Health Management System
OOM Project

Introduction :

Our application “Health Management System” is designed to help a user to keep track of the daily fitness goals along with keeping track of different health related vitals.

It can be used for simple fitness tracking as well as a precautionary tool to alert users of any impending health condition.

Currently the application supports 3 different user profiles i.e.

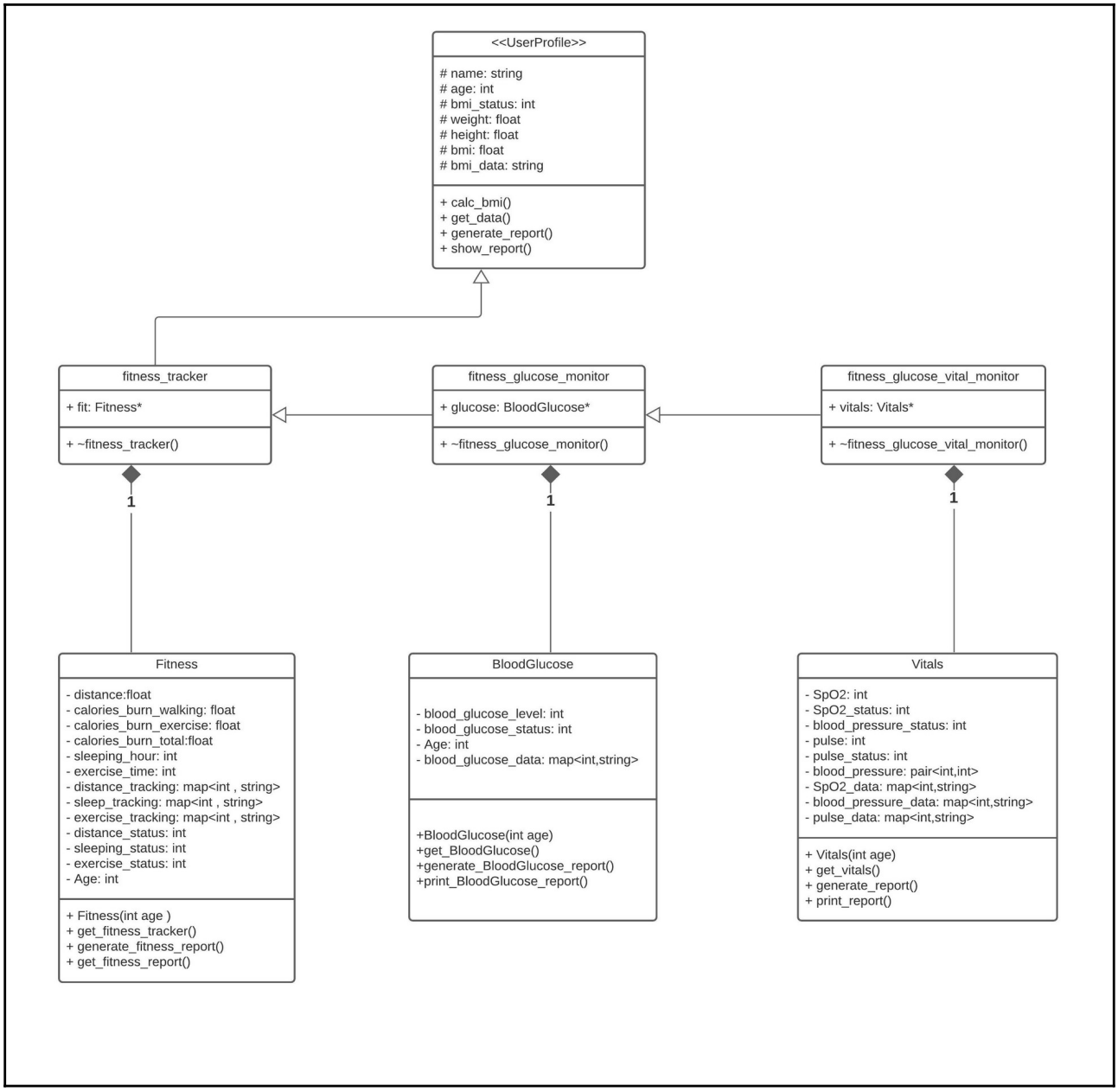
- Normal User (Basic Fitness Tracking)
- Diabetic user
- User with some major illness.

The application features allow users to monitor certain daily fitness goals such as exercise time, distance walked, sleep time, BMI which are evaluated according to their age group letting users to know whether they meet minimum daily requirements for staying healthy or not. It also helps users to keep track of Blood Glucose Levels, Blood Pressure, SpO2, etc.

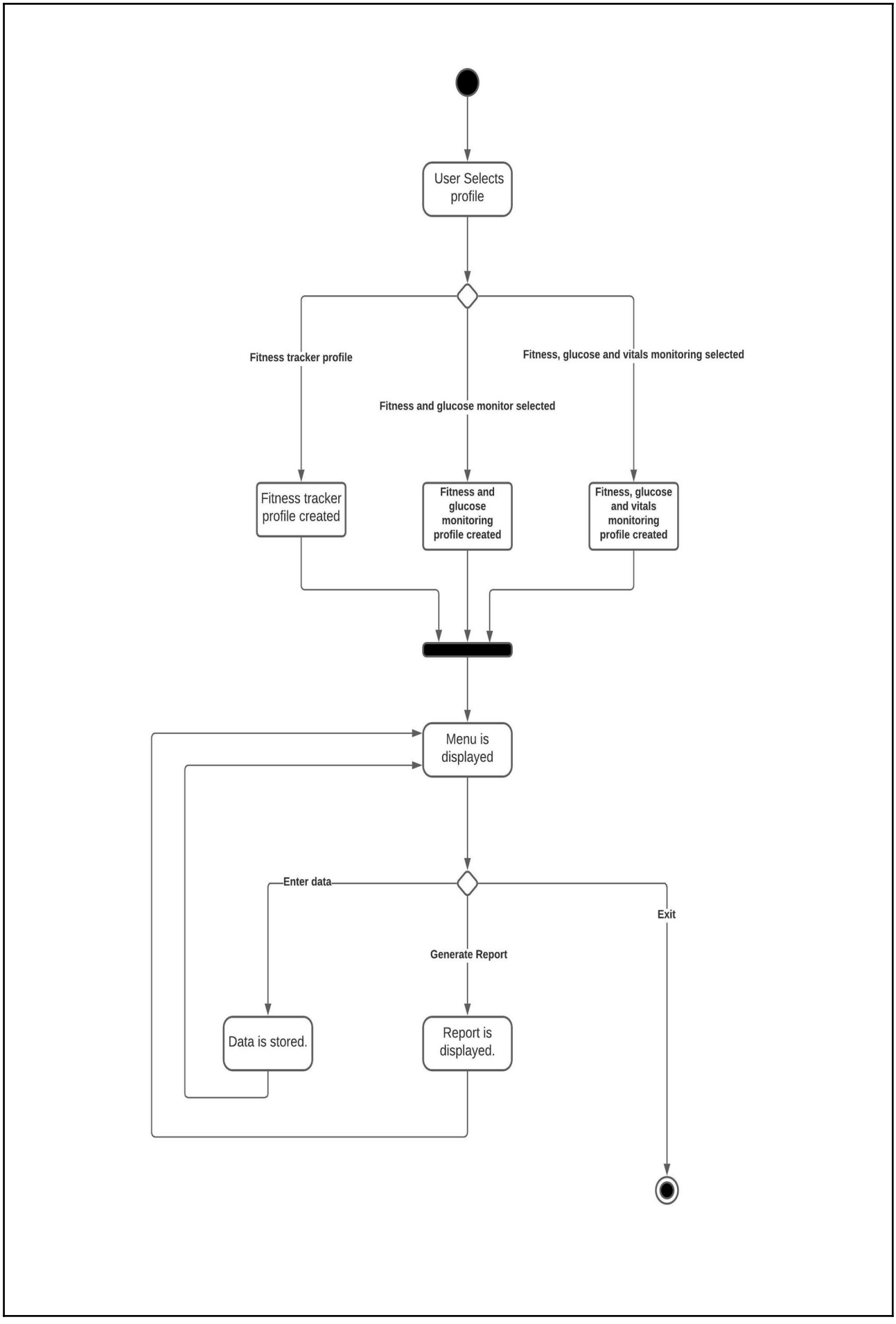
Requirements :

- To build a software which allows users to have systematic records of Health & Fitness.
    - Allowing the user to select the profile he/she is interested in.
        - Making a menu to give the user the choice of profile.
    - Taking input from user based upon his/her selected profile
    - Generating health reports for a user.
        - Storing diagnosis data and comparing it with data entered by the user to generate reports.
    - Allowing users to print the generated reports.

Class Diagram :



Activity Diagram :



Working :

Here is detailed working of our program.

- vitals.h and vitals.cpp
  - Contains class **Vitals** : has fields to store data like SpO2, blood pressure, pulse and methods to retrieve them and generate reports.
- bloodGlucose.h and bloodGlucose.cpp
  - Contains class **BloodGlucose** : has a field to store blood glucose data and methods to retrieve them and generate reports.
- fitness.h and fitness.cpp
  - has fields to store data like distance walked, calories burned, sleeping hour, exercise time and methods to retrieve them and generate reports.
- classes.h and classes.cpp
  - Contains
    - Class **UserProfile** (Abstract class) : It is our base class and contains fields to store user data such as name, age, height, weight, BMI and method to calculate BMI and other virtual functions used by derived class.

      There are 3 derived classes from this base class which helps simulate different user profile.
    - Class **fitness_tracker** : It is derived from class UserProfile.

      Contains pointer of type Fitness which carries out all the functionalities.
    - Class fitness_glucose_monitor : It is derived from class fitness_tracker.

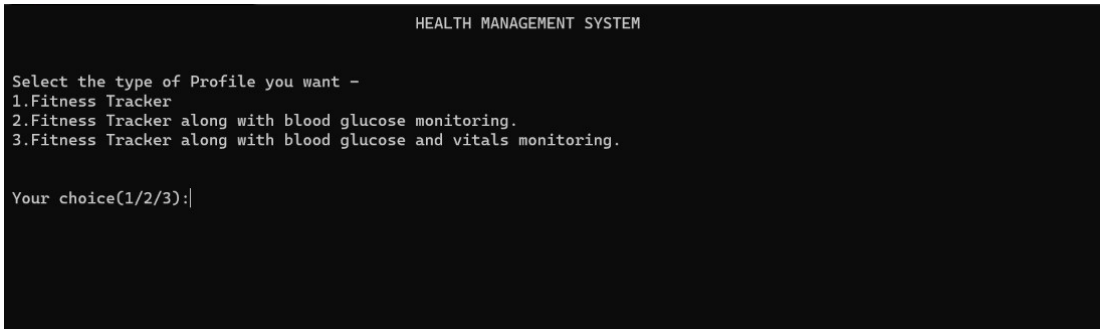
      Contains pointer of type BloodGlucose which add over the functionalities of class fitness_tracker.

    - Class fitness_glucose_vital_monitor : It is derived from class fitness_glucose_monitor.

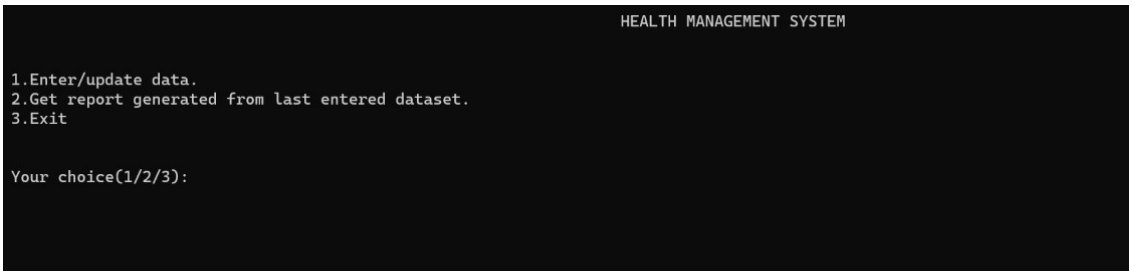
      Contains pointer of type Vitals which add over the functionalities of class fitness_glucose_monitor.

- **main.cpp** : It carries out the overall functionalities of the program at the run-time.
    - It has pointer of type UserProfile.
    - Program then takes input from user to decide upon the profile they want and then accordingly assigns new object of same type to the pointer.
    - User is displayed with the functionalities of the selected profile which are carried out by virtual function of the class UserProfile.
    - The user can then exit the program.

Screen-shots from Program :



Main Menu



Menu after selecting Profile

Group Members :

1. Ravi Maurya : LCS2020043
2. Nikhil Verma : LCI2020044
3. Kirti Goyal : LCI2020039
4. Sankalp Sahu : LCI2020016