Group 2 – COVID-19 ASSESSMENT & APPOINTMENT APPLICATION

Gajan Satkunanathan

Ritesh Panday

Christopher Owoicho

Shailesh Sankdasariya

GitHub Repository: https://github.com/owiekris/COVID-19-ASSESSMENT-APPOINTMENT-APPLICATION.git

.

Developer's Manual

COVID-19 ASSESSMENT & APPOINTMENT APPLICATION

This software allows users to perform COVID-19 self-assessment as well as book appointments at timeslots of their liking. To accomplish this, the user will need to provide various inputs to the program, which in turn fulfills various conditions and triggers functions within the program. A flowchart explaining the program's logic and model can be found in Appendix A.

Once the program starts, the user is asked if they are a new or returning user. The program will accept 0 for old and 1 for new users. If the user specified that they are a new user, the program will take them to the New User interface. Here, the user will be asked to enter their First & last name, Postal Code, e-mail address, and their Health Card Number. If the user specified that they are old, the program will ask for their Health Card number to check if they are in fact already registered in the program's database. The program will then perform a simple search for the entered Health Card number in the CSV file where all data is stored and return an error message if not found. If the health card number is not found, an error message will be displayed, and the user will be directed to the New User interface.

Now the user will be able to access the program's main interface, which includes 4 options: Book a vaccination appointment, COVID-19 self-screening, manage your appointments, and exit the program. The user will be required to enter a number from 1-4 accordingly.

- If the user selects option 1, they will be taken to the vaccine booking interface. Here, the program will display the earliest available appointment slot, and ask the user if they'd like to choose another date. The program will take inputs of 0 and 1 for no and yes, respectively. If the user responds with yes, the user will be prompted with the available dates and asked to select a date. Selecting a date will reveal the timeslots available for that date.
- If the user selects option 2, they will be directed for self-screening. In this screen, the user will be prompted with a set of COVID-19 screening questions used by Health Canada and Government of Ontario to determine if the user is symptomatic. The determination will be made based on the guidelines provided by Health Canada. If the user is found to be symptomatic, the program will recommend the user to self-isolate and see their healthcare provider and return in 2 weeks to book an appointment. The user will then be taken back to the main interface of the program.
- If the user selects option 3, they will be taken to the appointment management section. Here, the user can view their current appointment, and choose to setup a new appointment. The program will first check if the user has an appointment, and if they do not, they will be taken to the appointment booking section. This part of the program will work similarly to option 1, where the earliest date is displayed and if the user wants to select a new date, they will be prompted with the available dates & asked to select one.
- If the user selects option 4, the program will cease running and close.

The final part of the program prompts a confirmation to the user on-screen once they have booked an appointment. It will also print a .txt file and pop it out to the user as confirmation. The program will then ask the user for the next task #, and cease running & close if the user chooses to exit the program. Please note that the program will store all information entered by the user in a .csv or .txt file.

The flowchart

The software is scheduled to be released in three separate versions, which comes together to create COVID: Assessment and Appointment. The first release will provide exclusively the completed COVID screening program as the releases' purpose is to provide self-assessment to the client. A database and custom functions are setup in the background to store data and perform verification. This program can function independently. This release includes requirements: 1 and 2. The second release will execute the appointment booking system, including requirement 3. The third and final release contains the program that provides the user the ability to select appointments with custom

dates and the appointment confirmation to the user's email. Requirement 4, 5, and 6 are included in this final release.

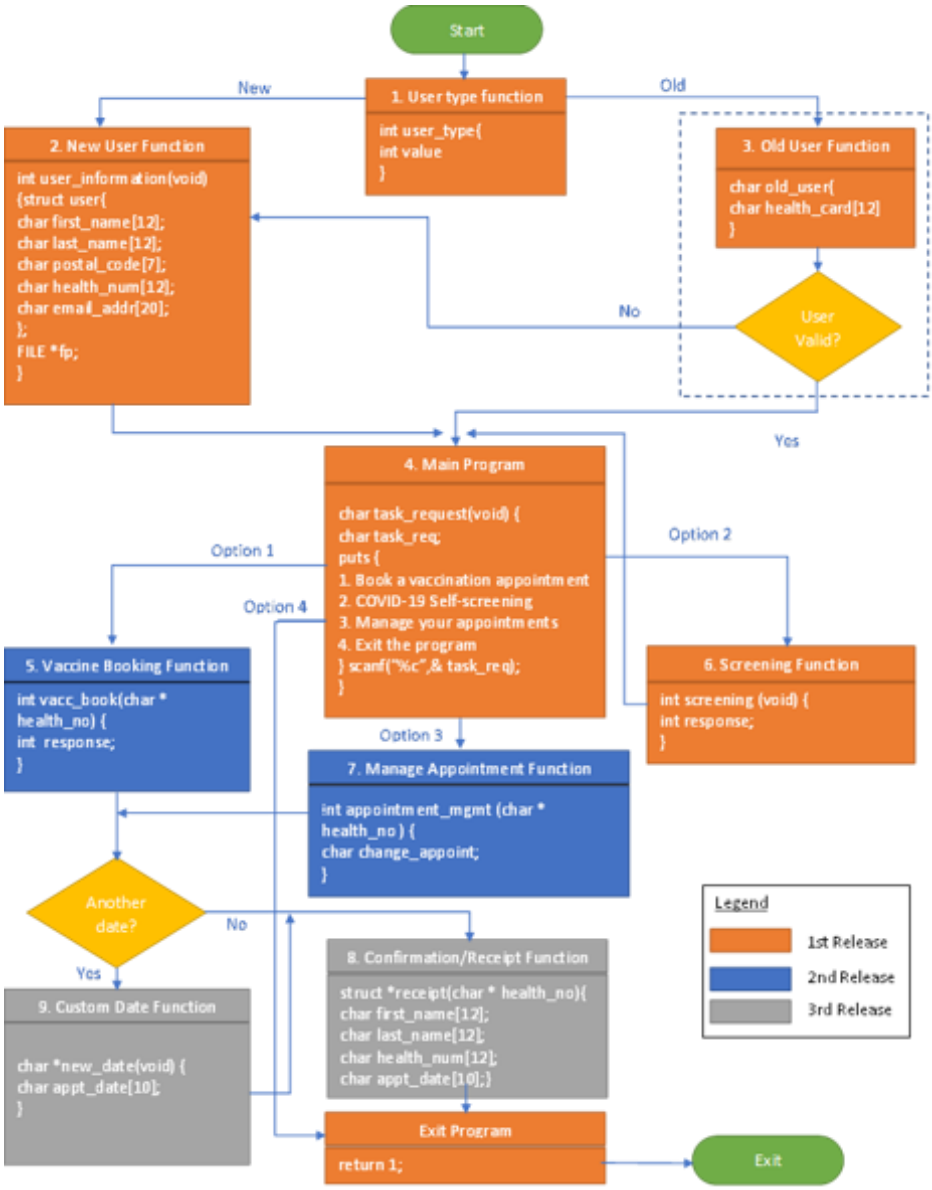

Figure 1

Modified and new functions in the final software:

Please note that the function headers and parameters in the flowchart required changes as the project evolved and required more functionality and constraints. Various sub-functions and some major background functions have been added to make the software more robust and provide extensive validation. The following functions and sub-functions were added and modified from the original proposal:

1. User type function
```
int identify_user(void);
```

2. New user function
```
int new_user(void)
```

3. Old user function
```
int old_user(void)
```

4. Task Selection function
```
char task_request(void)
```

5. Screening function
```
int screening_task(void)
```

6. Appointment booking function
```
void book_appointment(int record_pointer);
```

7. Manage appointment function
```
void manage_appointment(int record_pointer);
```

8. Read database function
```
int read_database(int read_rec);
```

9. Insert database function
```
int insert_database(User);
```

10. Validate input function
```
int validate_input(char *str,char *cmp_str,int max_length,int min_length);
```

11. Validate e-mail function

```
int validate_email(char *str);
```

12. Print receipt function

```
void print_receipt(User cur_user);
```

File Structure

/Code

- /build
  - /
- /cmake-build-debug
- /data
  - /database
  - /receipt
  - /report
  - /Database Structure.docs
  - /db.zip
- /doc
- /documents
- /include (all the header files included in the project)
  - /appointment.h
  - /database.h
  - /debug.h
  - /receipt.h
  - /task.h
  - /user.h
  - /validation.h
- /lib
- /src (all the .c files)
  - /appointment.c
  - /database.c
  - /receipt.c
  - /task.c
  - /user.c
  - /validation.c
- /test (all tests performed)
  - /data
  - /include
  - /src
- /CMakeLists
- /main
- /README.txt