**PROJECT REPORT**

Introductors: Prof. Nguyen Thi Thu Trang

**GROUP 3**

*Phạm Thanh Trường 20194460*

*Phan Mạnh Tuấn 20194461*

[Company name] [Company address]

**Demonstration of types of viruses and its mechanism**

*Object – oriented Programming*

# Asignment of members

Pham Thanh Truong 20194460:

* UseCase diagram
* Build screen
* Build controller
* Fix some parts in model parts

Phan Manh Tuan 20194461:

* Class Diagram
* Build package Virus
* Build package Cell
* Fix some parts in GUI

# Mini – project description

# Describe in detail about mini-project requirement

Specifications:

**- GUI:** Design the GUI with your favor

**- Design:** the application must have these functions:

*+ On the main screen:* Title of the application, options to choose between virus with lipid envelop and virus without lipid envelop, help menu and quit.

• User can choose to investigate one of the two types of viruses in the main menu to start the application.

• After choosing the desired type, the application will show a variety of viruses in order for user to select.

• The help menu shows basic usage and aim of the application.

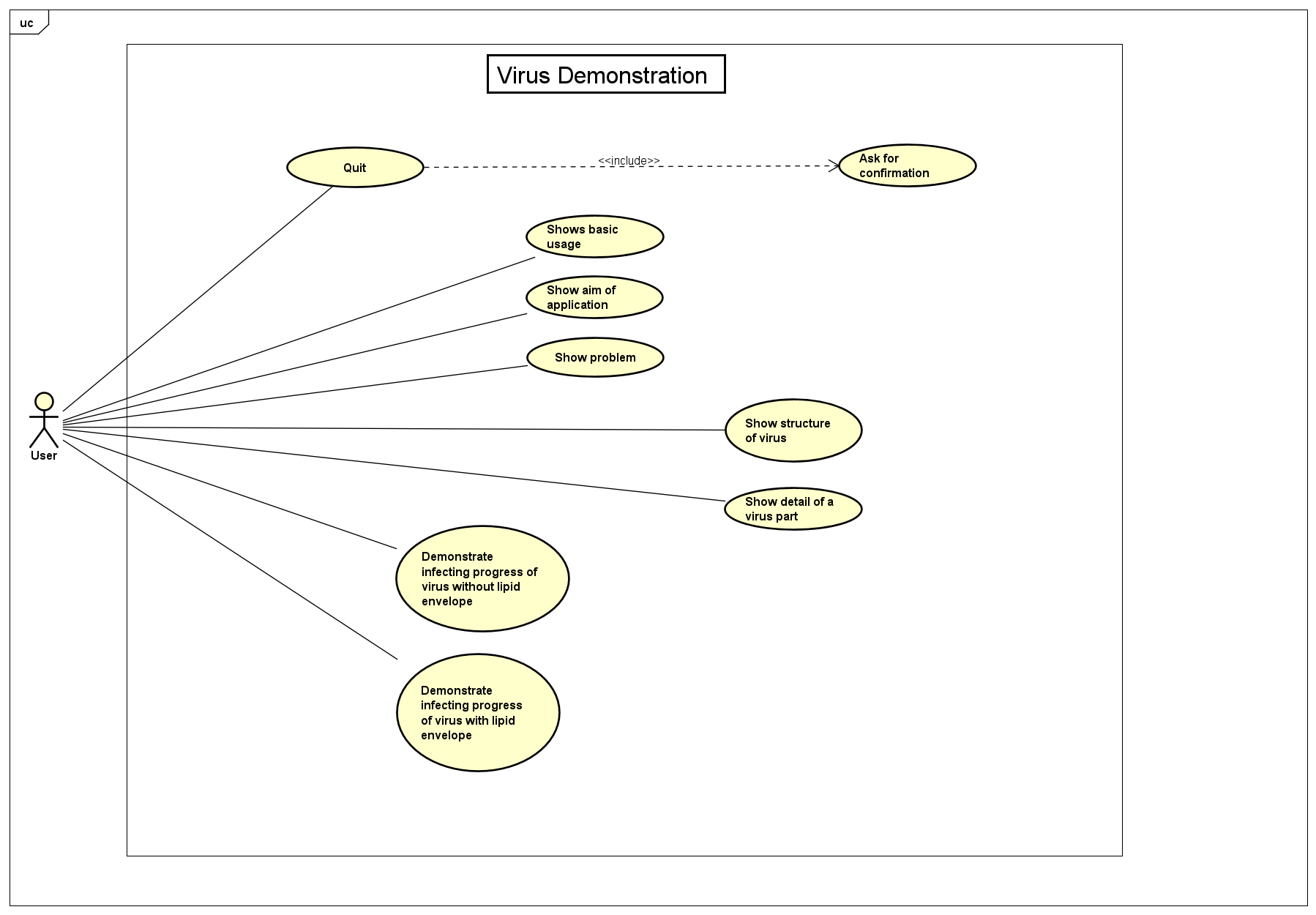
• The quit button exits the application. Be sure to ask for confirmation.

*+ In the demonstration:*

• Display the structure of the virus. Note that each virus has different structure, you should clearly display and explain them.

• One button to start demonstrating the progress of virus infecting the host cell. Different viruses have the same basic mechanism of spreading with minor difference - remember to show that There is always return button for user to get back to the main menu at any time.

# Usecase and explanation



*Explaination:*

* First of all, our window always have a quit button with ask for confirmation.
* In the main screen, we has “Help” menu which contains “Usage”, “Aim of application” and “Problem” items on the top of scene. In the center has a split pane contain 2 list of virus (with and without envelope). Choose a virus to “show structure of virus”.
* In the StructureScreen, if you choose a part, screen will show the detail scene of that part.
* In the StructureScreen, you also can choose to demonstrate infecting process to move to InfectingScreen.
* Every Screen have “Back” button or “Quit” programme.

# Explanation of design ideas

## *General class diagram*

## https://media.discordapp.net/attachments/794488057562529802/848583042976317470/class_diagram_topic7.png?width=561&height=630

**General class diagram:**

Packages:

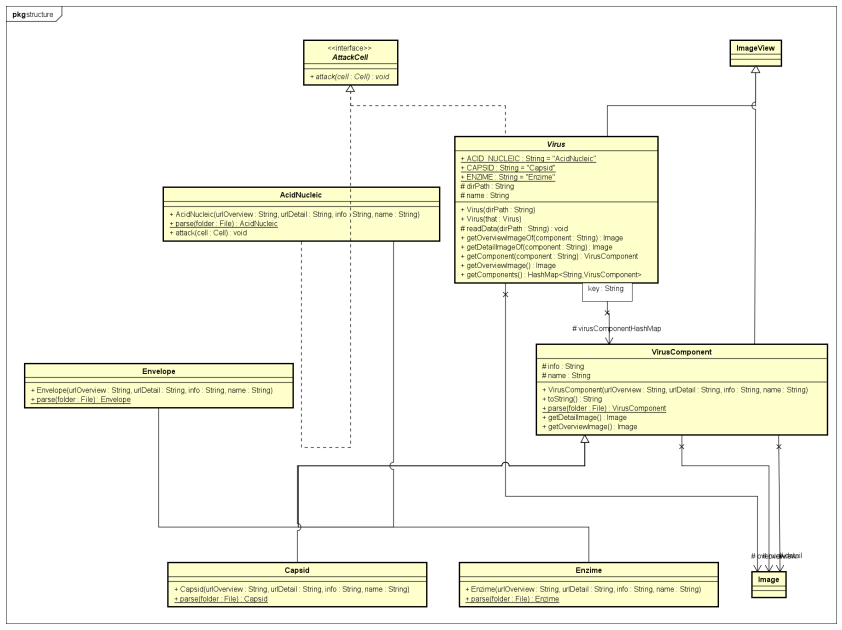
+ Package “model”: store all components of viruses and cell class.

+ Package “controllers”: store all the screen controllers.

+ Package “screen”: store the main screen class of the application.

## *Class diagram for each package*

* VirusStructure

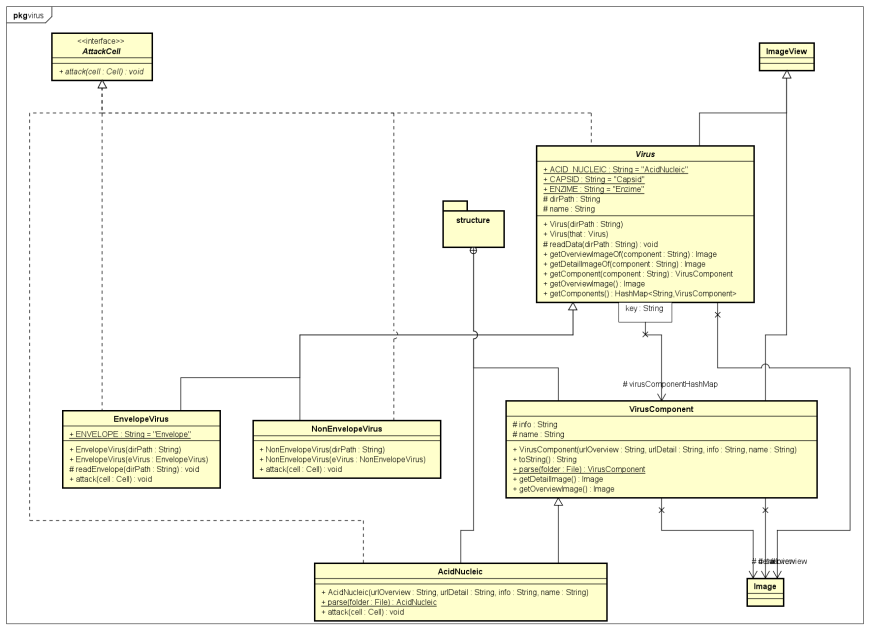


“Viruscomponent” stands for a part of real Virus.

Parts of Virus: AcidNucleic, Enzime, Capsid, Envelope, ... Inheritate from VirusComponent and has some specific methods. (Inheritant)

For example: AcidNucleic has specific method “attack” to model the process virus attack, then virus inject to nucleus and start spreading.

* Virus

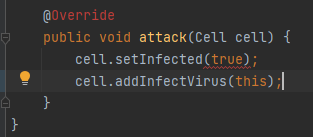


“Virus” will have a virusComponentHashMap to store all the components of a virus.

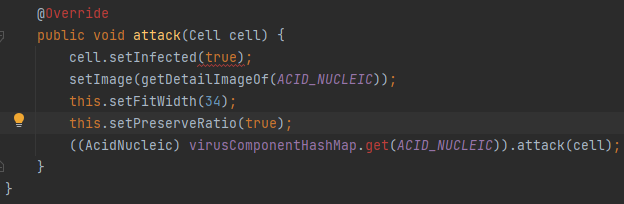
NonenvelopeVirus, EnvelopeVirus inheritate from Virus and implement “AttackCell”. (Inheritant)

+ attack : Many way to attack depends on Virus with envelope or Virus without envelope or Nucle (Polimophism).

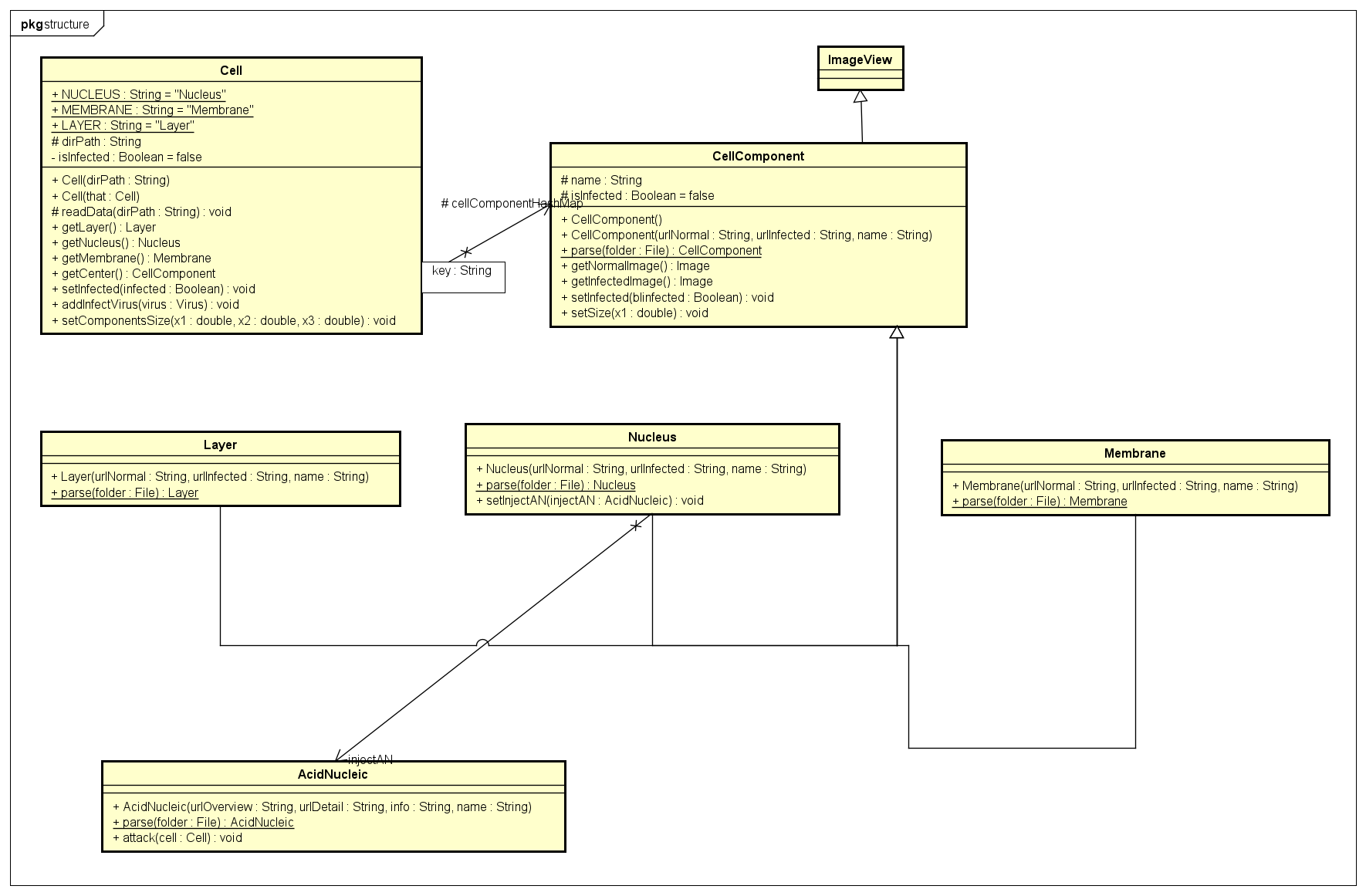
* Virus with envelope attack: Virus will break into cell and start spreading.



* Virus without envelope attack: Virus will use its acid nucleic to incorporate with opponent’s nucleus. When the cell divide into two or more, the number of infection cell will increase.



* **CellStructure**



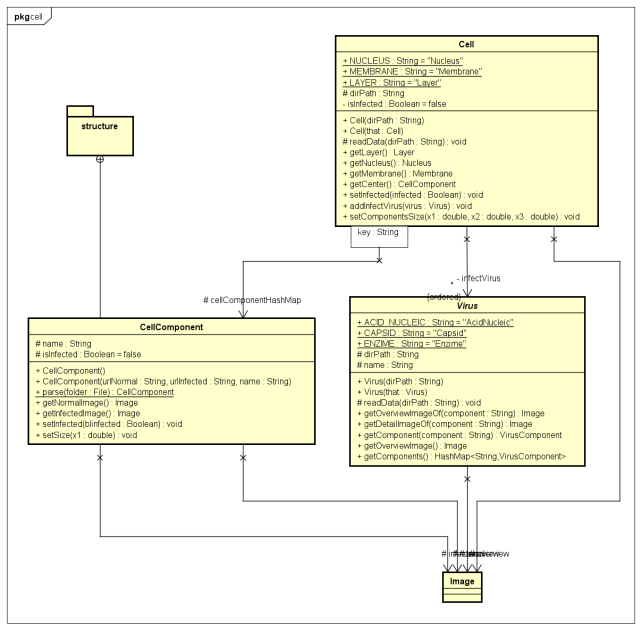
Similar to Virus, we also create class name CellComponent to abstract CellComponent in real life. Parts of Cell: Layer, Membrance, Nucleus inheritate from CellComponent with some specific attributes and methods. (Inheritant)

CellComponent inheritate from ImageView, we set the image for CellComponent base on the state of the Cell.

+ Image normal : Store normal image of CellComponent

+ Image infect : Store infected image of CellComponent

* **Cell**



“Cell” will have a cellComponentHashMap to store all components of cell.

+ dirPath: path to the directory contains all the datas of cell.

+ isInfected: State of Cell is being attacked or not.

+ cellComponentHashMap: contains all Cell component.