

TERM PROJECT PART 2: Design with UML

The purpose of this exercise is to practice modeling with UML. The assignment builds on last week’s scenario, utilizing a system you are familiar with so that you can concentrate on UML



February 8, 2024

Boston university

Mahim Choudhury

Table of Contents

[4.1 Selected Use Case, Revised 2](#_Toc158627326)

[A 4.1 (ChatGPT regarding Selected Use Case – Revised) 3](#_Toc158627328)

[4.2 Sequence Diagram 5](#_Toc158627329)

[Figure 1: Sequence Diagram 5](#_Toc158627331)

[4.3 Class Model 6](#_Toc158627332)

[Figure 2: Class Model Diagram 6](#_Toc158627333)

[4.4 Documenting Classes and Relationships 7](#_Toc158627334)

[4.4.1 First Business Class Selected: User 7](#_Toc158627335)

[4.4.2 Second Business Class Selected: Sensors. 7](#_Toc158627336)

[4.4.3 Third Business Class Selected: PetMonitoring. 8](#_Toc158627337)

[4.4.4 Non-Business Class Selected: <<Design>>. 8](#_Toc158627338)

[Appendix 9](#_Toc158627339)

[References 10](#_Toc158627340)

[Evaluation 11](#_Toc158627341)

MET CS682 ASSIGNMENT 4

TERM PROJECT PART 2: Design with UML

*Mahim Choudhury*

The purpose of this exercise is to give you practice modeling with UML. The assignment builds on last week’s scenario, utilizing a system you are familiar with so that you can concentrate on UML. USE THIS TEMPLATE: Please respond by using this Word template, including the text boxes, leaving the headings and the gray text unchanged—but exclude the *hints* section (at the end) from your solution.

1. OBSERVE LIMITS; 2. USE AI GENERATION: as before but only on Question 4.1. 3. OUTSIDE RESEARCH; 4. NAME YOUR FILE; 5. EVALUATION; 6. HINTS: as before

You are to specify a UML design for *EmbeddedAI*. The characteristics listed in Assignment 3 (Part 1 of the term project) apply. *EmbeddedAI* has a potentially large scope, but your solution should focus on aspects that you select. These are assumed to include your requirements for Assignment 3 (Part 1 of the term project), except where you indicate otherwise.

# Selected Use Case, Revised

## Select Improve and update the one of the two use cases from Term Project Part 1. This will be used for the rest of the assignment, specifically the Sequence Diagram in section 4.

| Step # | Actor | System |
| --- | --- | --- |
| 1 | Pet Owner activates pet monitoring mode | System initiates AI-enabled pet monitoring functionalities, including smart cameras, motion sensors, and environmental sensors, to track pet movements and behaviors. |
| 2 | Pet Owner receives activity alert | System utilizes AI algorithms to detect pet movements, sounds, or irregular behaviors, and sends real-time alerts and notifications to the pet owner's mobile device. |
| 3 | Pet Owner views live monitoring feed | System provides access to live video streams and environmental data through the mobile app, allowing the pet owner to observe their pets' activities and surroundings remotely. |
| 4 | Pet Owner interacts with pets remotely | System enables the pet owner to engage with their pets through interactive features, such as remotely operated toys, treat dispensers, or two-way audio communication channels. |
| 5 | Pet Owner ensures pet safety | System allows the pet owner to monitor environmental conditions, such as temperature, humidity, and air quality, and take necessary actions to ensure the safety and comfort of their pets. |
| 6 | Pet Owner receives guidance on mitigating risks or contacting emergency services (Alt 6) | If the system detects unusual behavior or potential hazards, it alerts the pet owner and provides guidance on mitigating risks or contacting emergency services. |
| 8 | System activates pre-configured safety protocols (Alt 8) | If the pet owner is unavailable to respond to alerts, the system can activate pre-configured safety protocols, such as contacting designated emergency contacts or veterinary services. |

The AI response was not useful at all as it failed to provide any new useful extension to my current use case. After carefully observing the use case myself, I am convinced it has enough functionalities and features to make out a complex system design. It has many classes and objects with dependencies which would be beneficial when designing the sequence and class diagram.

## A 4.1 (ChatGPT regarding Selected Use Case – Revised)

Show your most relevant prompt to chatGPT and the response which was most relevant to your solution.

Prompt: Please refine, improve, and extend the pet monitoring use case I defined in part 1 to if necessary.

Response: To extend and improve the pet activity monitoring use case, we can introduce additional functionalities and interactions between the Pet Owner and the Smart Home System. Here's the revised and extended version of the use case:

Use case Name: Pet Activity Monitoring Actor: Pet Owner Description: The integrated EmbeddedAI system empowers pet owners to monitor their pets remotely, ensuring their safety and well-being even when the owners are away from home. Pre-condition: The system is online and connected to IoT devices.

| Step # | Actor | System |
| --- | --- | --- |
| 1 | Pet Owner activates pet monitoring mode | System initiates AI-enabled pet monitoring functionalities, including smart cameras, motion sensors, and environmental sensors, to track pet movements and behaviors. |
| 2 | Pet Owner receives activity alert | System utilizes AI algorithms to detect pet movements, sounds, or irregular behaviors, and sends real-time alerts and notifications to the pet owner's mobile device. |
| 3 | Pet Owner views live monitoring feed | System provides access to live video streams and environmental data through the mobile app, allowing the pet owner to observe their pets' activities and surroundings remotely. |
| 4 | Pet Owner interacts with pets remotely | System enables the pet owner to engage with their pets through interactive features, such as remotely operated toys, treat dispensers, or two-way audio communication channels. |
| 5 | Pet Owner ensures pet safety | System allows the pet owner to monitor environmental conditions, such as temperature, humidity, and air quality, and take necessary actions to ensure the safety and comfort of their pets. |
| 6 | Pet Owner receives guidance on mitigating risks or contacting emergency services (Alt 6) | If the system detects unusual behavior or potential hazards, it alerts the pet owner and provides guidance on mitigating risks or contacting emergency services. |
| 7 | System activates pre-configured safety protocols (Alt 7) | If the pet owner is unavailable to respond to alerts, the system can activate pre-configured safety protocols, such as contacting designated emergency contacts or veterinary services. |

# Sequence Diagram

## Provide a detailed sequence diagram for one of your use cases above. Your solution should have at least five objects. Make sure to represent these correctly. For example, if you were to create an instance of an *VideoChat* class (i.e., to model a new video chat instance) it may be called *newVideo:Video* or just *:Video* if the instance name is irrelevant.

A diagram with green and blue squares

Description automatically generated

### Figure 1: Sequence Diagram

Sequence Diagram depicts the interaction between an actor and the system for a use-case scenario [1]. The sequence diagram above includes everything, but the live monitoring feature described in the use case (to avoid diagram complexity). It starts with the pet owner clicking the pet-monitoring tab in the smart home system’s home screen (non-entity class). It enables the system to activate sensors and start monitoring. The detection class is an instance of the sensors class. The AI will help the system to detect pet movements continuously. If it detects unusual activities, it automatically goes to the Activity function and enables the alert which triggers a notification to be sent to the owner. The alt course to send the notification to emergency contact as mentioned in the use case is again avoided in the sequence diagram to reduce complexity. The user can click interaction and safety options from the pet monitoring display and involve them in the respective features as depicted in the diagram. Notice the loops, it indicates the sensor detection and interaction is a continuous action unless the user breaks out of it. Another key aspect to keep in mind here is that I made all the returns to the pet monitoring screen assuming the user will use it for the interaction except the notification.

# Class Model

Provide a class model for the system, maintaining the system scope you determined. Your solution should have roughly 12 classes. (When complete, a real design typically may contain hundreds of classes, but your submission must focus on the scope you have chosen.) Your class model should show classes and their relationships. To add clarity to your diagram, provide key attributes and methods. (You do not have to list every attribute and method, just the most important ones.) Label clearly.

A diagram of a computer

Description automatically generated

### Figure 2: Class Model Diagram

The class diagram illustrates the pet monitoring aspect of the system, where EmbeddedAI enhances functionality. It begins with a non-entity stereotype class named Design, conforming to the UML User Guide's definition of stereotypes as labels for similar classes within a specific domain, essentially creating a metatype [2]. The model showcases aggregation, composition, and inheritance relationships among its classes. The Detection class serves as the integration point for EmbeddedAI to analyze data. Further elaboration on key classes and their relationships is provided in the subsequent section.

# Documenting Classes and Relationships

Using the format below, select three most important nontrivial business[[1]](#footnote-1) classes and one non-business design class from the class model and the format below, explain its importance for your design, and its relationships with other classes. Select classes which are complex enough to require explanation.

## 4.4.1 First Business Class Selected: User

**Importance for the Design**: The “User” class defined in the class model diagram is one of most important business classes in the system. Although the main beam of light in this system is focused around the implementation of Iot devices and AI analysis, ultimately the destination is the User. User class defines what the system should know about the human users of it. User has their own id, username. It also allows them to create new users and update their information. The class has some relationships with other important classes which will be described later. It is the root of regular user, emergency contact, interactions with pet, and selecting what dispenser to use for interaction. Although the AI can help in this area, I still believe the pets need their human friend to interact with them instead of a cloud.

**Relationship with other classes**: The “User” class in the class diagram shows that it is inherited by two other sub classes named “EmergencyContact” and “Interaction”. From my previous experience, I could tell the inheritance of classes means they are the same class with additional features. Meaning the EmergencyContact and interaction classes already have the attributes of “User” class and then they have their own unique attributes and behavior. In this case, the interaction class have it’s own ID and the EMergencyContact has two more attributes, Email and Cell and their behaviors. For example, each object of Interaction class can have it’s own specific userID, username, and type id. Similarly, EmergencyContact can have it’s userID, username plus the email and cell per instances of the class itself. Interaction class has it’s own relationship with “Dispenser” class, which is also a branch coming out of the root “User” class.

## 4.4.2 Second Business Class Selected: Sensors.

**Importance for the Design**: The “Sensors” class is another important business class for. For this system, it works as the hub class for all the IoT devices to provide the data to the AI for analysis. The class is designed to hold information about the IoT devices such as Id, location, and behaviors like activate(), deactivate(). There could be more attributes and behaviors, however, we are only focusing on some. It is clearly visible in the class diagram that this class has many relationships with some other important classes in the system. This emphasizes the importance of this class in this system. Although it can connect to many other IoT devices, the diagram is only showing the thermostat, and camera as the main two sensors. The relationships are described in the next section.

**Relationship with other classes**: The “Sensors” class has aggregation relationship with Thermostat and Camera classes. From my understating from past learning, aggregation relationship means that on class is part of or has another class. For the Thermostat, it can control one and only one thermostat and a thermostat can only be controlled by that one sensor class. I chose only one thermostat because in most houses, there is only one thermostat to control the temperature. For the Camera class however, the Sensor class can control 1 to many cameras since a house can have many of them. Cameras can only be controlled by that one Sensor class though. The Aggregation relationship also tell us that the Thermostat and Camera classes can exist without the Sensors class. It has the same relationship with “PetMonitoring” class. However, the “detection” class cannot exist without the Sensor class and therefore, it has a composition relationship with it.

## 4.4.3 Third Business Class Selected: PetMonitoring.

**Importance for the Design**: The “PetMonitoring” class is another most important business class for this system. This class works as the main brain of the pet monitoring feature of the smart home system. It connects to various important classes and works as the manger. It holds information of the pet(s), monitors their activity, detects anomalies, checks the system health, etc. Without the “PetMonitoring”, the main purpose of this pet monitoring feature of smart home system will not function as required. It is required to communicate with other classes to gather data and work with it. The many relationships it has with other classes are described in the section below.

**Relationship with other classes**: The “PetMonitoring” class has aggregation relationship with “Sensors”, “Safety”, “User”, “Pet”. This way, it can get information from them and their related classes. It can have 1 to many Sensors, Users, and Pets. Those classes however, can only be associated with only one “PetMonitoring” class. The “PetMonitoring” class can be associated with 0 to 1 Safety class. The reason I chose 0 to 1 is because I think even without the information from safety class, it can still function with the activity detected from the sensors and keep recording it. On the other hand, the “Safety” class should only be associated with the on and only “PetMonitroing” class.

## 4.4.4 Non-Business Class Selected: <<Design>>.

**Importance for the Design**: As depicted by “<< >>”, the Design class is the non-busineess class for this system. It is pictured as the main home screen from where the user can interact with the system to enable pet monitoring mode. It can also display valuable information such as graphs of previous usage, pets weekly sleep time, etc from previous monitoring. Although it is a non-business class, it is important to the user from user interface of the pet monitoring system’s standpoint. It determines how user-friendly and easy to use pet monitoring feature would be to the users. It has a dependency with “PetMonitoring” class which is described in the section below.

**Relationship with other classes**: The dependency described in the class model is pretty self-explanatory. It displays the “PetMonitoring” class. As mentioned earlier, it need the data from the “PetMonitoring” class in order to display the information in the GUI. “PetMonitoring” class gather the data from various other classes across the design but it all needs be presented to the user through he non business class <<Design>>.

# Appendix

Since Term Project Part 2 builds on Term Project Part 1, Provide requirements you have outlined last week in Term Project Part 1. Not everything needs to be included, only the parts you feel are helpful in supporting the UML design. This section is not graded.

Scope:

Scenario:

Imagine a Smart Home Automation System designed to manage various aspects of home living, such as lighting, temperature control, security, and appliance automation. This system utilizes IoT devices and sensors to enable remote monitoring and control via mobile apps or voice commands.

Scope: The scope entails integrating EmbeddedAI into an existing Smart Home Automation System to enhance user experience, efficiency, and security. This integration will leverage generative AI like ChatGPT and IoT elements to provide personalized assistance, intelligent automation, and proactive home management capabilities. Key focus areas include personalized home assistance, natural language interaction, predictive home automation, and enhanced security and safety measures.

The project will involve analyzing user behavior, preferences, and environmental data collected by IoT sensors to offer tailored recommendations and automate routine tasks. It will enable natural language interactions with EmbeddedAI, allowing users to communicate with their smart home system using voice commands or text input. Predictive capabilities will anticipate user needs and potential issues, while enhanced security measures will detect anomalies and trigger automated responses to ensure the safety of the home environment.

Initial Use Case:

|  |  |  |
| --- | --- | --- |
| **Use case Name** | Pet activity monitoring | |
| **Actor:** | Pet owner | |
| **Description:** | The integrated EmbeddedAI system can aid the smart home system to empower the pet owners with the ability to leave their pets home without worry. AI-enabled monitoring system enhances pet safety and allows pet owners to stay connected with their pets, even when they are not physically present at home. | |
| **Pre-condition:** | The system is online and connected to IoT devices. | |
| **Step #** | **Actor** | **System** |
| **1** | Pet Owner activates pet monitoring mode | System initiates AI-enabled pet monitoring functionalities, including smart cameras, motion sensors, and environmental sensors, to track pet movements and behaviors. |
| **2** | Pet Owner receives activity alert | System utilizes AI algorithms to detect pet movements, sounds, or irregular behaviors, and sends real-time alerts and notifications to the pet owner's mobile device. |
| **3** | Pet Owner views live monitoring feed | System provides access to live video streams and environmental data through the mobile app, allowing the pet owner to observe their pets' activities and surroundings remotely. |
| **4** | Pet Owner interacts with pets remotely | System enables the pet owner to engage with their pets through interactive features, such as remotely operated toys, treat dispensers, or two-way audio communication channels. |
| **5** | Pet Owner ensures pet safety | System allows the pet owner to monitor environmental conditions, such as temperature, humidity, and air quality, and take necessary actions to ensure the safety and comfort of their pets. |
| **Alternate Courses:** | [System Alt 6] If the system detects unusual behavior or potential hazards, it alerts the pet owner and provides guidance on mitigating risks or contacting emergency services.  [System Alt 7] If the pet owner is unavailable to respond to alerts, the system can activate pre-configured safety protocols, such as contacting designated emergency contacts or veterinary services. |  |

# References

Show that you used a wide variety of resources by listing them below and clearly indicating in the body above where you used each one (inserting [1], [2] etc.). We suggest using APA format, but other formats are fine as long as your paper clearly distinguishes your work from work of others, being mindful of plagiarism rules.

[1] Whitten, B. (2007). *Systems analysis & design methods*. (7th ed.). New York, NY: McGraw-Hill Irwin.

[2] Booch G, Rumbaugh, J, Jacobson I (2014). The Unified Modeling Language User Guide. Upper Saddle River, NJ: Addison-Wesley.

# Evaluation

A picture containing text, screenshot, number, parallel

Description automatically generated

1. also called “domain” or “entity” class [↑](#footnote-ref-1)