Chris Moorhead
cam0071@auburn.edu

Protec Command Center

1. Concept Statement:

The Protec Command Center is a home security system that permits the home-owner to monitor the exterior and interior of their property either directly through a PC or remotely via the internet. Command Center incorporates various devices, supplied by Protec, that a home-owner can install in order to ensure the prevention of theft and the safety of their family. Most commonly these devices are sensors.

Among these sensors are external cameras. These are installed around home entry points and provide constant surveillance. The Command Center allows direct access to any of the camera feeds for the home-owner. When cameras detect motion, they immediately notify the user and send a picture of the intrusion. The cameras use classification to determine whether the motion is caused by a person or something else such as an animal.

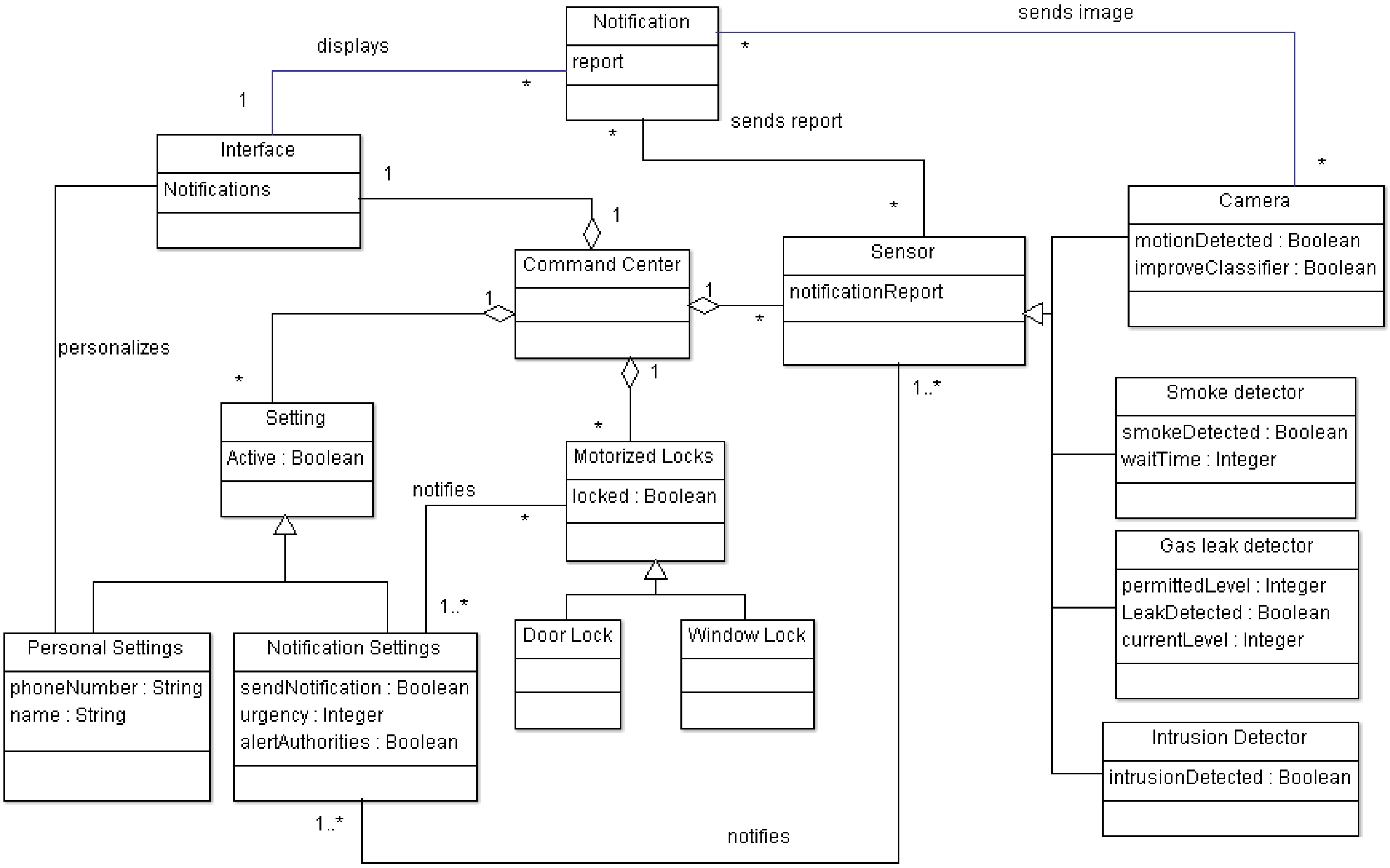
Other sensors that the Command Center includes are smoke detectors, gas leak detectors, and intrusion detectors. Intrusion detectors are motion sensors detecting if a door or window is open or if there is movement coming through the door or window. Similarly, to the cameras, abnormal readings for these sensors default to sending notifications to the user through system notifications.

The Command Center also includes motorized locks for doors and windows. Through the Command Center interface, the home-owner can both see what doors and windows are unlocked. The homeowner can then choose to lock them all through a “Lockdown” command or lock them all individually. The doors can be manually unlocked as well without the application but the home-owner will be notified if any doors or windows open without using the application.

The Command Center application gives the home-owner control of what the see and when they see it. Upon entering the application, the home-owner is shown the most recent and important notifications. If there are images associated with the notification, such as via the video feed, then the home-owner can easily expand the notification to view it. From the main menu, the home-owner can access settings or speak with a company security specialist if they have any specific questions. This specialist will either be called directly or sent a notification to contact the home-owner. The specialist can provide information pertaining to the system as well as assist setting up components and sensors remotely.

There are many settings within the Command Center that the home-owner can adjust. The primary setting is how they want to receive notifications and which ones they prioritize. Relating to the video feed, the user may wish to only be notified for intrusions that they consider a threat. If the system classifies the intrusion as a deer, then the person would not be notified directly but the notification would still be archived. If the intrusion is a person, however, then the notification would be sent directly to the home-owner. Similarly, the home-owner can set up the system to automatically notify the authorities for specific sensor readings. If a gas leak is detected or a smoke alarm stays on for a certain duration of time, then the fire department can be alerted. Or if a person is detected attempting to break into the home then the police department can be immediately notified. If the user gives Protec permission, the image notifications from their video feed will be used to continue to improve the classification model used in order to provide the most effective home protection possible.

Conceptual Domain Model:



Linguistic Analysis:
Some of the classes I decided to eliminate include:

- **Door/window/entry point** – since these are not actually part of the system (and the locks and sensors are), it would be irrelevant.
- **Protec Devices** – Sensors and motorized locks are types of devices so the complete blanket term would be too vague. I considered eliminating **sensor** due to it being a bit vague, but I felt that it correctly described what the class is. Sensors all have shared methods and so making them all their own class would be redundant itself. Instead, I decided to rely on inheritance here.
- **Detected intrusion:** The intrusion detection is done by the camera. I felt that this would be considered implementation and thus a bad class to add to the system.

Actors:

I decided to make the security specialist and Protec itself actors. The security specialist directly interacts with the system settings and sets up devices. Further the specialist and the home-owner interact with each other. The ‘Protec’ actor refers to the classifier using the system’s data, if allowed, to further train their model as well as any other data the home-owner is willing to provide.

Constraints/Invariants:

Note: I was unsure if these were supposed to be connected directly to the diagram or if they were to be written separately. One the first line for each, I listed the class that the constraint applies to.

1. **(on gas leak detector)**

permittedLevelSet

{{OCL} self.permittedLevel > -1}

2. **(on gas leak detector)**

detectLeak

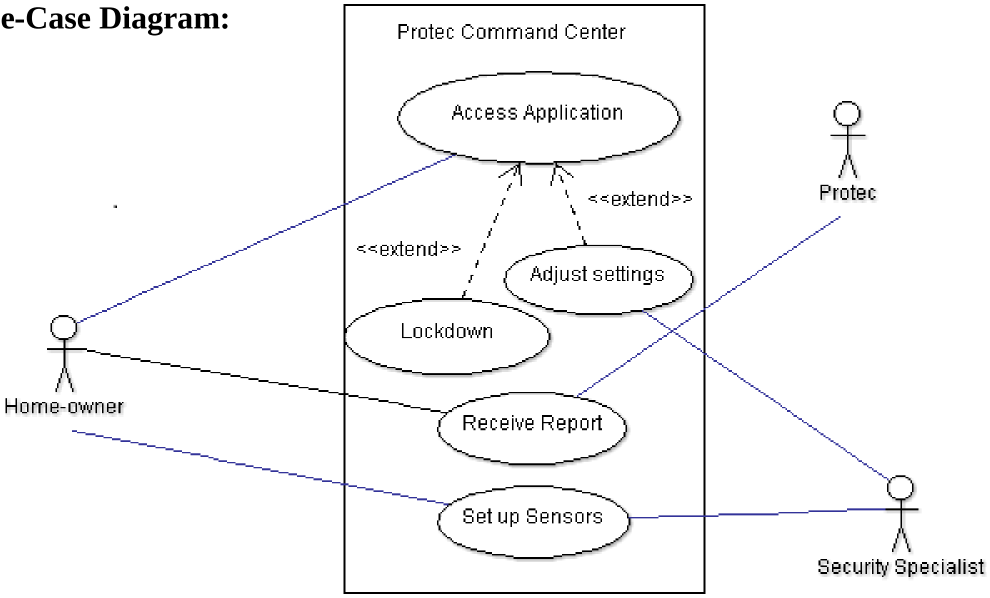
{{OCL} self.leakDetected = self. permittedLevel <= self.currentLevel}

3. **(on personal settings)**

phoneNumberNumeric

{{OCL} self.phoneNumber.matches(‘[0-9]*’)}

2. Use-Case Diagram:



3. Detailed Use Case:

Use Case Name: Lockdown
Participants: Home-owner
Pre-condition: Customer needs to be logged into the application in order to control the locks.

Typical Course of Events:

Actor Intentions	System Responsibility
1. Home-owner accesses the application and uses the "Lockdown" button.	
	2. Prompts the home-owner for confirmation.
3. Accepts lockdown confirmation.	
	4. Activates the motorized locks that are currently not already locked.
	5. Once all entries have been locked, stops the motor and notifies the home-owner that the entries are locked.
	6. Return user to main application menu.

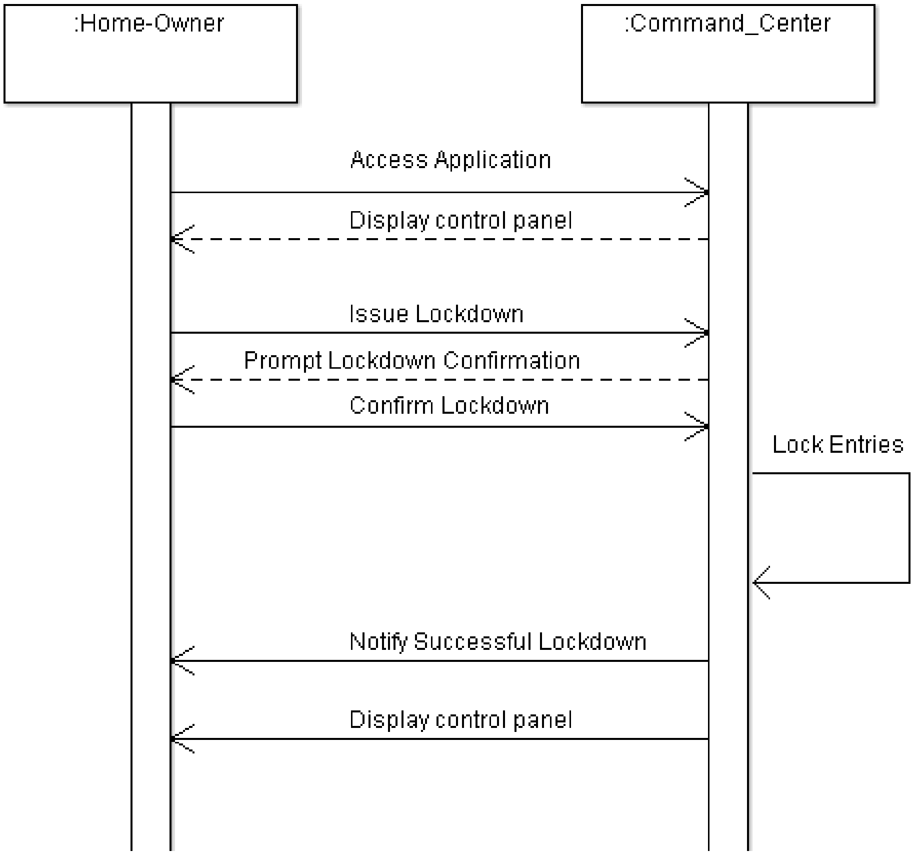
Alternative Courses:

- **Step 5:** If an entry was not closed then the lock cannot secure. Notifies the home-owner that the lockdown was not completed, and which entry needs to be securely shut. They can close the door and then restart the process to lock the final entry (or lock it manually).
- **Step 3:** The home-owner accidently clicked the "Lockdown" button. They cancel the lockdown request and the system skips to step 6.

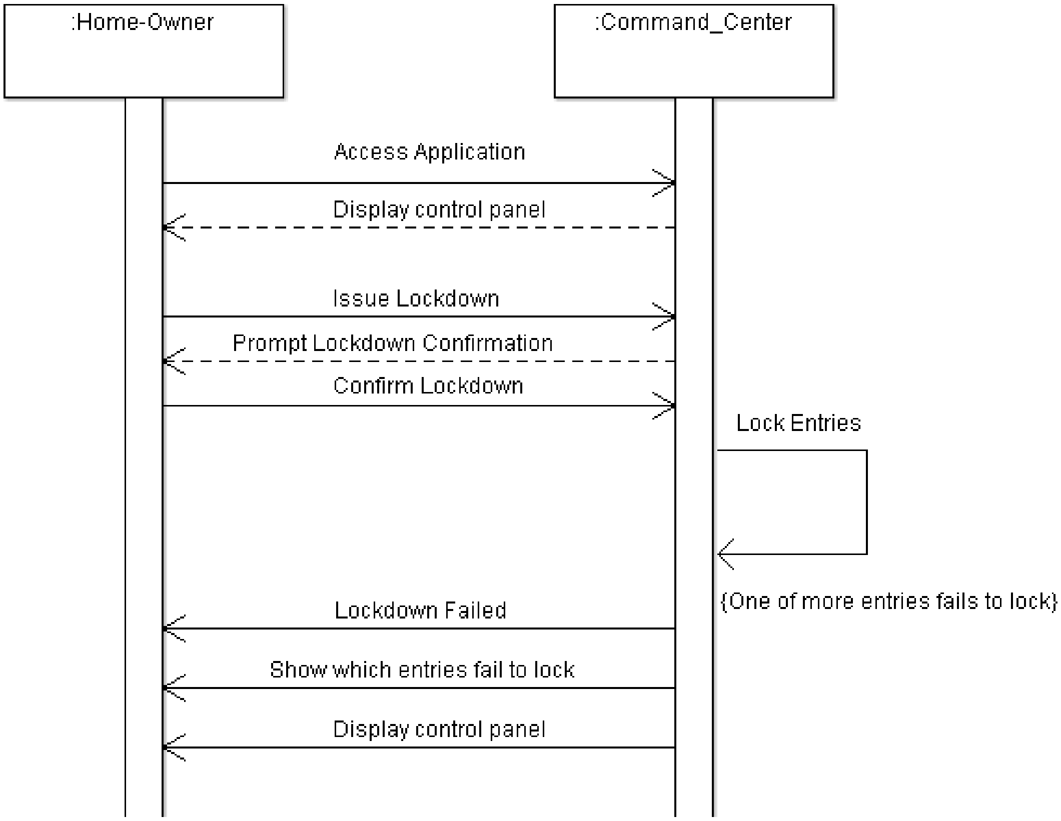
Post-condition: All the entries are securely locked or the home-owner has been notified as to which entry needs to be closed for the lockdown to be completed.

4. High-Level System Sequence Diagrams:

Scenario 1: Typical flow of events from above. The home-owner uses the interface to access the application. From within the application, the home-owner presses the “Lockdown” button. The system responds with a confirmation. The home-owner confirms the lockdown and the system proceeds to activate the motorized locks for unlocked entries. Once the doors are locked, the motors stop and the home-owner is notified that the lockdown is complete. The user is then returned to the main application menu.



Scenario 2: The first alternative course of action where the lockdown cannot be complete due to an open door. The events start the same way. The home-owner uses the interface to access the application. From within the application, the home-owner presses the “Lockdown” button. The system responds with a confirmation. The home-owner confirms the lockdown and the system proceeds to activate the motorized locks for unlocked doors. The system activates the motors for the entry locks and locks them. Some of the locks fail to fully secure themselves because some entries are not fully closed. The system sends the home-owner a notification that the lockdown cannot be complete and that some entries are still unlocked due to being ajar. The home-owner is returned to the application control panel and can try again to finish the process.



Sources:

https://help.eclipse.org/oxygen/index.jsp?topic=%2Forg.eclipse.ocl.doc%2Fhelp%2FOCLExamplesforUML.html

I used this page for guidance on the OCL.