Week 4

CS2212

Introduction to Software Engineering

Class-Based Modeling

Ask Questions Live

cs1.ca/ask

Some images and content from Software Engineering: A Practitioner's Approach 9th Edition By Roger Pressman and Bruce Maxim

Class-Based Modeling

- **Class-based modeling** represents:
  - **Objects** that the system will manipulate.
  - **Operations** *(also called methods or services)* that will be applied to the objects to affect the manipulation.
  - **Relationships** *(some hierarchical)* between the objects.
  - **Collaborations** that occur between the classes that are defined.

The elements of a **class-based model** include **classes** and objects, **attributes**, **operations**, **CRC models**, UML **class diagrams**.

Identifying Analysis Classes

Examining the **usage scenarios** developed as part of the requirements model and perform a "**grammatical parse**“.

- **Classes** are determined by **underlining each noun** or noun phrase and entering it into a simple table.
- **Synonyms** should be noted.
- If the **class (noun)** is required to implement a solution, then it is part of the **solution space**; otherwise, if a class is necessary only to describe a solution, it is part of the **problem space**.

*But what should we look for once all of the **nouns** have been isolated?*

Potential Analysis Classes

- **External Entities:** *(for example: other systems, devices, people) that produce or consume information.*
- **Things:** *(for example: reports, displays, letters, signals) that are part of the information domain for the problem.*
- **Occurrences:** *events that occur within the context of system operations.*
- **Roles:** *played by people who interact with the system.*
- **Organizational Units:** *that are relevant to an application.*
- **Places:** *that establish the context of the problem and overall function.*
- **Structures:** *(for example: sensors, four-wheeled vehicles, or computers) that define a class of objects or related classes of objects.*

Identifying Analysis Classes: Example 1

Identify the **nouns** by underlining them.

The SafeHome security function enables the homeowner to configure the security system when it is installed, monitors all sensors connected to the security system and interacts with the homeowner through the Internet, a PC or a control panel.

During installation, the SafeHome PC is used to program and configure the system. Each sensor is assigned a number and type, a master password is programmed for arming and disarming the system, and telephone numbers(s) are input for dialing when a sensor event occurs.

When a sensor event is recognized, the software invokes an audible alarm attached to the system. After a delay time that is specified by the homeowner during system configuration activities, the software dials a telephone number of a monitoring service, provides information about the location, reporting the nature of the even that has been detected. The telephone number will be redialed every 20 seconds until telephone connection is obtained.

The homeowner receives security information via a control panel, the PC, or a browser, collectively called an interface. The interface displays prompting messages and system status information on the control panel, the PC or the browser window. Homeowner interaction takes the following form….

Identifying Analysis Classes: Example 1

The SafeHome security function enables the homeowner to configure the security system when it is installed, monitors all sensors connected to the security system and interacts with the homeowner through the Internet, a PC or a control panel.

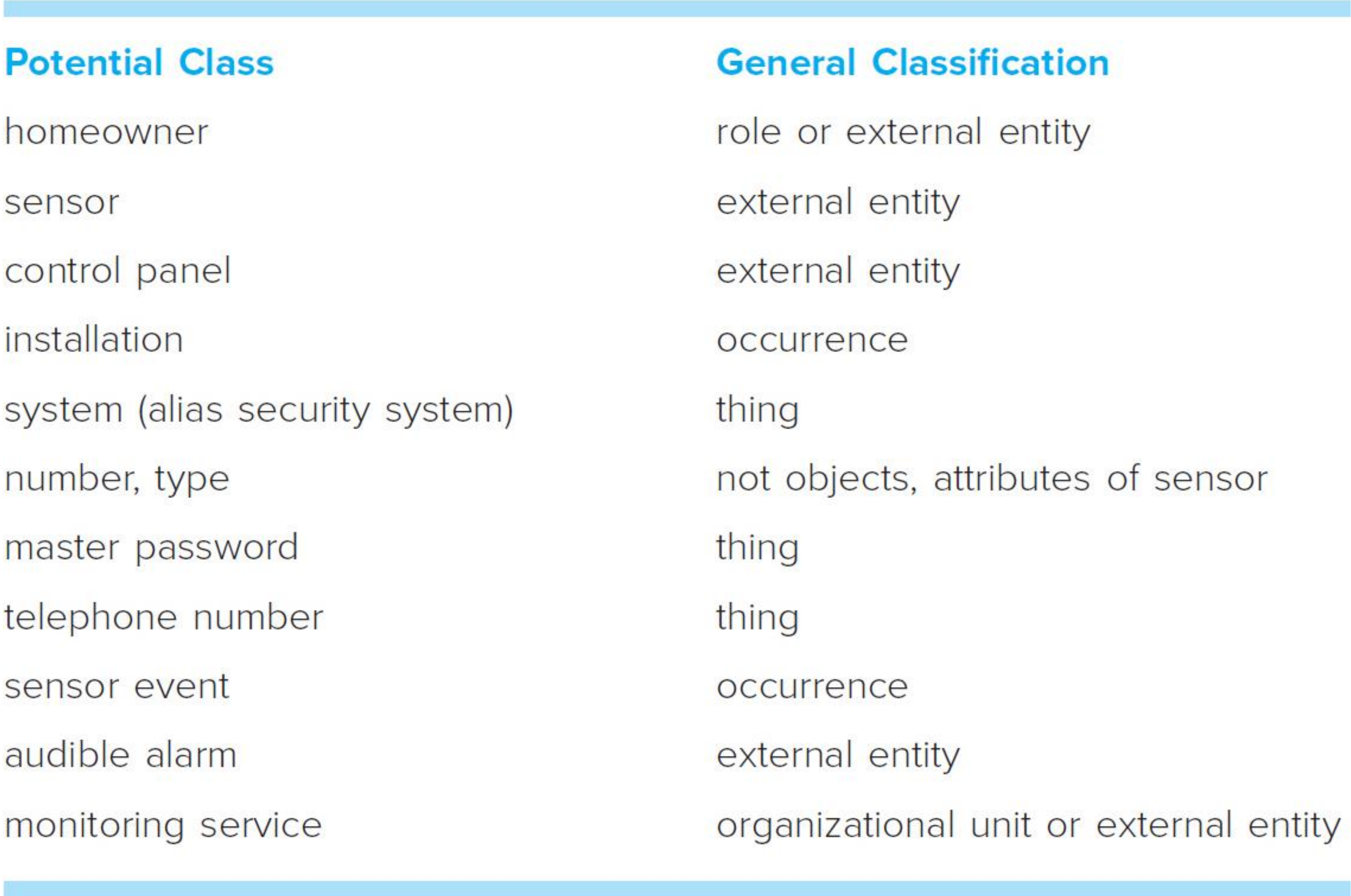
During installation, the SafeHome PC is used to program and configure the system. Each sensor is assigned a number and type, a master password is programmed for arming and disarming the system, and telephone numbers(s) are input for dialing when a sensor event occurs.

When a sensor event is recognized, the software invokes an audible alarm attached to the system. After a delay time that is specified by the homeowner during system configuration activities, the software dials a telephone number of a monitoring service, provides information about the location, reporting the nature of the even that has been detected. The telephone number will be redialed every 20 seconds until telephone connection is obtained.

The homeowner receives security information via a control panel, the PC, or a browser, collectively called an interface. The interface displays prompting messages and system status information on the control panel, the PC or the browser window. Homeowner interaction takes the following form….

Identifying Analysis Classes: Example 1

Potential classes that are part of the solution space.



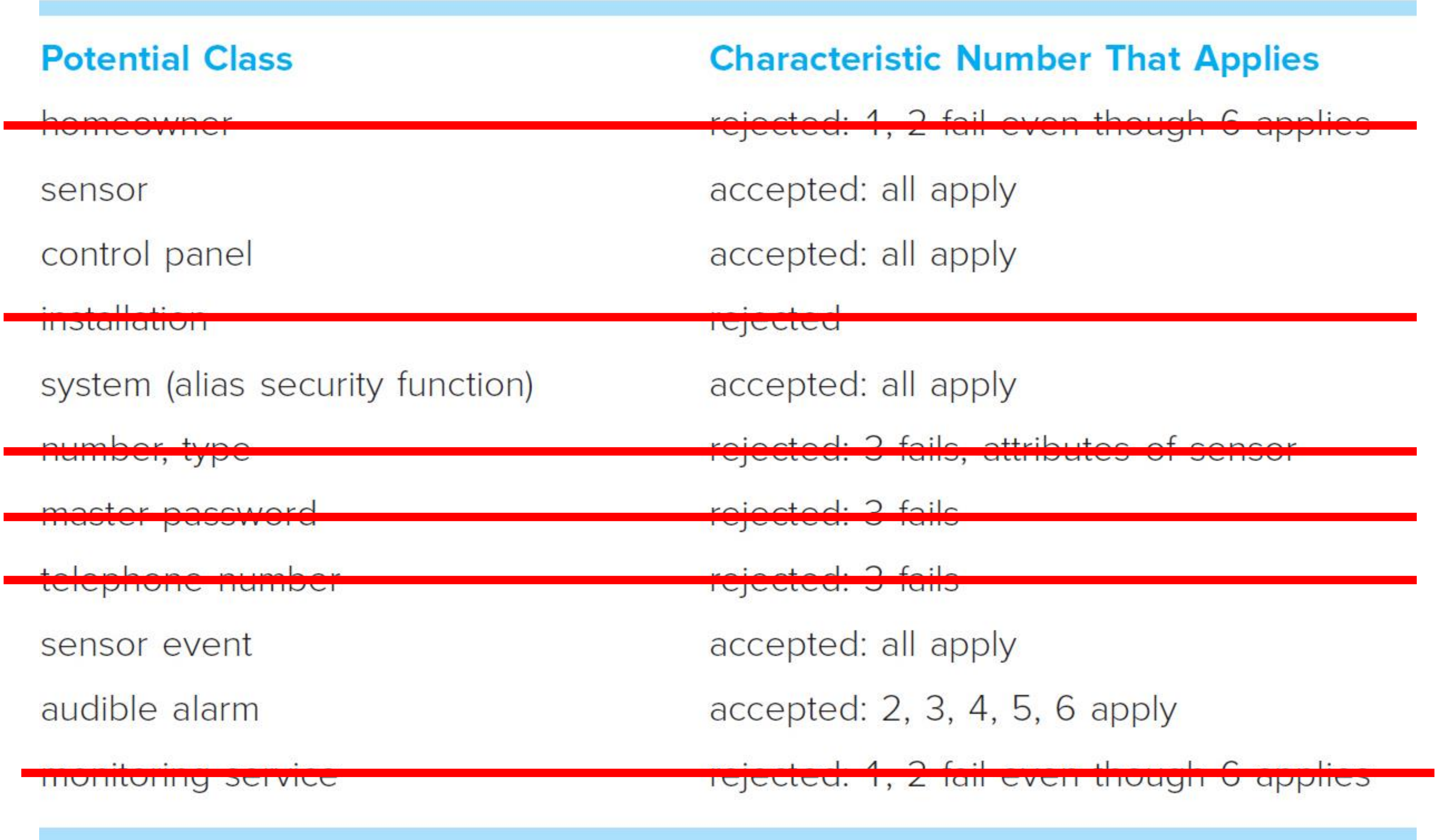
Potential Class	General Classification
homeowner	role or external entity
sensor	external entity
control panel	external entity
installation	occurrence
system (alias security system)	thing
number, type	not objects, attributes of sensor
master password	thing
telephone number	thing
sensor event	occurrence
audible alarm	external entity
monitoring service	organizational unit or external entity

Analysis Class Selection

Characteristics that should be used as you consider each potential class for inclusion:

1. **Retained information**
2. **Needed services**
3. **Multiple attributes**
4. **Common attributes**
5. **Common operations**
6. **Essential requirements**

Identifying Analysis Classes: Example 1



Potential Class	Characteristic Number That Applies
~~homeowner~~	~~rejected: 1, 2 fail even though 6 applies~~
sensor	accepted: all apply
control panel	accepted: all apply
~~installation~~	~~rejected~~
system (alias security function)	accepted: all apply
~~number, type~~	~~rejected: 3 fails, attributes of sensor~~
~~master password~~	~~rejected: 3 fails~~
~~telephone number~~	~~rejected: 3 fails~~
sensor event	accepted: all apply
audible alarm	accepted: 2, 3, 4, 5, 6 apply
~~monitoring service~~	~~rejected: 1, 2 fail even though 6 applies~~

Defining Attributes

- **Attributes** *(also called properties or fields)* **describe** a **class** that has been selected for inclusion in the analysis model.

- It is the **attributes** that **define** the **class**—that clarify what is meant by the class in the context of the **problem space**.

- To develop a meaningful set of **attributes** for an analysis **class**, you should study each **use case** and select those "**things**" that reasonably "**belong**" to the **class**.

Defining Operations

- **Operations** *(also called methods)* define the **behaviour of an object**.
- **Operations** they can generally be divided into four broad categories:
  1. Operations that **manipulate data** in some way.
  2. Operations that **perform a computation**.
  3. Operations that **inquire about the state**.
  4. Operations that **monitor an object** for the **occurrence of a controlling event**.

- These functions are accomplished by operating on attributes and/or associations.
- Therefore, an **operation** must have “knowledge” of the **class attributes** and associations.

Identifying Analysis Classes: Example 1

Identify the **verbs** by marking them in italics.

The SageHome security function enables the homeowner to configure the security system when it is installed, monitors all sensors connected to the security system and interacts with the homeowner through the Internet, a PC or a control panel.

During installation, the SafeHome PC is used to program and configure the system. Each sensor is assigned a number and type, a master password is programmed for arming and disarming the system, and telephone numbers(s) are input for dialing when a sensor event occurs.

When a sensor event is recognized, the software invokes an audible alarm attached to the system. After a delay time that is specified by the homeowner during system configuration activities, the software dials a telephone number of a monitoring service, provides information about the location, reporting the nature of the even that has been detected. The telephone number will be redialed every 20 seconds until telephone connection is obtained.

The homeowner receives security information via a control panel, the PC, or a browser, collectively called an interface. The interface displays prompting messages and system status information on the control panel, the PC or the browser window. Homeowner interaction takes the following form….

Identifying Analysis Classes: Example 1

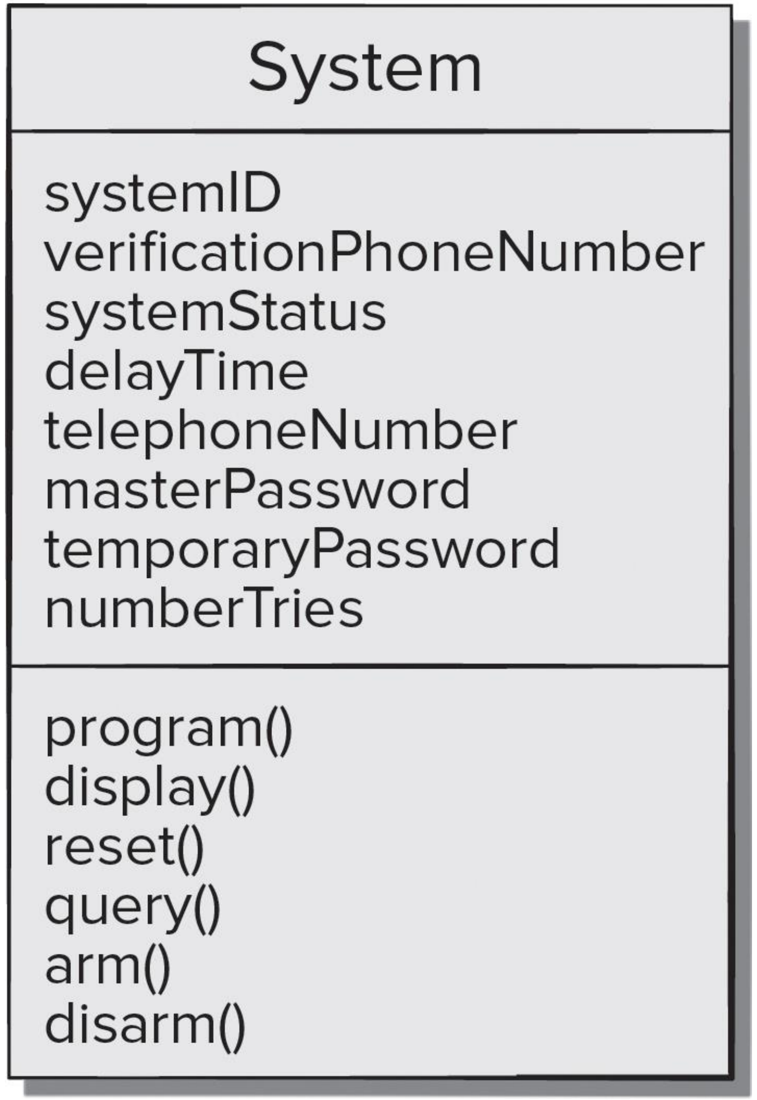
The SageHome security function *enables* the homeowner to *configure* the security system when it is *installed*, *monitors* all sensors *connected* to the security system and *interacts* with the homeowner through the Internet, a PC or a control panel.

During installation, the SafeHome PC is used to *program* and *configure* the system. Each sensor is assigned a number and type, a master password is programmed for *arming* and *disarming* the system, and telephone numbers(s) are *input* for *dialing* when a sensor event occurs.

When a sensor event is *recognized*, the software *invokes* an audible alarm attached to the system. After a delay time that is *specified* by the homeowner during system configuration activities, the software dials a telephone number of a monitoring service, *provides* information about the location, *reporting* the nature of the even that has been detected. The telephone number will be *redialed* every 20 seconds until telephone connection is *obtained*.

The homeowner *receives* security information via a control panel, the PC, or a browser, collectively called an interface. The interface *displays* prompting messages and system status information on the control panel, the PC or the browser window. Homeowner interaction takes the following form….

Example: System Class



System

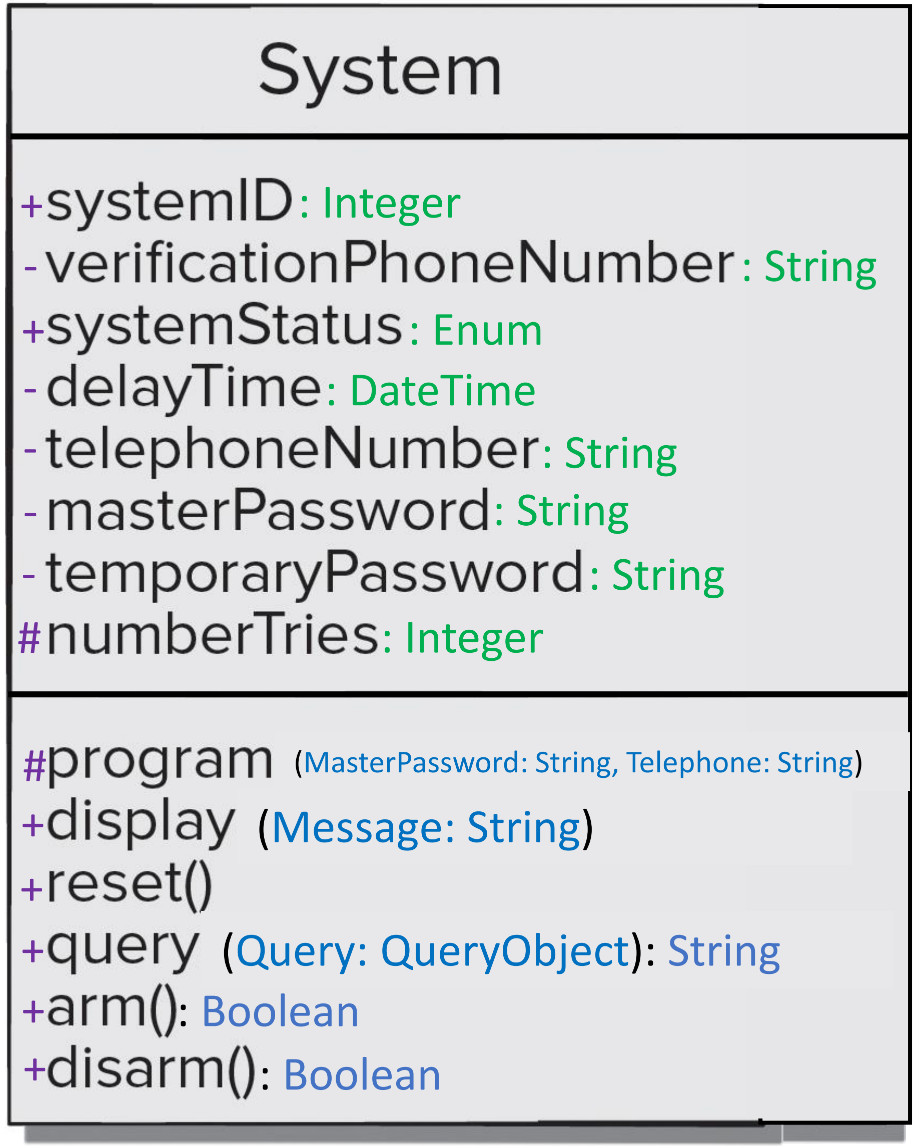
systemID
verificationPhoneNumber
systemStatus
delayTime
telephoneNumber
masterPassword
temporaryPassword
numberTries

program()
display()
reset()
query()
arm()
disarm()

- Analysis classes should contain a class name, list of attributes, and a list of operations.
- Many implementation details are still omitted:
  - Not all required attributes and operations are listed yet.
  - Attribute types, parameter types, and return types may not yet be listed.
  - Operations may not yet have parameters listed.
  - Visibility (public, private, protected, etc.) may not yet be listed).

Example: System Class

As we move from **analysis modeling** to **design modeling** we start to fill in the details of each class.



System

+systemID: Integer
-verificationPhoneNumber: String
+systemStatus: Enum
-delayTime: DateTime
-telephoneNumber: String
-masterPassword: String
-temporaryPassword: String
#numberTries: Integer

#program (MasterPassword: String, Telephone: String)
+display (Message: String)
+reset()
+query (Query: QueryObject): String
+arm(): Boolean
+disarm(): Boolean

Activity: Identify Potential Classes

Example Scenario: Web-based Pothole Tracking and Repair System (PHTRS)

Citizens can log onto a website and report the location and severity of potholes. As potholes are reported they are logged within a “public works department repair system” and are assigned an identifying number, stored by street address, size (on a scale of 1 to 10), location (middle, curb, etc.), district (determined from street address), and repair priority (determined from the size of the pothole). Work order data are associated with each pothole and include pothole location and size, repair crew identifying number, number of people on crew, equipment assigned, hours applied to repair, hole status (work in progress, repaired, temporary repair, not repaired), amount of filler material used, and cost of repair (computed from hours applied, number of people, material and equipment used). Finally, a damage file is created to hold information about reported damage due to the pothole and includes citizen’s name, address, phone number, type of damage, and dollar amount of damage. PHTRS is an online system; all queries are to be made interactively.

Task 1

Based on these nouns, create a list of potential classes.

Rule out nouns that should be attributes of a class, classes that are in the problem space, and classes do not have the required characteristics we discussed.

Citizens can log onto a website and report the location and severity of potholes. As potholes are reported they are logged within a “public works department repair system” and are assigned an identifying number, stored by street address, size (on a scale of 1 to 10), location (middle, curb, etc.), district (determined from street address), and repair priority (determined from the size of the pothole). Work order data are associated with each pothole and include pothole location and size, repair crew identifying number, number of people on crew, equipment assigned, hours applied to repair, hole status (work in progress, repaired, temporary repair, not repaired), amount of filler material used, and cost of repair (computed from hours applied, number of people, material and equipment used). Finally, a damage file is created to hold information about reported damage due to the pothole and includes citizen’s name, address, phone number, type of damage, and dollar amount of damage. PHTRS is an online system; all queries are to be made interactively.

Task 2

Identify the attributes and operations for the Pothole class.

Mark verbs and potential operations in italics (or any other way).

Example Scenario: Web-based Pothole Tracking and Repair System (PHTRS)

Citizens can log onto a website and report the location and severity of potholes. As potholes are reported they are logged within a "public works department repair system" and are assigned an identifying number, stored by street address, size (on a scale of 1 to 10), location (middle, curb, etc.), district (determined from street address), and repair priority (determined from the size of the pothole). Work order data are associated with each pothole and include pothole location and size, repair crew identifying number, number of people on crew, equipment assigned, hours applied to repair, hole status (work in progress, repaired, temporary repair, not repaired), amount of filler material used, and cost of repair (computed from hours applied, number of people, material and equipment used). Finally, a damage file is created to hold information about reported damage due to the pothole and includes citizen's name, address, phone number, type of damage, and dollar amount of damage. PHTRS is an online system; all queries are to be made interactively.

Task 3

Create a Class Diagram for the Pothole class.

Only include a class name, operations, and attributes (don’t need types, parameters, or visibility).

Citizens can log onto a website and report the location and severity of **potholes**. As potholes are reported they are logged within a “public works department repair system” and are *assigned* an **identifying number**, stored by **street address**, **size** (on a scale of 1 to 10), **location** (middle, curb, etc.), **district** (*determined from street address*), and **repair priority** (*determined from the size of the pothole*). Work order data are associated with each pothole and include pothole location and size, repair crew identifying number, number of people on crew, equipment assigned, hours applied to repair, hole status (work in progress, repaired, temporary repair, not repaired), amount of filler material used, and cost of repair (computed from hours applied, number of people, material and equipment used). Finally, a damage file is created to hold information about reported damage due to the pothole and includes citizen’s name, address, phone number, type of damage, and dollar amount of damage. PHTRS is an online system; all queries are to be made interactively.

CRC Modeling

- Class-Responsibility-Collaborator (CRC) modeling provides a simple means for **identifying** and **organizing** the **classes** that are relevant to system or product requirements.

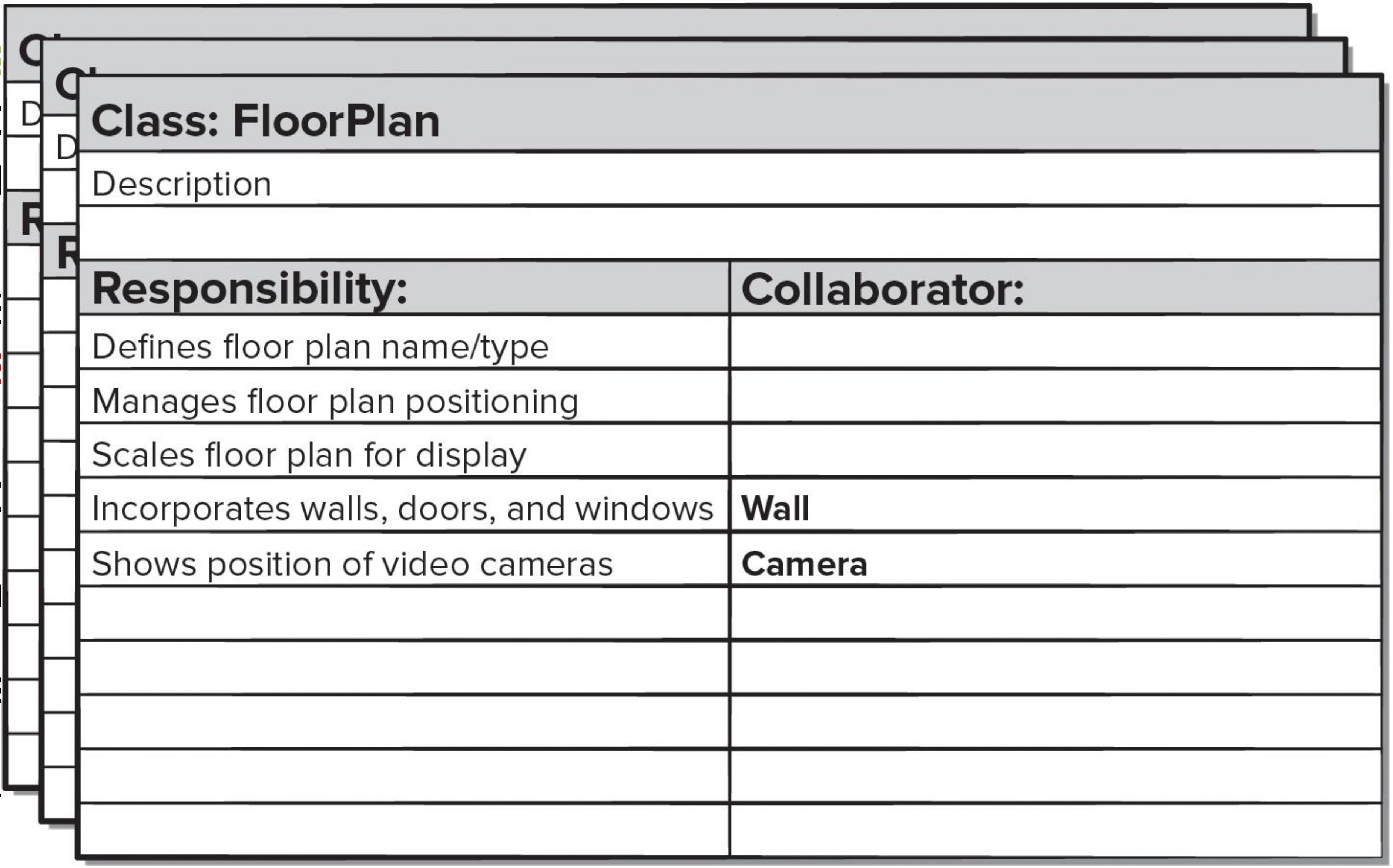
- A CRC model is really a **collection of standard index cards** that **represent classes**.

- **The cards are divided into three sections:**

  1. Along the top of the card you write the **name of the class**.
  2. List the **class responsibilities** on the left.
  3. List the **collaborators** on the right.

CRC Modeling

- Class-Respons means for ident system or produ
- A CRC model is represent class
- **The cards are d**
  1. Along the top
  2. List the **class**
  3. List the **colla**



Class: FloorPlan

Description

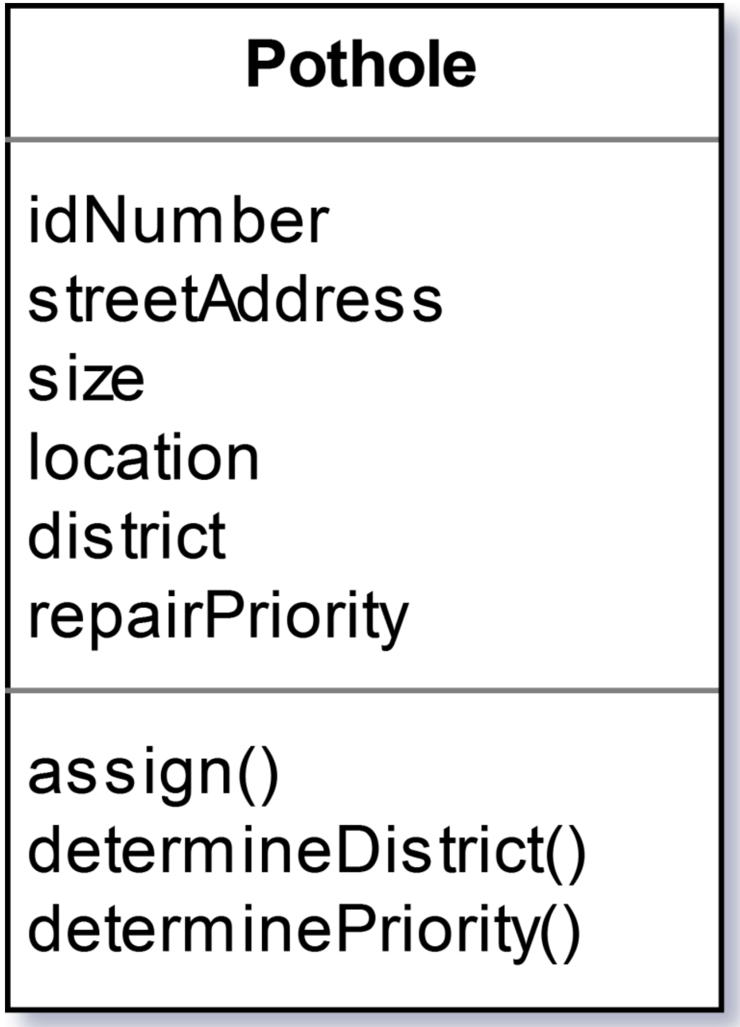
Responsibility:	Collaborator:
Defines floor plan name/type	
Manages floor plan positioning	
Scales floor plan for display	
Incorporates walls, doors, and windows	**Wall**
Shows position of video cameras	**Camera**

Responsibilities

- **Responsibilities** are attributes and operations that are relevant for the class.

- **Collaborators** are those classes that provide a class with information needed or action required to complete a responsibility.

CRC Card Example: Pothole



Pothole

idNumber
streetAddress
size
location
district
repairPriority

assign()
determineDistrict()
determinePriority()

Class: Pothole	
The pothole class represents all instances of potholes that will be tracked by PHTRS. Potholes are initially reported and logged with the system by Citizens.	
Responsibility:	**Collaborator:**
Assigned an identifying number.	
Defines a pothole including street address, size, and location.	
Determines district.	
Determines repair priority.	
Reported by citizens.	**Citizen**
Associated with Work Order.	**Work Order**

CRC Model Review Process

1. **All stakeholders** in the review *(of the CRC model)* are **given a subset of the CRC model index cards**. No reviewer should have two cards that collaborate.

2. The **review leader reads the use case** deliberately. As the review leader comes to a named object, they **pass a token to the person holding the corresponding class index card**.

3. When the token is passed, the **holder of the class card is asked to describe the responsibilities** noted on the card. The group determines whether one of the responsibilities **satisfies the use case requirement**.

4. If an error is found, **modifications are made to the cards**. This may include the definition of new classes *(CRC index cards)* or revising lists of responsibilities or collaborations on existing cards.