CCPROG3 Term 3, AY2019-2020
Machine Project Specifications

With COVID-19 becoming a pandemic in a short period of time, it has caught the Philippines (and the world) unprepared. People in the tech industry (or who just want to help) have risen up to propose and/or implement contact tracing applications. Apps that track geo-locations of mobile phone users as a means to identify places (and times) that may have been "contaminated", when the user eventually exhibit COVID symptoms or eventually tested positive, are receiving a push back for violating data privacy. Note that data privacy laws in each country may be different. In general, regardless that it is for public interest, a person's right to data privacy should not be violated. The government, however, has the right to protect its citizens. Therefore, it is the government's mandate to do the contact tracing. Any information needed by the government cannot be withheld by the citizen. Of course this means that personal data (e.g., name and address) and sensitive personal data (e.g., birthday, age, and health information) has to be shared by the individual to the government but the government is also responsible in using these information only for the purpose of contact tracing, as well as responsible in keeping the data secure and confidential.

Even with geo-locations not being tracked in the applications, the general public are wary of using contact tracing applications in fear of their personal information being leaked out and they (and their families) become discriminated. This is mainly due to undisclosed or vague details of who has access to the data and how will these be used in the applications, as well as reports of discriminated people (including health workers) being locked in their homes or denied entry to their home/community and to the extreme of being hurt (bleach splashed on them).

Now, with still a considerable number of new COVID-19 cases, establishments are tasked to maintain their customer information to facilitate contact tracing efforts of the government. Customers are asked to write their name and contact information for every establishment they visit. These are usually kept on a sheet of paper, where other customers will also be able to see (as they are filling up their info in the same page). [You might have heard of news of establishment employees posting customer's information in social media, either to bash / rant on the customer's attitude or to brag they have their celebrity customer's contact number.]

In short, balancing privacy and confidentiality with public health interest is important. For your project, I outline here my proposed solution. My premise is, in the ideal world, only the people who should have access to the data do so and that as much as possible, there is little to no reduplication of efforts. Here are the details of my [simplified] proposal (that you have to design and implement in Java, following object-oriented concepts):

Users of the program may be normal citizens (customers/visitors of an establishment), a contact tracer (person assigned by DOH), or a government official (maybe from IATF, DOH, etc.). Note that contact tracers and government officials are also citizens (that is they have attributes and behaviors that are associated with normal citizens).

Each type of user of the system also has an account information: username, password, and role. The role specifies what information and features this account has and can access. A citizen who is also a contact tracer (or a government official) will only have 1 account. Each user can change his password, but once the username is acceptable (this should be unique) and the role is indicated, these values cannot be changed. For simplicity, it is assumed that it is the government official who creates the accounts for the contact tracer (so no random person will just claim to be a contact tracer and there is no need for an approval process). The government official is also the one who can suspend/terminate the contact tracer's account from the system (in the case where the contract tracer no longer works as one, due to resignation or lapse of employment contract). Also for simplicity, let us assume that a government official account is created for the Admin of this program (username Admin2020, with initial password @Dm1n0202). So, government official accounts can also create other government official accounts, as well as suspend/terminate other government official accounts too (but not own

account → this also ensures that there is at least one government official account that can access the system). Suspending/terminating accounts will "demote" the account into a citizen account.

Each citizen of the country has the following information: name (first name, middle name, surname), home address, office address, phone number, email address, and establishment visit record. The record is a collection of at least 30 days worth of entries. Each entry consists of the code of the establishment visited, date of visit, and time of visit.

Assume that the program will be part of a mobile app, where eventually data of each citizen are uploaded into the server for consolidating into reports, etc. For this project, to simulate the different accounts, the user logs in and logs outs.

To simplify the features and the interaction, let us say that it is the citizen who reports that he has tested positive. The government official will have access to the set of positive cases. This access is only "read only", that is, the government official cannot modify any information of the citizen. The government official only uses the information to assign the case to a contact tracer and to generate reports. The contact tracer is the one that generates the list of people who may be exposed and notifies them of the possible exposure to a positive case. The contact tracer also cannot modify any information about other accounts. In fact, the contact tracer should not even know the name (among others) of the positive case.

The following is an outline of the features of the application:

Main Menu

1. Register – User inputs his chosen username, program checks if it is a unique username. If it is, the program asks user to input his personal and contact information. A password that the user chooses will also be asked. The program should check if the password is at least 6 characters and contains at least 1 character that is not a letter (could be a digit or could be a special character that is not a space). Once the password is acceptable, the citizen's account is created. Program goes back to the Main Menu.
2. Log in – username and password is asked from user and verified by the program prior to showing the appropriate menu options available for that particular user.

   If user's role is a government official, the following options are available:

   a. Show Unassigned Cases – this option loads the file of unassigned cases and displays the case numbers
   b. Show Contact Tracing Updates – this option shows the list of positive cases from a specific input date range (of when the reports were made) and their status (that is, who is the assigned contact tracer and if traced already or pending).
   c. Analytics – this option generates and shows reports. The program should be able to (1) show number of positive cases of a given city (from home address of positive case) in a given duration; (2) show number of positive cases in a given duration; (3) show number of positive cases in a given city
   d. Create Government Official Accounts – creates / registers a Government Official account by asking the current user to input a username for the new account. If the username is not unique, the account is not created. If valid username, an initial password, with at least 6 characters with 1 special character or digit is generated. Should the new account already have a citizen account, the user name is just used to populate all the information. If there is no existing account, personal information and contact information is asked as input and used to initialize.
   e. Create Contact Tracer Accounts – similar to the creation of government official accounts, but for a contact tracer.
   f. Terminate Account – gets the username of the account that will be made into a citizen account. A citizen account cannot be terminated.
   g. Log Out

If the user's role is a contact tracer, the following options are available:

a. Show Cases – shows a list of case numbers assigned to this contact tracer that have not undergone contact tracing yet.
b. Trace Specific Case - Shows a listing of citizen's usernames who could have come in contact with the case number being handled. Note that the username of the positive case being handled should not be included in the listing. The program determines possible exposure by looking at the date and time the positive case (case being handled) and checks all citizens who had been at the same establishment code at the same date and the time might intersect (for example, the citizen is possibly exposed if his check-in time is same or after the positive case time OR if the citizen's next check in (to a different establishment) is before the next check in of the positive case (in another establishment → think of your algo for schedule overlaps when enrolling or planning a meeting—consider only for the same day)
c. Inform Citizens Possibly Exposed – this option automatically sends a notice to the users account a notice that he may have been in contact with a positive patient on <date> in <establishment> and is advised to be tested too. The positive patient should not be sent a new notice. Advise too that, should his test result be positive, he should report via his app. Once informed, the case is flagged as contact tracing complete.
d. Log Out

If the user's role is a citizen, the following options are available*:

a. Check In – the user is asked to input the establishment code. The program also records the date and time of this check in (by retrieving the machine's date and time).
b. Report Positive Test Result – once this option is chosen, the date of this notice and username is recorded, a case number is issued.
c. Update Profile Information – Allows modification of personal information and contact information.
d. Log Out* - If there is an update on the profile, the text file has to be updated. Otherwise, only the check-in records done while logged in will be added to the text file.

*Prior to showing the options during log-in and before the user is logged out, if there is a notice that the current user may be exposed, the message should be shown. [Better, if the message appears all the time the user logs in. This message will not appear anymore, once the user reports that he has tested positive, there is a new message of another possible exposure (at a later date), or it is 14 days after the last message (and the user has not reported to test positive).

3. Exit – program terminates

Phase 1 requirements: Due: August 31 (M) 07:30AM in AnimoSpace

1. Draw the UML class diagram to represent the government official, contact tracer, and citizen accounts. The diagram should include aggregation, composition, and association relationships, as applicable. It is not expected that hierarchy (in terms of inheritance) is included in the diagram.
2. Implement only the Main Menu, all except feature (b) for the government official, and all features of the citizen account. For all features that require files, for phase 1, assume that data will be retrieved from the main memory (i.e., from the current run). Thus, check-ins will also be getting the date as user input (rather than retrieve from the machine), to test the features.
3. Phase 1 implementation requires all displays to be on the console (no GUI).

Phase 2 requirements: Due: September 22 (T) 11:59PM in AnimoSpace

1. Complete UML class diagram for the entire program to be implemented, including those for the GUI. If the diagram is too big, it is suggested that there be 2 separate UML diagrams: 1 diagram to show the class names and the interactions and the other diagram to show the details of each of the classes.

2. Implement all features of the program, including saving and loading from file.
3. Phase 2 implementation requires GUI. The screen design is up to the programmer. The design and the program should follow MVC.
4. All text file formats are indicated towards the end of this specifications. Unless explicitly stated so in the specs, the programmer is to decide on when data should be loaded and saved, considering the accessibility of the data and this simulation (e.g., it might not be a good idea to always have the saving done only during exit; maybe some of the saving will be done upon log-out).
5. Prior to showing the Main Menu at the beginning, allow for previous session data to be pre-loaded. This could be part of the initializing.
6. Additional features can be implemented by the programmer. However, these should not contradict the requirement. Whatever implemented additional features should be reflected in the UML class diagram too.

Deliverables per phase:

1. UML class diagram
2. Java source files of implemented classes, including the internal documentation
3. Meaningful program documentation generated via Javadoc
4. Test script of each method per class. Exclude methods that displays only (no checking) or that returns values. Setters may be included only if they perform checking. Checking constructors should be included in the test script.

Note:

1. If there are questions regarding the specs, check “MP QA.gdoc” file in the Shared Drive (same shared drive where our lecture videos are uploaded). If a similar question is not previously posted (by other students), you may raise your question in the Discussion page in AnimoSpace.
2. For the minimum requirements of this MP, all the requirements written in this document should be present and working. Incomplete (but running) submissions will incur deductions.
3. Do not forget to include internal documentation (comments) in your code. At the very least, there should be an introductory comment and a comment before every class and every method. This will be used later to generate the required External Documentation for your Machine Project. You may use an IDE or the command prompt command javadoc to create this documentation, but it must be PROPERLY constructed.
4. Statements and methods not taught in class can be used in the implementation. However, these are left for the student to learn on his or her own.
5. You are required to create and use methods and classes whenever possible. Make sure to use Object-Based and Object-Oriented Programing concepts properly. No brute force solution.
6. This project is at most done in groups of 2. A person may work alone. A person / group cannot discuss or ask about design or implementation with other persons, with the exception of the teacher and his/her groupmate. Copying other people's work and/or working in collaboration with other teams are not allowed and are punishable by a grade of 0.0 for the entire CCPROG3 course and a case may be filed with the Discipline Office. In short, do not risk it; the consequences are not worth the reward. (Also, what is a measly passing grade compared to a life-long burden on your conscience?)
7. Submission of the project for each phase was announced during first day of classes, and indicated also in the syllabus. Late submission will not be accepted and will therefore result to 0 for that phase.
8. The students are responsible in submitting in the Canvas assignment submission page the correct version. Only the last uploaded version by the deadline will be used as basis for assessment. As part of the requirement, the proponent/s should make pertinent back-ups of his/her/their own project.
9. During the demo, all members of the group should be present. The group should know how to generate the bytecode file and to run the said file in the command prompt. Apart from question-answer, there is a demo problem that each member will have to work on individually as part of the demo of phase 2.
10. A student or a group who cannot answer questions regarding the design and implementation of the submitted project convincingly will incur a grade of 0 for that project phase.

11. During the MP demo, it is expected that the program can successfully be interpreted into bytecode file and will run. If the program does not run, the grade for that phase is 0. However, a running program with complete features may not necessarily get full credit, as implementation (i.e., code) will still be checked.
12. All sources should have proper citations. Citations should be written using the APA format. Examples of APA-formatted citations can be seen in the References section of the syllabus.
13. For the students' reference, the rubric for assessing each phase is accessible in the syllabus (in Canvas).
14. Parts of the requirement may be expected as a class activity / assignment, due even before the deadline. If such is the case, an announcement of the expectation and due date will be announced prior.
15. The outputs of this project may be used in reports and publications.

Text File Formats:

1. Master List – This is a text file of all usernames and their roles
   <username><space><role><newline>
   <username><space><role><newline>
   ::
   <username><space><role><newline>
   <end of file>

   <role> can be "citizen", "official", or "tracer".

2. Account Information – Each text file only contains one account information. Each account is saved with the username as the filename and the extension is ".act".
   <password><newline>
   <first name>,<middle name>,<surname><newline>
   HOME:<home address><newline>
   OFFICE:<office address><newline>
   PHONE:<phone number><newline>
   EMAIL:<email address><newline>
   <end of file>

3. Establishment Visit Record – All account's establishment visit record is saved into 1 file only. The file is in the following format:
   <username1><newline>
   <establishment code><space><checkIn date><space><checkIn time><newline>
   <establishment code><space><checkIn date><space><checkIn time><newline>
   : :
   <establishment code><space><checkIn date><space><checkIn time><newline>
   <newline>
   <username2><newline>
   <establishment code><space><checkIn date><space><checkIn time><newline>
   <establishment code><space><checkIn date><space><checkIn time><newline>
   : :
   <establishment code><space><checkIn date><space><checkIn time><newline>
   <newline>
   :::
   <usernameN><newline>
   <establishment code><space><checkIn date><space><checkIn time><newline>
   <establishment code><space><checkIn date><space><checkIn time><newline>

: :
<establishment code><space><checkIn date><space><checkIn time><newline>
<newline>
<end of file>

Where:

1. Assume <establishment code> is just 1 word (example: JolTaft123, DLSU-M)
2. <checkIn date> is in the format of MM,DD,YYYY
3. <checkIn time> is in military format HHMM, assumed to be Manila timezone (GMT+8)
4. Data can be trimmed to past 30 days (from current system date) prior to saving.

4. Positive Cases – All reported positive cases are saved into 1 file only. The file is in the following format:
<casenum1><space><username><space><reportDate><space><tracer username><space><status><newline>
<casenum2><space><username><space><reportDate><space><tracer username><space><status><newline>
: : :
<casenumN><space><username><space><reportDate><space><tracer username><space><status><newline>

<reportDate> is in MM,DD,YYYY format. <tracer username> is “000” (string containing 3-zeroes) if no contact tracer assigned yet. For the project, <status> is “T” for traced and notified and “P” is pending.