Final Report

Safety Critical Computer Systems

Author:
Tran Hoc Phuc – 1235133
Le Thanh Tung – 1235175
Nguyen Tran Minh Luan – 1235081
Szu-Chi Huang – 1358609
Duong Thanh Minh – 1235094

Supervisor:
Prof. Dr. Matthias F. Wagner

March 5th , 2021

Table of Contents

List of Figures

List of Tables

I. Status report

a. Features:

As of March 5th, the development of the Vital Covid-19 Information application of our project is finished. The application's features are:

- User account database system to store user's specific information (state, district) for quicker viewing of desired information. User can also modify their previously saved info: password, state or district.
- Viewing of Covid-19 statistics on a global, German states or German districts scale. Information is retrieved from a third-party online database through third-party APIs.
- Self diagnosis function for the viewer to learn how likely it is that the user has been affected by Covid-19 through symptoms. If the user's symptoms are not found in this self-diagnosis function, it means that Covid-19 does not cause those symptoms.
- View Covid-19 regulations information, however, only information for several most populous cities of Germany are included.
- View general healthcare tips that one should do to lower the chance of getting infected with Coronavirus or the course of action that one who suspects they may be affected by Covid-19 should take.

b. Improvements:

There are many ways to improve this application that we couldn't implement in time:

- Online database: We first decided to implement the user account and database system to make the usage of functions faster and easier for users and backup user's data in case of data loss on the local machine or if the users need to switch devices. This means it would make much more sense to use an online database over a local database, and only that online database is more challenging and more time-consuming to set up.
- Save the user's last answers and results to the self-diagnosis questions and results.
- Graphical icons to decorate the different functions of the user interface.
- A feature that uses the user's postal code to determine their city and state so that the user does not have to enter it manually.
- Authorized accounts to help better monitoring of the user accounts database.

c. Design problems:

- At first, we planned the area regulations function of the project to be able to cover more areas, however, as we proceed with the development, we learned that unlike the Covid-19 related specific statistics, no information source covers the Covid-19 related regulations for a large scale area/region. So we decided to hard code in the information on Covid-19 of only several of Germany's most populous cities.
- We planned to use the user account system and database to save more of user's information, such as user's name, age, address. However, it turns out that some of this personal informations are irrelevant to the functions of this application, so we decided not to include it in the final application build.

II. Design model

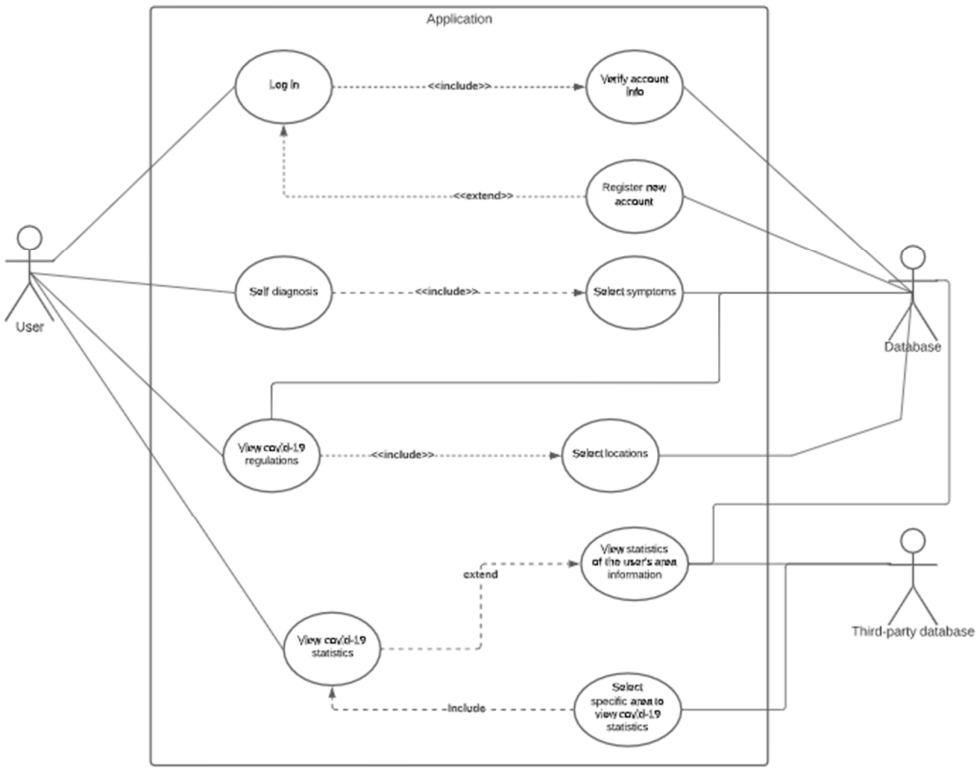

Figure 1.Use case diagram

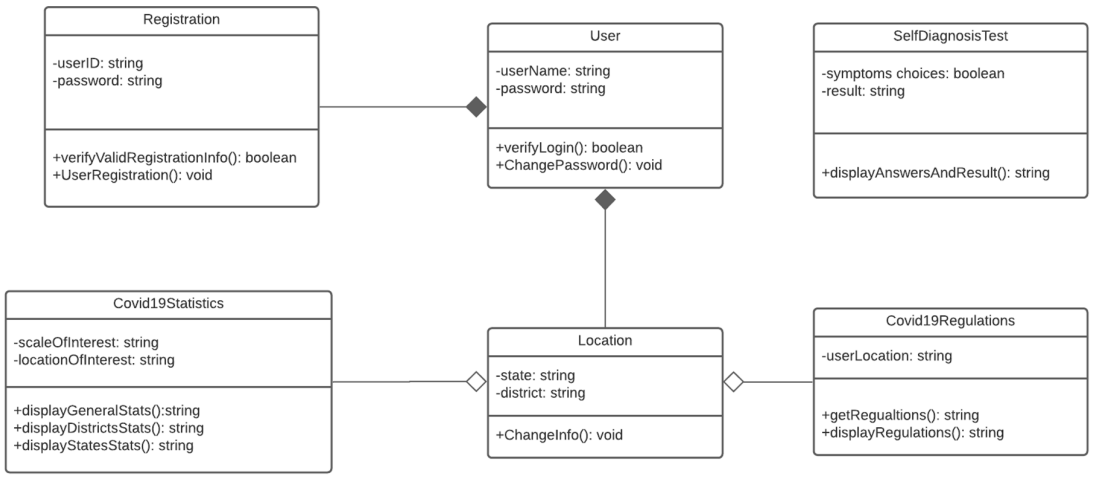

Figure 2. Class diagram

III. HMI design

a. Purpose of the Human-Machine Interface:

The application does the analysis base on the information provided by users. In order to make the data collection process more understandable and convenient for the users, a simple graphical user interface of the application is required.

b. The tool for GUI design:

For this application, we used Netbeans IDE. The tool provides a visual preview of the GUI that makes it easier for testing and debugging. The GUI component (i.e. Checkbox, input text field, etc) can also be placed and modified with drag & drop.

c. The fundamental component for the data collection process:

1. Text field: the information can be input using a keyboard.
2. Combobox: the box that included some information for the users to choose.
3. Check box: often use to collect data for Boolean questions.
4. Button: often use to submit the data or move between the frames.

d. Frames of GUI:

+ There is only one frame show up at a time, with the help of buttons, the user can switch between frames.

+ The application starts with login/register frame. After successful login, the main frame shows up with 3 options for the different types of analysis.

+ Base on the user's selection, the respective frame for the analysis is displayed.

IV. Software plan

Due to the submission deadline's extension, we are given more time to develop the project more thoroughly. The UI development and user database (along with the Login/Register/Logout options) took a significant amount of time compared to other tasks. Typically, all other features took the same amount of 14 days, while the UI development and user database took 26 and 13 days, respectively. The testing and debugging procedures are finished, and we are expected to complete the documentation on 05.03.2021.

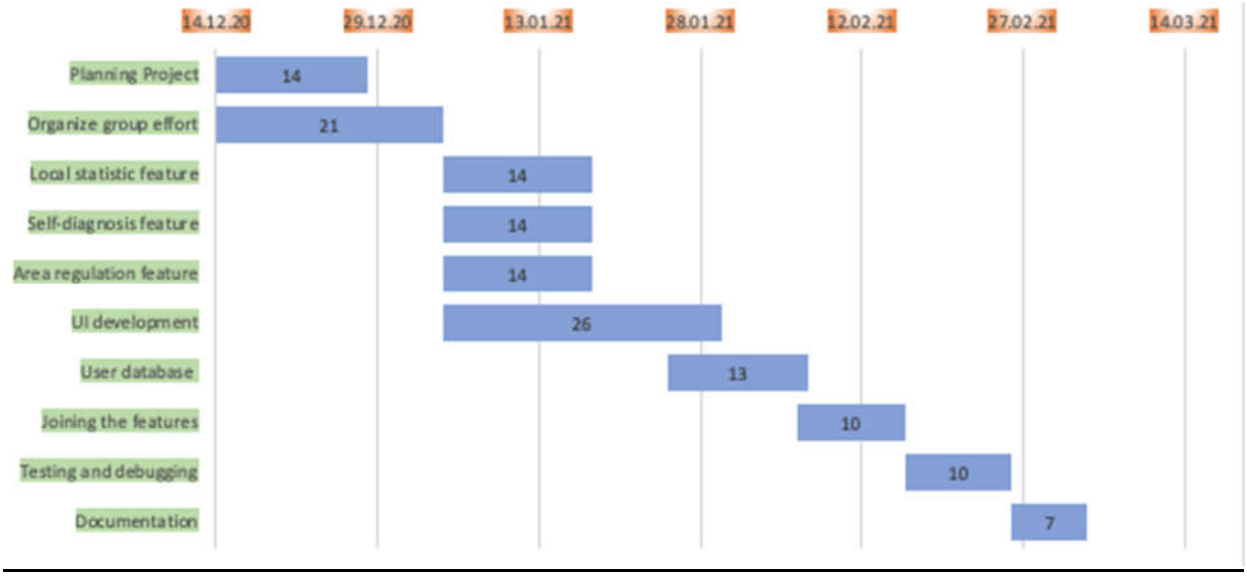

Figure 3. Updated software plan

As shown in the figure above, the blue bars indicate how long it took to finish a task (in days), the green text explicitly specifies each task, and the orange one gives a rough estimation about the date.

V. Hazard analysis

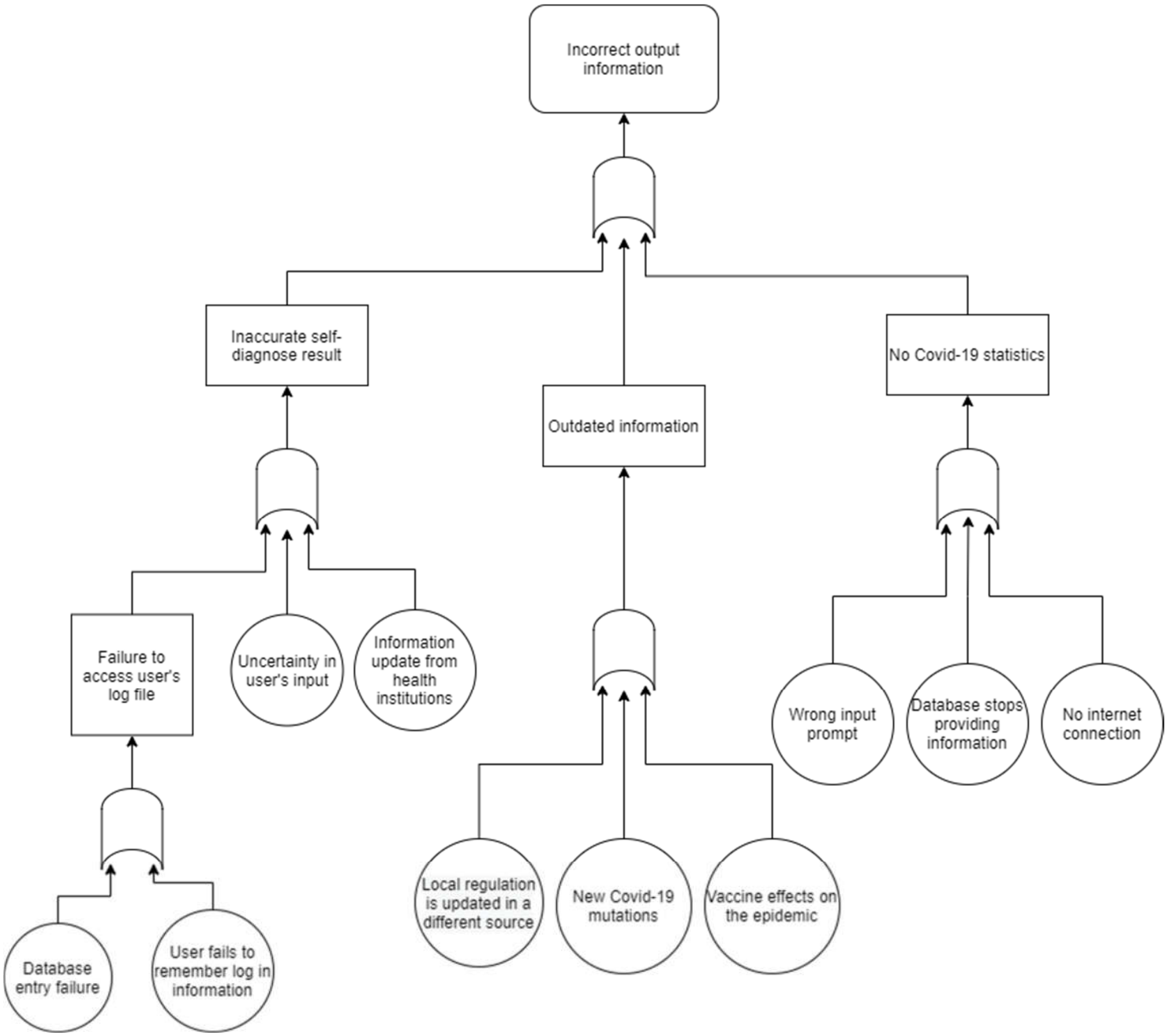

Figure 4. Fault tree analysis of the application

VI. Safety plan

a. User Privacy:

For the safety of the user, their personal information like location, name, and age are kept private. We implemented a local H2 database, which means that the user database is created on the device itself and not connected to the internet, thus reduces the possibility of it being hacked. The database contains user information and provides login/registration features for the application. Each user can register with a sole username that belongs only to them and a password for that username.

b. Network issues:

In the case of the internet connection failure, a warning should be prompted for the user. No information is retrieved if there is no connection, thus the Covid local statistics, self-diagnosis and area regulation features should not be accessible.

c. Self-diagnosis result accuracy:

There are many elements influencing the result: Covid-19 vaccinations, new mutations discovered (thus symptoms and lethality of the virus significantly change), and the vast differences in symptoms experienced by each infected person. These reasons could lead to failure in analysis results, which could lead to a dangerous outcome.

A false negative result from the application could make users feel complacent that they do not get Covid-19 while they do, thus unknowingly, they became a source of infection. There should be a warning that the result is not entirely accurate, and the user should take the diagnosis as a reference only, further check with physicians is necessary if the user felt unwell. The application should also have the feature for users to update their health status for better accuracy of the analysis.

VII. Security plan

a. Application/System identification:

Application requirements	High	Medium	Low
Confidentiality	User's registration/database		User's login/logout
Integrity		Self-diagnosis test	Area regulation
Availability		User interface	Local area statistic

Table 1.List the types of sensitive information the application/system accesses

b. Management controls:

Planning for Security in the Life Cycle: Determine which phases of the life cycle the application/system, or parts of the application/system, are in. Identify how security has been handled during each of the listed applicable life cycle phases.

- Initiation: Planning Project
- Development: Programming code
- Implementation: UI development
- Operation/Maintenance: Testing and Debugging
- Disposal: Documentation and Presentation

c. Operational controls:

Application/System hardware and software maintenance controls:

- Are there restrictions/controls on those who perform hardware and software maintenance and repair activities? → Only group members can perform hardware and software maintenance and repair activities.
- Are software warranties managed to minimize the cost of upgrades? → The data of the application are updated to ensure the result is the latest
- Was the application/system software developed in-house or under contract? → The application software was developed in-house.

d. Technical controls:

Identification and authentication:

- The application user authentication control mechanisms: User's name and password. Password system is used in the application:

-- password length.
-- allowable character set.
-- If the user login successfully, he/she also could change the password and login again.

- How does the access control mechanism support individual accountability and audit trails: Each user has an individual account and password to record his/her data.
- If the user uses the wrong password or login name, it would be invalid access so that the data would not be shown.

VIII. Fault Tolerance Analysis

Fault tolerance is the property that enables the system to continue normal operation when certain components (or one or more of them fail).

- No single point of failure: If the application fails, it is expected to continue to run without interruption during the repair process.
- Main Objective: The main objective of fault tolerance while applying is the primary consideration of this criterion.
  Data Management: The data size and information quality are the primary considerations. Data transmission must be reduced without affecting information quality, which can be achieved through storage.

To deal with software faults:

1. Prevention: These features and functions could prevent faults from being introduced into the application.
2. Before the application implementation, we test and debug to check the implementation and remove any exposed faults.

IX. Prototype

Initial UI design and ideas, though much has changed in our final products.

+ The Login Frame is displayed first for the user to login into their account.

Figure 5. Login page

+ If the users are new to the application, they are required to provide some necessary information to create an account (such as user's name, age, city).

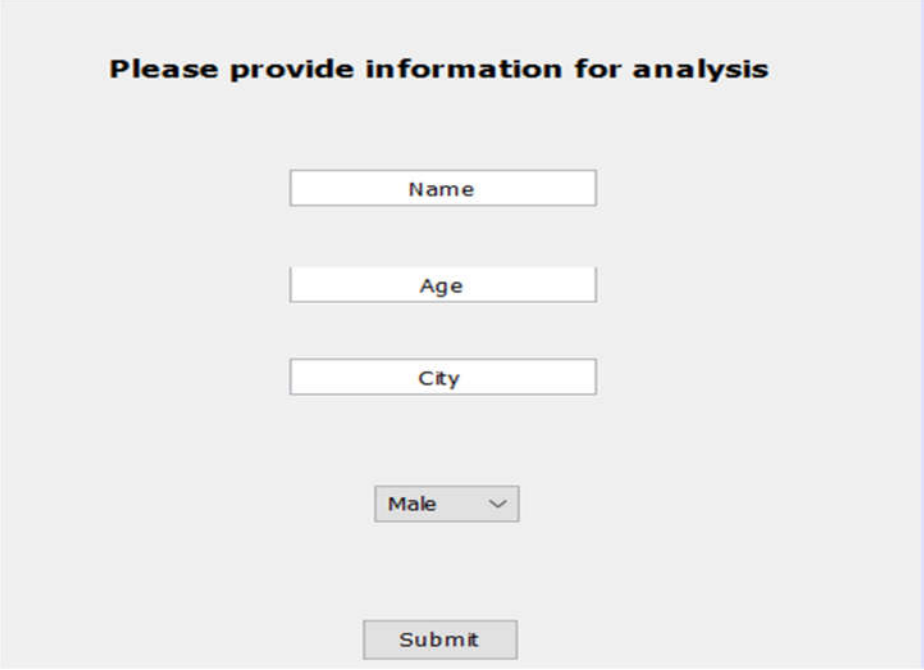

Figure 6. Snippet of the register step.

+ After successfully login, the software provides 3 different options for the user to choose from, including self-diagnosis test, area statistic and local area regulation.

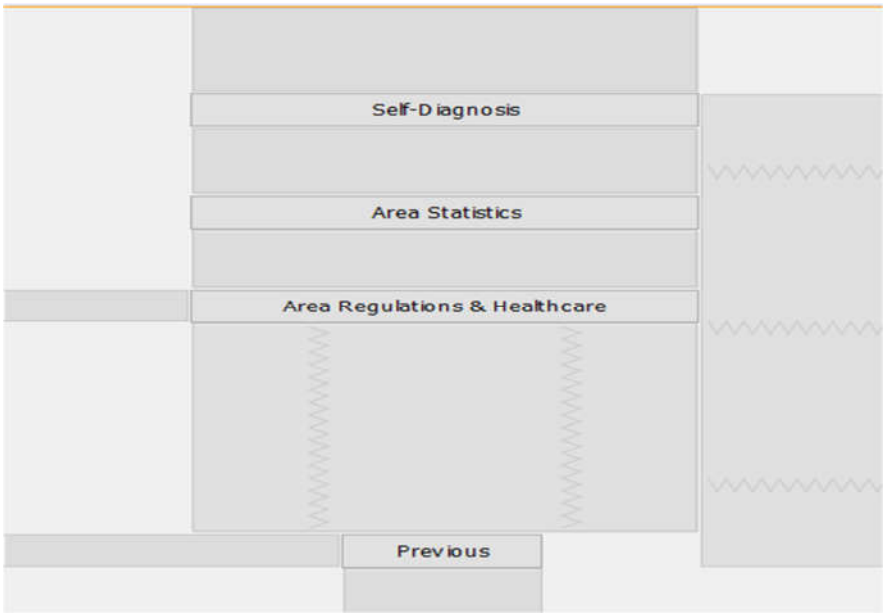

Figure 7. Homepage

+ For the self-diagnosis function, the users are required to answer the displayed questions and provide necessary information for the test.

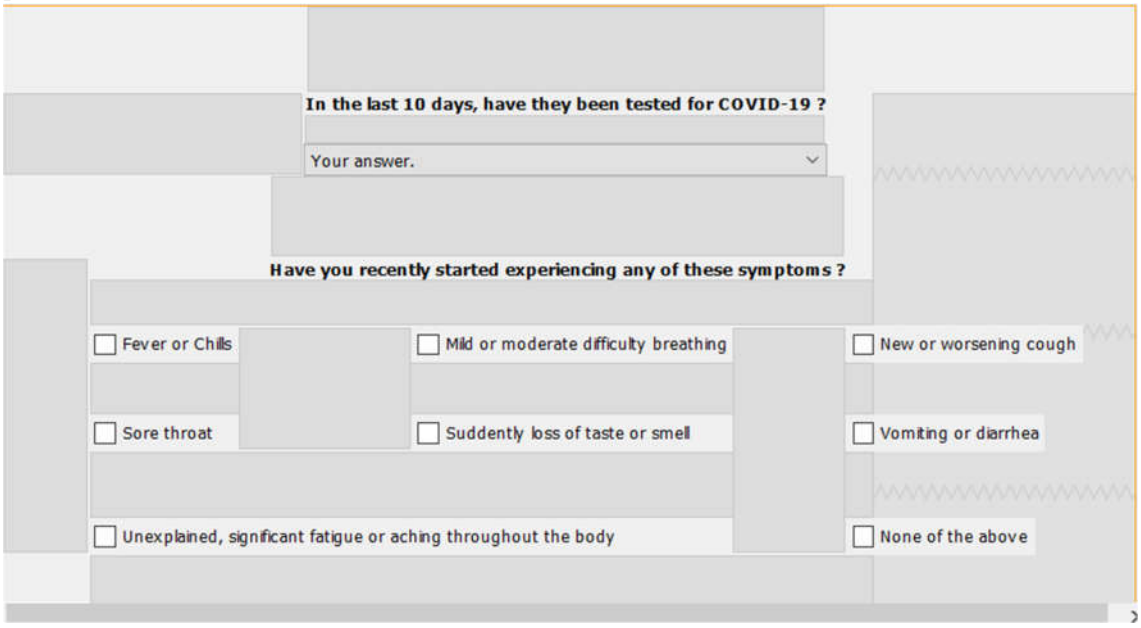

Figure 8. Self-diagnosis test function

+ For the area statistic, users need to enter both the location they want to check and its corresponding region for the program to output the result.

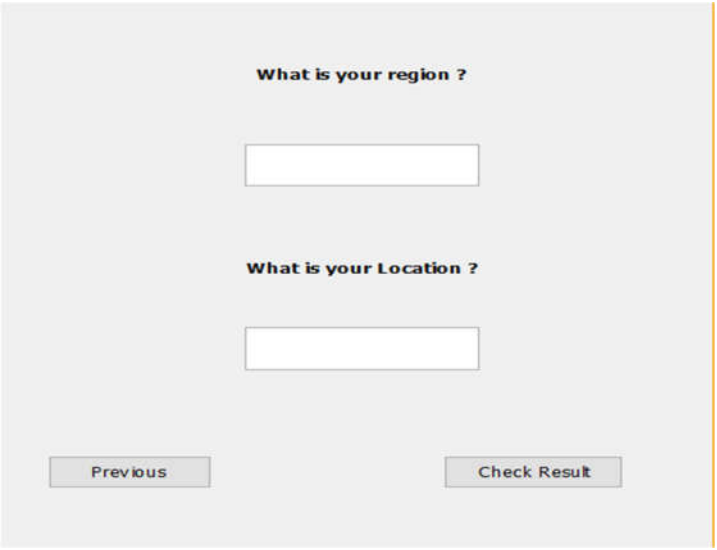

Figure 9. Area statistic fill-in information

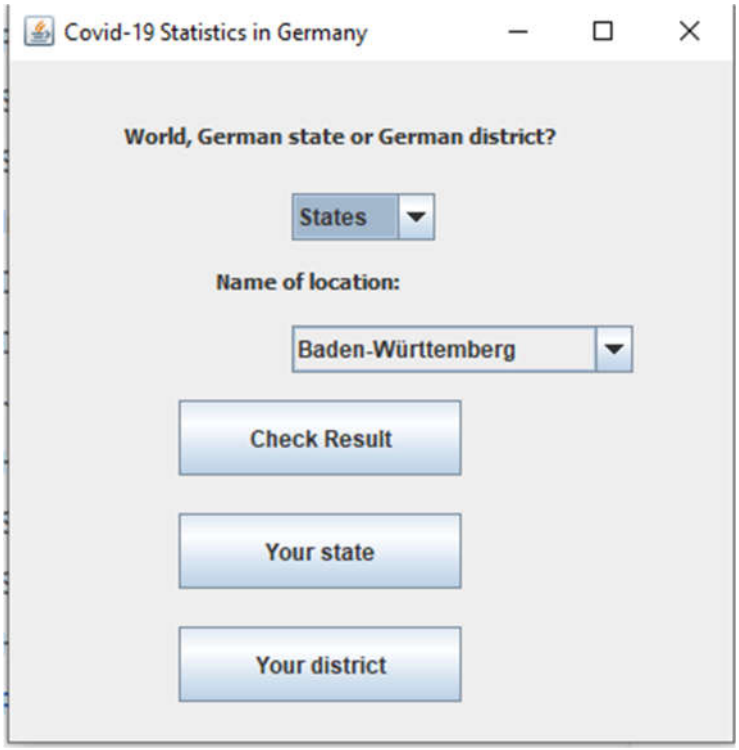

Figure 10. Area statistic function's options

X. Appendix

a. Report distribution:

Duong Thanh Minh	Status report and design model.
Tran Hoc Phuc	Software plan, application demonstration and formatting.
Le Thanh Tung	Hazard analysis and safety plan.
Szu-Chi Huang	Security plan and fault tolerance analysis.
Nguyen Tran Minh Luan	HMI design and prototypes.

b. Software development distribution:

For the program development process, each individual is responsible for different tasks, including:

+ **Szu-Chi Huang**: responsible for the area regulation feature.
+ **Nguyen Tran Minh Luan**: responsible for the UI of the overall system.
+ **Tran Hoc Phuc**: responsible for the self-diagnosis test feature and the user's login/logout feature.
+ **Le Thanh Tung**: responsible for the local area statistic feature and the user's registration/database feature.
+ **Duong Thanh Minh**: responsible for joining all the features along with testing the program.