**Gym Management  
System**

**Phase 1: Software Engineering Project**

**Submitted To**

Dr. Reham Adel

TA. Mai Mohamed

**Submitted By**

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Student Name** | **Section Number** |
| 42110049 | Adham Hassan Sayed Elagaty | B1 |
| 42110074 | Ibrahim Mostafa | B1 |
| 42110505 | *Abdelrahman Khaled Aboarab* | B1 |
| 42110103 | Ahmed Mostafa Mahmoud | A2 |
| 42110244 | Youssef Omar Elzgzouk | A2 |
| 42110339 | Ahmed Ashraf Sayed | B3 |
| 42110040 | Ziad Ahmed Hegazy | B3 |

**Contact Email:** 42110049.adham@acu.edu.eg

# **Introduction**

In response to the evolving needs of gym management, this project introduces a cutting-edge Gym Management System (GMS). The GMS is poised to revolutionize the way gym facilities handle employee administration, member interactions, communication, and overall operational efficiency.

The primary objective of the Gym Management System is to offer a user-friendly and efficient platform that caters to the diverse requirements of both administrators and gym members. From creating and managing employee profiles to facilitating seamless member subscriptions and class enrollments, the GMS aims to streamline the entire gym management process.

The system places a strong emphasis on communication and engagement, allowing gym staff to effortlessly post announcements and members to subscribe to private trainers, gym packages, and specific months. Advanced search functionalities empower administrators to quickly access specific information, ensuring an optimal user experience.

To fortify security, the system implements robust encryption measures, safeguarding user data and ensuring protection against unauthorized access. Usability is a key focus, with a commitment to reducing task completion times and making the software inherently user-friendly.

Availability is paramount, and the system aims to be accessible 24/7, ensuring uninterrupted service for both administrators and gym members. Furthermore, the system is designed with mutability in mind, ensuring easy maintenance and updates to meet evolving organizational needs.

In summary, the Gym Management System is positioned to redefine gym administration by providing a comprehensive, secure, and user-friendly solution. Through its diverse functionalities, the system aims to empower gym facilities to efficiently manage their operations, enhance member experiences, and adapt to the dynamic demands of the fitness industry.

* 1. **Purpose**

The purpose of this document is to outline the functional and non-functional requirements for the development of a comprehensive fitness management system. The system aims to facilitate the management of employee profiles, member information, subscriptions, classes, announcements, and various administrative tasks within a fitness organization.

* 1. **Scope**

The scope of the fitness management system includes the following key features and functionalities:

Employee Management: Create, view, and search employee profiles. Allow administrators to add employees and modify their roles or status.

Member Management: Enable members to create profiles with personal information. Provide search functionalities for members.

Announcement and Communication: Allow employees to post announcements. Enable administrators to view, modify, and delete announcements.

Subscription Management: Provide functionalities to handle automatic payments, view payment records, and search for specific payments. Allow members to subscribe to private trainers, packages, and specific months.

Package and Offer Management: Allow administrators to add, modify, and search for packages and month offers. Enable members to subscribe to packages and specific months.

Trainer Management: Permit administrators to add new trainers to the system. Allow members to subscribe to private trainers.

Email Management: Enable administrators to view email information, save details about email send operations, and automatically send emails for subscription or profile creation. Provide search functionalities for specific emails.

Class Management: Allow administrators to add, view, and search for classes. Provide lists of available and open classes. Enable members to subscribe to classes.

Attendance and Subscription Tracking: Provide functionalities to calculate class and month attendance for members. Allow updating freeze periods, viewing all package subscriptions, and updating the number of invitations associated with a package.

* 1. **Definitions, acronyms, and abbreviations**
* FR: Functional Requirement
* NFR: Non-Functional Requirement
* must: Indicates a mandatory requirement
* want: Indicates a desired, but not mandatory, requirement

# **Requirements**

* 1. **Functional Requirements**
     1. Create Member Profile

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR001 | Allows employees to create profiles for members with personal information. | must |  |

* + 1. Search for Member

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR002 | Enables searching for a member within the system. | must |  |

* + 1. Add Employee

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR003 | Only accessible by administrators, it allows them to add new employees to the system. | must |  |

* + 1. View Employees

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR004 | Permits administrators to view employees on the system. | must |  |

* + 1. Search Employee

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR005 | Enables administrators to search for employees within the system. | must |  |

* + 1. Change Employee Status

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR006 | Allows the modification of employee roles or status (e.g., active, inactive) by administrators. | must |  |

* + 1. Add Announcement Post

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR007 | Enables employees to create and post announcements. | must |  |

* + 1. Record Payments

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR008 | Make Function to record Automatic payments subscriptions. | must |  |

* + 1. View All Payments

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR009 | Enables administrators to view All Payments Records. | must |  |

* + 1. Search About Payment

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR010 | Enables administrators to searching for specific payments using their unique identifiers. | must |  |

* + 1. Subscribe Private Trainer

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR011 | Allows members to subscribe to private trainers by Employee. | must |  |

* + 1. Add Package

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR012 | Allows the addition of new packages by administrators. | must |  |

* + 1. Change Package Status

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR013 | Permits administrators to enable or disable packages. | must |  |

* + 1. Search Package

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR014 | Allows searching for specific packages using their unique identifiers. | must |  |

* + 1. View All Package

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR015 | Provides a list of all available packages. | must |  |

* + 1. Subscribe Package

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR016 | Enables members to subscribe to packages by Employee. | must |  |

* + 1. Add Month Offers

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR017 | Allows the addition of new month offers by administrators. | must |  |

* + 1. View All Months Offer

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR018 | Provides a view of all available offers for the month. | must |  |

* + 1. Search for Month Offer

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR019 | Enables searching for specific month offers using their unique identifiers | must |  |

* + 1. Subscribe Month

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR020 | Allows members to subscribe to specific months by Employee. | must |  |

* + 1. Add Trainer to System

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR021 | Permits administrators to add new trainers to the system | must |  |

* + 1. Subscribe Private Trainer

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR022 | Allows members to subscribe to private trainers by employee. | must |  |

* + 1. View Emails

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR023 | Allows admin to view emails info. | must |  |

* + 1. Save Information of Sending Email

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR024 | Automatic save details about email send operations. | must |  |

* + 1. sending emails

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR025 | Automatic sending emails when subscription or create new profile. | want |  |

* + 1. Search Email

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR026 | Enables searching for specific emails using their unique identifiers. | must |  |

* + 1. Add Class

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR027 | Allows the addition of new classes by administrators. | must |  |

* + 1. View All Classes

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR028 | Provides a list of all classes. | must |  |

* + 1. View Available Classes

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR029 | Shows classes that are currently open for enrollment. | must |  |

* + 1. Search for Classes

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR030 | Enables searching for specific classes. | must |  |

* + 1. Subscribe Class

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR031 | Allows members to subscribe to classes by employee. | must |  |

* + 1. Post Announcement

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR032 | Allows Employee to post new announcements. | must |  |

* + 1. Delete Announcement

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR033 | Permits administrators to remove announcements. | must |  |

* + 1. View All Announcements

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR034 | Provides a comprehensive list of all announcements. | must |  |

* + 1. View All Class Subscriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR035 | Provides a list of all class subscriptions by administrators. | must |  |

* + 1. Calculate Number of Class Attendance

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR036 | Enables tracking and calculation of member class subscription attendance. | must |  |

* + 1. View All Month Subscriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR037 | Provides a list of all month subscriptions. | must |  |

* + 1. Calculate Number of Month Attendance

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR038 | Enables tracking and calculation of member month attendance. | must |  |

* + 1. Update Freeze Number

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR039 | Allows updating the number of freeze periods | must |  |

* + 1. View All Package Subscriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR040 | Provides a list of all package subscriptions. | must |  |

* + 1. Update Number of Invitations

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR041 | Allows updating the number of invitations associated with a package. | must |  |

* + 1. Take Member Attendance

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR042 | Allows marking attendance for members. | must |  |

* + 1. Show All Attendance of Member

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR043 | Provides a list of all class attendance show by administrators. | must |  |

* + 1. Take Trainer Attendance

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR044 | Allows marking attendance for trainer. | must |  |

* + 1. Show All Attendance of Trainer

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| FR045 | Provides a list of all class attendance show by administrators. | must |  |

* 1. **Non Functional Requirements**
     1. Security

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| NFR001 | Security refers to the protection of the system and its data from unauthorized access or malicious attacks. One measure of security is the encryption strength, by encrypt users date by there id. | must |  |
| NFR002 | By Doing encryption must do function to decryption | must |  |

* + 1. usability

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| NFR003 | Usability focuses on the ease of use and user satisfaction when interacting with the system. Task completion time is a measure of usability that quantifies the time taken by users to complete specific tasks within the system. It is typically measured in seconds, and a lower task completion time indicates better usability and make this software user Frendly. | must |  |

* + 1. Availability

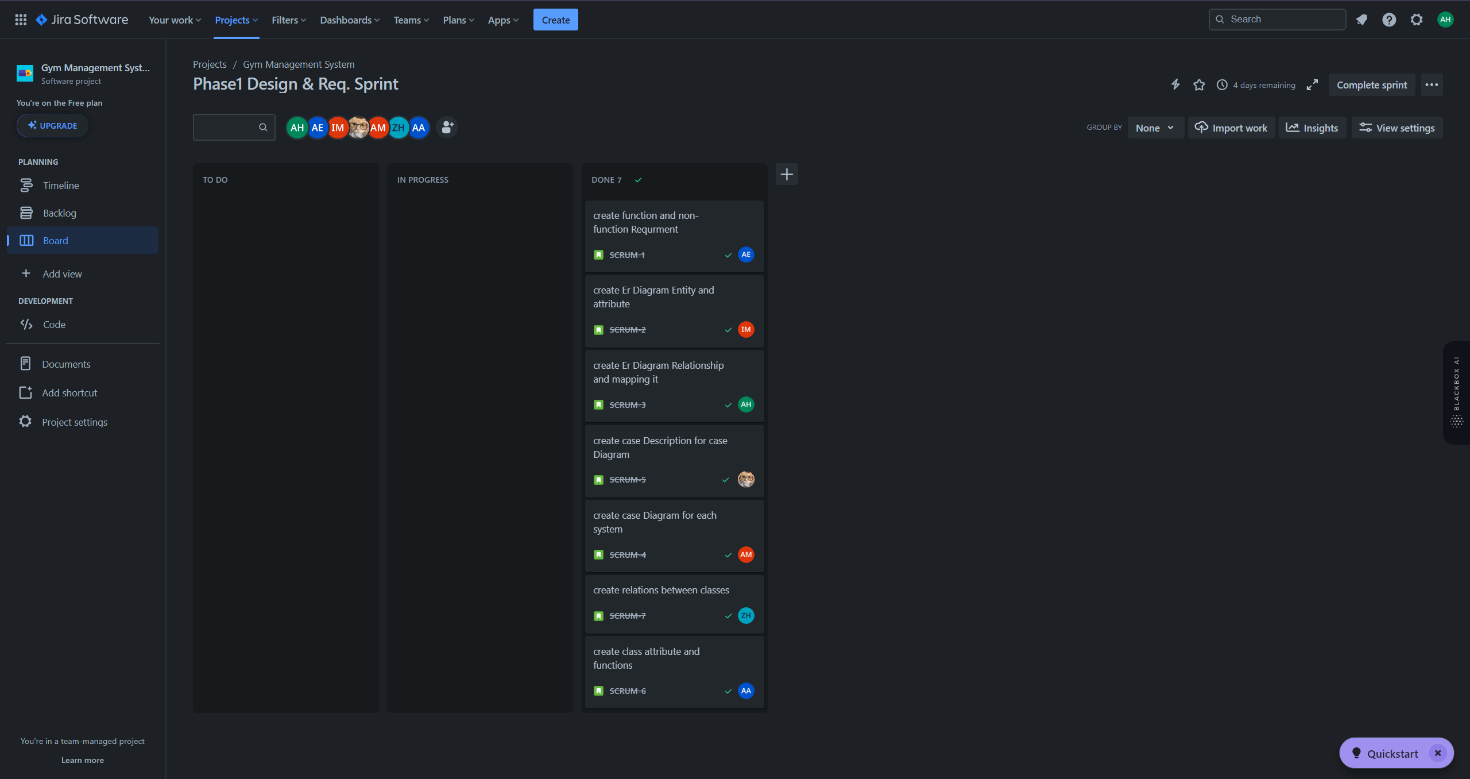
|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| NFR004 | The system should be available to its users 24/7. This can be measured by conducting regular availability tests and calculating the percentage of system downtime. | must |  |

* + 1. Maintainability

|  |  |  |  |
| --- | --- | --- | --- |
| **code** | **Requirements statement** | **Must/want** | **Comments** |
| NFR005 | easy to maintain and update. This can be measured by assessing the code quality | must |  |

# **Used Technologies**

* **Jira**: Is a website to make a board, put all tasks on it and assign each task to each member. We used it to manage our project and tasks upon the team members by organizing the tasks, creating sprints, and keeping track of our progress.



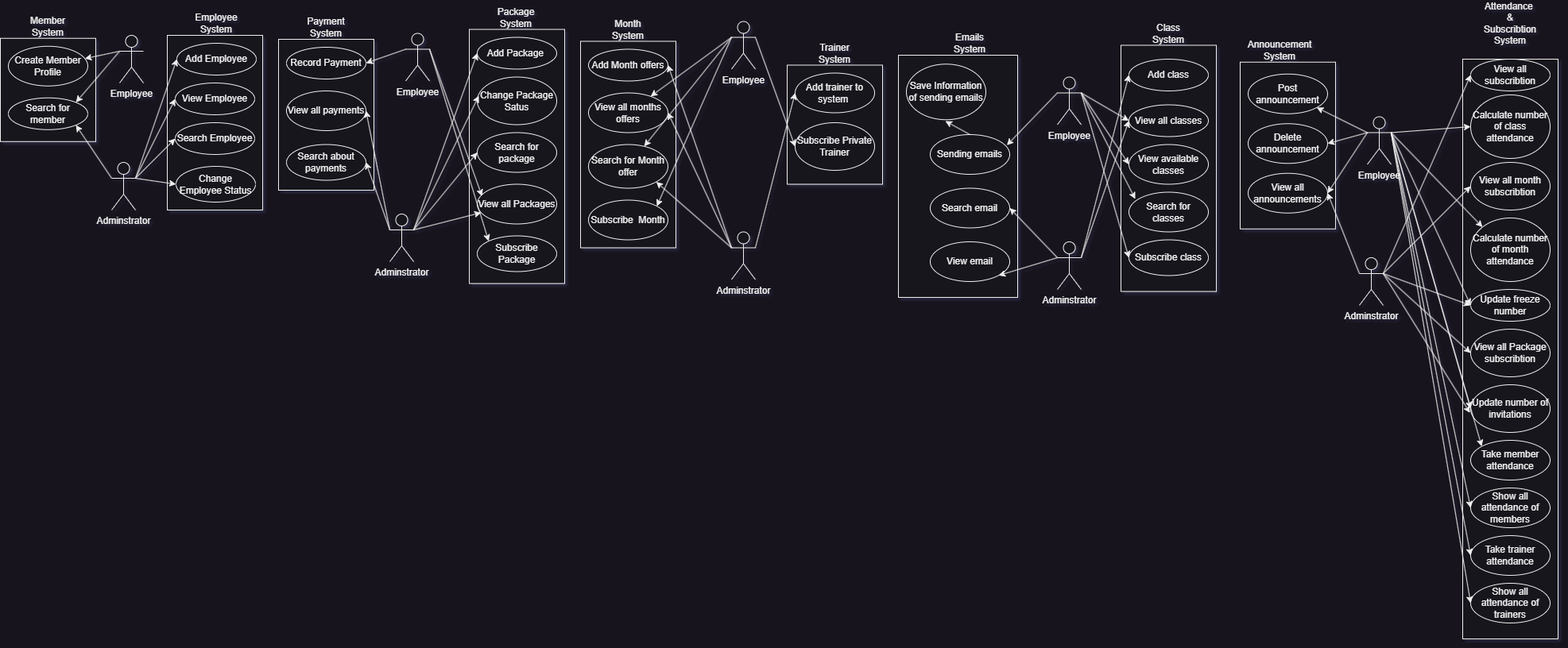
* **GitHub repository link**: This repository is made to make it easy for our team to communicate with each other. Every member will upload his copy of code with his modification to the code, so every member will always know the updates. This repository will help avoiding and reducing the reasons of errors.

https://github.com/AdhamElagaty/gym\_mangement\_system

* **Frontend technology:**
  + Desktop Application with C#
  + C# framework is .Net Windows form Application
* **Backend technology:**
  + MySql Database.
  + Write SQL Query in C#.

# **System Models**

* 1. **Use Case Model**

****

* + **Use Case Description Tables**

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Member System.Diagram#1.used-case#1.1 | |
| Use Case Name: | Create Member Profile | |
| Actors: | Employee | |
| Overview: | The employee creates a profile for member to assign information about member (e.g. Name - Phone- Address) | |
| Related use cases: |  | |
| Event(Stimulus): | **User Action** | **System Action** |
| Employee creates profile for member | System receives personal information in database |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Member System. Diagram #1.used-case#1.2 | |
| Use Case Name: | Search for Member | |
| Actors: | Employee - Administrator | |
| Overview: | Allow searching for a member in system | |
| Related use cases: | Create member profile | |
| Event(Stimulus): | **User Action** | **System Action** |
| Employee looks for member and member information | System gets this information from database |
| Exceptions: | If the system cannot connect to database server we try to save member information offline | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Employee System.Diagram#1.usedcase#1.3 | |
| Use Case Name: | Add employee | |
| Actors: | Administrator | |
| Overview: | Administrator can add employee to the system | |
| Related use cases: |  | |
| Event(Stimulus): | **User Action** | **System Action** |
| Administrator can add employee to the system | System receives personal data of employee |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Employee System.Diagram#1.usedcase#1.4 | |
| Use Case Name: | View employee | |
| Actors: | Administrator | |
| Overview: | Administrator can view employee information in the system | |
| Related use cases: |  | |
| Event(Stimulus): | **User Action** | **System Action** |
| Administrator can view employee information in the system | System shows personal data of employee in system |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Empoyee System.Diagram#1.usedcase#1.5 | |
| Use Case Name: | Search employee | |
| Actors: | Administrator | |
| Overview: | Administrator can Search about employee in the system | |
| Related use cases: |  | |
| Event(Stimulus): | **User Action** | **System Action** |
| Administrator can search about employee in the system | System looks for personal data of employee in the system |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Employee System.Diagram#1.usedcase#1.6 | |
| Use Case Name: | Change employee Status | |
| Actors: | Administrator | |
| Overview: | Administrator can change employee status in the system | |
| Related use cases: |  | |
| Event(Stimulus): | **User Action** | **System Action** |
| Administrator can change employee status in the system | System changes status of employee |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Payment System.Diagram#1.usedcase#1.8 | |
| Use Case Name: | Record payments | |
| Actors: | Employee | |
| Overview: | Employee can record member payment in the system | |
| Related use cases: |  | |
| Event(Stimulus): | **User Action** | **System Action** |
| Employee can record member payment in the system | System records member payment in the system |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Payment System.Diagram#1.usedcase#1.9 | |
| Use Case Name: | View payments | |
| Actors: | Administrator | |
| Overview: | Administrator views if employee records payment or not | |
| Related use cases: | Record payments | |
| Event (Stimulus): | **User Action** | **System Action** |
| Administrator can view payments in the system | System shows recorded payments  in the system |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Payment System.Diagram#1.usedcase#1.10 | |
| Use Case Name: | Search about payments | |
| Actors: | Administrator | |
| Overview: | Administrator searches about member payment | |
| Related use cases: | Record payments | |
| Event (Stimulus): | **User Action** | **System Action** |
| Administrator looks for member payments in the system | System looks for payment in  in the database |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Package System.Diagram#1.usedcase#1.12 | |
| Use Case Name: | Add package | |
| Actors: | Administrator | |
| Overview: | Administrator adds new package | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Administrator adds new package to the system | System adds the new package to the system |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Package System.Diagram#1.usedcase#1.13 | |
| Use Case Name: | Change package status | |
| Actors: | Administrator | |
| Overview: | Administrator controls if package is enabled or not | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Administrator controls if packages is enabled or not | System updates package status |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Package System.Diagram#1.usedcase#1.14 | |
| Use Case Name: | Search package | |
| Actors: | Administrator, Employee, | |
| Overview: | Administrator and employee looks for package using package ID | |
| Related use cases: | Add package | |
| Event (Stimulus): | **User Action** | **System Action** |
| User adds package | System shows result of the search |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Package System.Diagram#1.usedcase#1.15 | |
| Use Case Name: | View All Packages | |
| Actors: | Administrator, Employee | |
| Overview: | Administrator and Employee view all enabled or disabled packages | |
| Related use cases: | Add package | |
| Event (Stimulus): | **User Action** | **System Action** |
| User view all packages | System shows all packages from database |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Package System.Diagram#1.usedcase#1.16 | |
| Use Case Name: | Subscribe Packages | |
| Actors: | Employee | |
| Overview: | Employee enables member to subscribe to package | |
| Related use cases: | Add package | |
| Event (Stimulus): | **User Action** | **System Action** |
| Employee make subscribe to user package | System records subscription  for member |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Month System.Diagram#1.usedcase#1.17 | |
| Use Case Name: | Add month offer | |
| Actors: | Administrator | |
| Overview: | Administrator adds month’s new offers to system | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Administrator adds offers to system | System records offers |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Month System.Diagram#1.usedcase#1.18 | |
| Use Case Name: | View all month offers | |
| Actors: | Administrator, Employee | |
| Overview: | View all month offers | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Employee view about all month offers | System shows all month offers |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Month System.Diagram#1.usedcase#1.19 | |
| Use Case Name: | Search for month offer | |
| Actors: | Administrator, Employee | |
| Overview: | Search for month offer using offer’s ID | |
| Related use cases: | Add month offer | |
| Event (Stimulus): | **User Action** | **System Action** |
| search about month using ID | System shows the result of the search |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Month System.Diagram#1.usedcase#1.20 | |
| Use Case Name: | Subscribe month offer | |
| Actors: | Employee | |
| Overview: | Employee makes subscribe for the member | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Employee adds member in month’s subscription | System records member’s subscription |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Trainer System.Diagram#1.usedcase#1.21 | |
| Use Case Name: | Add trainer | |
| Actors: | Administrator | |
| Overview: | Adds new trainer to system | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Add new trainer data to system | System add trainer data to the database |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Trainer System.Diagram#1.usedcase#1.22 | |
| Use Case Name: | Subscribe private trainer | |
| Actors: | Employee | |
| Overview: | Member subscribe to private trainer by employee | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Employee attaches member to private trainer | System records private trainer of member |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Emails System.Diagram#1.usedcase#1.23 | |
| Use Case Name: | Save information of sending emails | |
| Actors: | Employee | |
| Overview: | Saves information to use it to send emails | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Save information | System saves information into database |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Emails System.Diagram#1.usedcase#1.24 | |
| Use Case Name: | Sending emails | |
| Actors: | Employee | |
| Overview: | Send emails to users | |
| Related use cases: | Save information to sending emails | |
| Event (Stimulus): | **User Action** | **System Action** |
| Employee sends emails to users | System sends email and saves its information |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Emails System.Diagram#1.usedcase#1.25 | |
| Use Case Name: | Search about emails | |
| Actors: | Employee | |
| Overview: | To search about specified email | |
| Related use cases: | Sending email | |
| Event (Stimulus): | **User Action** | **System Action** |
| Looks for an email | System shows results of search |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Emails System.Diagram#1.usedcase#1.26 | |
| Use Case Name: | View email | |
| Actors: | Employee | |
| Overview: | Employee show email information | |
| Related use cases: | Sending email | |
| Event (Stimulus): | **User Action** | **System Action** |
| Employee views email information | System shows emails information |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Class System.Diagram#1.usedcase#1.27 | |
| Use Case Name: | Add Class | |
| Actors: | Administrator | |
| Overview: | Administrator adds new class to the system | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Add new class to the system | Sets new class |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Class System.Diagram#1.usedcase#1.28 | |
| Use Case Name: | View all classes | |
| Actors: | Administrator, Employee | |
| Overview: | Administrator and Employee can view all available and not available classes | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Views all classes | Shows all classes |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Class System.Diagram#1.usedcase#1.29 | |
| Use Case Name: | View available classes | |
| Actors: | Employee | |
| Overview: | Employee can view all available classes | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Views all available classes | Shows all available classes |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Class System.Diagram#1.usedcase#1.30 | |
| Use Case Name: | Search for classes | |
| Actors: | Employee | |
| Overview: | Employee can search about classes | |
| Related use cases: | Add class | |
| Event (Stimulus): | **User Action** | **System Action** |
| Searches about specified class | Shows results of search |
| Exceptions: |  | |
| Comments: |  | |

|

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Class System.Diagram#1.usedcase#1.31 | |
| Use Case Name: | Subscribe class | |
| Actors: | Employee | |
| Overview: | Employee can subscribe class to member | |
| Related use cases: | Add classSubscribtion | |
| Event (Stimulus): | **User Action** | **System Action** |
| Subscribe in class | Attaches user to class |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Announcement System.Diagram#1.usedcase#1.32 | |
| Use Case Name: | Post announcement | |
| Actors: | Employee | |
| Overview: | Employee can post an announcement | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Posts announcement | Uploads announcement |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Announcement System.Diagram#1.usedcase#1.33 | |
| Use Case Name: | Delete announcement | |
| Actors: | Employee | |
| Overview: | Employee can delete an announcement | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Delete announcement | remove announcement |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Announcement System.Diagram#1.usedcase#1.34 | |
| Use Case Name: | View all announcements | |
| Actors: | Employee, Administrator | |
| Overview: | Employee Administrator can view all announcements | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| view all announcements | Shows all announcements |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Attendance and subscription System.Diagram#1.usedcase#1.35 | |
| Use Case Name: | View all class subscriptions | |
| Actors: | Administrator | |
| Overview: | List all class subscription | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| view all class subscription | List all class subscription |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Attendance and subscription System.Diagram#1.usedcase#1.36 | |
| Use Case Name: | Calculate number of class attendance | |
| Actors: | Employee | |
| Overview: | Employee Calculates number of class attendance | |
| Related use cases: | Take trainer attendance, take member attendance | |
| Event (Stimulus): | **User Action** | **System Action** |
| Calculates number of class attendance | Saves and shows number of class attendance of class |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Attendance and subscription System.Diagram#1.usedcase#1.37 | |
| Use Case Name: | View all month subscription | |
| Actors: | Administrator | |
| Overview: | Administrator lists all month subscription | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| lists all month subscription | Show all month subscription |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Attendance and subscription System.Diagram#1.usedcase#1.38 | |
| Use Case Name: | Calculate number of month attendance | |
| Actors: | Employee | |
| Overview: | Employee calculates number of month attendance | |
| Related use cases: | Calculate number of class attendance | |
| Event (Stimulus): | **User Action** | **System Action** |
| Calculates number of month attendance | Saves and shows number of class attendance |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Attendance and subscription System.Diagram#1.usedcase#1.39 | |
| Use Case Name: | Update freeze number | |
| Actors: | Administrator, Employee | |
| Overview: | Administrator and Employee update freeze number | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Records freeze days for member | Updates freeze days |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Attendance and subscription System.Diagram#1.usedcase#1.40 | |
| Use Case Name: | View all packages subscription | |
| Actors: | Administrator | |
| Overview: | Administrator views all packages subscription | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| views all packages subscription | Shows packages subscription |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Attendance and subscription System.Diagram#1.usedcase#1.41 | |
| Use Case Name: | Update number of invitations | |
| Actors: | Administrator, Employee | |
| Overview: | Administrator and Employee add invitation to packages member | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Adds invitation to package | Attach member to pacakge |
| Exceptions: |  | |
| Comments: |  | |

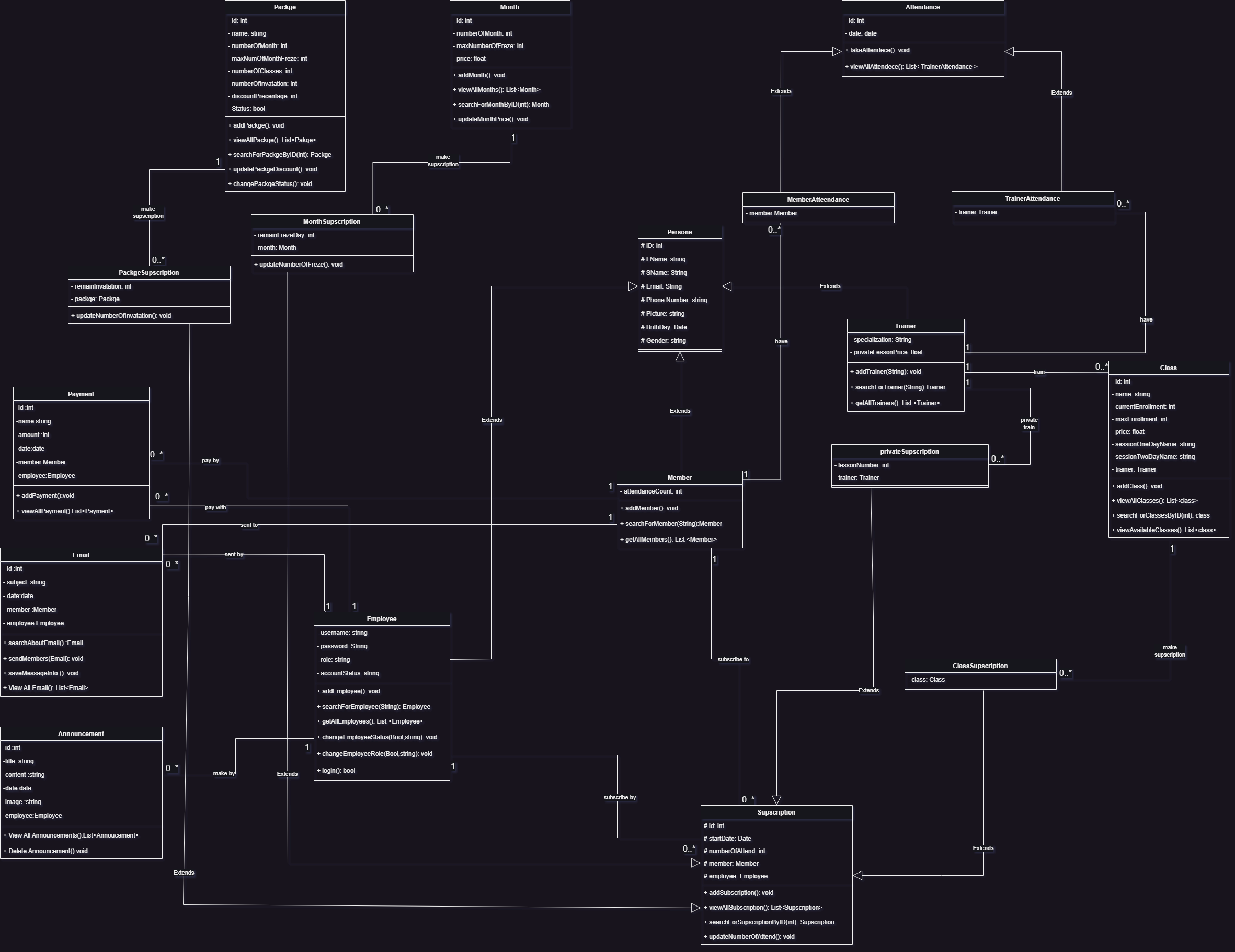
|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Attendance and subscription System.Diagram#1.usedcase#1.42 | |
| Use Case Name: | Take member attendance | |
| Actors: | Employee | |
| Overview: | Employee assigns attendance of member | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Assigns attendance for member | Saves attendance of member |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Attendance and subscription System.Diagram#1.usedcase#1.43 | |
| Use Case Name: | Show all attendance of members | |
| Actors: | Employee | |
| Overview: | Employee views all attendance of members | |
| Related use cases: | Take member attendance | |
| Event (Stimulus): | **User Action** | **System Action** |
| View attendance of members | Shows attendance of members |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Attendance and subscription System.Diagram#1.usedcase#1.44 | |
| Use Case Name: | Take trainer attendance | |
| Actors: | Employee | |
| Overview: | Employee take attendance from trainer and sets it | |
| Related use cases: |  | |
| Event (Stimulus): | **User Action** | **System Action** |
| Sets attendance of trainer | Saves attendance of trainer |
| Exceptions: |  | |
| Comments: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case Number: | #Attendance and subscription System.Diagram#1.usedcase#1.45 | |
| Use Case Name: | Show all attendance of trainer | |
| Actors: | Employee | |
| Overview: | Employee views all attendance of trainer | |
| Related use cases: | Take trainer attendance | |
| Event (Stimulus): | **User Action** | **System Action** |
| View attendance of trainer | Shows attendance of triner |
| Exceptions: |  | |
| Comments: |  | |

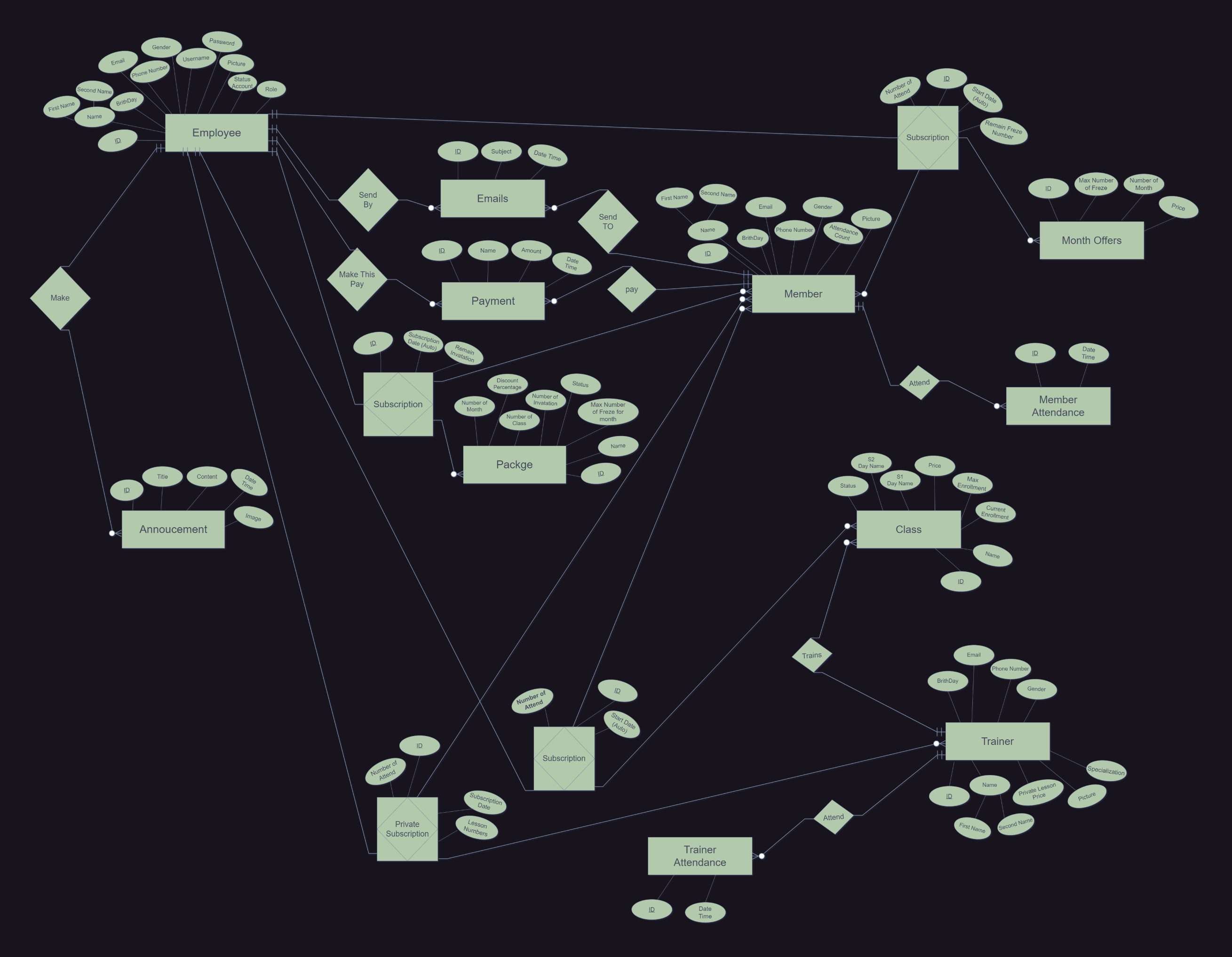
# **4.2** **Class diagrams**

****

* + **Use Classes Description Tables**

|  |  |  |
| --- | --- | --- |
| **Class ID** | **Class Name** | **Description** |
| 1 | Person | Represents a generic person with attributes such as first name, last name, email, phone, picture, birthday, and gender. |
| 2 | Member | Inherits from Person. Represents a gym member with additional attributes like attendance count. Provides methods for adding, searching, and retrieving members. |
| 3 | Trainer | Inherits from Person. Represents a gym trainer with attributes like specialization and private lesson price. Provides methods for adding trainers and retrieving a list of trainers. |
| 4 | Employee | Inherits from Person. Represents an employee with attributes like username, password, role, and account status. Provides methods for adding employees, searching, retrieving all employees, changing roles, and login functionality. |
| 5 | Class | Represents a gym class with attributes such as name, current enrollment, max enrollment, price, session days, and associated trainer. Provides methods for viewing all classes, searching by ID, and retrieving available classes. |
| 6 | Month | Represents a subscription month with attributes like the number of months, max freeze days, and price. Provides methods for adding months, viewing all months, searching by ID, and updating prices. |
| 7 | Packge | Represents a gym package with attributes like name, number of months, max number of freeze days, number of classes, number of invitations, discount percentage, and status. Provides methods for adding packages, viewing all packages, searching by ID, updating discounts, and changing status. |
| 8 | Subscription | Represents a gym subscription with attributes like start date, number of attendances, associated member, and employee. Provides methods for adding subscriptions, viewing all subscriptions, searching by ID, and updating attendances. |
| 9 | PrivateSubscribtion | Represents a private lesson subscription with attributes like lesson number and associated trainer. Provides a method for updating the number of invitations. |
| 10 | ClassSubscription | Represents a class subscription with an associated class. |
| 11 | PackageSubscription | Represents a package subscription with attributes like remaining invitations and associated package. Provides a method for updating the number of invitations. |
| 12 | MonthSubscription | Represents a month subscription with attributes like remaining freeze days and associated month. Provides a method for updating the number of freeze days. |
| 13 | Payment | Represents a payment with attributes like name, amount, date, associated member, and employee. Provides methods for adding payments and viewing all payments. |
| 14 | Email | Represents an email with attributes like subject, date, associated member, and employee. Provides methods for searching, sending to members, saving message info, and viewing all emails. |
| 15 | Announcement | Represents an announcement with attributes like title, content, date, image, and associated employee. Provides methods for viewing all announcements and deleting an announcement. |
| 16 | Attendance | Represents attendance with attributes like date. Provides methods for taking attendance and viewing all attendance. |
| 17 | MemberAttendance | Represents member attendance with an associated member. |
| 18 | TrainerAttendance | Represents trainer attendance with an associated trainer. Provides methods for viewing all trainer attendance. |

# **4.3 Physical Entity-Relationship Diagram**

****

# **Ownership Report**

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Student Name** | **What the student done** |
| 42110049 | *Adham Hassan Sayed* | *create Er Diagram Relationship and mapping it* |
| 42110074 | *Ibrahim Mostafa* | *create Er Diagram Entity and attribute* |
| 42110505 | *Abdelrahman Khaled Aboarab* | *create function and non-function Requrment* |
| 42110103 | *Ahmed Mostafa Mahmoud Khater* | *create case Diagram for each system* |
| 42110244 | *Youssef Omar Elzgzouk* | *create case Description for case Diagram* |
| 42110339 | *Ahmed Ashraf Sayed Abdelrahman* | *create class attribute and functions* |
| 42110040 | *Ziad Ahmed Hegazy* | *create relations between classes* |