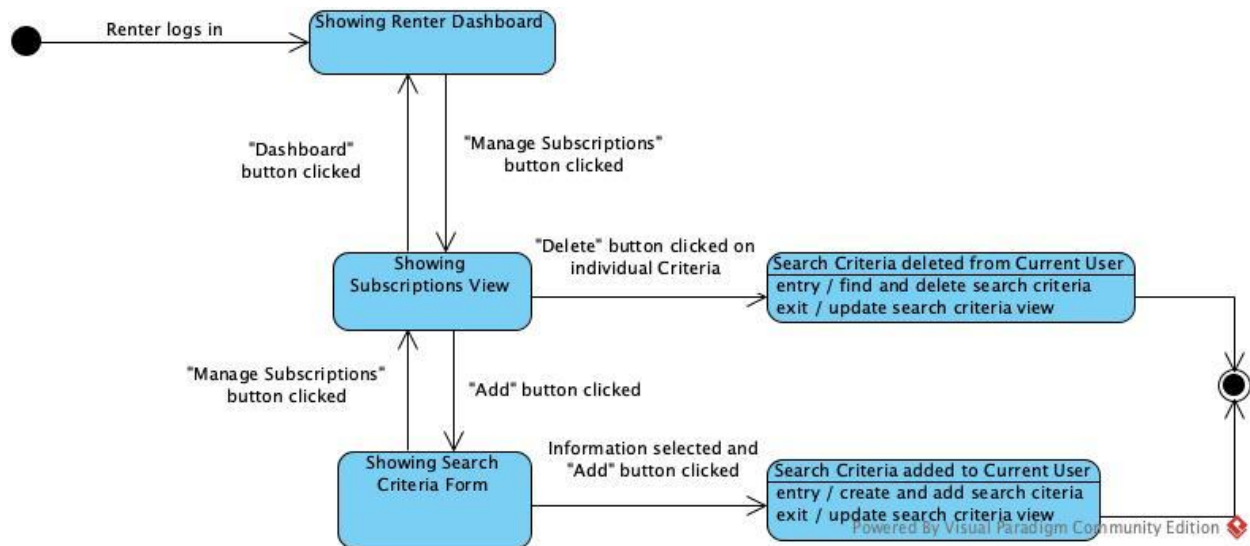
Figure 1.6.3 Subscribe and Unsubscribe State Transition Diagram
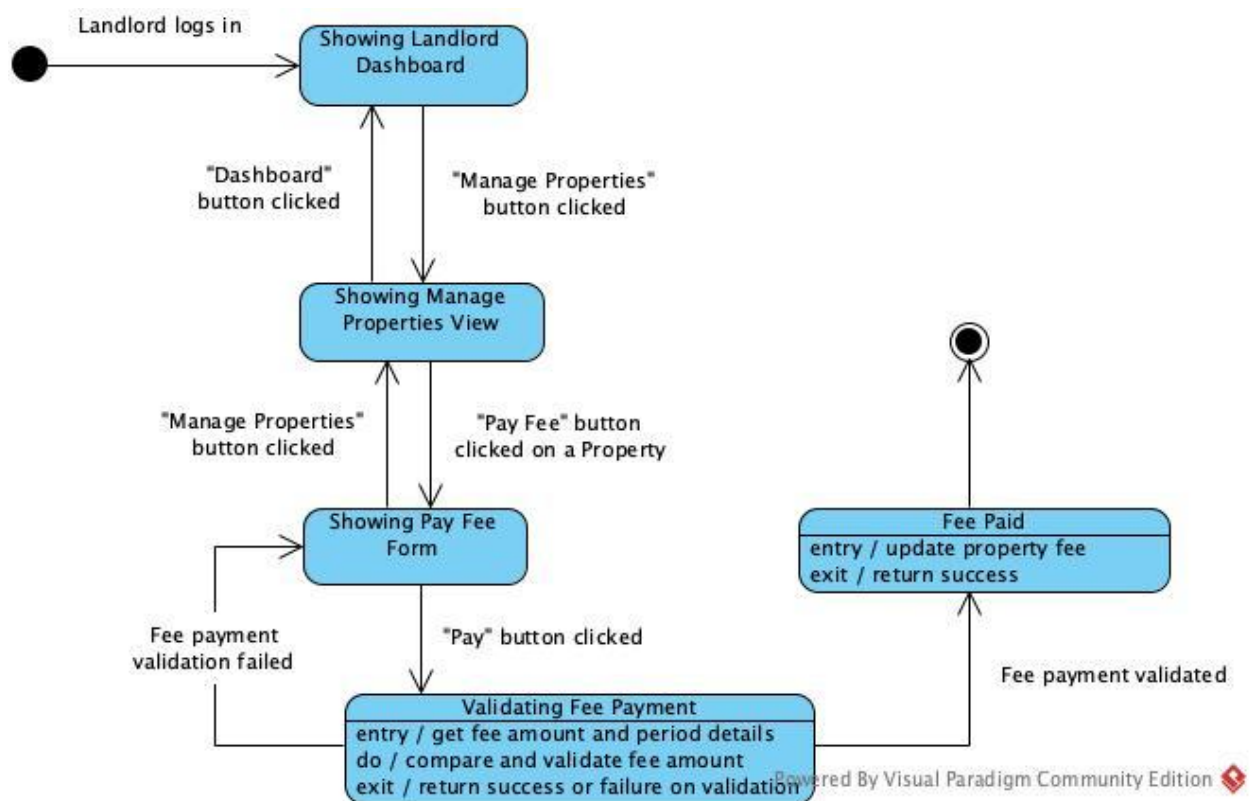
1.6.4 Pay Fee State Transition Diagram
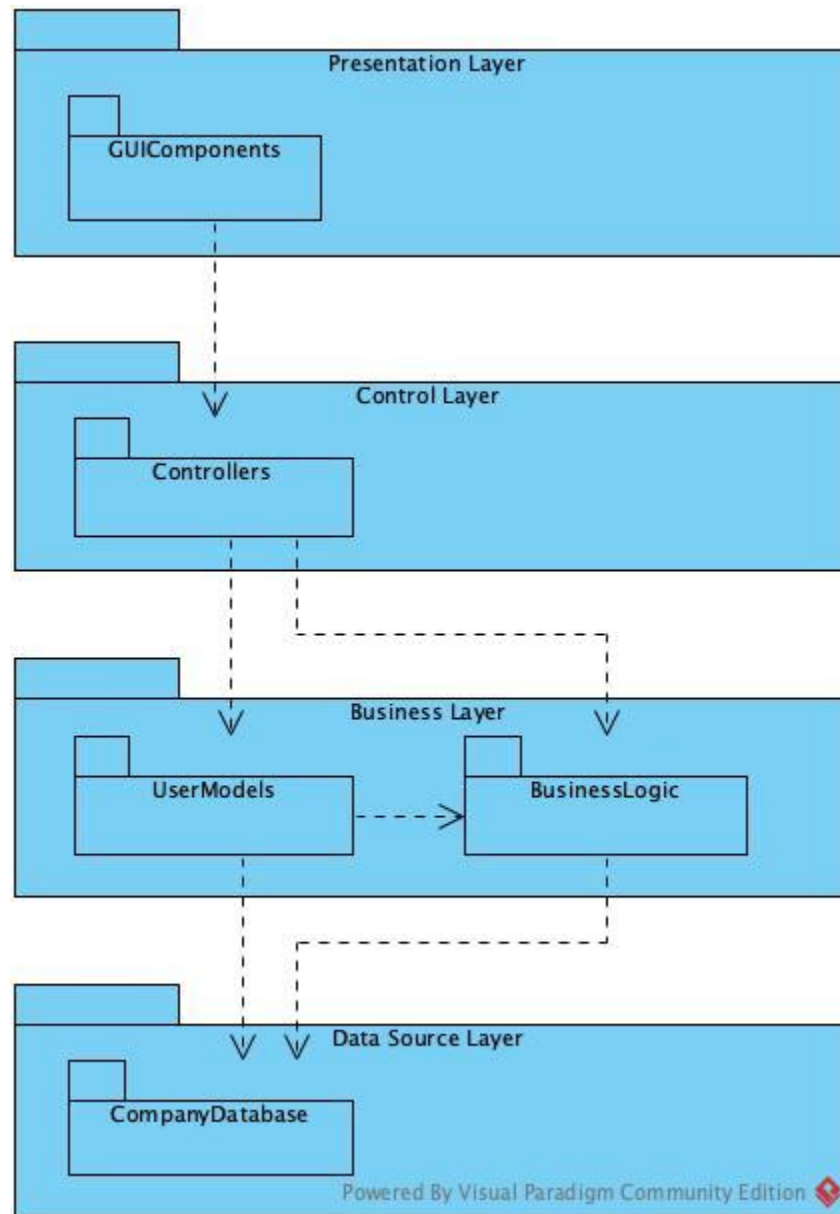

Figure 1.6.4 Pay Fee State Transition Diagram
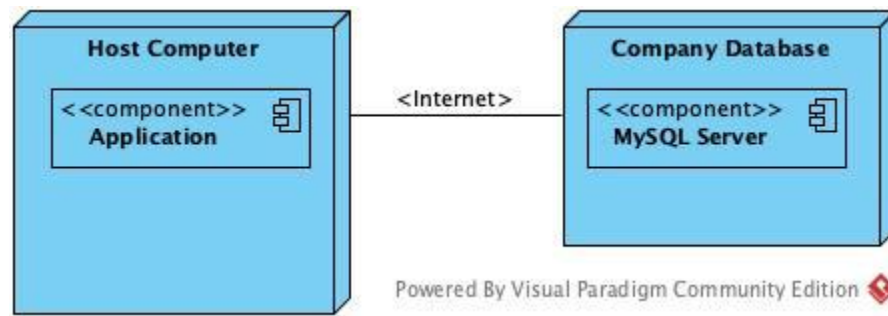
1.7 Package Diagram

Figure 1.7 Package Diagram

1.8 Deployment Diagram

Figure 1.8 Deployment Diagram