Use-case diagram:

In UML, use-case diagrams are made to create a model for the behaviour of the system to help the developer in capturing and eliciting the requirements of the system. Also use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors (in our case the actors are the Pharmacist/User and the Doctor/Admin). The use cases (the tasks that take place in the system) and actors in use-case diagrams describe what the system’s purpose is and how the actors use the system, but not how the system operates internally.

Use-case diagrams illustrate and define the context and requirements of either an entire system or the important tasks of the system. In our Pharmacy system, our important tasks include: “Adding a new Medicine”, “Deleting Existing Medicine”, “Searching for Medicine using the input data”, “Updating Existing Medicine”. You would typically develop your use-case diagram in the early phases of your project and refer to it throughout your development process.

Our “Use-case” diagram is meant to help us in the following situations:

Before starting a project, we have created our use-case diagram to model our new Pharmacy system so that all participants in the project share the same understanding of the workers, customers, and activities of the business.

While we were gathering our system’s Functional requirements, we created our use-case diagram in order to capture the system’s main function to present it to the developers for them to know what the system should do.

During the analysis and design phases, we are planning on using the use cases and actors from our use-case diagrams to identify the classes that the system requires.

During the testing phase, we can use the use-case diagram to identify the tests we need to take for our system.

If you are having difficulty understanding our Use-case diagram, Read the following description of our use-case diagrams:

Use cases  
A use case represents all the tasks included in the system. These tasks should also be stated in the functional requirements of the business. Our tasks include:

1. Login: The login acts like a gate way into the system. It takes an input for the username and password and must verify them before letting the user into the system. However, if the input had and error a message will pop up saying: “Invalid input”.
2. Input Medicine Information: In our system we will include a place for the user to input all the necessary information needed to perform the other tasks such as adding, deleting, and updating for medicine.
3. Clear Medicine Information: we want a button to clear all the information instead of deleting them one by one.
4. Exit: a button to exit the search area.
5. Add Medicine, after filling up the medicine information, we plan to have a button that adds a new medicine into the system. However, the button wont work unless the Admin gives his approval.
6. Update Existing Medicine: When we select a specific medicine, we expect the medicines information to appear. When the information appears, we want to be able to edit it and then click the update button in order to save the information.
7. Search for Medicine using data: after filling up the medicine information, we will make a search button in order to find our medicine faster.
8. Search Option: if our user doesn’t know all the information about the medicine, they will be able to filter the medicines in order to limit the medicine options. (Search for Medicine by filtration).
9. Delete Existing Medicine: after search for the medicine using the medicine information, we plan to have a button that deletes an existing medicine from the system. However, the button won’t work unless the Admin gives his approval.

[Actors](https://www.ibm.com/docs/en/SS8PJ7_9.6.1/com.ibm.xtools.modeler.doc/topics/cactor.html)  
An actor represents a role of a user that interacts with the system that you are modelling. The user can be a human user, an organization, a machine, or another external system. Our actors include:

1. Doctor/Admin
2. Pharmacist/User

[Relationships in use-case diagrams](https://www.ibm.com/docs/en/SS8PJ7_9.6.1/com.ibm.xtools.modeler.doc/topics/crelsme_ucd.html)  
In UML, a relationship is a connection between model elements. A UML relationship is a type of model element that adds semantics to a model by defining the structure and behaviour between the model elements. Our relation include:

1. Include relations:
   1. Between login and the verification of the login information.
   2. Between adding a medicine and the admins approval.
   3. Between deleting a medicine and the admins approval.
2. Extend relations:
   1. Between login and the “Invalid Input” message.