

Maintenance & Renovation  
Abhay Solanki  
4/16/23

David Chan  
4/17/23

# Peer Review

**Feature Name: Maintenance & Renovation**  
**Developer Name: Abhay Solanki**  
**Date Developer Submitted: 4/16/23**  
**Reviewer Name: David Chan**  
**Date Review Completed: 4/17/23**

**Issued by:**

Algorithmic Alchemist

**Team Lead**

Sierra Harris

**Team Members**

Abhay Solanki

Anh Huynh

Aster Lee

David Chan

Sierra Harris

Github: <https://github.com/abhay772/491B>

## Version History

Version #	Date	Reason for Change
Version 1	4/17/2023	Original Document

## **Table of Contents**

<b>Version History</b>	<b>2</b>
<b>Major Positives</b>	<b>4</b>
<b>Major Negatives</b>	<b>4-5</b>
<b>Unmet Requirements</b>	<b>5-8</b>
<b>Design Recommendations</b>	<b>8</b>
<b>Test Recommendations</b>	<b>9</b>
<b>Maintenance &amp; Renovation BRD Requirements</b>	<b>10-15</b>
<b>Abhay's Maintenance &amp; Renovation Design</b>	<b>16-24</b>

## Major Positives

- The objects are clearly being listed along with the names and types.
- The design reuses code made for other parts of the project which simplifies the design making it easier to read while reducing repetition in design and code.
- The design uses the user's authentication claims to verify the ownership of projects being added or altered which is a more secure and efficient way of validation of ownership.
- There is a good amount of validation on inputs before they reach the database level.
- The front end design lists all the required data needed to be shown and the important information is located mostly around the one-third mark of the screen which is the eye-level where users pay the most attention first.

## Major Negatives

### Frontend

- The front end design flow is a little confusing when first looking at it. There are two flows showing from login to the feature but it is not labeled clearly as different flows.
- There is inconsistency with showing which button needs to be clicked to change the view as homepage to Maintenance and Renovation views both don't specify that the side button with the same name needs to be pressed.
- Many buttons are not detailed as buttons and what view it should lead to for example the top-right view is shown to be a detailed view of the service with an "X" to close it back but it is not detailed as a button and what it should do.
- For the list of services or items it is not clear to the user if 5 is limit or if it is scrollable or if more is able to be loaded in. There is also no data showing the total amount and how many are being shown. For example if the user has 20 services in the list it should show the phrase "5 out of 20 being shown".

- Each view is not clear on its purpose as some have names underneath and others do not.
- The search bar and services to be compared are not shown how it would function with user interaction from the front end design.

## **Backend**

- The design shows the MnRController always existing before it is called or initiated so it should get initiated when the first ajax call to the controller is made by the frontend. The last return to the client side view should be where it ends also for the diagram.
- The controller layer is the entry point so it should do some basic validation also along with the client side validation to stop calls being made to the manager with invalid data.
- Many of the backend design requirements that require a timer to check if an action is done under a certain time are not implemented in the current design so a developer coding based on the design alone would not know to set a time restriction for certain actions.
- Many of the pop up notifications or view updates do not have any way of being verified in the current designs.
- There are a lack of failure cases that are required to show the design flow of when the system has to handle any kind of errors.

## **Unmet Requirements**

### **Compare Missing Requirements**

- There is no design shown to keep track that the system must create the list within 5 seconds or that the notification bubble must be created in under 5 seconds
- There are no failure cases to show what would happen if there was an error in the system.
- There are no designs for when a user enters a service name and does not get a list of services even though they exist. It should show when services are found in

the database and then on the client side there are no services then it should log it as an error.

- There are no designs for when a user enters a service name and it does not have any matches but a pop up is not shown. For checking if the popup was shown make a flag in the popup code and if it is run then it successfully made the popup but if not then there should be an error that is logged.
- There are no designs for when the user enters a service name and the system returns a list of completely different services that don't match the entered name. The system should check the name and description for if it matches part of what is searched and if none is matching then it should log it as an error.

### **Estimate Project Missing Requirements**

- The design does not show anything that requires it to show the price to the user in US dollars.
- It needs to show what happens if the property value is not calculated and displayed to the user. It could show that if there was an error calculating it should log and then return a message back to the user to try again.
- There could be a check if the property value is calculated and not displayed to the user so when the value is calculated it could return an success message and if there is a success message then it should check if the code to post the value ran and if it did not then it should log an error.
- If there was an error when property value estimation is being added to a project then it should log the error and return an error message to the view.

### **Price Charts Missing Requirements**

- The period of time for the price chart can be adjusted to weekly, monthly, or yearly and is not shown as being able to be changed in the charts..
- There are no design specifications that the price chart must be displayed as a line graph. So the design should specify that so that other chart styles are not used.

- The price and period of time is not stated for which to be on which axis. Clearly labeling which will be the x-axis and which will be the y-axis would clarify that the price and period of time should be on the opposing axis.
- If the price chart is not developed then it should log and return an error but that is not shown or if the price chart is developed it should mark a success flag but if it was not displayed then a success flag for display would not be run meaning it should lead to an error.
- If the price chart shows incorrect information then there should be a design to show it being logged.

### **Add Project Missing Requirements**

- There is no timer requiring the user's project to be added to account within 5 seconds and displayed to the user.
- The design doesn't have a failure case for when the user project is added to the system within 5 seconds but not displayed to the user.
- There should also be a design for when the user project is not added within 5 seconds but displayed to the user or if the user project is not added and not displayed to the user. The user should receive an "Unable to Add Project" error message.

### **Edit Project Missing Requirements**

- When the user project modifications are updated and displayed to the user there should be a timer checking if it is under 5 seconds in the design
- If a project modification took more than 5 seconds to update the design should show what will happen next like logging the error and showing the user an error message.
- When project modifications are updated within 5 seconds but not displayed to the user then the design should show what should happen next.

- The design doesn't show how the system deals with when there are project modifications that are not updated and the user is displayed old data or it's not displayed to the user.

### **Delete Project Missing Requirements**

- The design is missing what the system should do when the user does not have the rights to delete a project and can not delete the project. There could be a part of the design that shows if the user's claim doesn't match the project then it logs the attempt and returns an error message to the user.
- The design should have a timer to check if the user project is not deleted within 5 seconds and if the user does not receive a success message.
- If there was an error at the database level and the user project could not be deleted then the user should receive an "Unable to Delete" message in the design.

## **Design Recommendations**

When a layer or object is never used then it should not be in the design as it could make it more cluttered and difficult to read the design. There should also be timers to restrict how long certain actions should take. An easy way to set a timer to validate how long an action took would be to have the action be asynchronous and then I would make a task call and a second task with a 5 second delay then use "completedTask = await Task.WhenAny" to compare which task finished first by checking if the completedTask equals the first task call.

I would also recommend that for all inputs that there would be input validation in both client side and in the entry point controller for backend to reduce calls further into the backend which would obviously fail. That would reduce useless calls and save processing power.

There are also a lack of failure design cases which is important to have when coding based on the design to know how to deal with errors and error handling. The design for handling any case other than the success case should be robust and



thorough. I recommend using flags or code to check if a certain section of code was called and ran. One way is if a function fails and returns a false for the result then I would require that the other function that needs to be run when the function fails to return a result of true.

## Test Recommendations

First test I would do would be to test all of the validation rules to make sure only valid inputs are placed by the user. To test this I would start with a test using invalid characters, then a test with too many characters, followed by below the minimum if there is a minimum set. Next I would test the authentication requirements by making a test with a type of user that should not have access for example if it requires a property manager user then make a test with a service provider user. Also test if the project belongs to a certain user that a different user can not access it. More tests can be made to check how the system deals with failure states. For example a test can be made with a dummy database that returns an error and then you can assert if the action that was planned to happen got executed. Another is to place invalid data in the database and check how the system deals with invalid data that would break the design flow.

Front end testing could be done with Nightwatch where you test the entire feature from end to end. I would start with making a test for each of the success cases in the BRD to test if that works as planned. Then next I would attempt to make test cases that follow the flow of what would happen if the user follows each of the failure cases. Finally the last set of tests I would recommend is to check all the popup and notifications in the designs. I would make a test that meets the requirements to make each popup or notification appear and then use nightwatch to check that it is shown.

## Maintenance & Renovation BRD Requirements

Use Case ID	<b>36</b>
Use Case Name	<b>Maintenance and Renovation - Compare</b>
Actor	- Property Managers.
Business Rule	<ul style="list-style-type: none"><li>- The service title must be under 50 characters.</li><li>- The service description must be under 250 characters.</li></ul>
Description	- This service will show a list of available services for the users to compare information
Preconditions	- The user has successfully logged into the website and selected the Maintenance and Renovation feature from the homepage.
Functional	<ul style="list-style-type: none"><li>- The user must enter a name of a type of service which must be under 50 characters.</li><li>- The system must create and display a list of services.</li><li>- Each service must show a price, title, and description.</li><li>- If there are no matching available services then the system must pop up a notification notifying the user.</li></ul>
Postcondition	<ul style="list-style-type: none"><li>- User sees a list of available services.</li><li>- If no matching available services then the user sees a notification bubble saying there are none available.</li></ul>
Nonfunctional	<ul style="list-style-type: none"><li>- The system must create the list within 5 seconds</li><li>- The notification bubble must be created in under 5 seconds</li><li>- Only 5 services should be displayed per page</li></ul>
Success Case	<ul style="list-style-type: none"><li>- The user enters a service name and a list of matching services is generated and displayed to the user</li><li>- The user enters a service name and if there are no matches it returns a pop up notification for it.</li></ul>
Failure Case	<ul style="list-style-type: none"><li>- The user enters a service name and does not get a list of services even though they exist.</li><li>- The user enters a service name and it does not have any matches but a pop up is not shown.</li><li>- The user enters a service name and the system returns a list of completely different services that don't match the entered name.</li><li>- More than 5 services were displayed per page</li></ul>

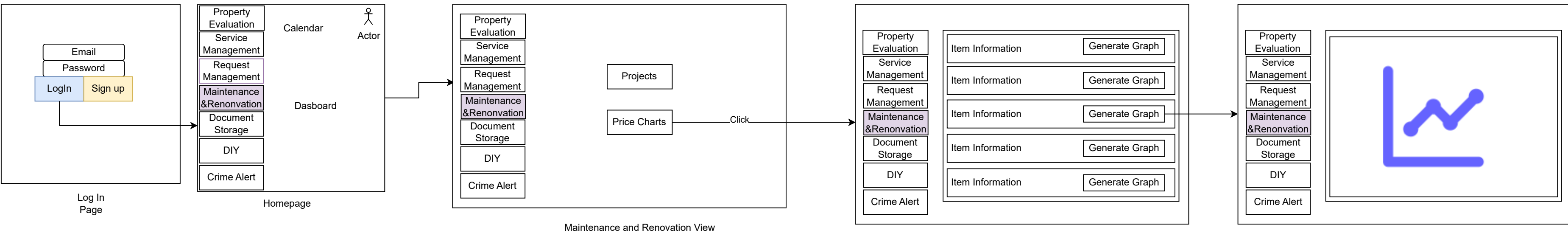
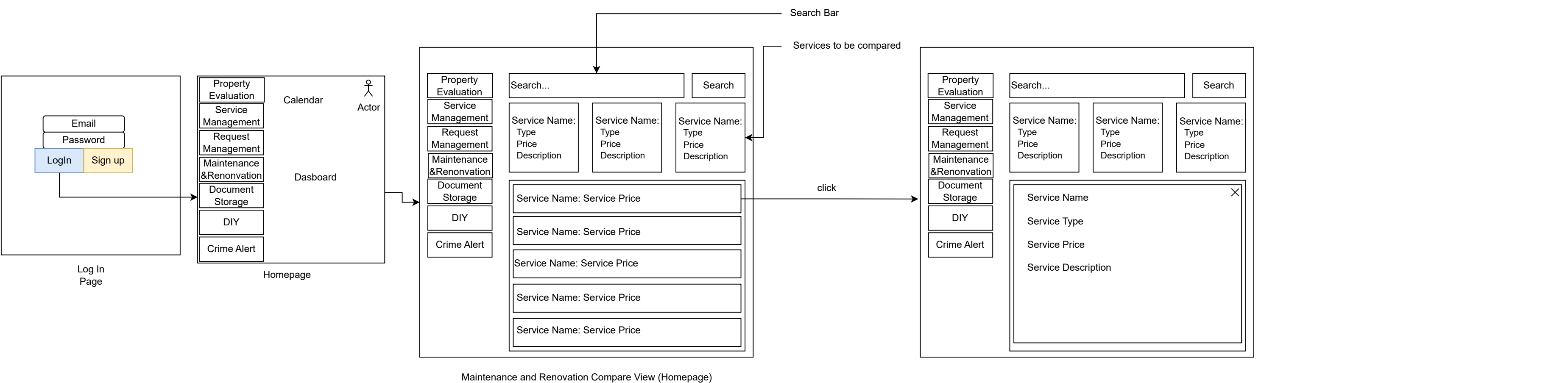
Use Case ID	<b>37</b>
Use Case Name	<b>Maintenance and Renovation - Estimate Project</b>
Actor	- Property Managers
Business Rule	- The user should be informed of any possible change in their property evaluation value due to a service
Description	- This service will give the users an estimate of how much their property evaluation will increase or decrease if a particular service is done to their property
Preconditions	- User has successfully logged into the account and navigated to the Maintenance and Renovation page. Users have select a service
Functional	<ul style="list-style-type: none"><li>- Calculates the property value using property evaluation considering the selected service</li><li>- Displays the change in property evaluation value</li></ul>
Postcondition	- Property value estimate is calculated and is display to the user
Nonfunctional	<ul style="list-style-type: none"><li>- Property value estimation is displayed to the user in US dollars</li><li>- Property value estimation is added to project</li></ul>
Success Case	- The property value is calculated and displayed to the user
Failure Case	<ul style="list-style-type: none"><li>- Property value is not calculated and displayed to the user</li><li>- Property value is calculated and not displayed to the user</li><li>- Property value estimation is not added to project</li></ul>

Use Case ID	<b>38</b>
Use Case Name	<b>Maintenance and Renovation - Price Charts</b>
Actor	- Property Managers
Business Rule	- User should be able to refer to price chart before committing to any service or DIY project
Description	- Price charts displays to the user the price of an project, service or item over a certain period of time
Preconditions	- User has successfully logged in and has navigated to the Maintenance and Renovation page
Functional	<ul style="list-style-type: none"><li>- Gather price information over a selected period of time</li><li>- Graph price information</li><li>- Display price chart to user</li></ul>
Postcondition	- Price chart is displayed
Nonfunctional	<ul style="list-style-type: none"><li>- The period of time for the price chart can be adjusted to weekly, monthly, or yearly.</li><li>- The price chart is display as a line graph</li><li>- The price and period of time on opposing axis</li></ul>
Success Case	<ul style="list-style-type: none"><li>- The price chart gathers information concerning the correct object or service</li><li>- The price chart is created and display</li></ul>
Failure Case	<ul style="list-style-type: none"><li>- The price chart is not developed</li><li>- The price chart is developed but not displayed</li><li>- The price chart shows incorrect information.</li></ul>

Use Case ID	<b>39</b>
Use Case Name	<b>Maintenance and Renovation - Add Project</b>
Actor	- Property Managers
Business Rule	- The user will be limited to having 3 different projects simultaneously.
Description	<ul style="list-style-type: none"> <li>- Users will be able to create project plans and view them on their account</li> <li>- Users will be able to choose the period in which the project will take place, their desired date and time for services, the planned services and budget</li> </ul>
Preconditions	- User is successfully logged in and navigates to their projects in the Maintenance and Renovation section
Functional	<ul style="list-style-type: none"> <li>- The system will create a project plan using the users inputted period, date and time, service, and budget <ul style="list-style-type: none"> <li>- Period and date will be accepted in the form of month/day/year to month/day/year</li> <li>- Time will be accepted in the form of "00:00"</li> <li>- Budget will be accepted in US dollars</li> </ul> </li> <li>- The user will enter the desired period duration of the project, date and time of services, and budget</li> <li>- The system will add the users property evaluation to the project</li> <li>- User will be able to choose from services within their budget</li> </ul>
Postcondition	<ul style="list-style-type: none"> <li>- A project plan will be created and displayed to the user</li> <li>- User receives "Project Added" message</li> </ul>
Nonfunctional	<ul style="list-style-type: none"> <li>- Users will be able to sort services by rating and/or price</li> <li>- Users project will be added to their profile within 5 seconds</li> <li>- Users project will show their estimated property evaluation value</li> <li>- Users project will the change in property evaluation</li> </ul>
Success Case	- Users project is added to account within 5 seconds and displayed to the user
Failure Case	<ul style="list-style-type: none"> <li>- User project is added to system within 5 seconds but not displayed to the user</li> <li>- User project is not added within 5 seconds but displayed to the user</li> <li>- User project is not added and not displayed to the user. User receives "Unable to Add Project" error message</li> </ul>

Use Case ID	<b>40</b>
Use Case Name	<b>Maintenance and Renovation - Edit Project</b>
Actor	- Property Managers
Business Rule	- Users must be able to edit their project only
Description	- The user will be able to edit a project that they created such as the period of when the project will take place, the desired time and dates for services, planned services, and budget.
Preconditions	- User is successfully logged into their account and navigates to Maintenance and Renovation project
Functional	<ul style="list-style-type: none"> <li>- User should be able to edit existing projects</li> <li>- The user can edit the planned services required for the project.</li> <li>- Users can edit the date and time they plan to have services.</li> <li>- Users can edit the period and desired duration of their project</li> <li>- Users will be able to increase or decrease their project budget</li> </ul>
Postcondition	<ul style="list-style-type: none"> <li>- Users project plan is updated in the system</li> <li>- User receives "Project Update Successful" message</li> </ul>
Nonfunctional	- User modifications are updated within 5 seconds and displayed to the user
Success Case	- User project modifications are updated and displayed to the user within 5 seconds
Failure Case	<ul style="list-style-type: none"> <li>- Project modification took more than 5 seconds to update</li> <li>- Project modifications are updated within 5 seconds but not displayed to the user.</li> <li>- Project modifications are not updated and user is displayed old data</li> <li>- Project modifications are not updated and not displayed to the user</li> <li>- User is able to modify another user's project</li> </ul>

Use Case ID	<b>41</b>
Use Case Name	<b>Maintenance and Renovation - Delete Project</b>
Actor	- Property Managers
Business Rule	- Users must be able to delete their project.
Description	- The user will be able to delete a project that they created.
Preconditions	<ul style="list-style-type: none"><li>- User has successfully logged into their account and navigated to the Maintenance and Renovation option.</li><li>- User has one more projects</li></ul>
Functional	- Users project plan and data is deleted from the database
Postcondition	<ul style="list-style-type: none"><li>- Project is deleted from the user's profile.</li><li>- The user can no longer access deleted projects.</li><li>- User receives and "Project Delete Successful" message</li></ul>
Nonfunctional	<ul style="list-style-type: none"><li>- User is able to delete project using the "Delete Project" option</li><li>- User project is deleted within 5 seconds</li></ul>
Success Case	<ul style="list-style-type: none"><li>- User project is deleted within 5 seconds and receives an "Project Delete Successful" message</li><li>- The user does not have the rights to delete a project and can not delete the project.</li></ul>
Failure Case	<ul style="list-style-type: none"><li>- User project is not deleted within 5 seconds and does not receive success message</li><li>- User project could not be deleted and receives an "Unable to Delete" message</li><li>- The user can delete a project that they should not be able to delete.</li></ul>





Service
+ ServiceID: int
+ ServiceName: string
+ ServiceType: string
+ ServicePrice: string
+ ServiceDescription: string

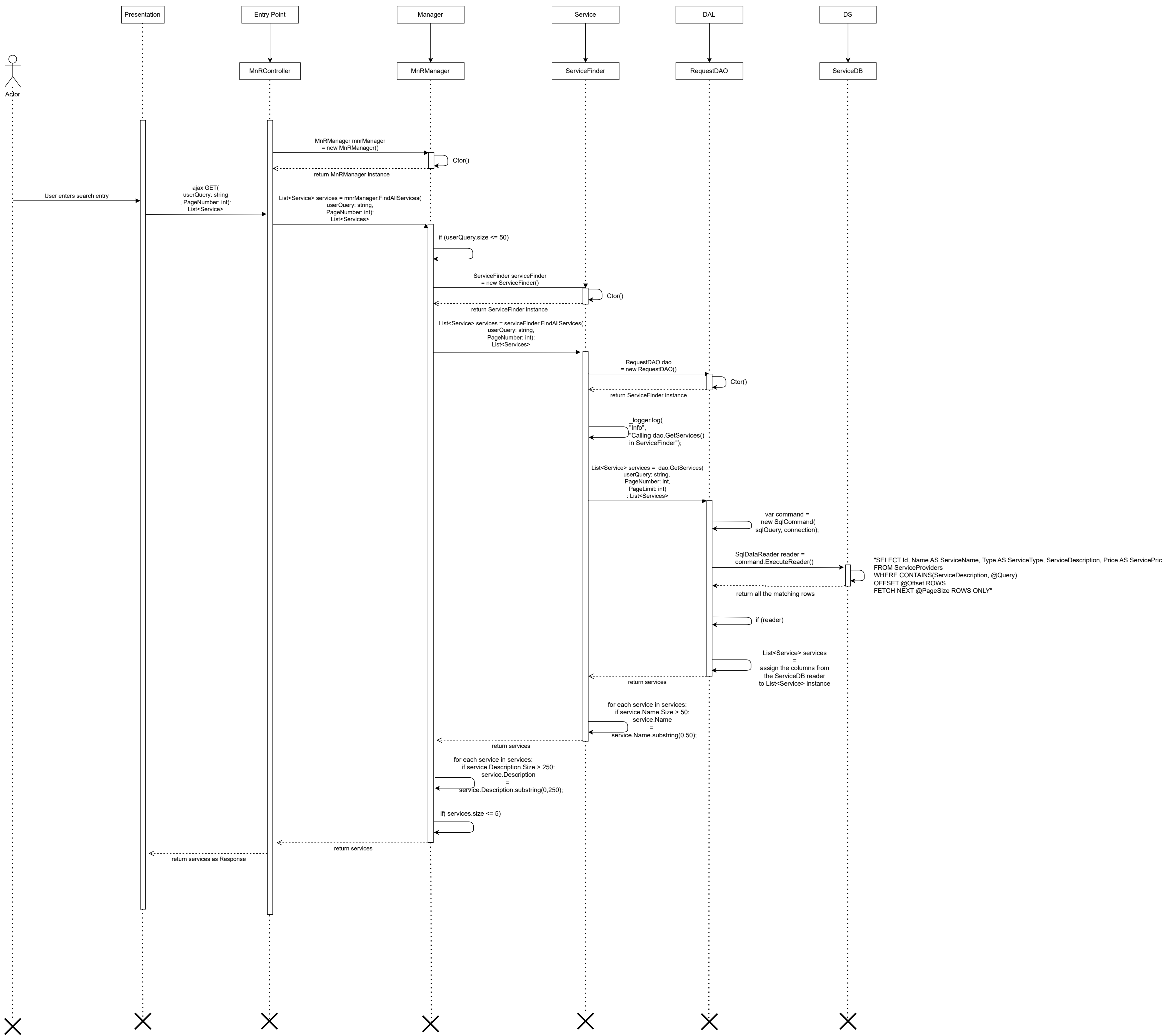
ProjectDetail
+ ProjectName: string
+ ServiceID: int
+ StartDate: Date
+ EndDate: Date
+ ServiceTime: Time
+ Budget: Double

ProfileChange
+ CNoOfBedrooms: int
+ CNoOfBathrooms: int
+ CSqFeet: int

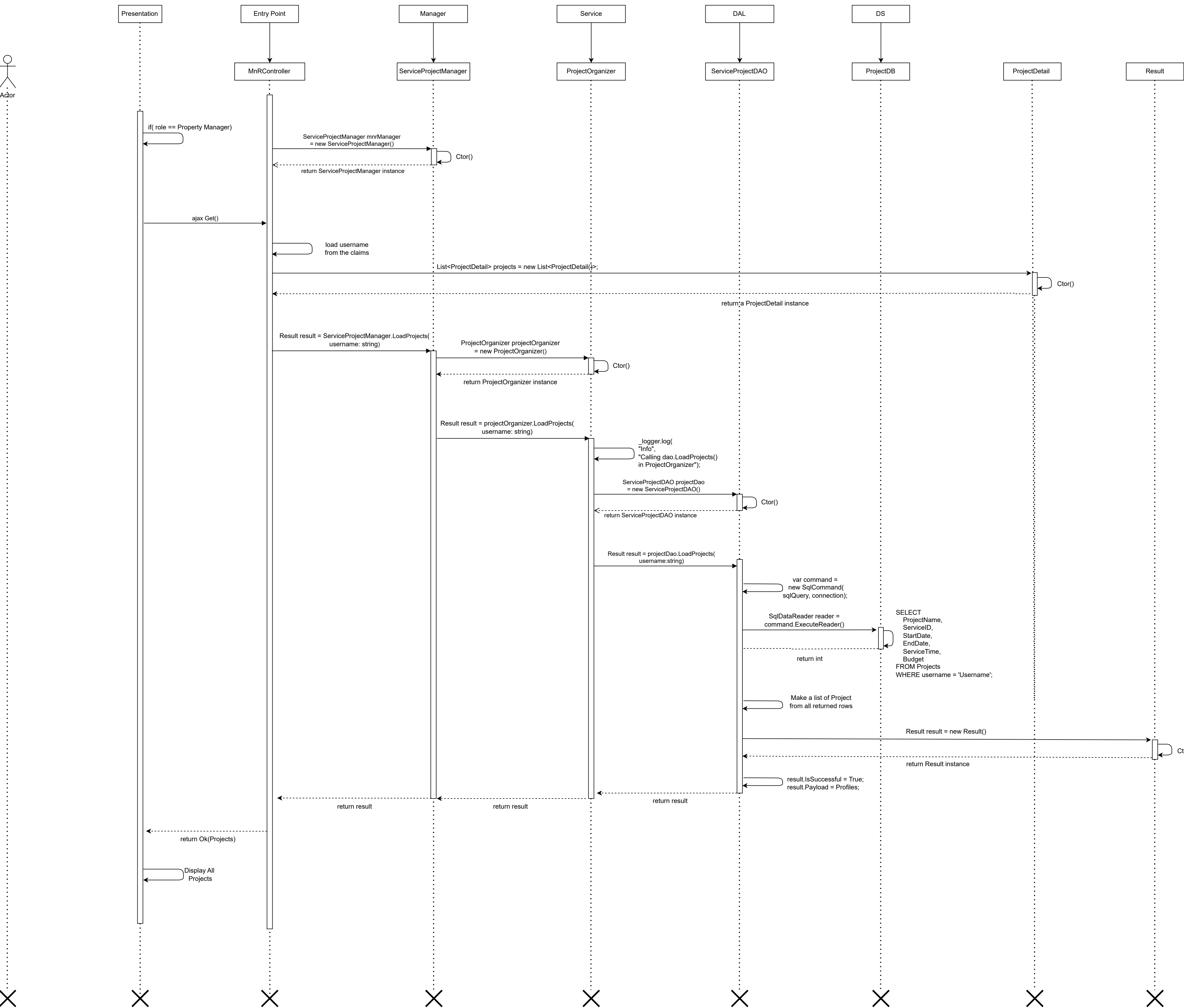
PropertyProfile
+ NoOfBedrooms: int
+ NoOfBathrooms: int
+ SqFeet: int
+ Zip: string
+ Description: string

EstimateDTO
+ EvalChange: double
+ OEvaluation: double

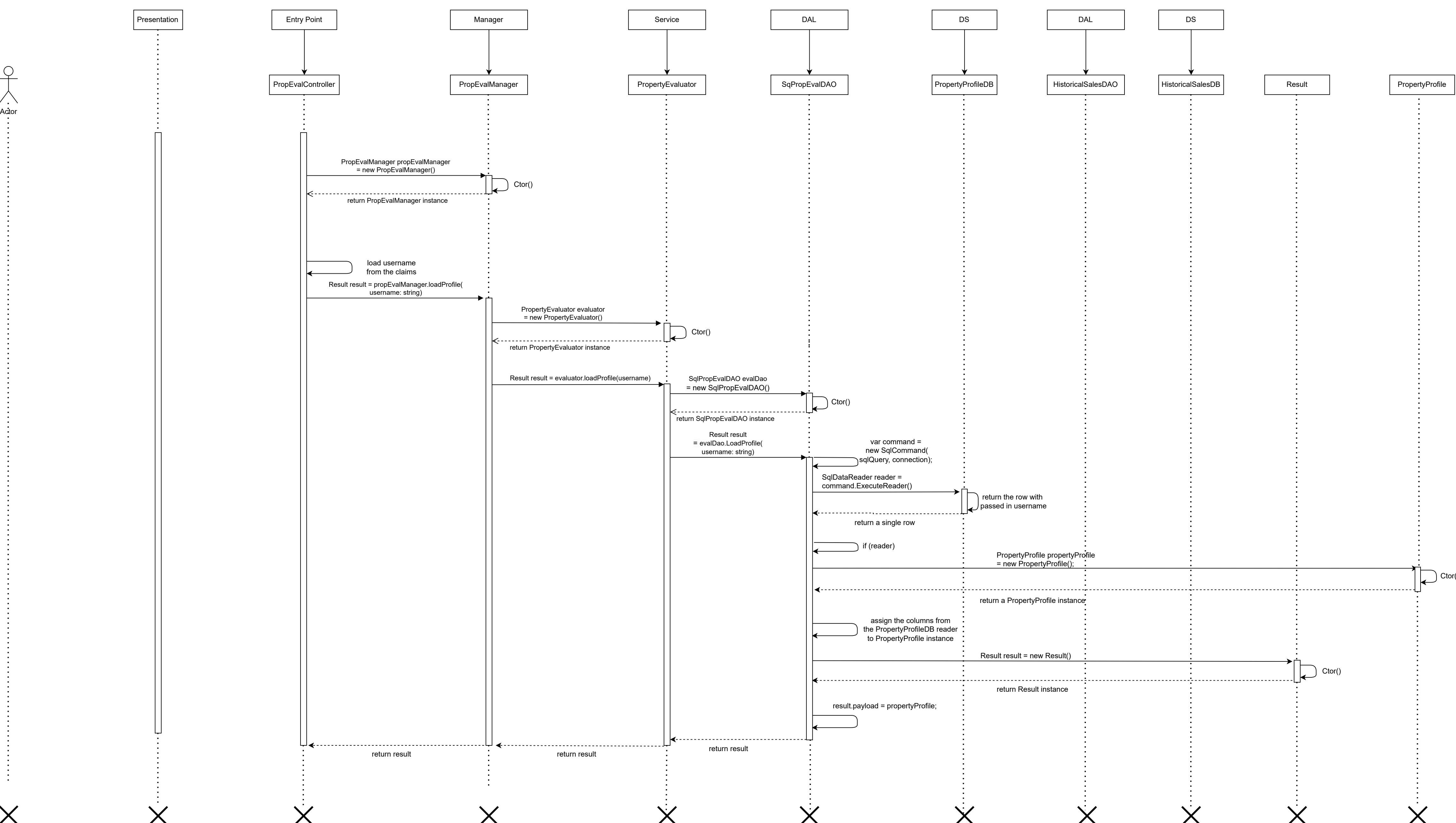
LoadServices

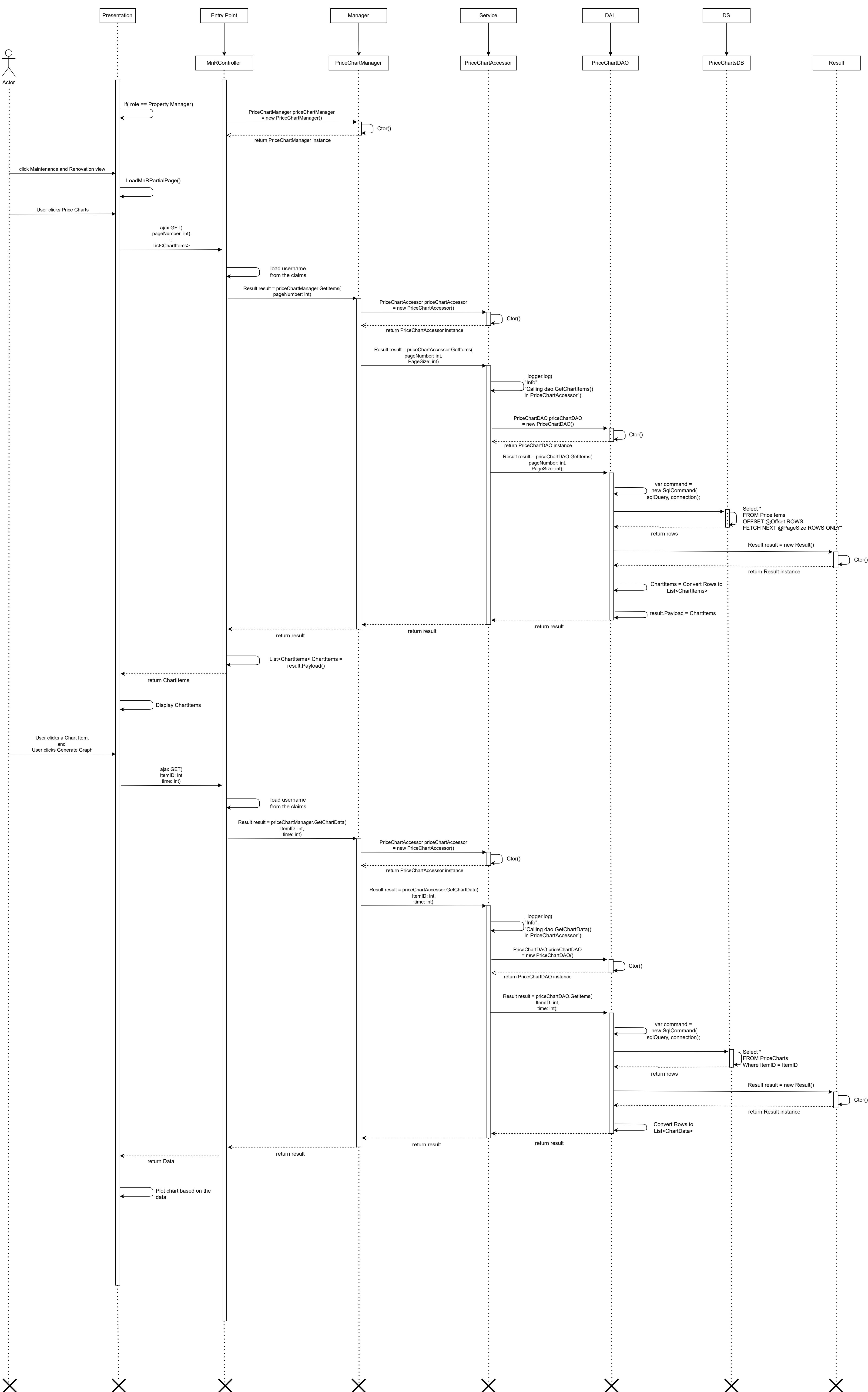


LoadProjects

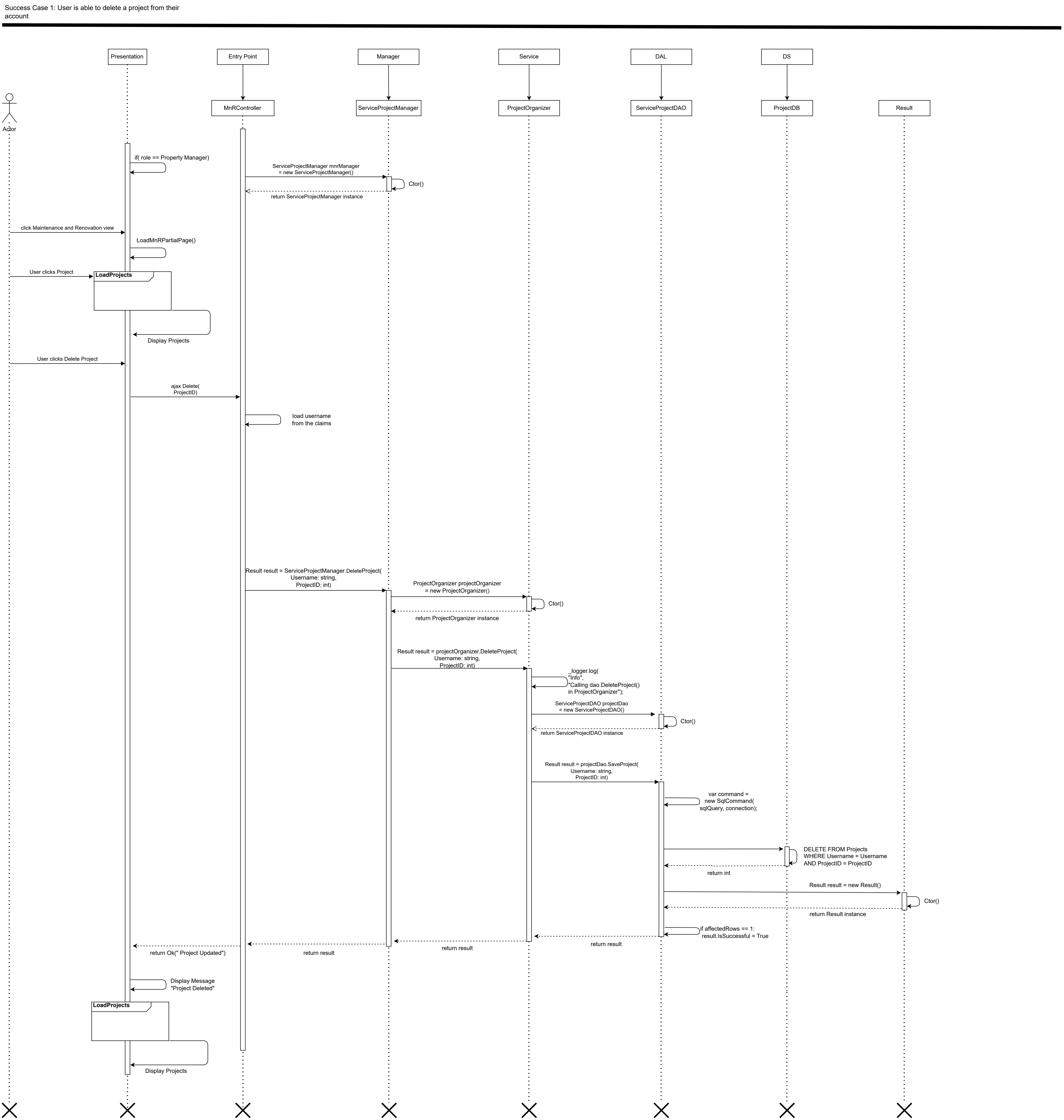


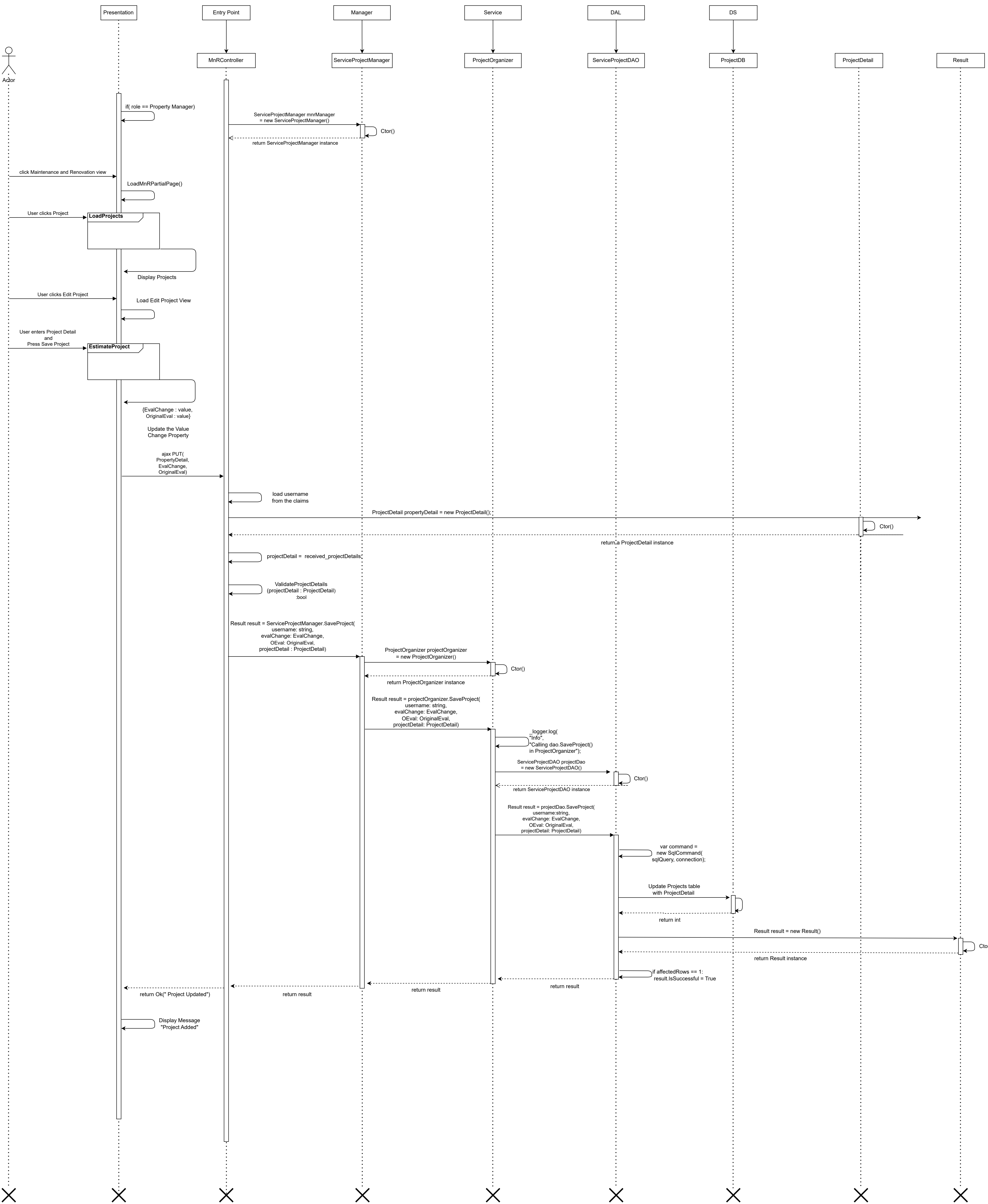
LoadProfile



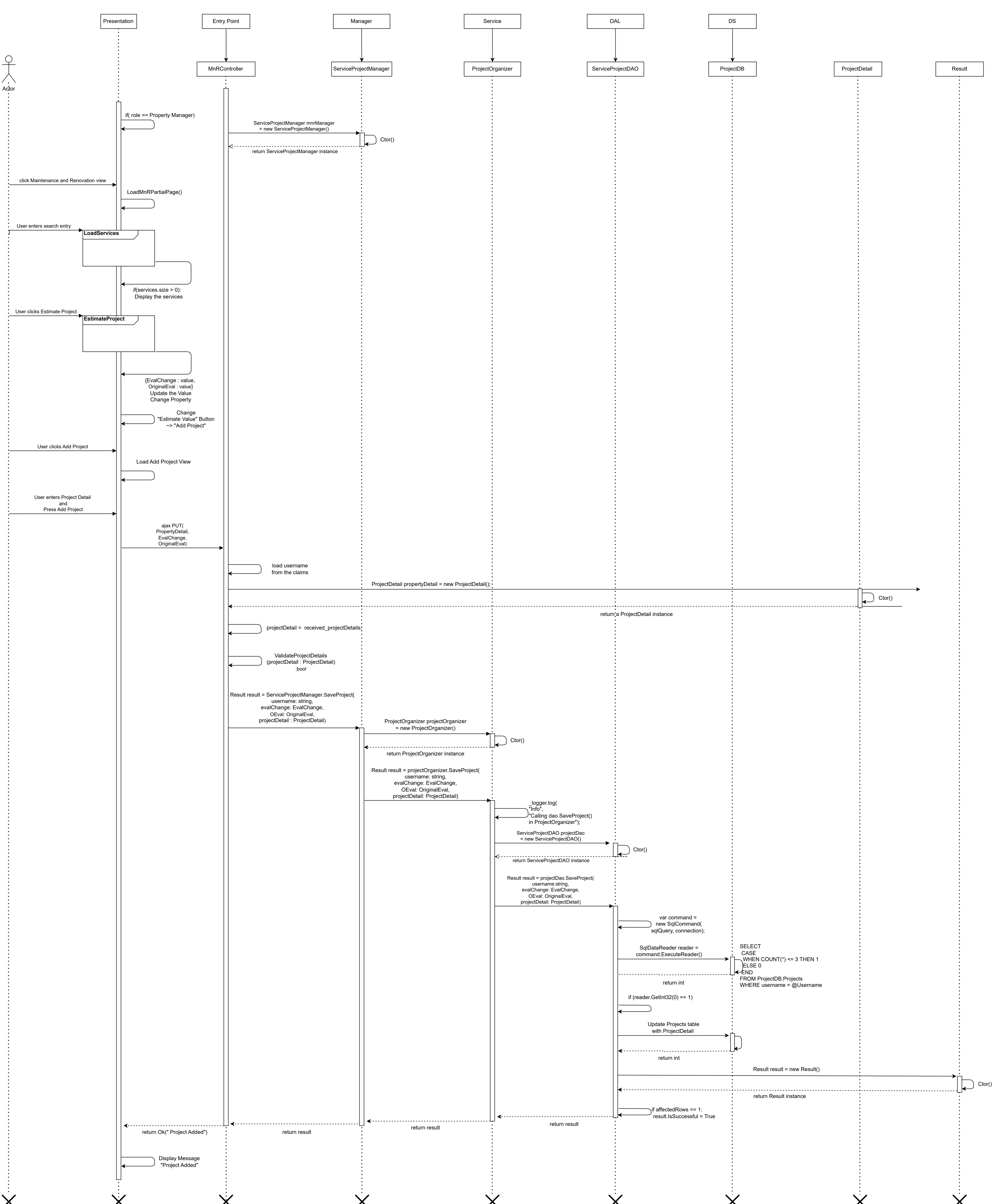


Success Case 1: User is able to delete a project from their account

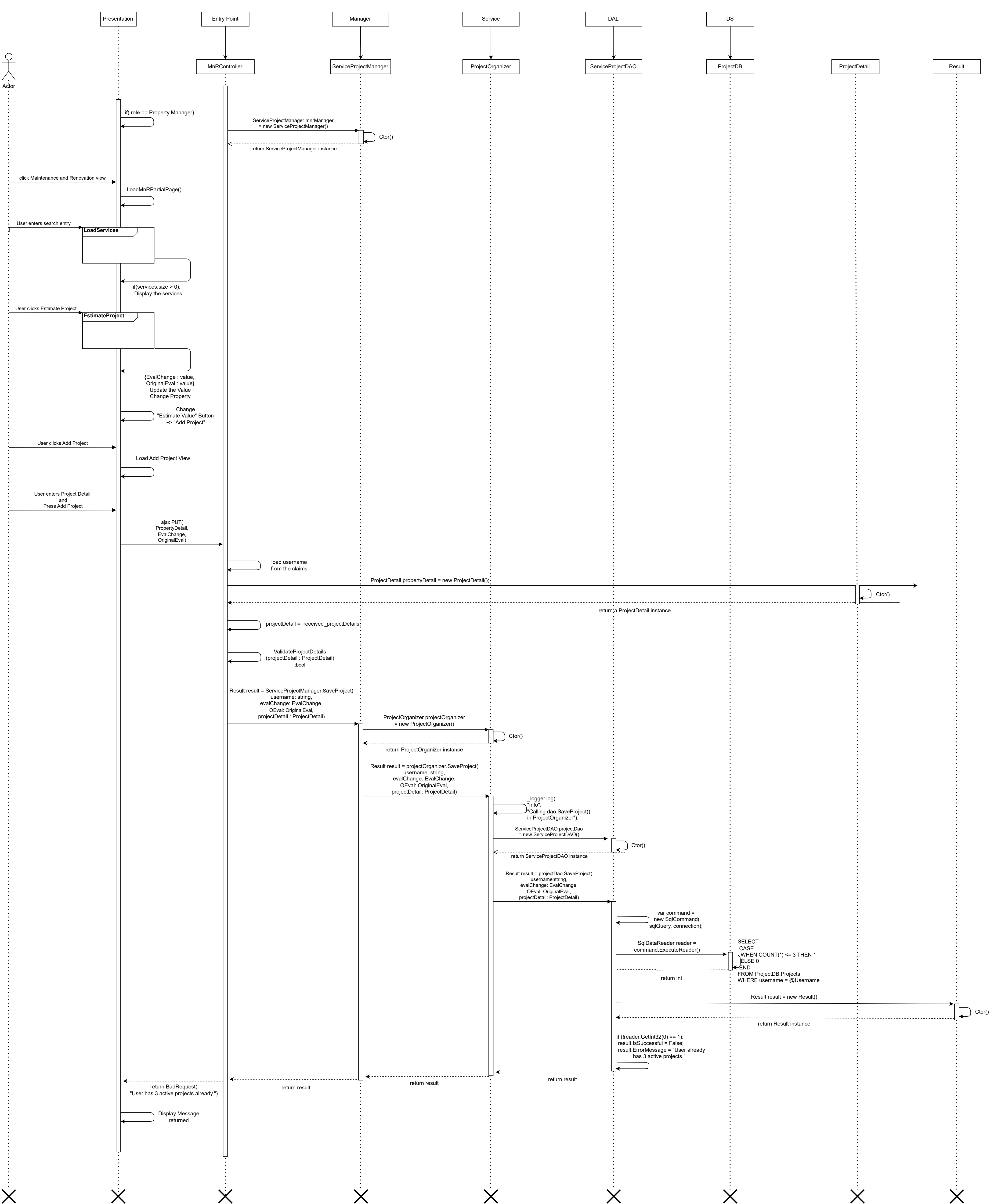




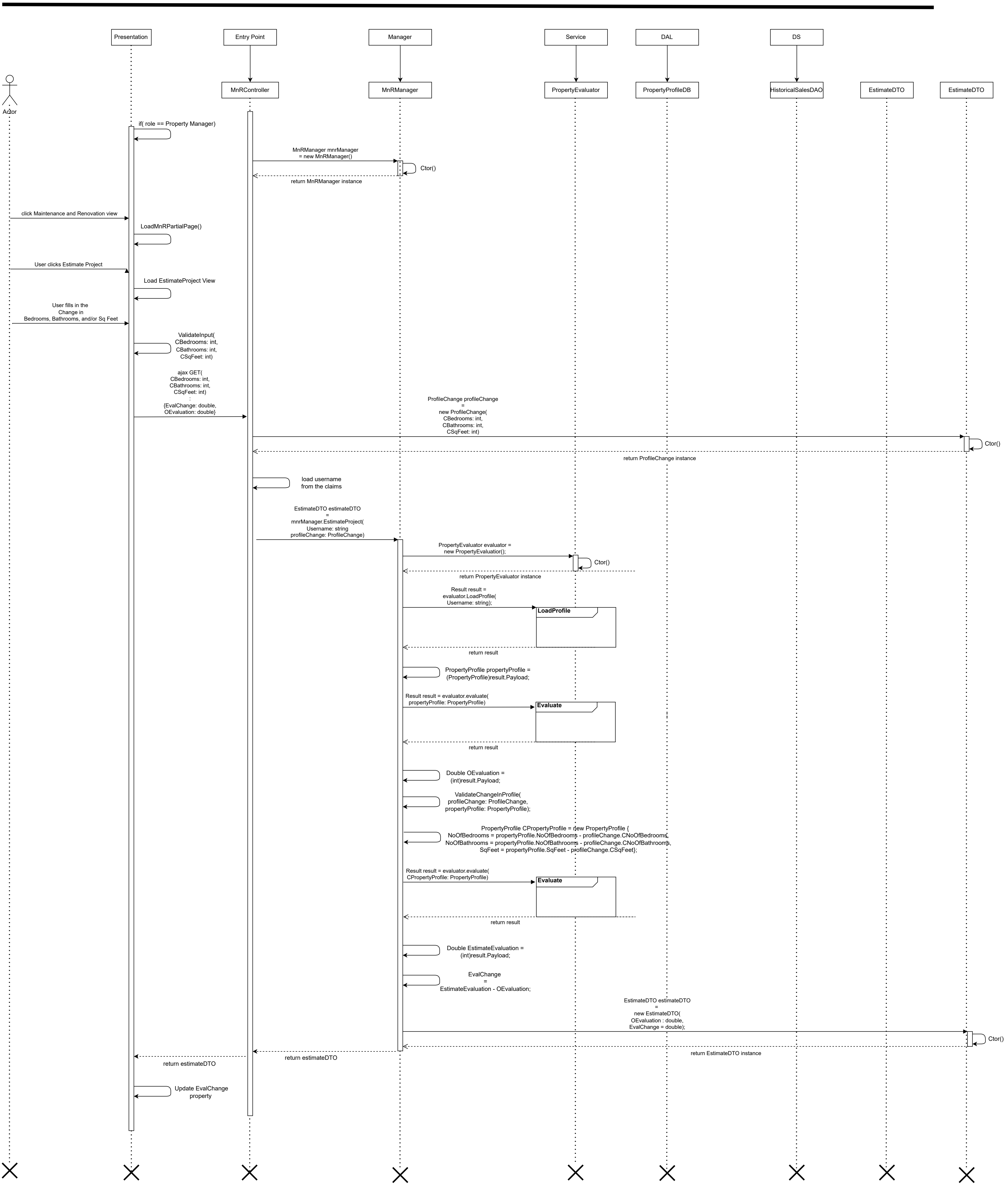
Process Case 1: Users project is added to account



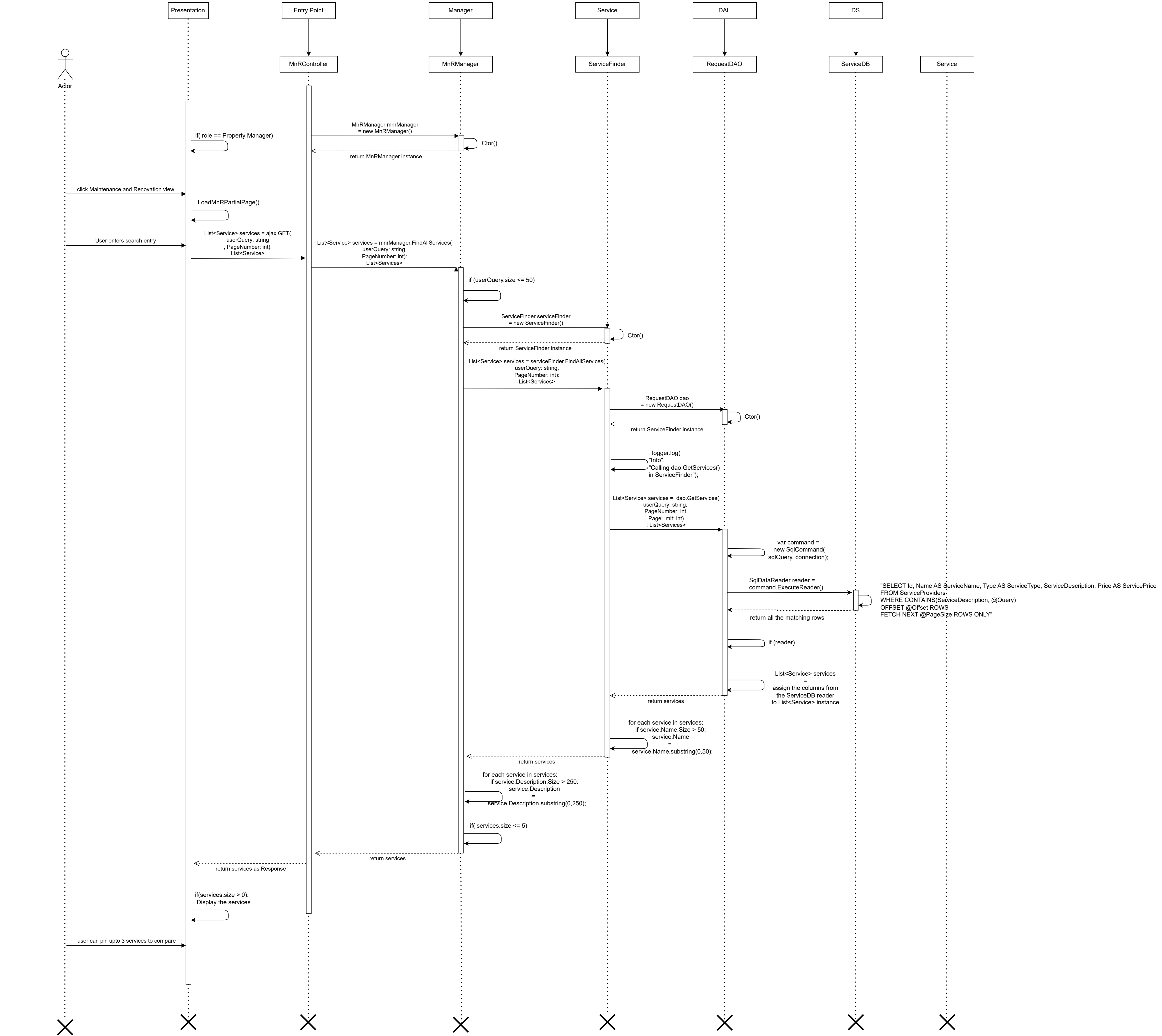
Failure Case 1: The user tries to add a Project while having 3 active Projects.



Success Case 1: The property value is calculated and displayed to the user



Success Case 1: The user enters a service name and a list of matching services is generated and displayed to the user.



Success Case 2: The user enters a service name and if there are no matches it returns a pop up notification for it.

