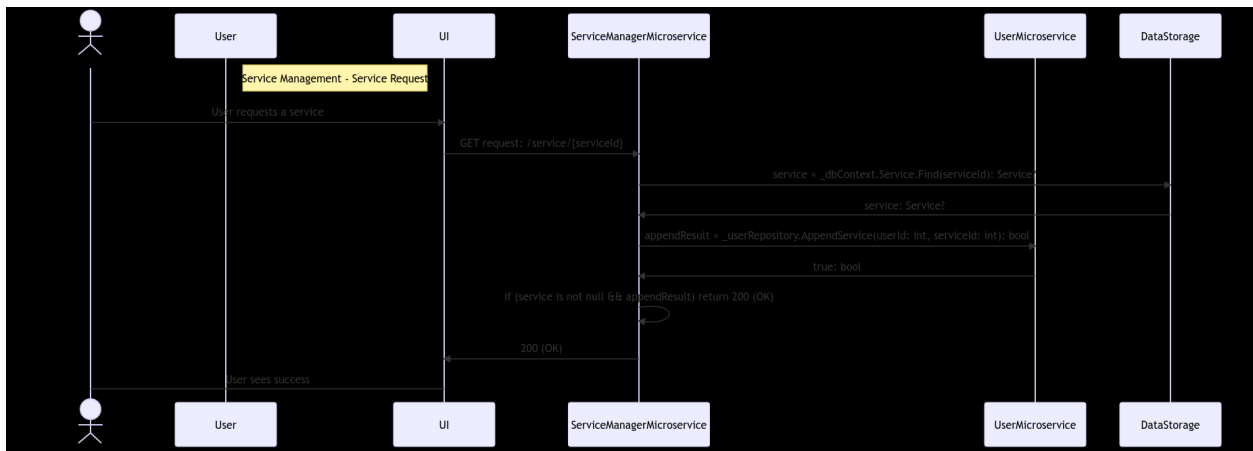


Use Case ID:	42
Use Case Name:	Service Management - Service Request
Actors:	- Property Manager
Business Rules:	- User must be a property manager with a known address.
Description:	- This feature will allow the user to requesting a service from a service provider
Preconditions:	- The user has successfully logged in to the website, and selected the Service Management feature from the homepage.
Service Request Functional	<ul style="list-style-type: none"> - Display services and service providers list - Allow the user to request services - Notifies service provider of request - Allow users to enter additional comments
Postconditions:	<ul style="list-style-type: none"> - A service request is created and sent, waiting for the service provider's response. - The user service list is updated to contain the pending confirmation service request
Nonfunctional	<ul style="list-style-type: none"> - The search bar should complete the search within 5 seconds. - Notification to property manager should be sent within 5 seconds - Notification to service provider should be sent within 5 seconds - Updating service data should take no more than 5 seconds. - Comment text allows only 150 characters - Only 5 services should be displayed on the view
Success Cases:	<ul style="list-style-type: none"> - User chooses a service and the system updates the service request of the service provider. - The system updates the users services with the pending service
Failure Cases:	<ul style="list-style-type: none"> - The service provider does not receive the request from a property manager - The service provider does receive the request but the information received is wrong - The search bar did not complete the search within 5 seconds. - It took more than 5 seconds for the property manager to get a notification - It took more than 5 seconds for the service provider to get a notification - Updating service data should take more than 5 seconds. - Comment text was more than 150 characters - More than 5 services were displayed per page - The users services list is not updated with the new pending service



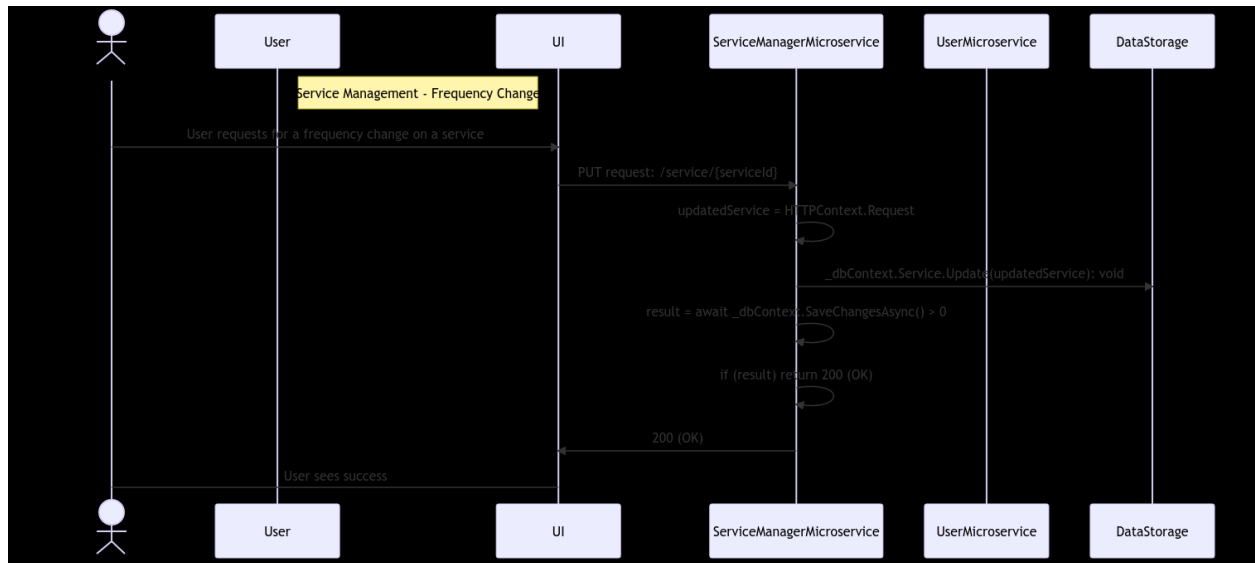
sequenceDiagram

```

actor U as User
Note right of User: Service Management - Service Request
participant UI
participant SMM as ServiceManagerMicroservice
participant UM as UserMicroservice
participant DS as DataStorage
U->>UI: User requests a service
UI->>SMM: GET request: /service/{serviceId}
SMM->>DS: service = _dbContext.Service.Find(serviceId): Service?
DS->>SMM: service: Service?
SMM->>UM: appendResult = _userRepository.AppendService(userId: int,
serviceId: int): bool
UM->>SMM: true: bool
SMM->>SMM: if (service is not null && appendResult) return 200 (OK)
SMM->>UI: 200 (OK)
UI->>U: User sees success
  
```

Use Case ID:	43
Use Case Name:	Service Management - Frequency Change
Actors:	- Property Manager
Business Rules:	- User should be able to request a frequency change of their services

Description:	<ul style="list-style-type: none"> - This feature will allow the user to manage all their services with service providers such as canceling, or changing the frequency of service.
Preconditions:	<ul style="list-style-type: none"> - The user has successfully logged in to the website, was directed to the homepage and selected the Service Management feature from the homepage.
Functional	<ul style="list-style-type: none"> - Display the users current services - Allow the user to Cancel or change the frequency of an service - Notify the property manager of service changes request - Update service database with a pending status for the users service change
Postconditions:	<ul style="list-style-type: none"> - The service change request is sent to the service provider and the service rating is uploaded.
Nonfunctional	<ul style="list-style-type: none"> - The search bar should complete the search within 5 seconds. - Notification to service provider should be within 5 seconds - Updating service data should take within 3 seconds. - Comment text box should only allow 250 characters
Success Cases:	<ul style="list-style-type: none"> - User navigates to the service they want to change and the service change request is sent to the service provider. - System updates user services with pending service change requests status.
Failure Cases:	<ul style="list-style-type: none"> - User chooses a service and the system does not update their services with a pending service change request status. - The system takes longer than 5 seconds to notify service provider, the systems send an "Unsuccessful Service Change Request" error message - User service change update takes longer than 3 seconds, the system send an "Unable to update" error message



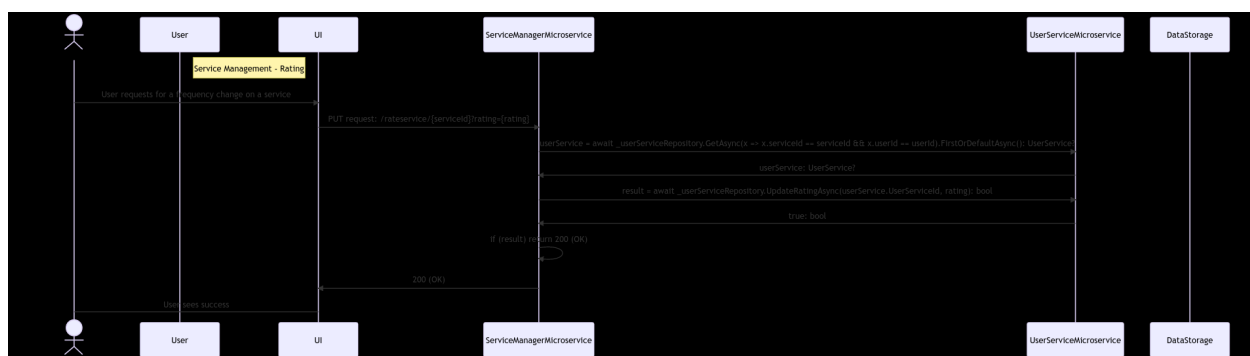
sequenceDiagram

```

actor U as User
Note right of User: Service Management - Frequency Change
participant UI
participant SMM as ServiceManagerMicroservice
participant UM as UserMicroservice
participant DS as DataStorage
U->>UI: User requests for a frequency change on a service
UI->>SMM: PUT request: /service/{serviceId}
SMM->>SMM: updatedService = HTTPContext.Request
SMM->>DS: _dbContext.Service.Update(updatedService): void
SMM->>SMM: result = await _dbContext.SaveChangesAsync() > 0
SMM->>SMM: if (result) return 200 (OK)
SMM->>UI: 200 (OK)
UI->>U: User sees success
  
```

Use Case ID:	44
Use Case Name:	Service Management - Rating
Actors:	- Property Manager
Business Rules:	- User must be a property manager with a known address.
Description:	- This feature will allow the user to rate a completed or closed service
Preconditions:	- The user has successfully logged in to the website, was directed

	to the homepage and selected the Rating feature from the homepage.
Functional	<ul style="list-style-type: none"> - Allow the user rate a service - Display a likert scale - Record user rating in a database - Property Managers can enter comments on the service provider.
Postconditions:	<ul style="list-style-type: none"> - The service request or change is sent to the service provider and the service rating is uploaded.
Nonfunctional	<ul style="list-style-type: none"> - The search bar should complete the search within 5 seconds. - Database should record user rating within 5 seconds. - Notification to property manager should be within 5 seconds - Notification to service provider should be within 5 seconds - Updating service data should take within 5 seconds. - Comment is limited to 240 words.
Success Cases:	<ul style="list-style-type: none"> - User is successfully able to rate the service that was performed. System updates the service provider's rating.
Failure Cases:	<ul style="list-style-type: none"> - The user was not able to rate a service - A likert scale was not displayed - User rating was not stored in a database - Property Managers were not able to enter comments on the service provider. - The search bar did not complete the search within 5 seconds. - Property manager did not get notified in 5 seconds - Service provider did not get notified in 5 seconds - Updating service data took more than 5 seconds. - Comment text was more than 240 words



```

sequenceDiagram
    actor U as User
    Note right of User: Service Management - Rating
    participant UI
    participant SMM as ServiceManagerMicroservice
  
```

```
participant USM as UserServiceMicroservice
participant DS as DataStorage
U->>UI: User requests for a frequency change on a service
UI->>SMM: PUT request: /rateservice/{serviceId}?rating={rating}
SMM->>USM: userService = await _userServiceRepository.GetAsync(x =>
x.serviceId == serviceId && x.userId == userId).FirstOrDefault():
UserService?
USM->>SMM: userService: UserService?
SMM->>USM: result = await
_userServiceRepository.UpdateRatingAsync(userService.UserServiceId,
rating): bool
USM->>SMM: true: bool
SMM->>SMM: if (result) return 200 (OK)
SMM->>UI: 200 (OK)
UI->U: User sees success
```