HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
GLOBAL ICT K67

VIRUS DEMONSTRATION

OBJECT-ORIENTED PROGRAMING

PROJECT REPORT

MEMBER LIST

Trinh Manh Quynh	20226064	Quynh.TM226064@sis.hust.edu.vn
Le Nhu Ngoc Son	20226066	Son.LNN226066@sis.hust.edu.vn
Vu Ngoc Dung	20226032	Dung.VN226032@sis.hust.edu.vn

Lecturer: Tran The Hung, PhD

Hà Nội — 2024

CONTENTS

1 Project Assignment

We will inform the tasks each member has completed by listing the work he involved in this project.

Number	Full Name	Student ID
1	Trinh Manh Quynh	20226064
2	Le Nhu Ngoc Son	20226066
3	Vu Ngoc Dung	20226032

Task	Responsible Member ID	Percentage Contribution
Class Diagram	Vu Ngoc Dung	70%
	Trinh Manh Quynh	30%
Use-case Diagram	Vu Ngoc Dung	100%
Components Package	Vu Ngoc Dung	70%
	Trinh Manh Quynh	30%
Main Frame Package	Trinh Manh Quynh	100%
Virus Data	Le Nhu Ngoc Son	90%
	Trinh Manh Quynh	10%
Report	Trinh Manh Quynh	100%
Slides	Le Nhu Ngoc Son	100%

2 Project Description

2.1 Mini-Project Overview

In this project, we choose to build a virus encyclopedia. A place like wikipedia, contains information about viruses, divided in two groups, Envelope and Non-envelope viruses. Users can look up for information of viruses, either in a dictionary or by search function. More detail structure about the program will be shown later on in a Use-case diagram and class diagram.

2.2 Mini-Project Requirement

For a better understanding of the project, here are the requirements that we need to follow during the process:

1. **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.

2. **In the demonstration**:
   - Display the structure of the virus.
   - One button to start demonstrating the progress of virus infecting the host cell.
   - Return button to return to the main menu.

Figure 1 illustrates a demo interface showcasing the features of the simulator that we have to develop.

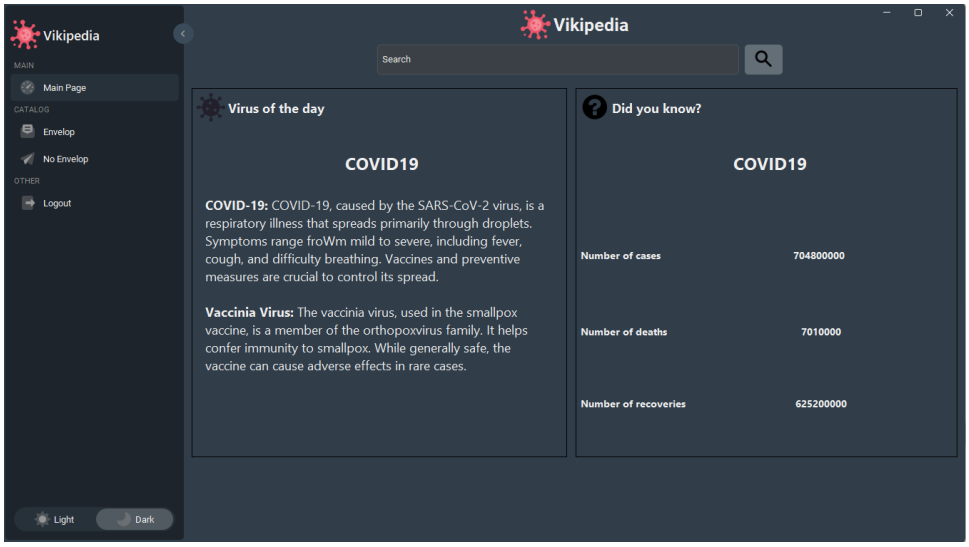

Figure 1: Sample GUI

2.3 Use-case Diagram

Figure 2: Use-case Diagram

2.3.1 Main screen

This is the main screen of our application. There is a side menu with different navigation, including Main menu, help menu and quit button.

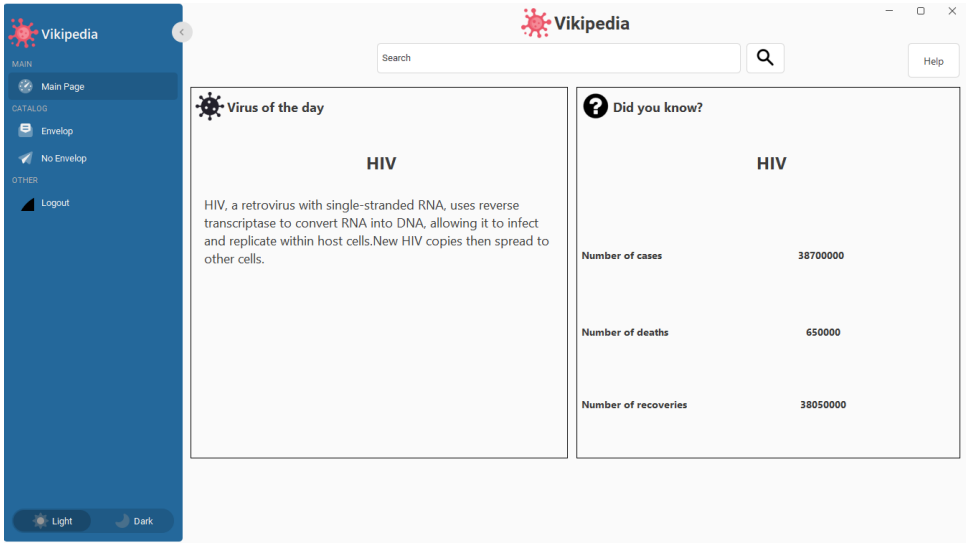

Figure 3: Main Menu

2.3.2 View help message

Firstly, users can choose Help button which gives them access to the help screen, where they can receive information on the aim of the application and how to use this program effectively.

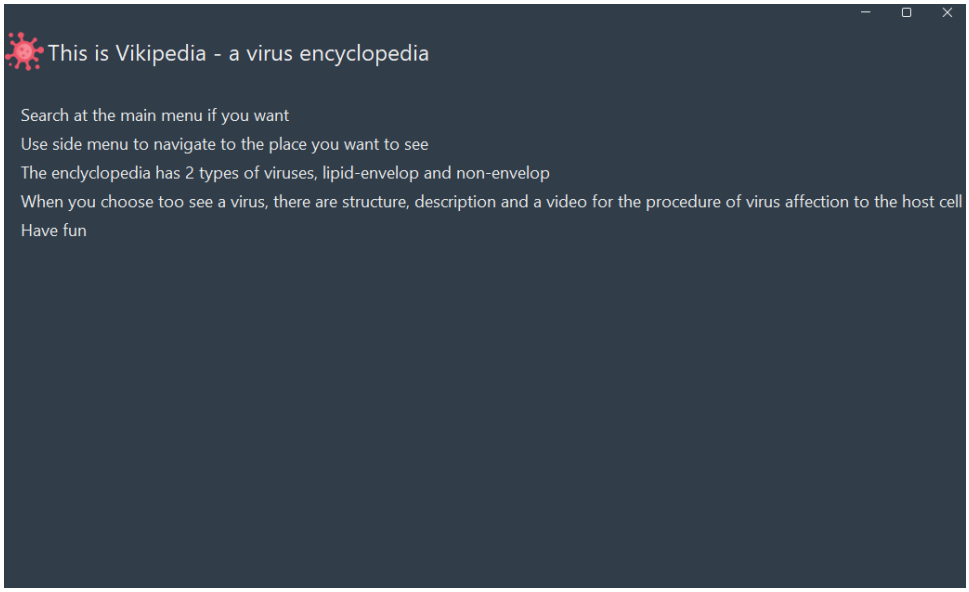

Figure 4: Help Screen

2.3.3 Operate on the dictionary

Either the user choose to search for the name of the virus, or use the side menu to navigate the desired virus. It will pop-up new frame of the virus information.

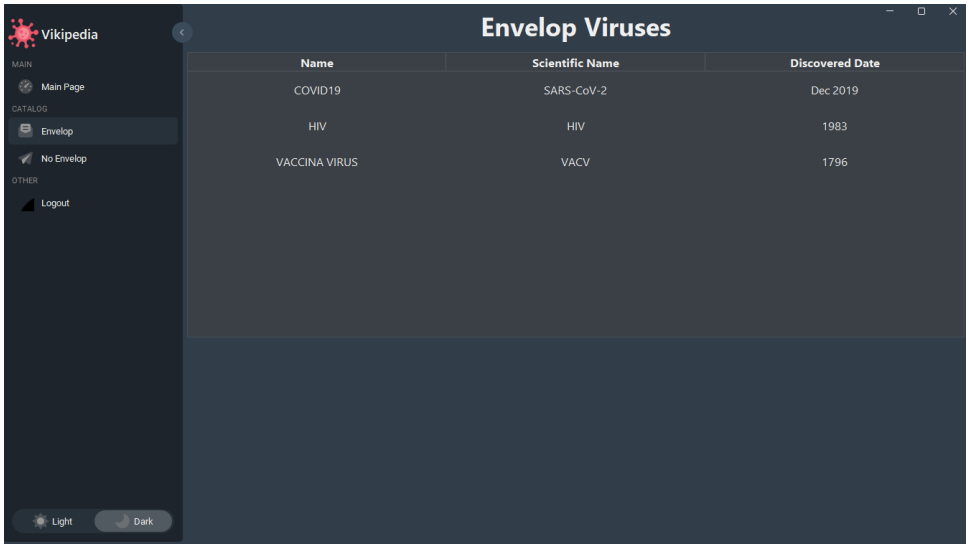

Figure 5: Choose Data Structures Screen

In the frame, there is video that represent the procedure of virus in affecting the host cell. User can choose to play, skip or rewind (10s).

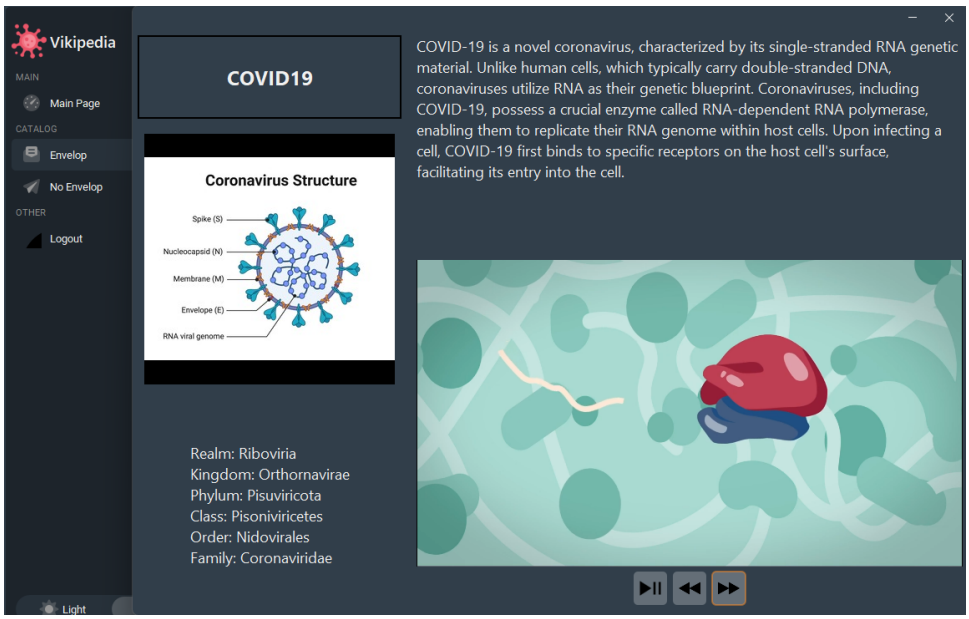

Figure 6: Initialize Data Structure Screen

3 DESIGN

3.1 Libraries

Since this application is an encyclopedia, regardless the GUI is not a requirement for the project, we notice that user experience in the program itself is a crucial part for an encyclopedia. Therefore, we need addition libraries. Below are the libraries we use:

- flatlaf: Java look and feel
- vlcj: video player for java swing

3.2 General Class Diagram

Our General Class Diagram consists of 5 main packages: **application**, **data**, **icon**, **menu** and **theme**.

The **application** package contains most of the main class of the program, but those are the GUI classes. The code under the hood responsible for the code logic is in **data** package, with some of the virus data (image, video). The **menu** and **icon** packages contain the images and icons for the GUI application.

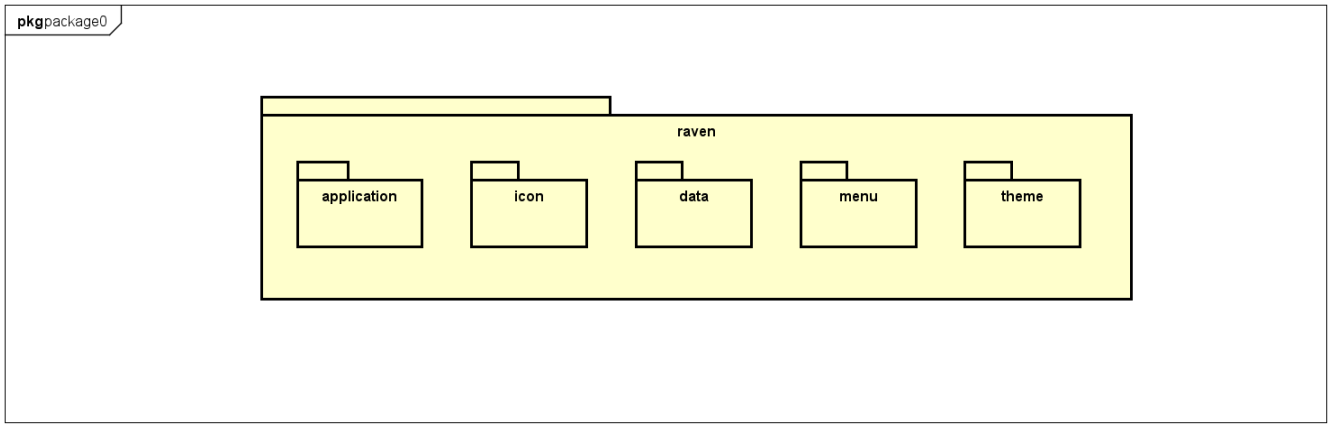

Figure 7: General Class Diagram

3.3 Package Details

3.3.1 raven.application package

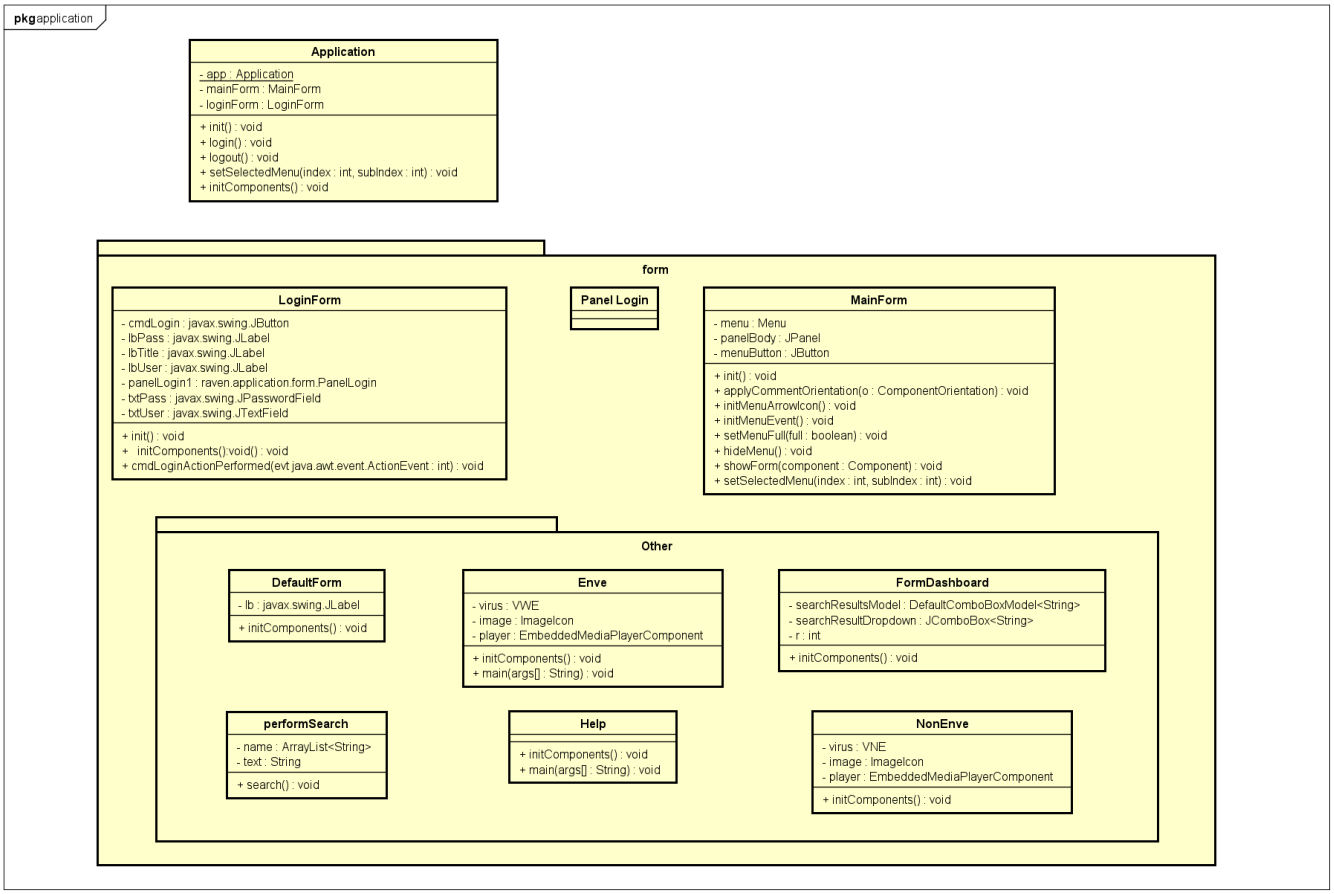

Figure 8: Package **raven.application** Class Diagram

In ***raven.application*** package, we have most of the program's GUI classes. Including LoginForm, MainForm, PanelLogin. Inside the package itself contains the form of the catalog, for different viruses. All of the classes use method **initComponent()** to initialize the required components.

3.3.2 raven.data package

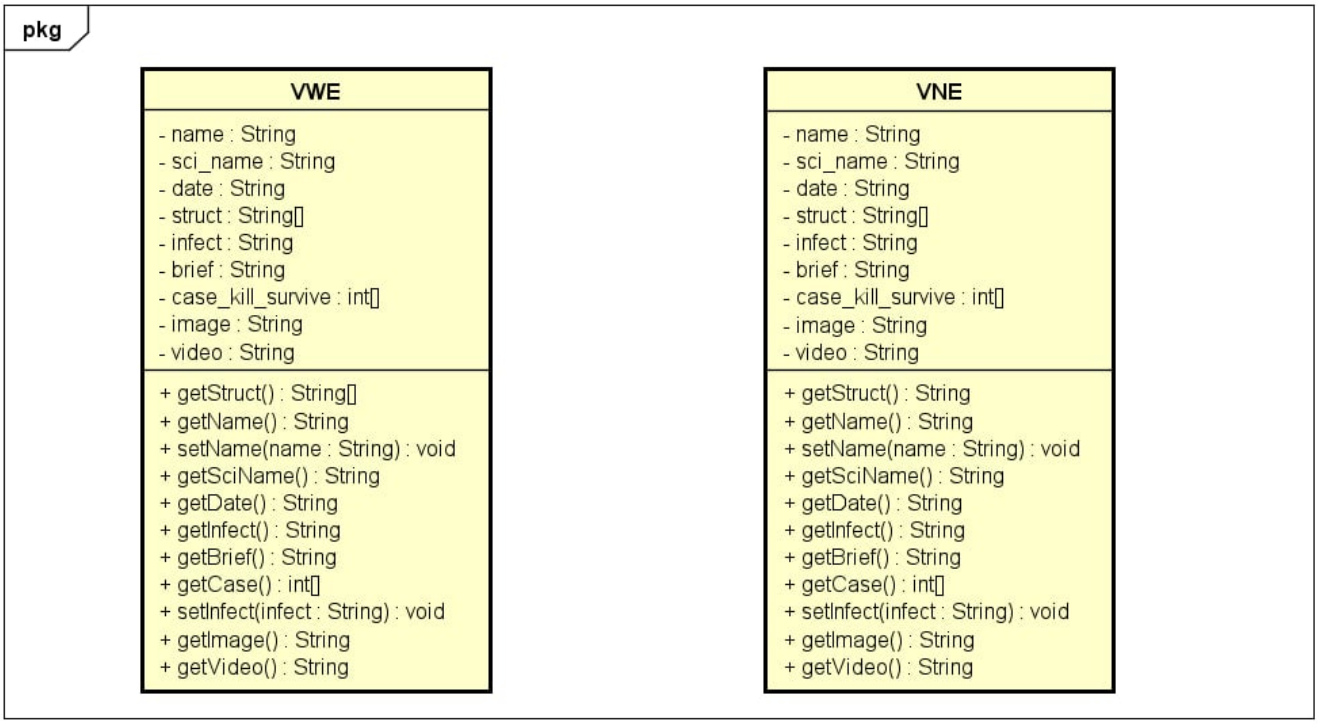

Figure 9: Package **raven.data** Class Diagram

This is the backbone of the program, in this package, beside of the images and videos of viruses, the main classes are VNE and VWE, stand for Non-envelop and With-Envelop, respectively. Both classes contain the atrribute:

- name: Virus's name
- sci_name: Scientific name of the virus
- date: Discovery date
- struct: The structure of the virus, provided by a brief description
- infect: A short paragraph of how a virus affect a cell.
- case_kill_survive: A list of int contains number of cases, death and survived.
- image, video: contain the path to the corresponding form of file.

Those viruses then initialized when the application is run.

3.3.3 the rest packages

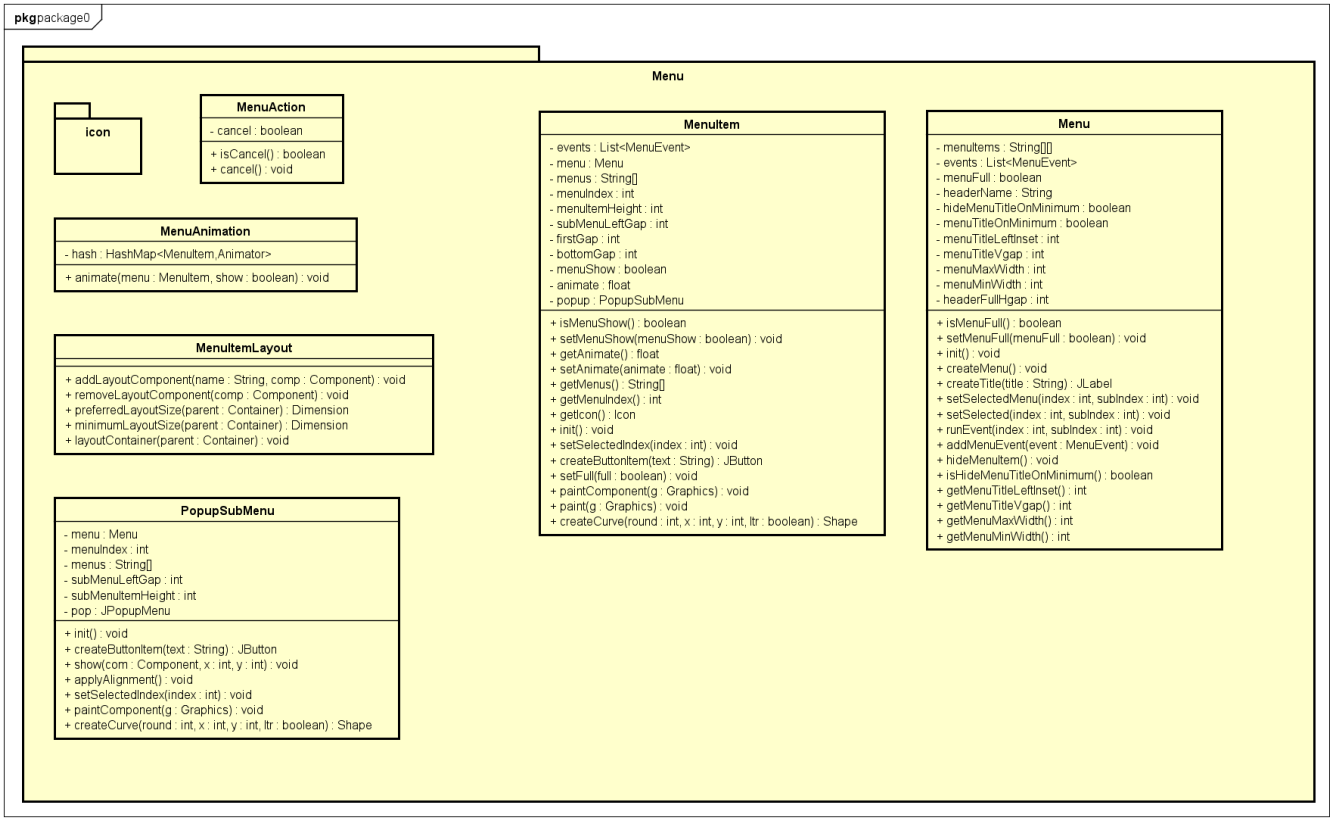

Figure 10: The **Menu** and **Theme** Class Diagram

The rest of the packages act as the utility of the program. **raven.icon** contains the icon of the menu application, in the SVG form. **raven.menu** contains classes for the menu of the application, with classes:

- Menu: The backbone of the program's menu. Here we need to specify that we use a Swing's look and feel to make the UX more dynamically. And contains the function to switch the screen mode between light and dark (this logic is contained in the mode package, inside the menu package).
- MenuAction: contains logic for cancel and open the side menu.
- MenuAnimation: The animation of the side menu
- MenuEvent: An interface for classes that emplement it (usually the menu classes)
- MenuItem: Responsible for the icons in the side menu. Listed in the array of String.
- MenuItemLayout: The design of the icons.
- PopupSubmenu: Submenu logic and design.

4 REFERENCES

This project greatly benefited from the knowledge and information provided by the following sources:

1. **FlatLaf** Library
   `https://www.formdev.com/flatlaf/`

2. **Vclj** Library
   `https://capricasoftware.co.uk/projects/vlcj`
   `https://github.com/caprica/vlcj`