

SOEN 390 - Software Engineering Team Design Project
Team 17

Architecture Description
For
Condo Management System

Department of Computer Science and Software Engineering
Gina Cody School of Engineering and Computer Science
Concordia University

Contents

Contents	1
1 Introduction	3
1.1 Scope	3
2 Stakeholders and concerns	3
2.1 Stakeholders	3
2.2 Concerns	4
2.3 Concern–Stakeholder Traceability	4
3 Viewpoints and Views	6
3.1 Domain Model	6
3.2 Component Model	7
3.3 Class Diagram	8
3.4 Deployment Diagram	9
3.5 Use Case Diagram	10
3.6 Activity Diagram (AD)	11
3.6.1 AD1 - Sign In/Sign Up	11
3.6.2 AD2 - Registering as Owner/Renter	12
3.6.3 AD3 - Reserving a Facility	13
3.6.4 AD4 - Submitting a Request	15
3.6.5 AD5 - Access Financial Report	15
Bibliography	17

1 Introduction

Many condos have property managers who manage the community's daily operations manually. They are responsible for managing finances, responding to customer complaints, managing documents and records, liaising between residents and board, etc. Some of these tasks are still done on paper and may not have a proper storage for documenting records. Thus, the need arises for a Condo Management System to automate these tasks and a proper storage of records making it simpler for both the condo owners/renters and managers to communicate and view their required information online.

In this report, our team has designed a software architecture for a mobile app/website for the condos in Montreal, Quebec, Canada. The main goal of this platform is to simplify daily community operations by automating processes such as payments, storing data, providing analytics and insights.

1.1 Scope

Considering the available resources, duration of the course and available human resources of our team, we have decided to constrain the scope of the project to all the condos in Montreal. As we progress further, In the future we may consider expanding to the whole of Canada.

2 Stakeholders and concerns

This section identifies the stakeholders, the individuals who will be accessing the system, and the concerns related to the system. The last subsection will be mapping the stakeholders to the concerns they have.

2.1 Stakeholders

The stakeholders we have identified are listed below with a brief description about them,

1. **Users**
 - **Condo Owners:** An individual who owns a condo in the system
 - **Condo Renters:** An individual who rents a condo in the system
 - **Condo Management Company:** An organization that has property's under its management
 - **Admin:** An individual responsible for responding to user requests
2. **Software Provider Organization**
 - **Project Manager:** An individual who is responsible for the planning and execution of the project
 - **Developer:** A group of individuals who develops software
 - **Tester:** An individual who tests the software built
3. **Property Developers:** Companies or individuals involved in the construction and sale of new condo developments.
4. **Investors:** An individual or entity who provides funds to the Project.
5. **Regulatory Bodies:** Government or industry organizations that set standards and regulations for property management and tenant rights.

2.2 Concerns

Some of the concerns considered fundamental to the architecture of Condo Management System have been listed below:

- **Concern 1:** Is all the information entered during registration validated?
- **Concern 2:** Is the system protected from unauthorized access?
- **Concern 3:** Is the system easy to operate and navigate?
- **Concern 4:** Is the correct information made visible to the correct user?
- **Concern 5:** Is it easy to get financial reports from the system?
- **Concern 6:** Is the analytics provided by the financial system accurate?
- **Concern 7:** Is it easy to navigate to and reserve a facility?
- **Concern 8:** Does the reservation system allow a single facility to be reserved at a point in time?
- **Concern 9:** Is the system quick and responsive?
- **Concern 10:** Will the software be delivered on time and within the budget?
- **Concern 11:** Is there enough skilled resources to develop the system?
- **Concern 12:** Which methodology is going to be used for the development? (Agile, waterfall, etc.)
- **Concern 13:** Is the system going to be easily modified?
- **Concern 14:** What is the environment for our system? (Hardware, Operating System)
- **Concern 15:** Are the features of the system able to interact with each other?
- **Concern 16:** Is the code easy to test?
- **Concern 17:** Are all the available resources being properly utilized?
- **Concern 18:** Cost of the System?
- **Concern 19:** Are the rules and regulation set being followed?
- **Concern 20:** Is it easy to verify information of the users?
- **Concern 21:** Is it easy to post requests on the system?

2.3 Concern–Stakeholder Traceability

The below table represents the association of each concern with the identified stakeholder having that concern:

Table 2.1: Example showing association of stakeholders to concerns in an AD

	Condo Owne r	Condo Rente r	Condo Mgmt. Co.	Admin	Project Manage r	Dev.	Teste r	Property Developer	Investor	Regulatory Bodies
Concern 1	X	X	X	X	-	-	-	-	-	-
Concern 2	X	X	X	X	-	-	-	-	X	-
Concern 3	X	X	X	X	-	-	-	-	-	-
Concern 4	X	X	X	X	-	-	X	-	-	-
Concern 5	X	-	X	-	-	-	-	-	-	-
Concern 6	X	-	X	-	-	-	X	-	-	-
Concern 7	X	X	-	-	-	-	-	-	-	-
Concern 8	-	-	X	-	-	-	X	-	-	-
Concern 9	X	X	X	X	-	-	-	-	-	-
Concern 10	-	-	-	-	X	-	-	-	X	-
Concern 11	-	-	-	-	X	-	-	-	-	-
Concern 12	-	-	-	-	X	-	-	-	-	-
Concern 13	-	-	-	-	-	X	-	-	-	-
Concern 14	-	-	-	-	-	X	-	-	-	-
Concern 15	-	-	-	-	-	X	-	-	-	-
Concern 16	-	-	-	-	-	-	X	-	-	-
Concern 17	-	-	-	-	-	-	-	-	X	-
Concern 18	-	-	-	-	X	-	-	-	X	-
Concern 19	-	-	-	-	-	-	-	-	-	X
Concern 20	-	-	-	-	-	-	-	-	-	X
Concern 21	X	X	-	-	-	-	-	-	-	-

3 Viewpoints and Views

We have used the Unified Modelling Language for describing our architecture design of the Condo Management System. The UML model helps system designers and developers visualize the structure of their system at a high-level and ensure that the application meets the user's needs.

3.1 Domain Model

This model is a real-world visualization of the system and it is derived from the project requirements. It is a visual representation of interconnected concepts of real-world objects that incorporates key concepts, behavior and relationships of all of its entities.

Based on the requirements, we found out the following users for our system: owners, renters, companies, employees. Each of them have been represented in the diagram below with their respective collections as we will store records of them. We also identified the need to have facilities and properties as components, as the companies can upload properties/facilities and owners/renters can buy/rent and reserve facilities. Thus in the diagram we have Facility and Property components with their collections.

Since we know from requirements that a owner/renter can buy/rent properties we showed this relationship with a directional arrow with the appropriate text above it. Similarly for facilities, the relationship has been shown with owner/renter using an arr directional arrow with appropriate text. The relationship between companies and employees is shown with the arrow named "sets" in the diagram. The relation between company and property is also shown as companies can add properties that are available to buy/rent.

From the requirements we know that the users have the ability to send requests to employees set by the company to respond to their request. This concept is shown between owner/renters and employees with the arrow labeled "messages" which is connected to the Messenger component. As we will also have notifications for users, the Notification component uses messenger and is connected to the respective users.

The system also has a financial system. This system includes a parent type financial calculator component that calculates the finances for users. This component depends on individual tax, condo fees and any extra expense calculator to calculate the total. This component is used by owner/renters and requires condo companies. This relation can also be seen in the diagram below.

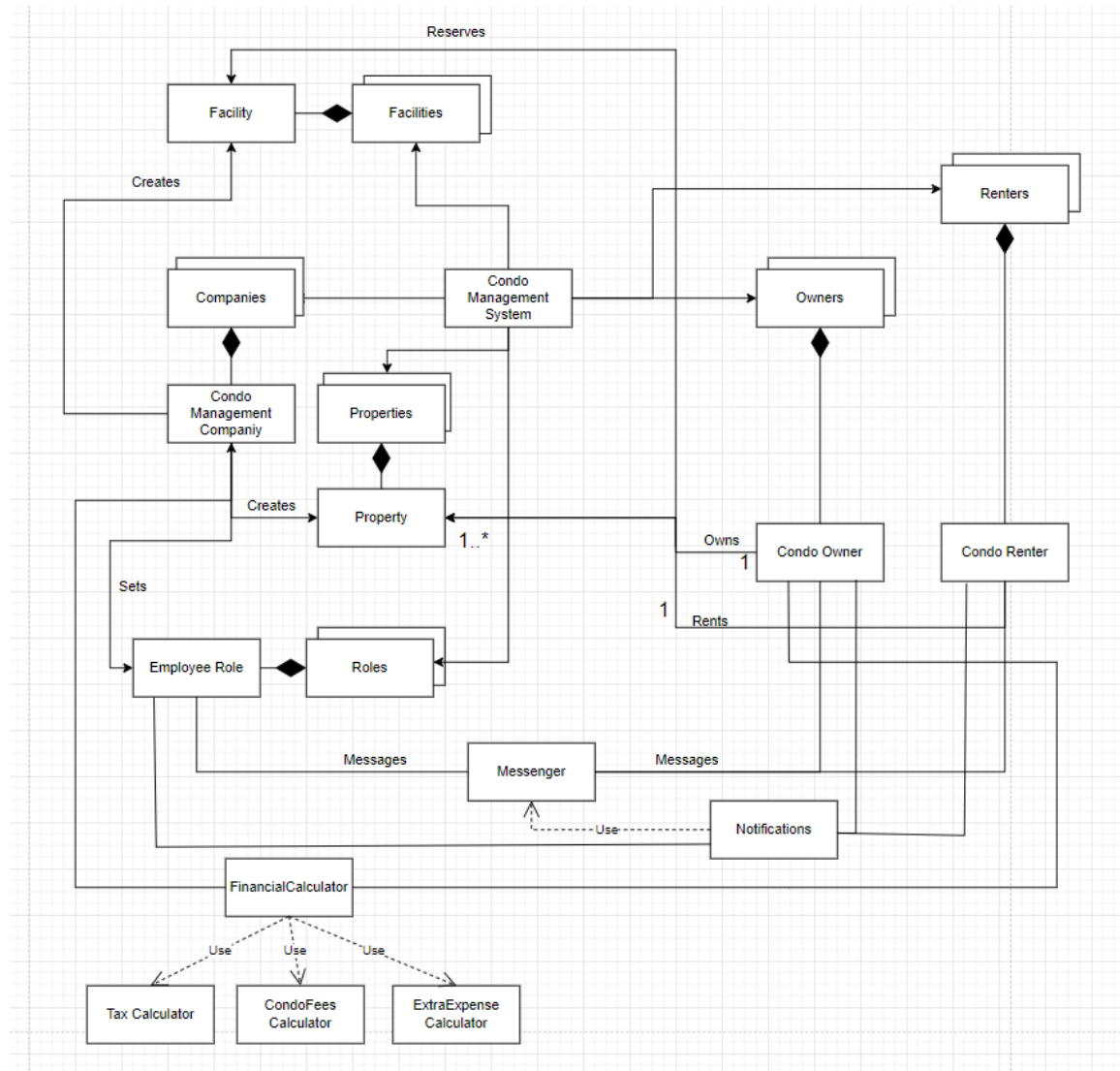


Figure 1: Domain Model of Condo Management System

3.2 Component Model

A component model, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. It shows the structure of the software system, which describes the software components, their interfaces, and their dependencies.

The below component diagram for the Condo Management System is divided into four subsystems namely, the webInterface(UI), the messaging system, reservation system, financial system and the database.

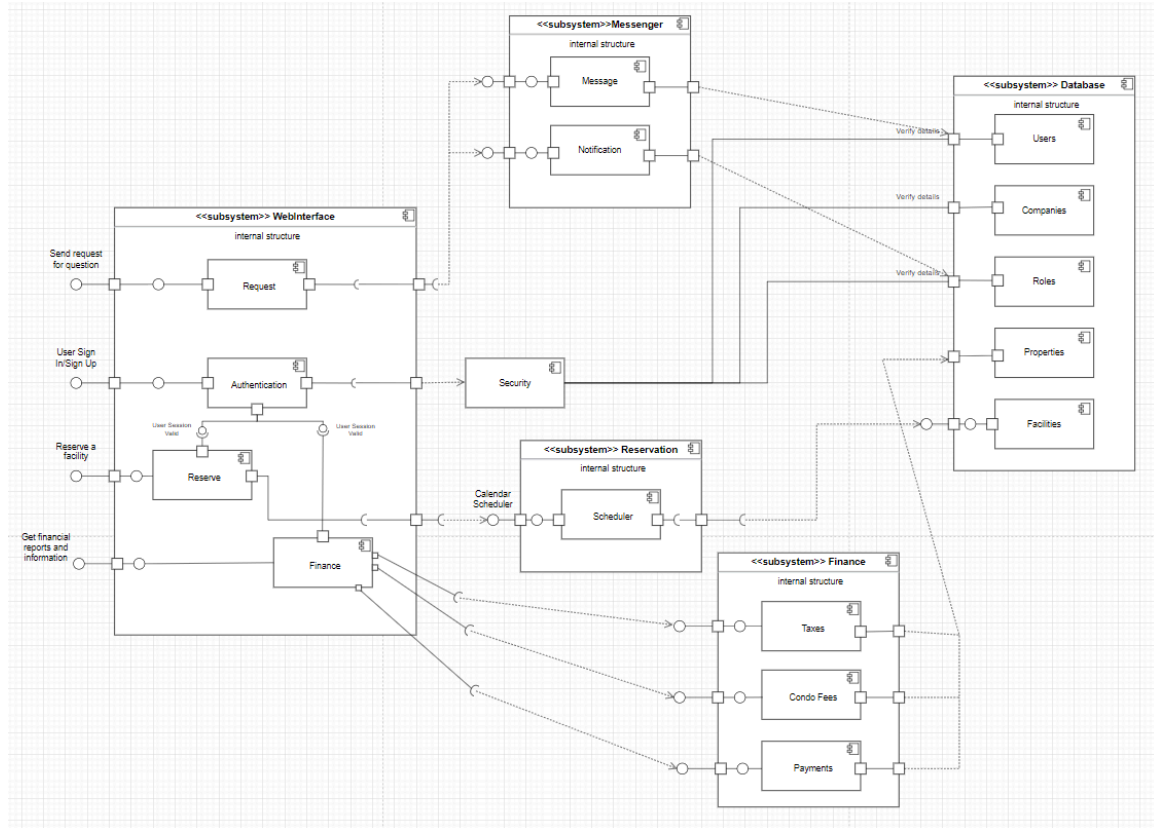


Figure 2: Component Model of Condo Management System

3.3 Class Diagram

Based on the domain model that was designed for our system, we have developed the following class diagram. It has all the classes identified in the domain diagram and the systems involved have been separated into classes to provide more details into its functioning. Using the GRASP principles we have identified the following patterns in our class diagram,

- The CondoManagementController class is the Controller Pattern. Since all the functionalities will be handled by this class. It also has the Information Expert pattern as it has access/copies of all users, companies, properties, facilities, etc.
- CondoManagementController class also has the Indirection pattern as it acts as a intermediary between UI and classes of the system.
- The FinanceCalculator class and Message class support the High Cohesion Pattern. We can have information regarding the messages received/sent and financial info of properties inside the CondoOwner/CondoRenter class, but then these classes will be focusing on

different functions and will lead to low cohesion. Thus, these classes were created for them to focus on separate tasks.

- Owners, Renters, Companies, Facilities, Properties, Roles have the Creator pattern with regards to CondoOwner, CondoRenter, CondoManagementCompany, Facility, Property, Role classes respectively.

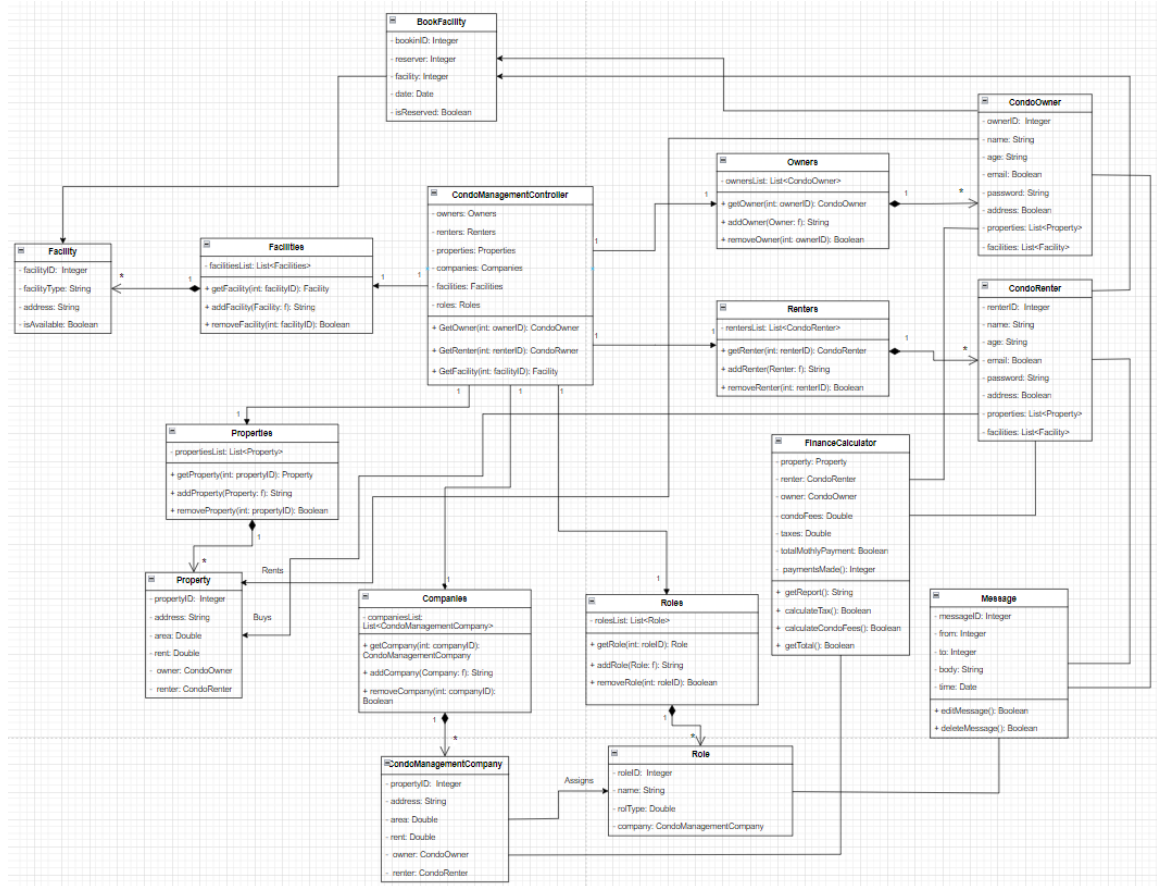


Figure 3: Class Model of Condo Management System

3.4 Deployment Diagram

The diagram below shows the deployment diagram for our system. We identified three nodes for our system. The first node is for our user i.e their devices on which they will access our application/system using a web browser/app which is shown as a sub-node.

The second node represents the web server for our application. Since for our backend we used node.js and express.js, these are shown as an executable environment node in our web server node. This node includes the logic for the different functions that our system will be performing like the reservation, financial, etc. logic. The user's browser will be able to connect to this server using the HTTP protocol.

The third node represents our database server. The database we used for this system is Firebase, a real-time database. We used the cylinder shape which is used in UML to represent the database. The

individual tables are also represented within the database. The web server connects to the database with the help of an API provided. This connection is made by importing the sdk into the application using express.js.

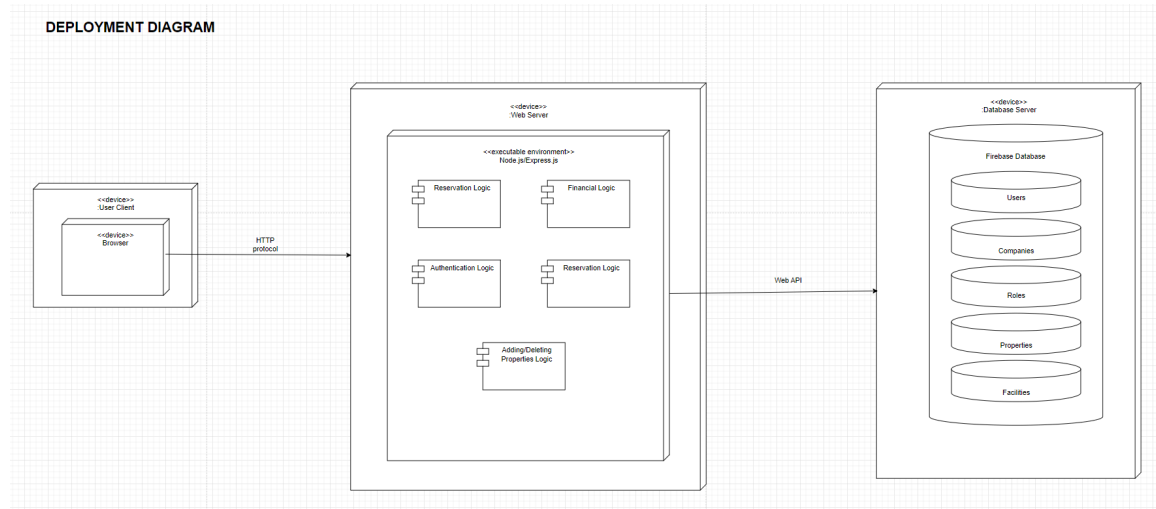


Figure 4: Deployment Diagram for Condo Management System

3.5 Use Case Diagram

In UML, Use Case diagrams are used to represent high-level functions and scope of a system. This diagram also identifies the interaction between the system and its actors. The use cases and the actors involved in the use cases describe what the system does and how the actors use it, without telling how the system works internally.

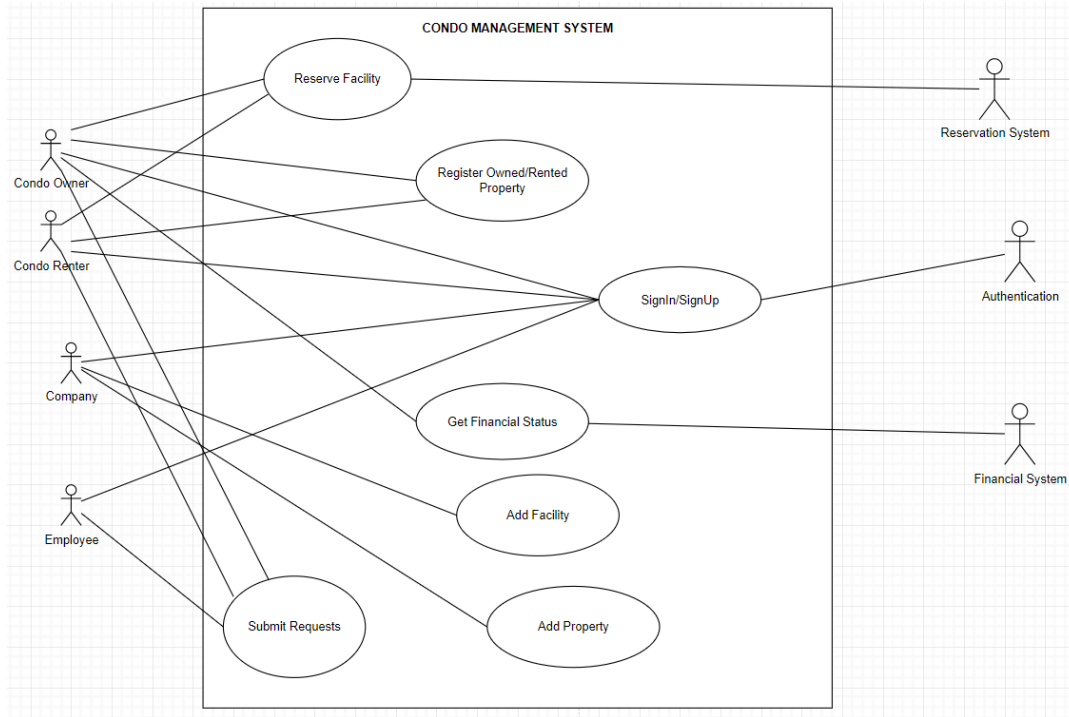


Figure 5: Use Case Diagram of Condo Management System

3.6 Activity Diagram (AD)

Activity Diagrams are a variation of the state diagram that show the workflow from a start point to the finish point detailing the many decision paths that exist in the progression of events contained in the activity. Below are some of the activity/function/use-case involved in the system.

3.6.1 AD1 - Sign In/Sign Up

This visualizes the activity of a user logging in or registering for the first time. The user starts at the login page. If the user is already registered, they enter their information to be validated and logged in. If they are a new user, they are taken to the registration page where they select their user category and enter the information to be registered.

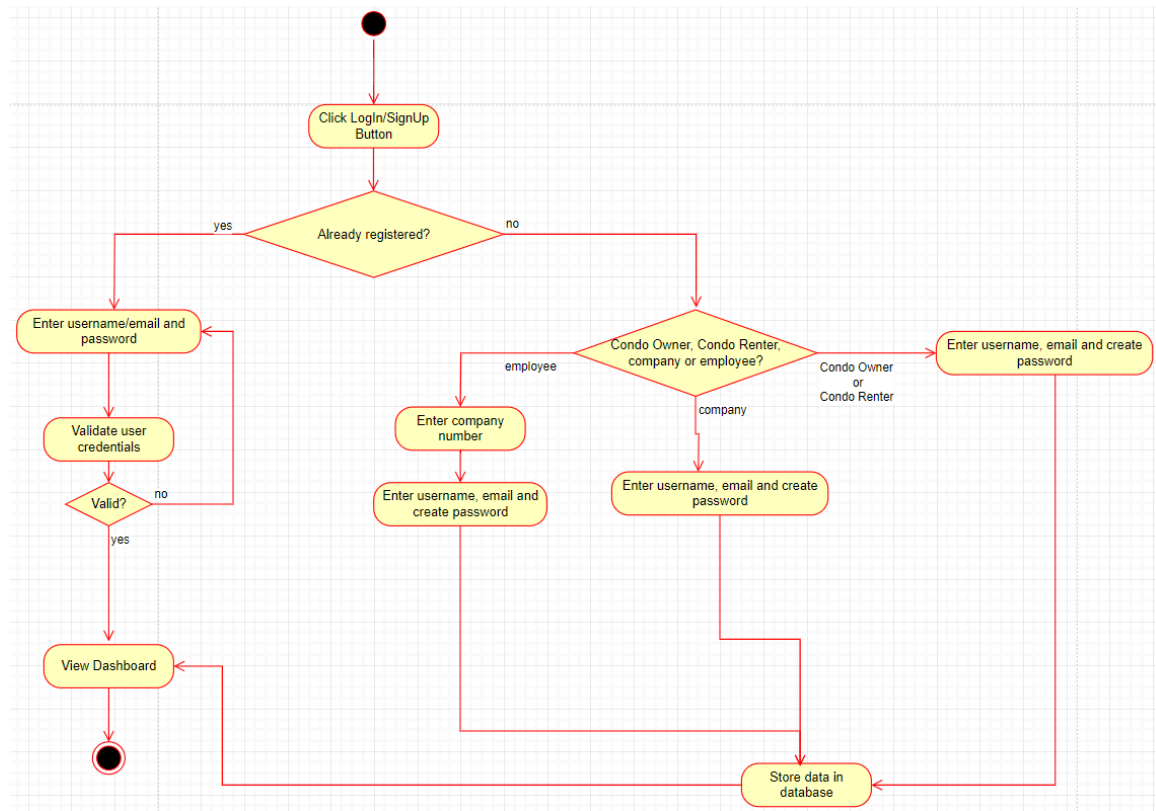


Figure 6: Activity Diagram for Sign In/Sign up

3.6.2 AD2 - Registering as Owner/Renter

This visualizes the activity of a user registering a property that they own or rent. From the profile dashboard the user clicks on the “Register Owner/Renter” button. On clicking, the user is displayed a field where they can enter the registration key they received from the condo management company. If they do not have a key, the user can request one from the company.

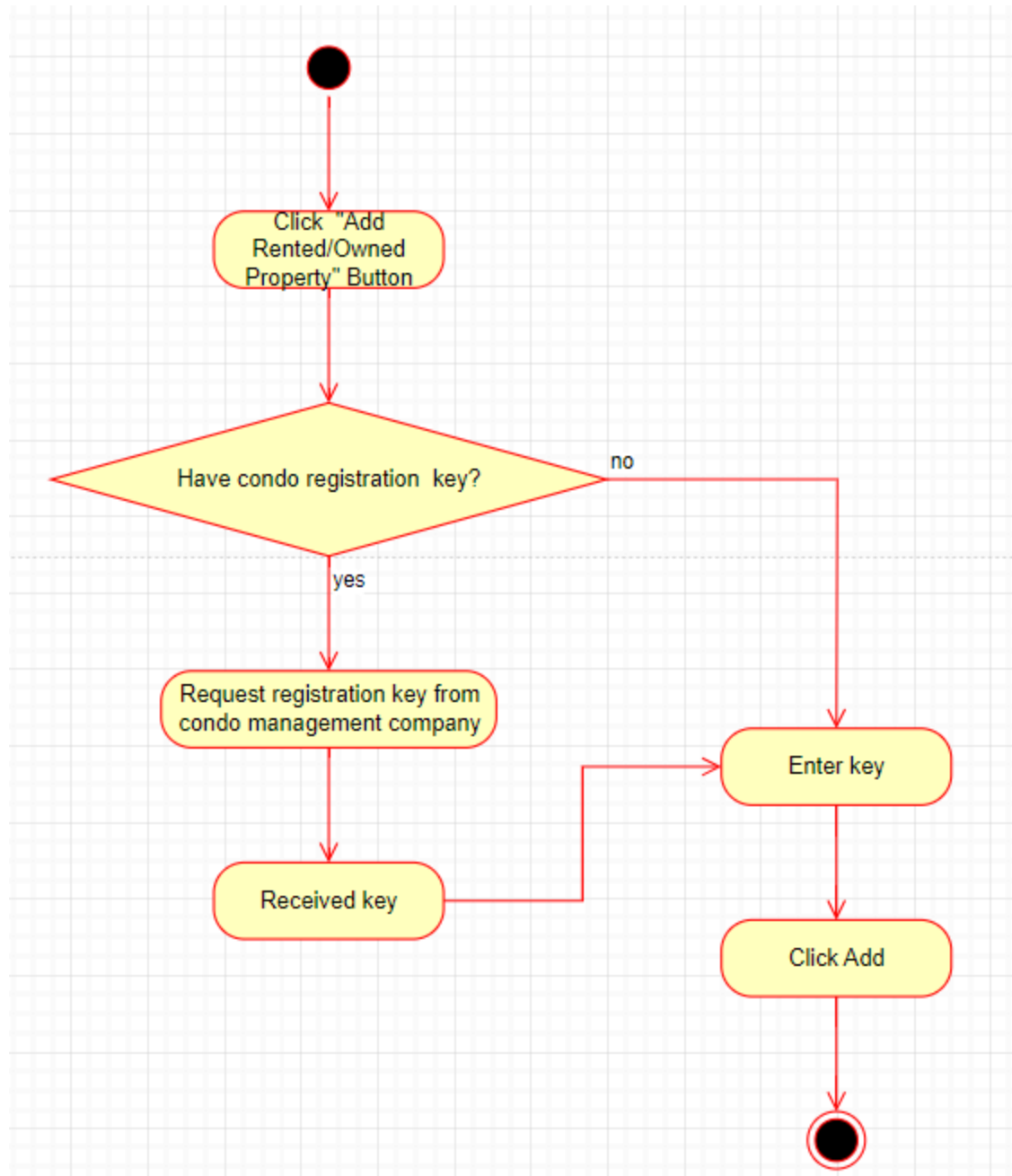


Figure 7: Activity Diagram for Registering as Owner/Renter

3.6.3 AD3 - Reserving a Facility

This diagram visualizes the activity of reserving a facility in the system. From the dashboard the user clicks on the "Reserve Facility" button. The user is taken to the reserve facility page where the user is asked to select the facility type and select from and to date from the calendar to reserve and then click the submit button. If the facility is available it is reserved, else an error is shown that the facility is reserved and the user is asked to repeat the process.

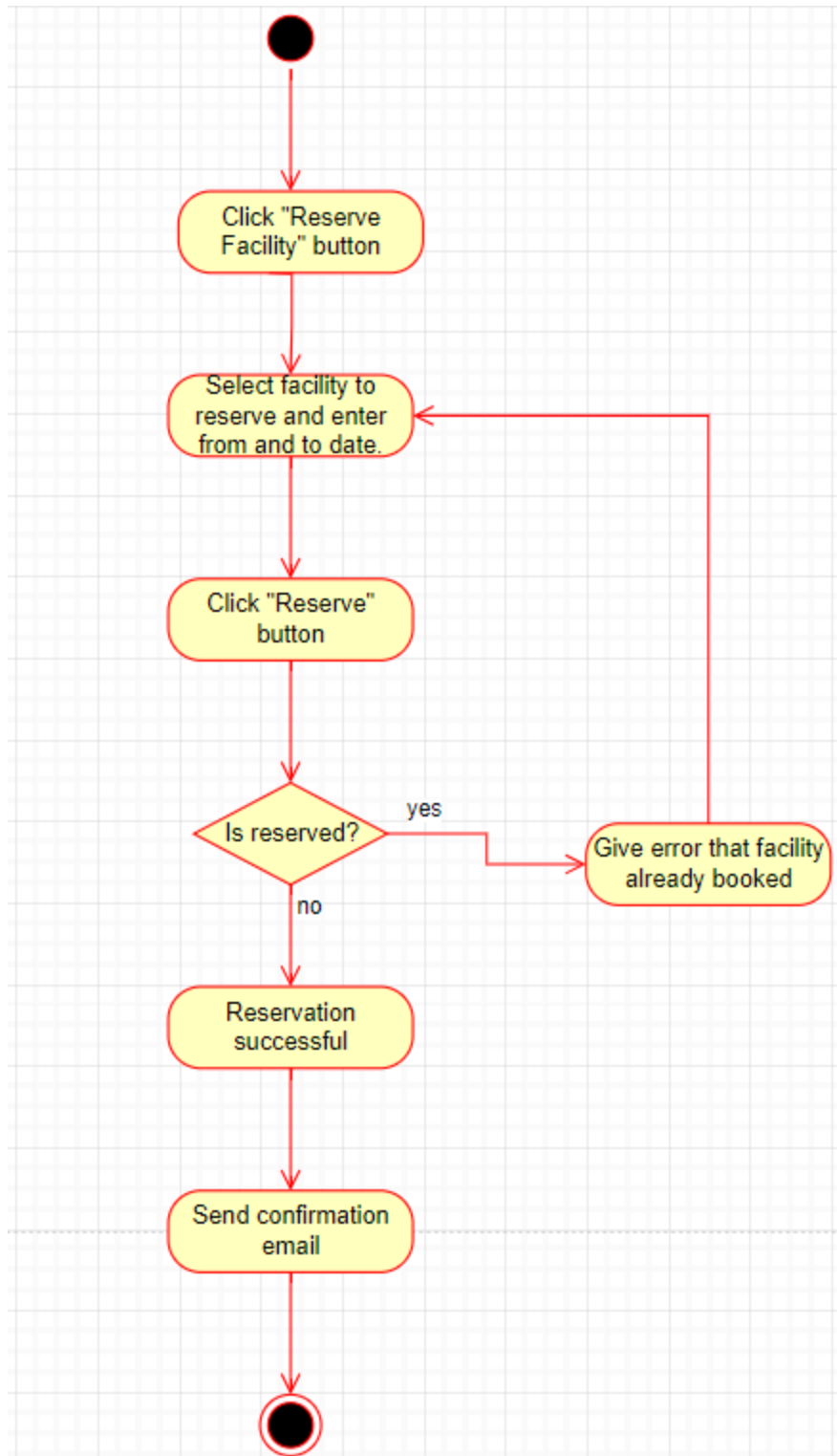


Figure 8: Activity Diagram for Reserving a facility

3.6.4 AD4 - Submitting a Request

This diagram visualizes the activity to submit a request in the system. User clicks on submit a request button and is taken to the request page. Here the user is asked to select the type of request from the dropdown and then enter his contact information. After entering the details, the user clicks on the submit button to submit the request.

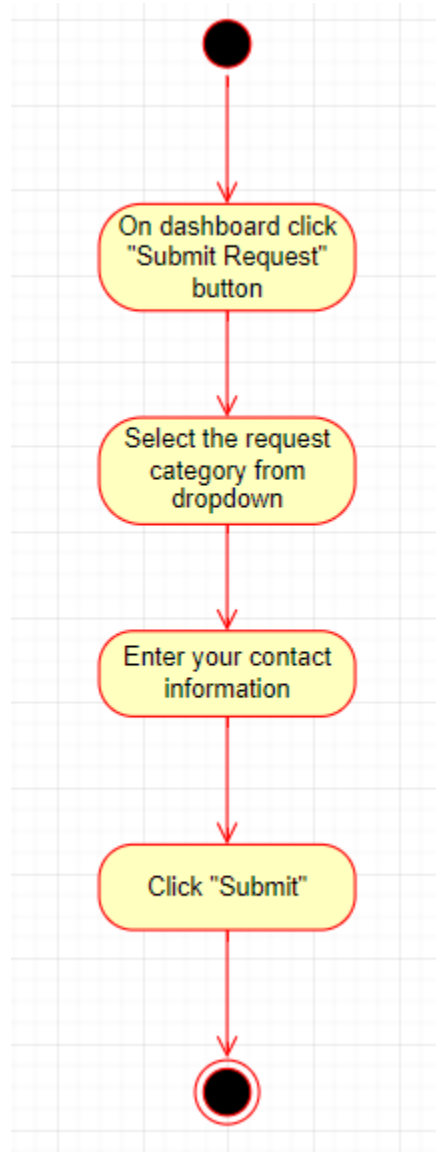


Figure 9: Activity Diagram for Submitting a request

3.6.5 AD5 - Access Financial Report

This diagram visualizes the activity of how a user gets a financial report of his total payments and payments made for a specific year or month. From the dashboard, the user clicks the "Financial Reports" button. The user then selects the year or month for which he wants to see the statistical analysis of. After selecting the user clicks "Get Report" and the information is displayed on the screen.

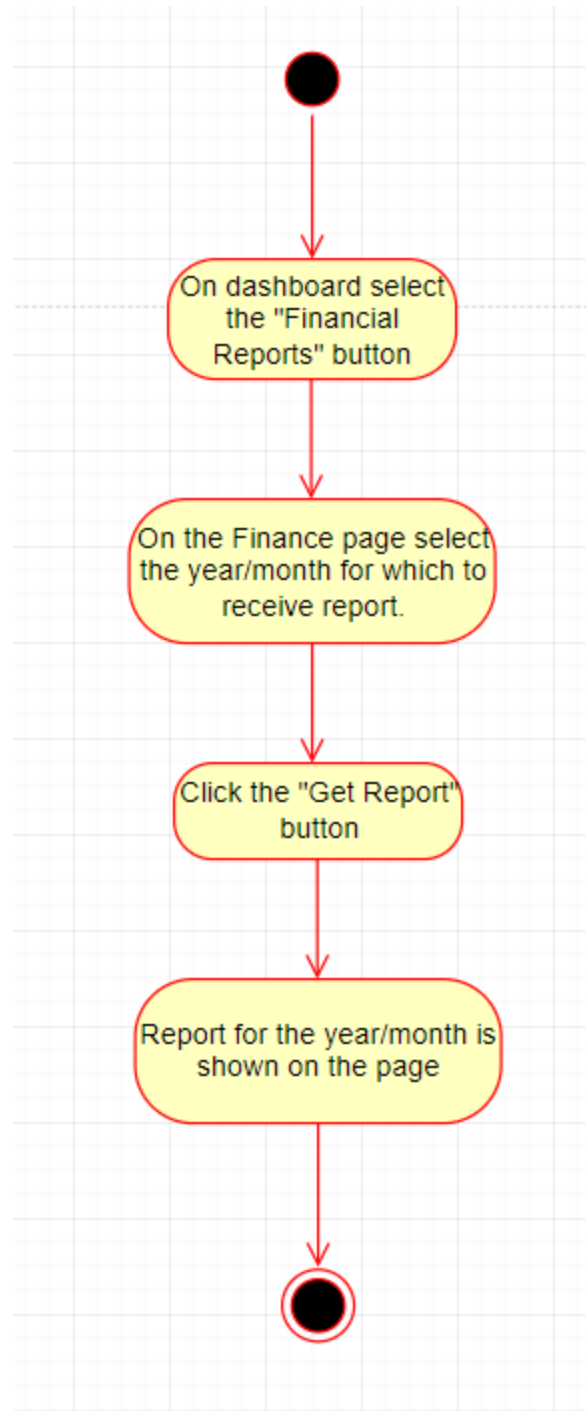


Figure 10: Activity Diagram for Accessing Financial Reports

Bibliography

[1]

'Domain Modeling: What you need to know before coding', *Thoughtworks*. Available:
<https://www.thoughtworks.com/en-ca/insights/blog/agile-project-management/domain-modeling-what-you-need-to-know-before-coding>. [Accessed: Feb. 07, 2024]

[2]

'IBM Documentation', Jul. 18, 2023. Available:
<https://www.ibm.com/docs/en/dma?topic=diagrams-component>. [Accessed: Feb. 07, 2024]

[3]

'IBM Documentation', Mar. 04, 2021. Available:
<https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case>. [Accessed: Feb. 07, 2024]

[4]

'What is Activity Diagram?' Available:
<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>. [Accessed: Feb. 07, 2024]

[5]

'Example: Software Architecture Document'. Available:
<https://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm>. [Accessed: Feb. 07, 2024]

[6]

'GRASP (object-oriented design)', *Wikipedia*. Jan. 08, 2024. Accessed: Mar. 03, 2024. [Online]. Available:
[https://en.wikipedia.org/w/index.php?title=GRASP_\(object-oriented_design\)&oldid=1194382900](https://en.wikipedia.org/w/index.php?title=GRASP_(object-oriented_design)&oldid=1194382900)

[7]

'Deployment Diagram Tutorial', *Lucidchart*. Available:
<https://www.lucidchart.com/pages/uml-deployment-diagram>. [Accessed: Mar. 21, 2024]

[8]

T. A. Gamage, 'Answer to "Having trouble with deployment diagram"', *Stack Overflow*, Mar. 18, 2017. Available: <https://stackoverflow.com/a/42874052>. [Accessed: Mar. 21, 2024]