



**Ain-Shams University**  
**Faculty of Engineering**  
**Computer Engineering and Software**  
**Systems Program**  
**CSE221: Object Oriented Analysis**  
**Project name:** System of a gym

Ayman Hesham Mohamed	16P3037
Rowan Hazem Wagieh	16P3023
Yara Hossam Mohamed	16P3002

## **Brief Description**

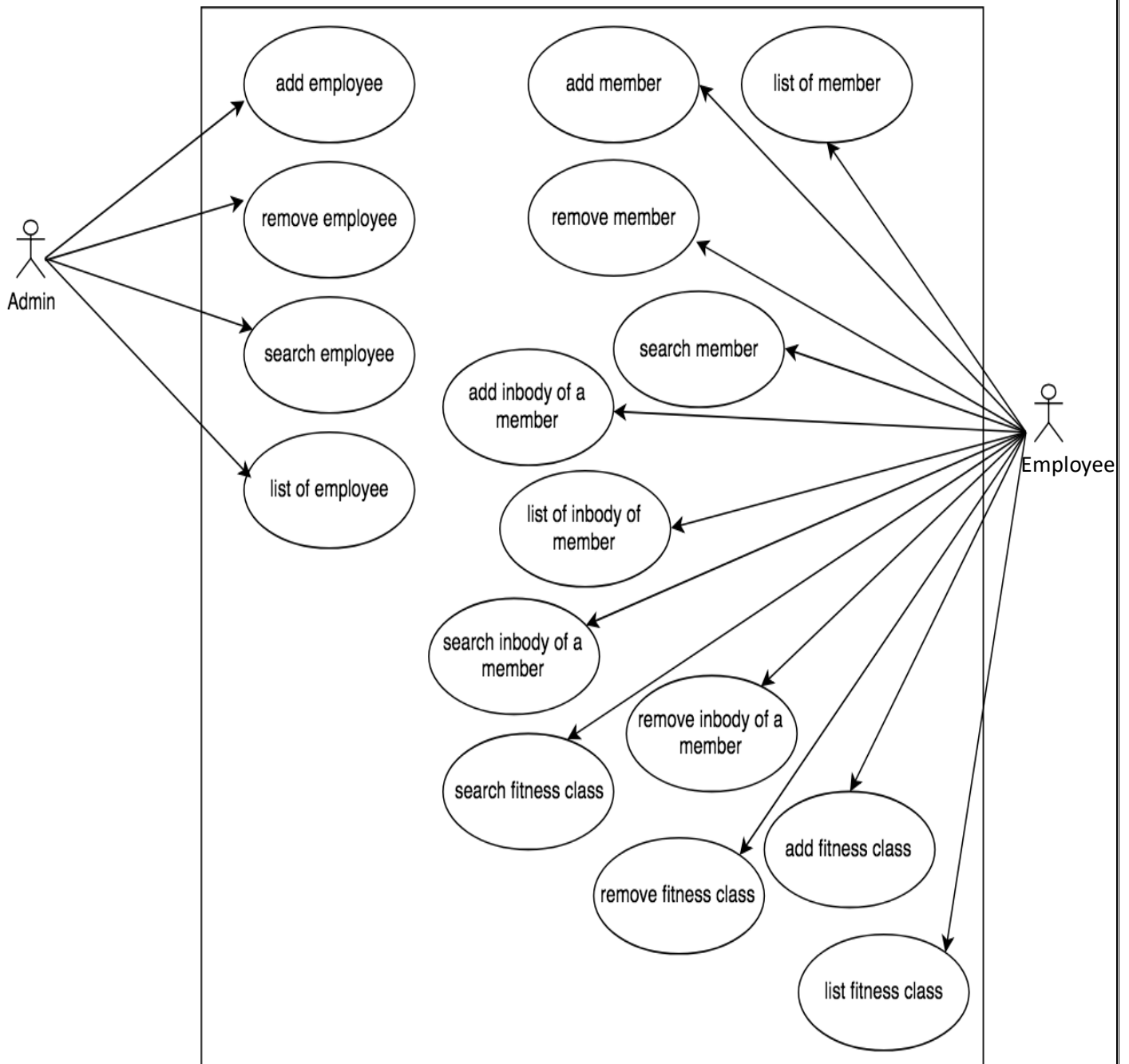
This project implement a java program for a system of a gym; once open the program, there is many use cases: add employee, member, fitness Class, InBody or removing, Searching, list on of them .. In this document we only list 8 of those use case which are: add member, employee, InBody and remove member, employee, InBody and searching for member, employee.

Assuming that the admin is able to add, remove, search or list employees. But the employee is able to add, remove, search, or list all members and fitness classes, inBody.

Also a use case diagram is mentioned to represent the relation between the actors and the use cases of the system and a description for each one of them.

Also a sequence diagram for each one and analysis class for each one representing: Boundry class, control class and entity class for each one.

## 1. Use Case Diagram:



## 2. Use Case Description:

### 1. Add employee:

Use Case Name	Add employee
Goal in context	A new employee apply to be added by the admin
Pre-Condition	The new employee doesn't exist in the system
Successful end condition	A new employee is added to the gym system
Failure End Condition	The employee is rejected
Primary actor	Admin
Secondary actor	None
Trigger	The admin see the new employee application and click at the add button to add the new employee
Main Flow	1-The admin clicks to Employee Jmenu 2-The admin click to the add button 3-The admin enter the new employee information 4-The system checks that it is not an old employee 5-A new employee is added to the system 6-The new employee information is saved in a file at the system
Extensions	4.1-The system discovers that the employee is old , displays "Repeated ID, Try again" and refuses to add the new employee.

	6.1-The system fails to save the information of the new employee.
Input	The new employee information
Output	None

## 2-Remove employee:

Use Case Name	Remove employee
Goal in context	The admin removes an old employee
Pre-Condition	The employee exists in the system
Successful end condition	The old employee is removed by the admin
Failure End Condition	The employee is not removed from the system
Primary Actor	Admin
Secondary Actor	None
Trigger	The admin click at the remove button at the system
Main Flow	1-The admin clicks at the Employee Jmenu 2-The admin clicks at the remove button 3-The admin enters the employee ID 4- The system checks that it is an old employee 5-The employee is removed from the system 6-The old employee information is deleted from the file
Extensions	4.1-The system discovers that the employee is not in the system and doesn't remove the employee from the system. 6.1-The system fails to remove the old employee information
Input	The employee ID

Output	None
--------	------

### 3. Search for employee:

Use case name	Search for employee
Goal in context	The admin searches for an specific employee
Pre-Condition	The employee exist in the system
Successful end condition	The admin have all the information of the employee
Failure end condition	The admin doesn't receive the employee information
Primary actor	Admin
Secondary actor	None
Trigger	The admin clicks at the search button and enters the employee ID
Main flow	1-The admin clicks at the Employee Jmenu 2-The admin clicks at the Search Button 3-The admin enters the employee ID 4-The system checks the employee ID 5-The system gets the employee information and displays it to the admin
Extensions	4.1-The system discovers that the employee ID does not exist at the system and displays "Not found" 5.1- The system fails to get the employee information

Input	The employee ID
Output	None

#### 4. Add member:

Use Case Name	Add member
Goal in context	A new member apply to be added by the employee
Pre-Condition	The new member doesn't exist in the system
Successful end condition	A new member is added to the gym system
Failure End Condition	The member is rejected
Primary actor	Employee
Secondary actor	None
Trigger	The employee see the new member information and click at the add button to add the new member
Main Flow	1-The employee clicks to Member Jmenu 2-The employee click to the add button 3-The employee enter the new member information 4-The system checks that it is not an old member 5-A new member is added to the system 6-The new member information is saved in a file at the system
Extensions	4.1-The system discovers that the member is old(exists at the system) , displays "Repeated ID, Try again" and refuses to add the new member.



	6.1-The system fails to save the information of the new member.
Input	The new member information
Output	None

## 5. Remove member:

Use Case Name	Remove member
Goal in context	The employee removes an old member
Pre-Condition	The member exists in the system
Successful end condition	The old member is removed by the employee
Failure End Condition	The member is not removed from the system
Primary Actor	Employee
Secondary Actor	None
Trigger	The employee click at the remove button at the system
Main Flow	1-The employee clicks at the Member Jmenu 2-The employee clicks at the remove button 3-The employee enters the member ID 4- The system checks that it is an old member 5-The member is removed from the system 6-The old member information is deleted from the file
Extensions	4.1-The system discovers that the member is not in the system and doesn't remove the member from the system. 6.1-The system fails to remove the old member information
Input	The member ID

Output	None
--------	------

#### 6. Search for member:

Use case name	Search for member
Goal in context	The employee searches for an specific member
Pre-Condition	The member exist in the system
Successful end condition	The employee have all the information of the member
Failure end condition	The employee doesn't receive the member information
Primary actor	Employee
Secondary actor	None
Trigger	The employee clicks at the search button and enters the member ID
Main flow	1-The employee clicks at the member Jmenu 2-The employee clicks at the Search Button 3-The employee enters the member ID 4-The system checks the member ID 5-The system gets the member information and displays it to the employee
Extensions	4.1-The system discovers that the member ID does not exist at the system and displays "Not found" 5.1- The system fails to get the member information

Input	The member ID
Output	None

#### 7. Add InBody:

Use Case Name	Add inbody
Goal in context	A new inbody to be added by the employee
Pre-Condition	The new inbody doesn't exist in the system
Successful end condition	A new inbody of an member is added to the gym system
Failure End Condition	The inbody is rejected
Primary actor	Employee
Secondary actor	None
Trigger	The employee see the new inbody information and click at the add button to add the new inbody of member
Main Flow	1-The employee clicks to Inbody Jmenu 2-The employee click to the add button 3-The employee enter the new inbody of member information 4-The system checks that it is not an old inbody of member 5-A new inbody of member is added to the system 6-The new inbody of member information is saved in a file at the system

Extensions	<p>4.1-The system discovers that the inbody of the member is old(exists at the system) , displays “Repeated ID, Try again” and refuses to add the new member.</p> <p>6.1-The system fails to save the information of the new inbody of the member.</p>
Input	The new inbody information
Output	None

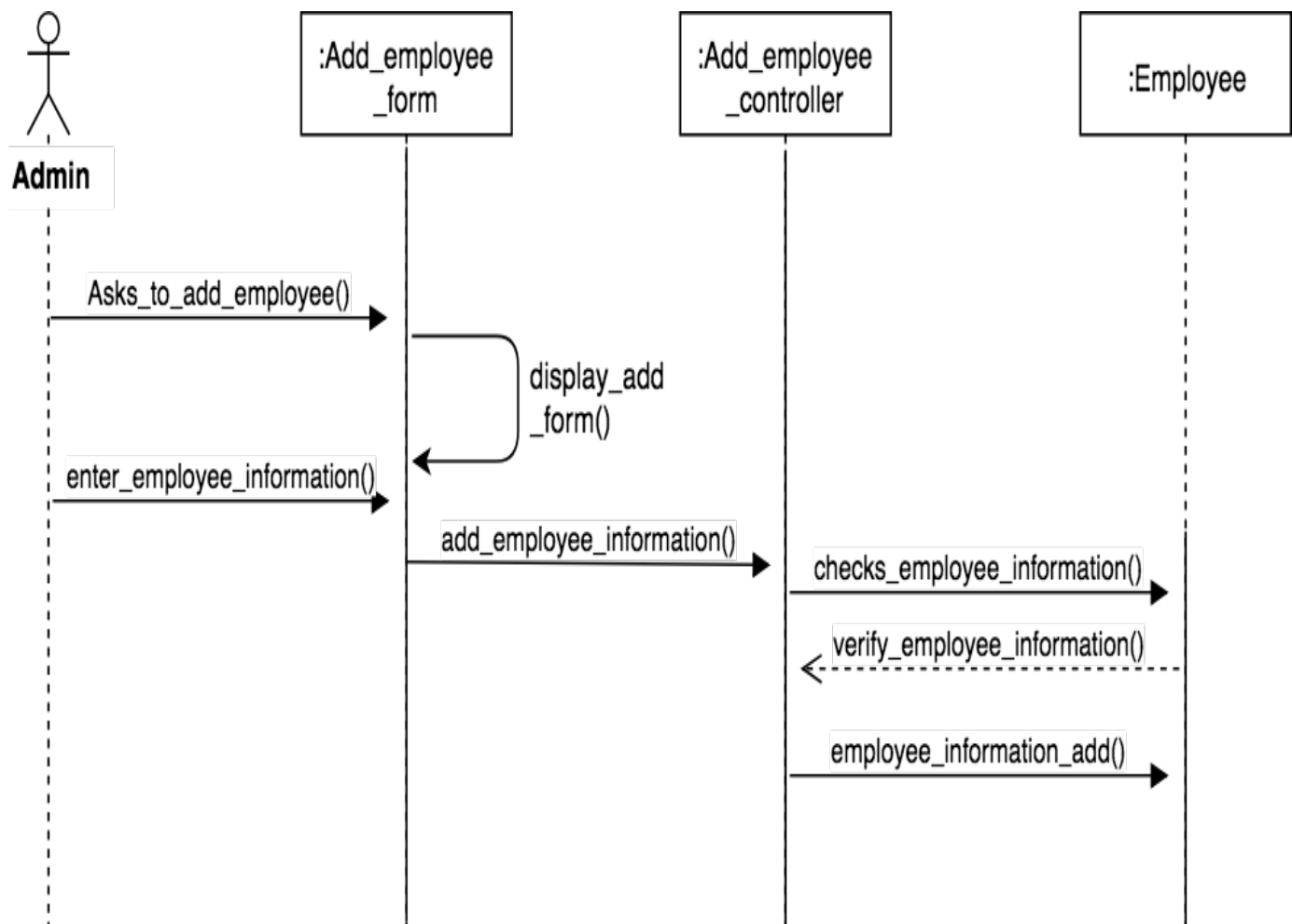
#### 8. Remove inbody:

Use Case Name	Remove inbody of a member
Goal in context	The employee removes an old inbody of a member
Pre-Condition	The inbody of the member exists in the system
Successful end condition	The old inbody of the member is removed by the admin
Failure End Condition	The inbody of the member is not removed from the system
Primary Actor	employee
Secondary Actor	None
Trigger	The employee click at the remove button at the system
Main Flow	1-The employee clicks at the Inbody Jmenu 2-The employee clicks at the remove button 3-The employee enters the inbody ID of the member 4- The system checks that it is an old inbody 5-The inbody is removed from the system 6-The old inbody of the member information is deleted from the file
Extensions	4.1-The system discovers that the inbody of the member is not in the system and doesn't remove the inbody from the system.

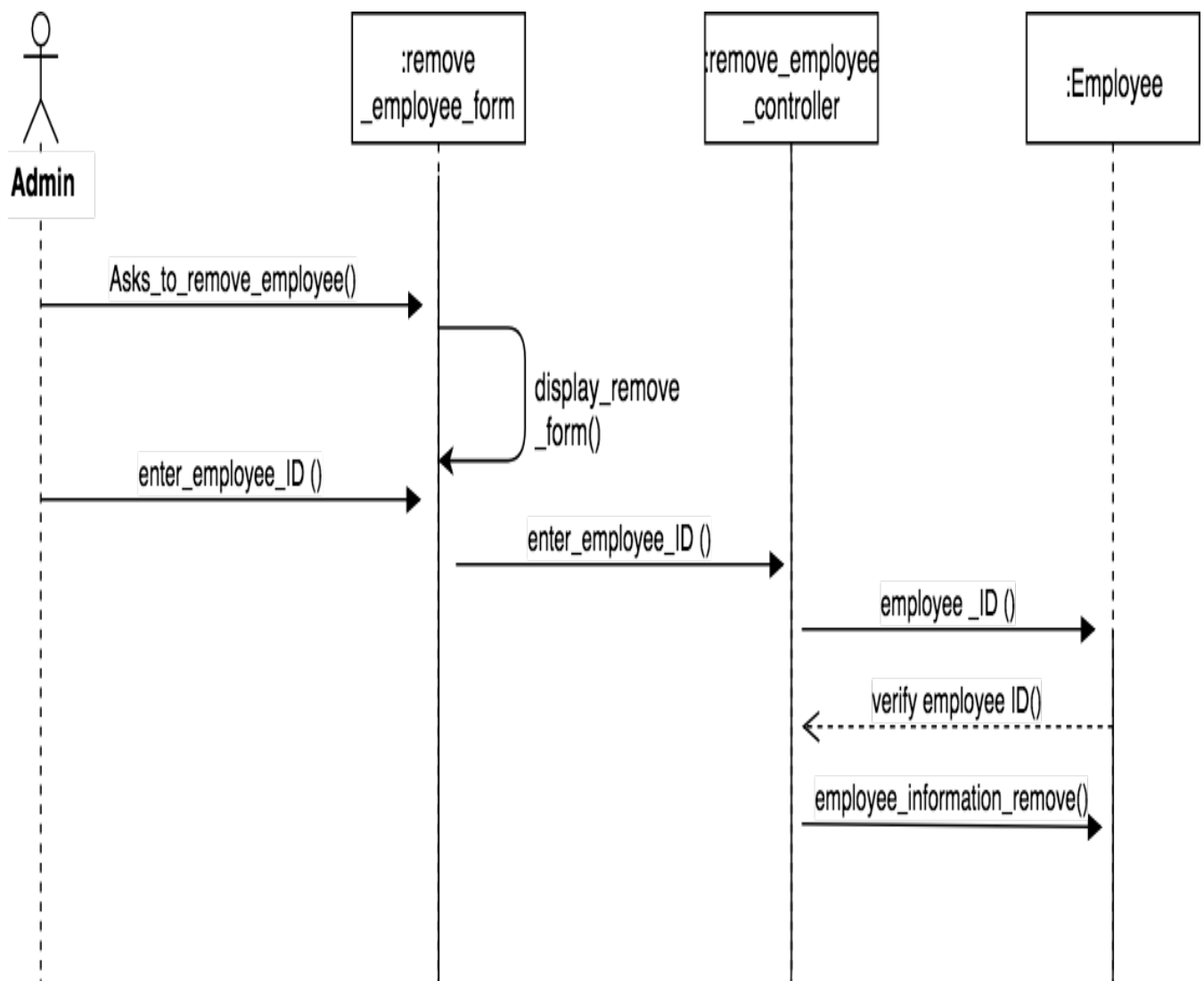
	6.1-The system fails to remove the old inbody of the member information
Input	The inbody ID of the member
Output	None

### 3. Time Sequence:

**Use case:** add employee

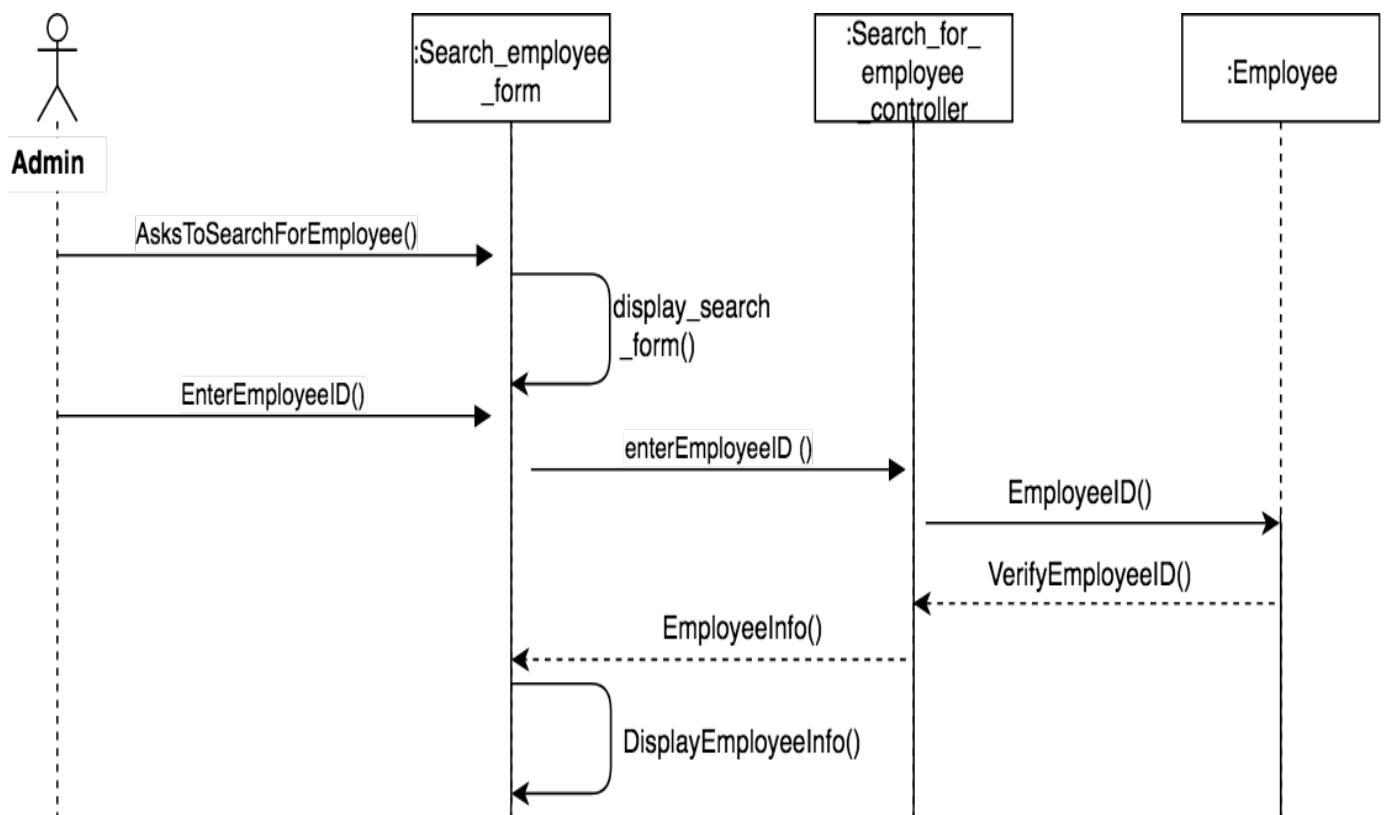


**Use case:** Remove employee

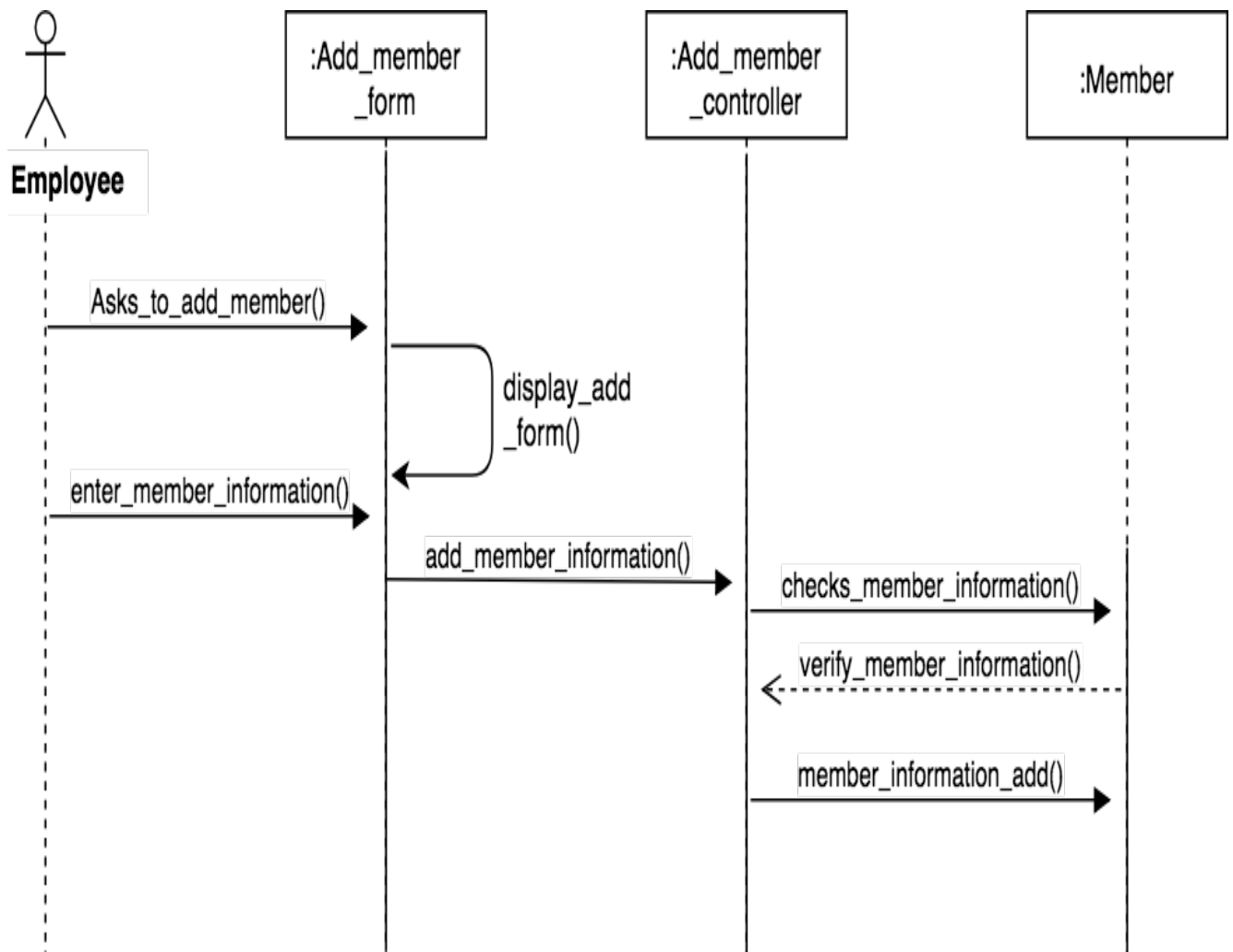




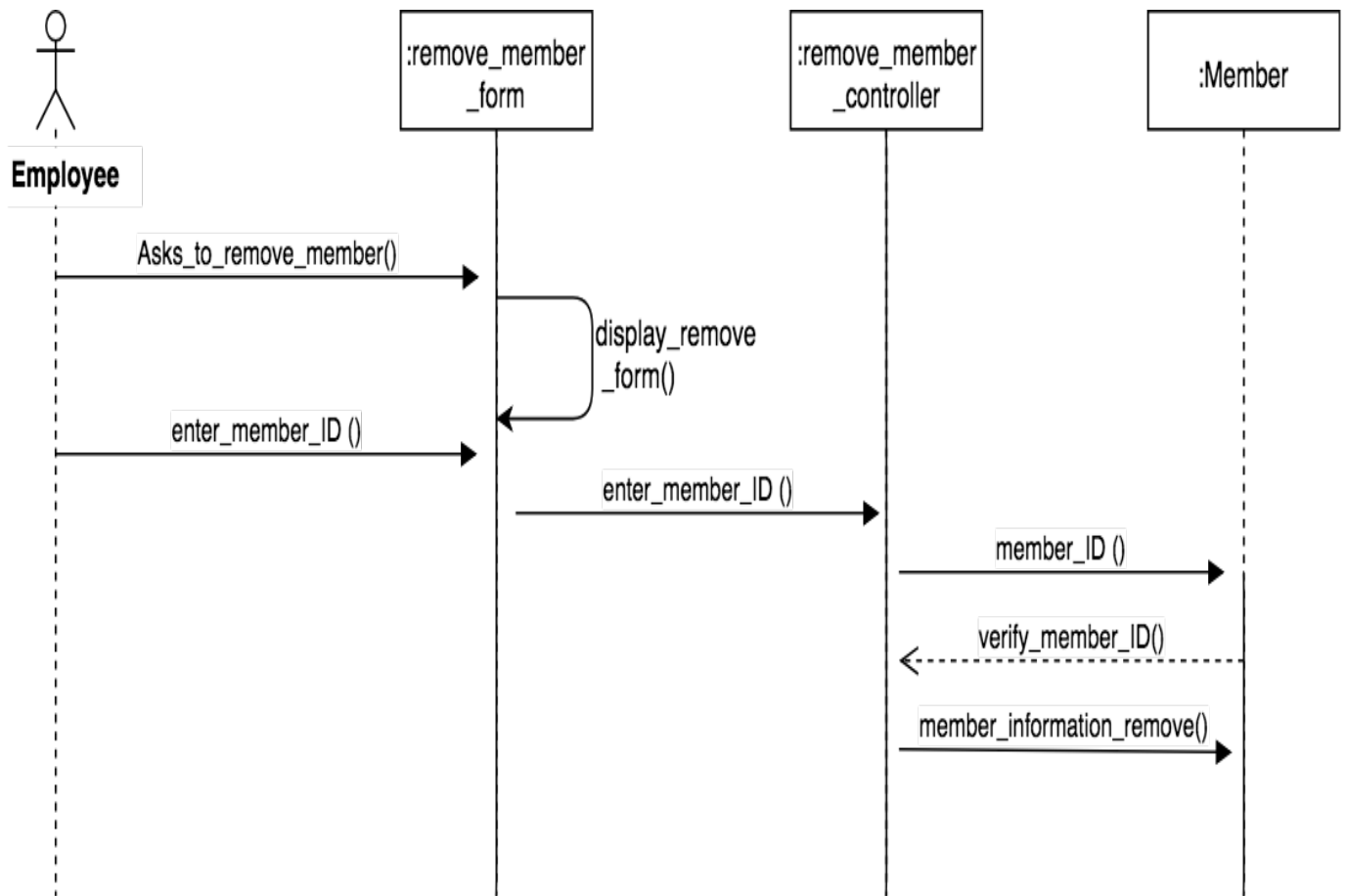
**Use case:** Search for employee



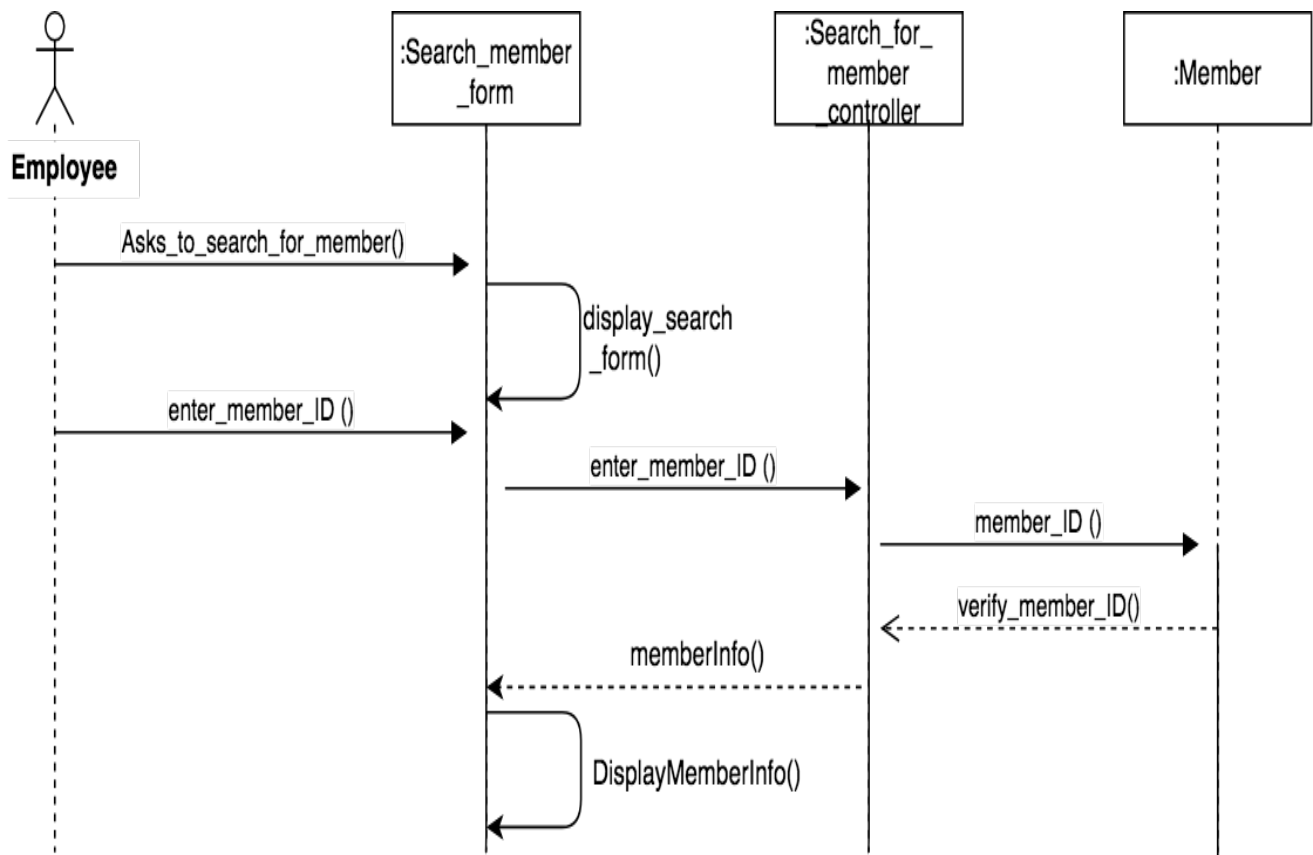
**Use case:** Add member



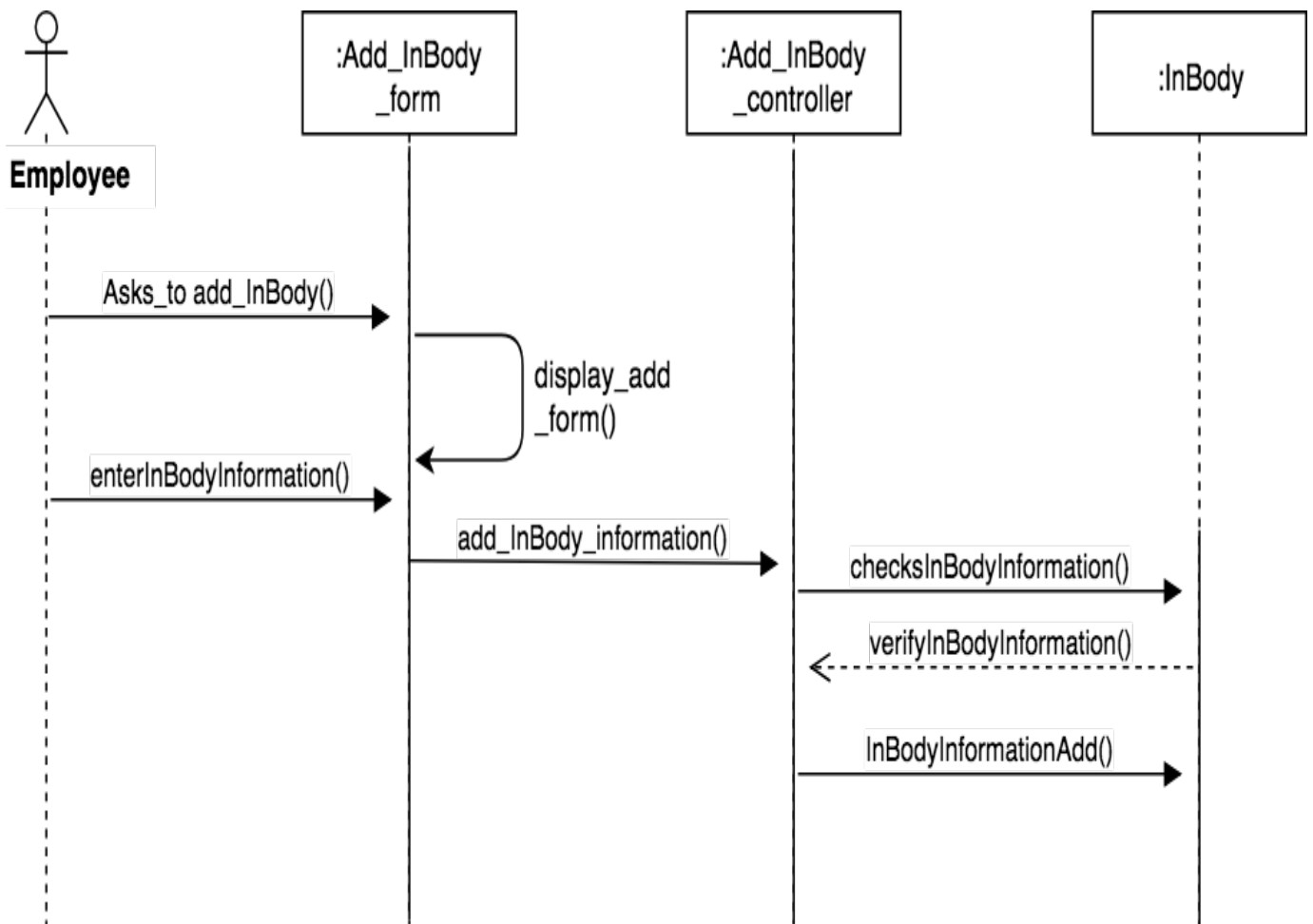
**Use case:** Remove member



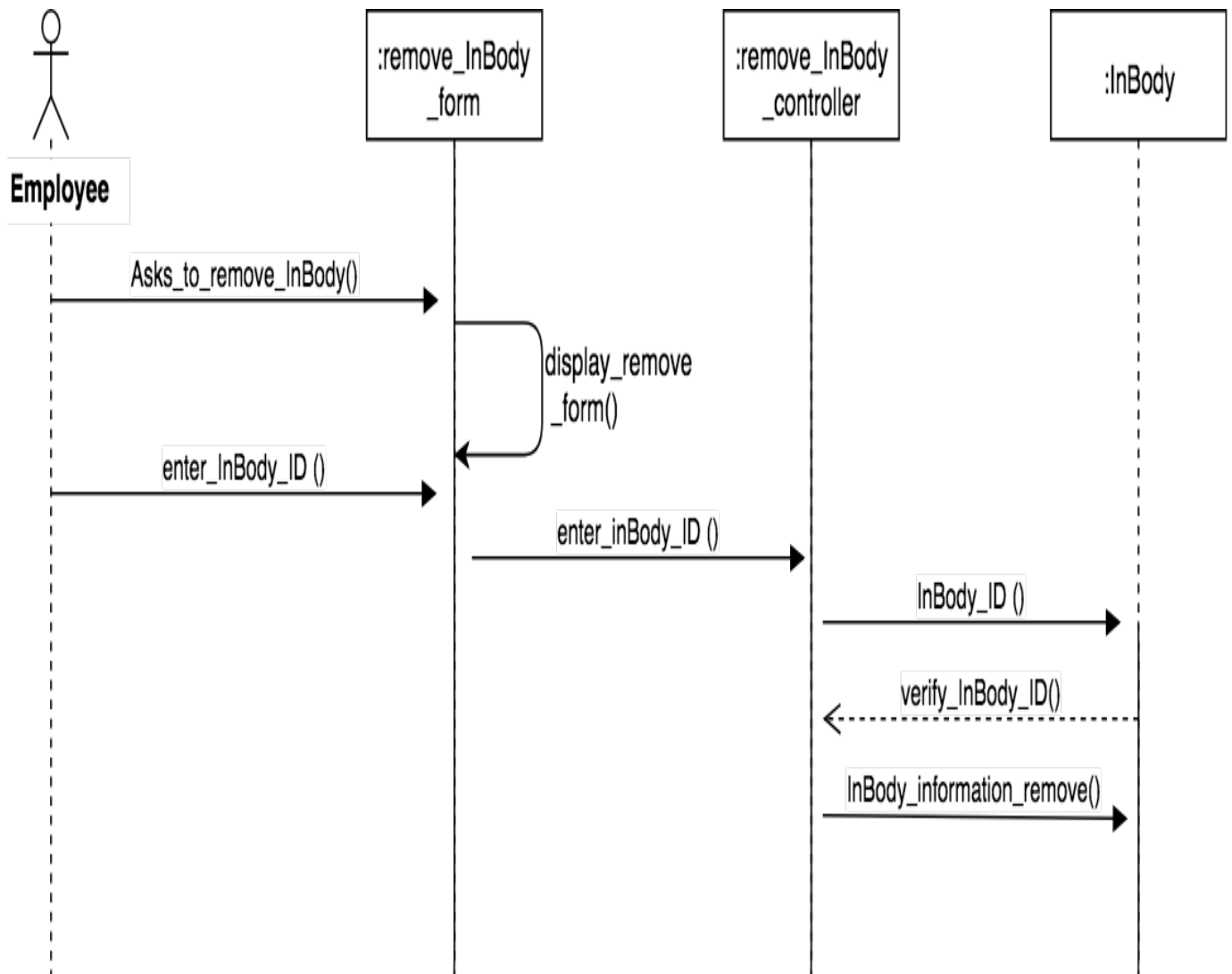
**Use case:** Search for member



**Use case:** Add InBody



**Use case:** Remove InBody



## 2. Analysis Classes

- Use Case name: Add member

Entity: Member

attributes (ID, Name, Address, Birthdate, Regdate)

Boundary: Add\_member\_form

Controller: Add\_member\_Controller

- Use Case name: remove member

Entity: Member

attributes (ID, Name, Address, Birthdate, Regdate)

Boundary: remove\_member\_form

Controller: remove\_member\_Controller

- Use Case name: Search for member

Entity: Member

attributes (ID, Name, Address, Birthdate, Regdate)

Boundary: Search\_for\_member\_form

Controller: Search\_for\_member\_Controller

- Use Case name: Add employee

Entity: Employee

attributes (Name ,ID ,Birthdate ,Salary ,Address,Date of employment,Age)

Boundary: Add\_employee\_form

Controller: Add\_employee\_controller

- Use Case name: Remove employee

Entity: employee

attributes (Name ,ID ,Birthdate ,Salary ,Address,Date of employment,Age)

Boundary: remove\_employee\_form

Controller: remove\_employee\_Controller

- Use Case name: Search for employee

Entity: employee

attributes (Name ,ID ,Birthdate ,Salary ,Address,Date of employment,Age)

Boundary: search\_for\_employee\_form

Controller: search\_for\_employee\_Controller

- Use Case name: add InBody

Entity: InBody

attributes (InbodyID,Height ,Weight, Age, bodyfatmass, BMI, PBF, Gender, testdate)

Boundary: add\_inbody\_form

Controller: add\_inbody\_Controller

- Use Case name: remove InBody

Entity: InBody

attributes (InbodyID,Height ,Weight, Age, bodyfatmass, BMI, PBF, Gender, testdate)

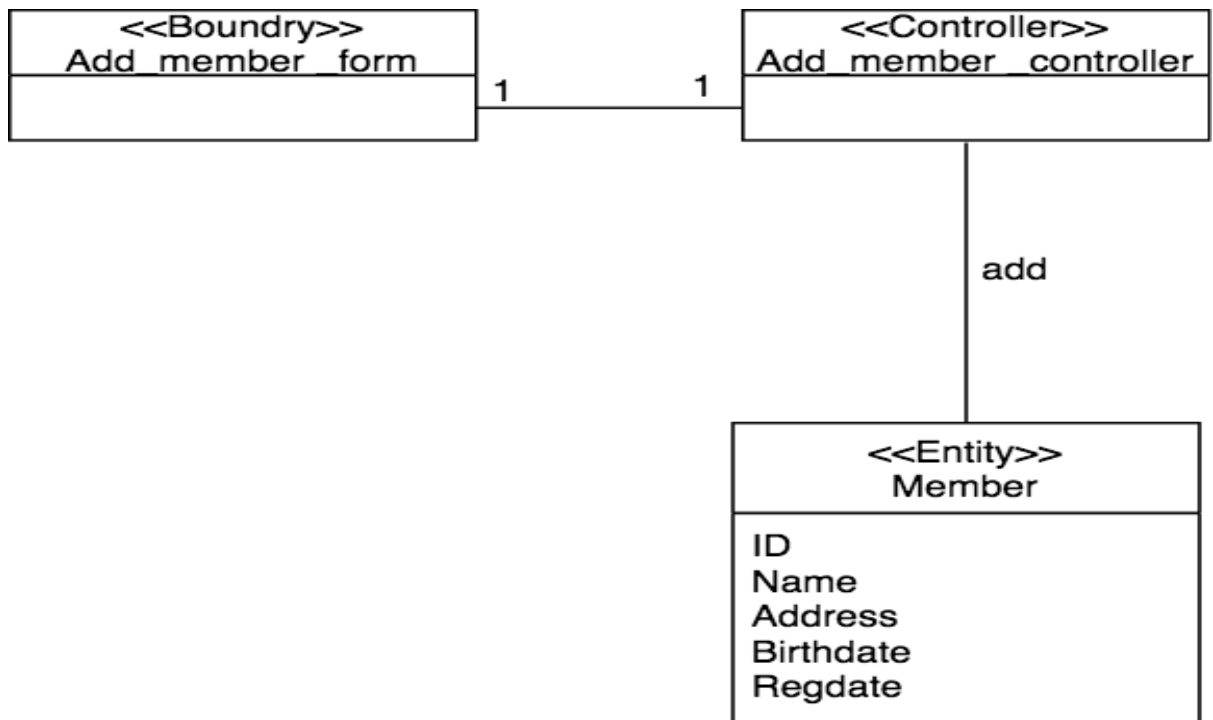
Boundary: remove\_InBody\_form

Controller: remove\_InBody\_Controller

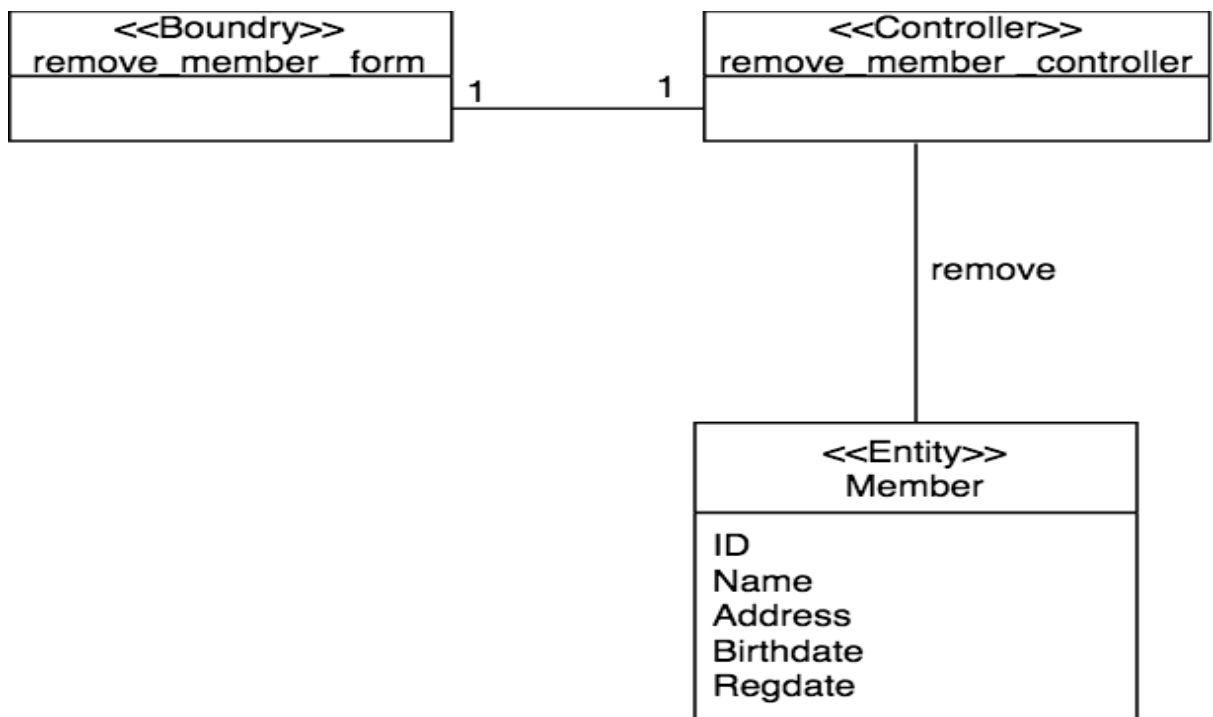


### 3. Analysis Model

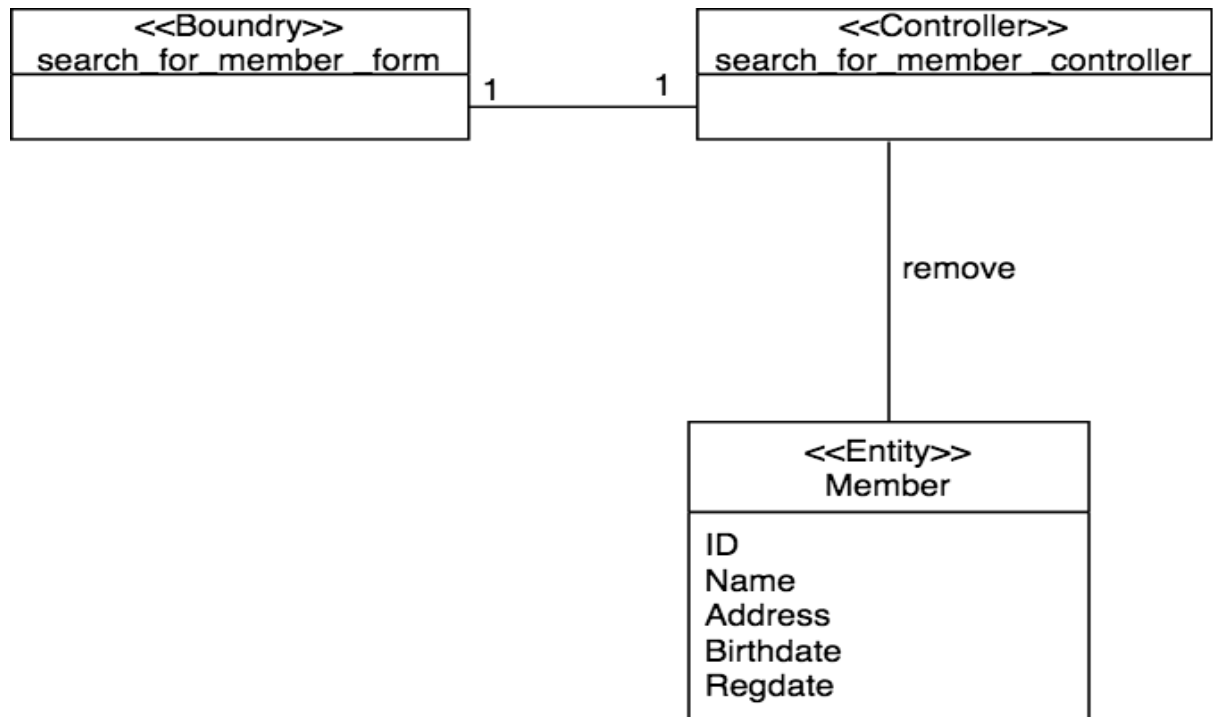
**Use case:** add member



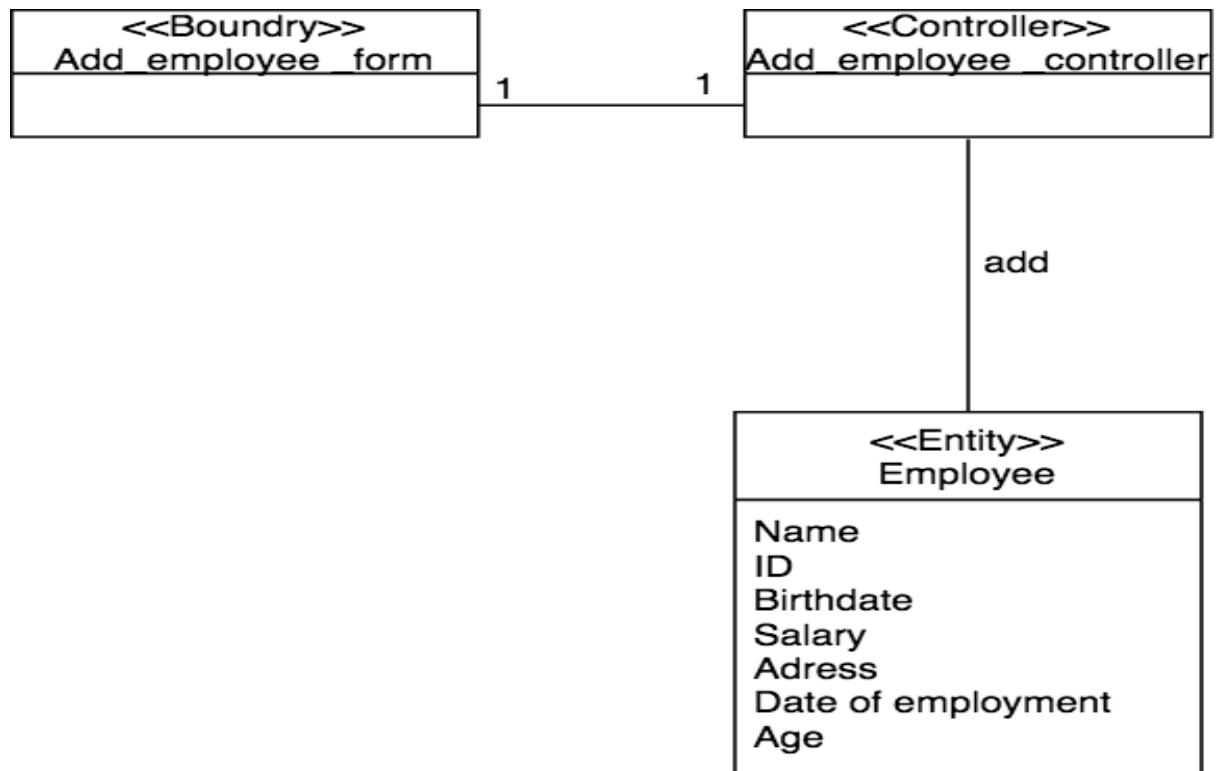
**Use case:** Remove member



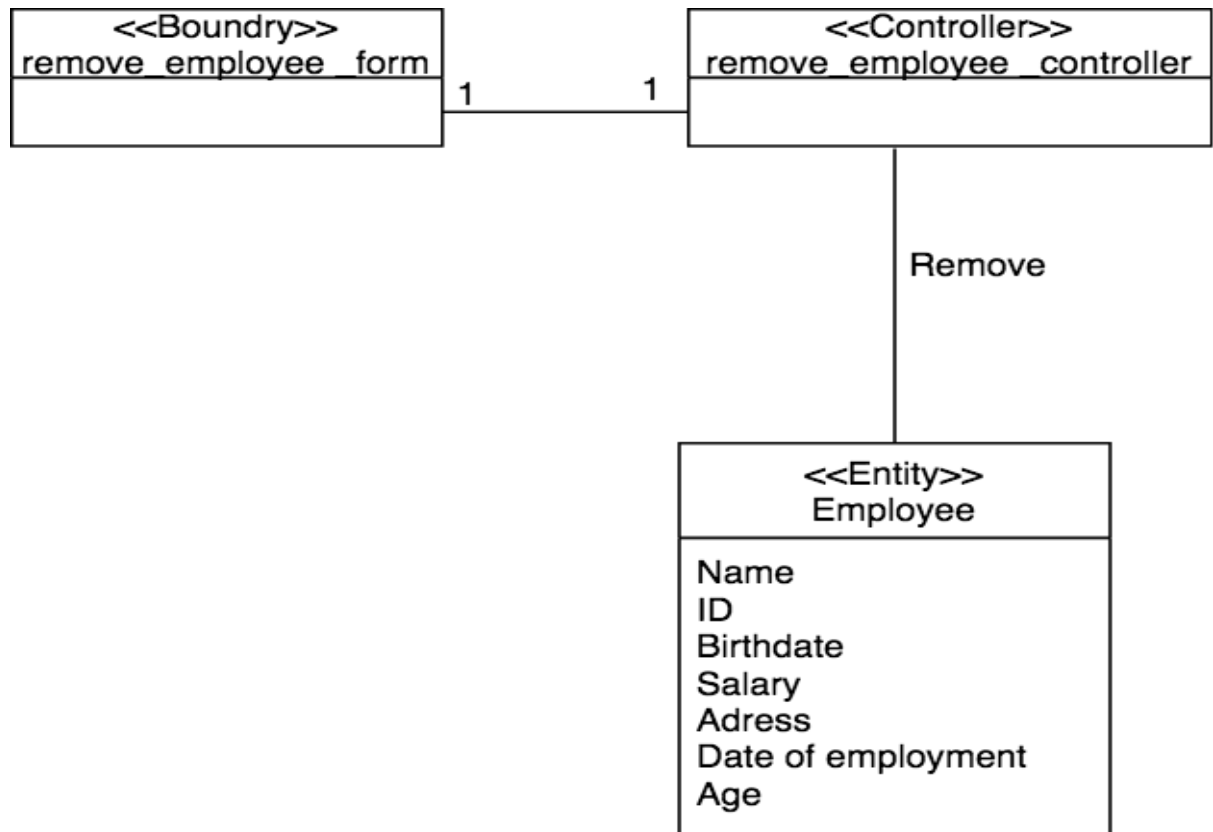
**Use case:** Search for member



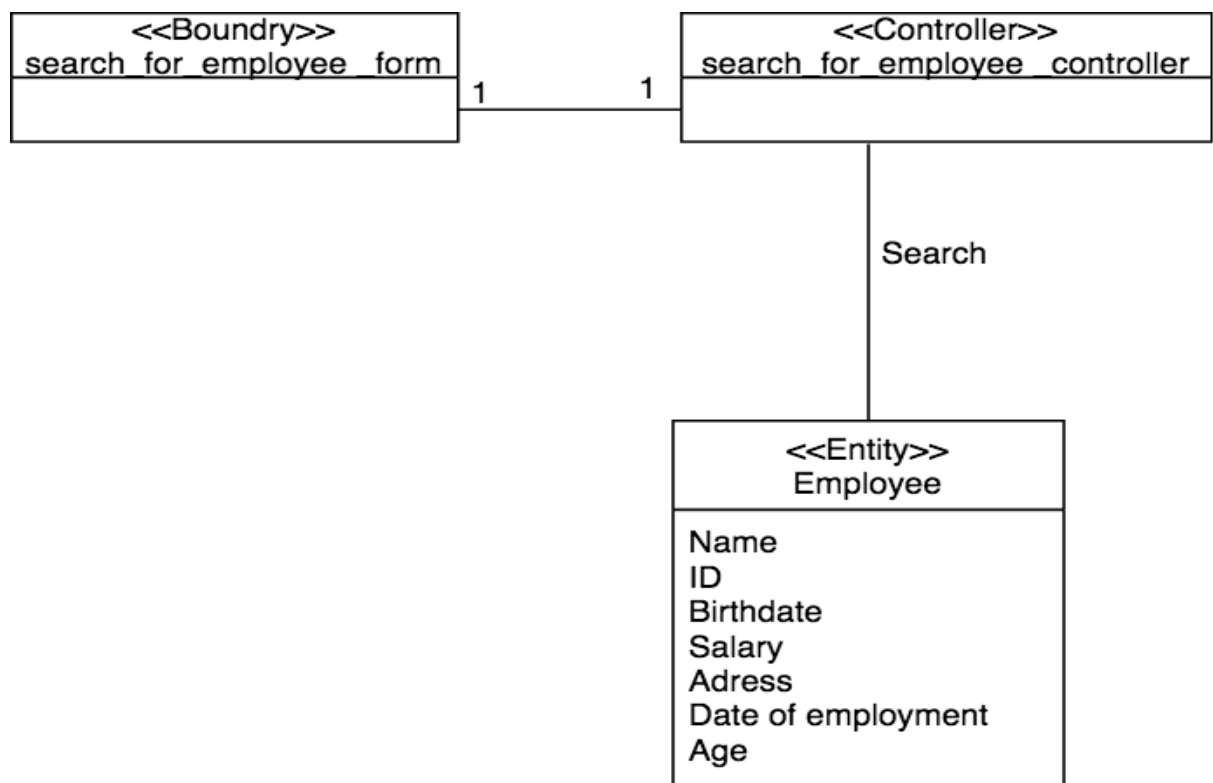
**Use case:** add Employee



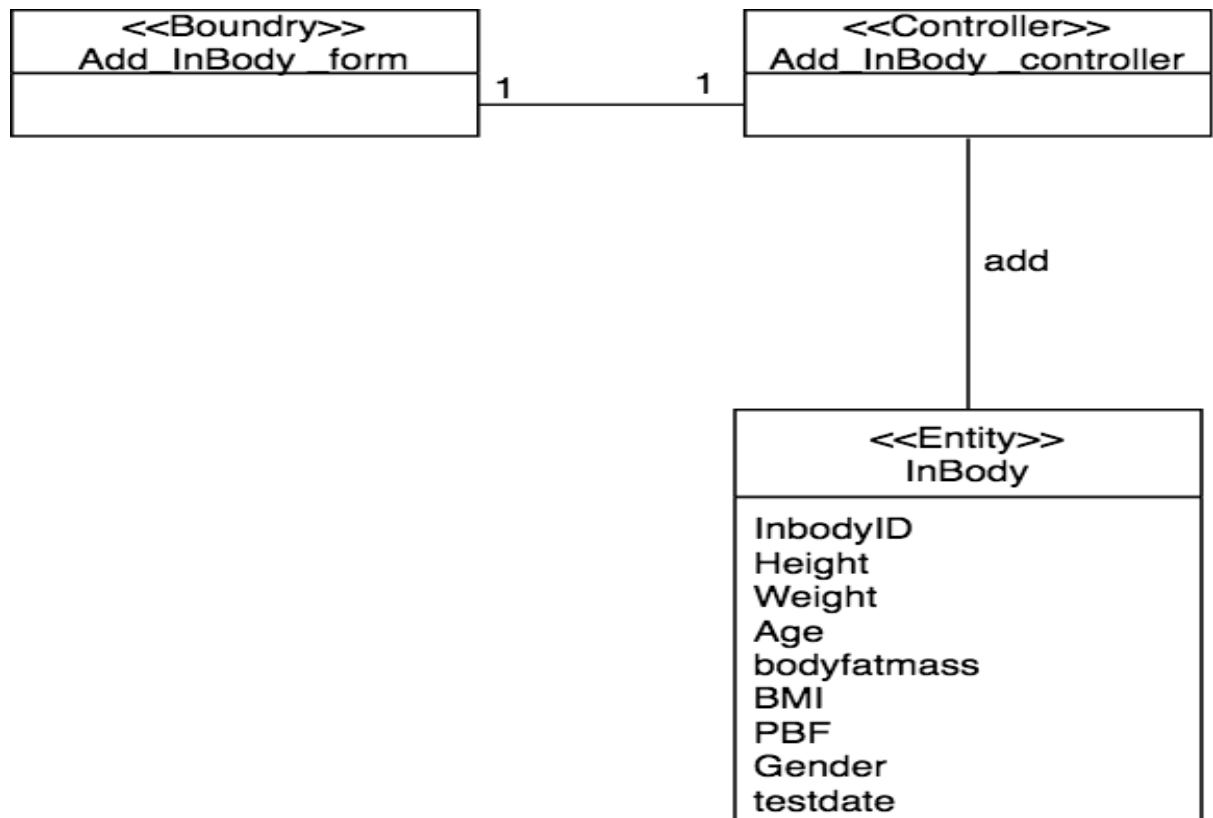
**Use case:** Remove employee



**Use case:** Search for employee



**Use case:** Add InBody



**Use case:** Remove InBody

