Solver: Kong Zhong Han

Email Address: KONG0110@e.ntu.edu.sg

1.

   a)

b)

Actor:	User
Description:	Users are able to upload a new photograph to the CloudPSI System
Preconditions:	Users must be a registered user with CloudPSI System
Postconditions:	1. The photograph will be uploaded onto the CloudPSI System. 2. Analysis of the photograph will be done by the machine learning algorithm to estimate the PSI. 3. Email will be sent to the user to notify user of acceptance or rejection of photograph.
Priority:	High
Frequency of Use:	High
Flow of Events:	**Actor Steps** 1. User uploads a photograph onto the System.	**System Steps** 2. The System gets the updated PSI from API. 3. System will analyze the photograph based on photo clarity and Singapore's current PSI. 4. Email will be sent to the user.
Alternative Flows:	If the user attempts to upload a photograph without logging in as a registered user: 1. User uploads a photograph onto the system. 2. The system requests user to login. 3. Photograph continues upload upon successful login.
Exceptions:	Nil
Includes:	Login

2. Note: The format for this question has changed from previous years, where class diagrams include boundary and controller classes. So I drew this sort of diagram in my exam, close to what my class diagram was in my project.

a)
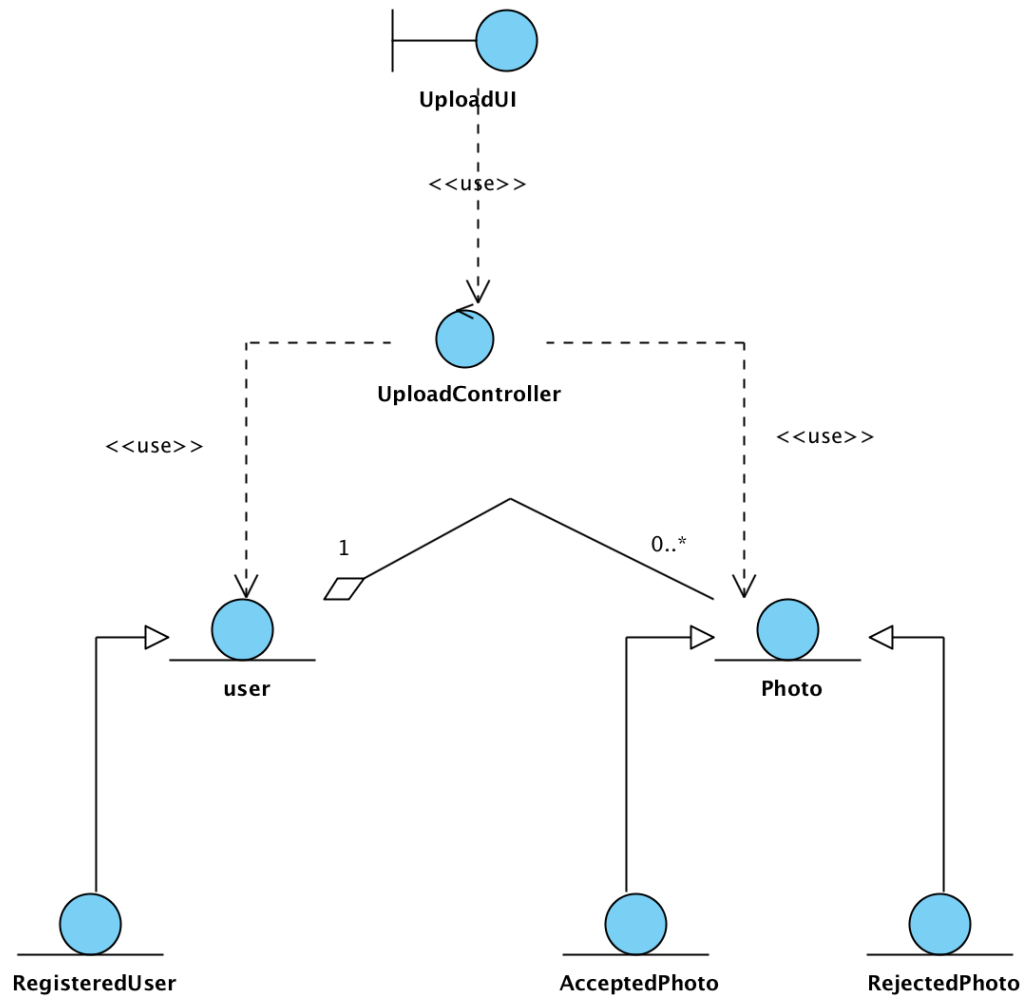
b)
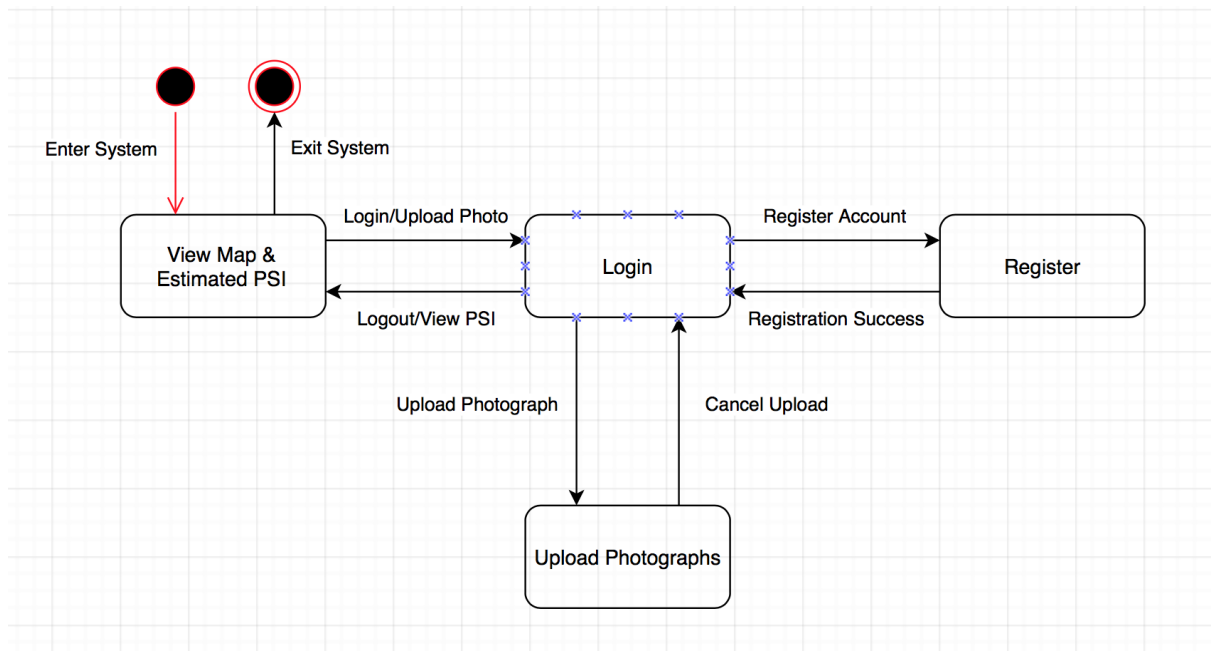
c) Note: Use the different methods of estimating the time to develop software elements to answer this question

I believe the use of Expert Judgment would be the most logical way to accurately estimate the difficulty and complexity of implementing this machine learning algorithm, aka getting programmers with a specialty in machine learning.

Also, the use of Wideband Delphi could also be used if there are many programmers with the experience in Machine Learning Algorithm.

3.

a) Propose the use of Model View Controller (MVC) Architecture. This allows easy change to the UI, while the presentation of data to web or mobile interface would be consistent as the UI is not tightly interwoven with core functionality and data.

b)

   i) Use of Observer Pattern to allow the System to send hazard alerts to the registered users.

   Use of Push + Pull mechanism to send alerts to the registered users. This allows users to receive an alert/notification from the System, and should they require more information, users can retrieve more details about the notification.

ii) Loose Coupling between the objects(Location) and it's dependent objects (Users). Allows users to continually change and update it's location, while being able to receive hazard alerts from a specific location which users are 'subscribed'/located at.
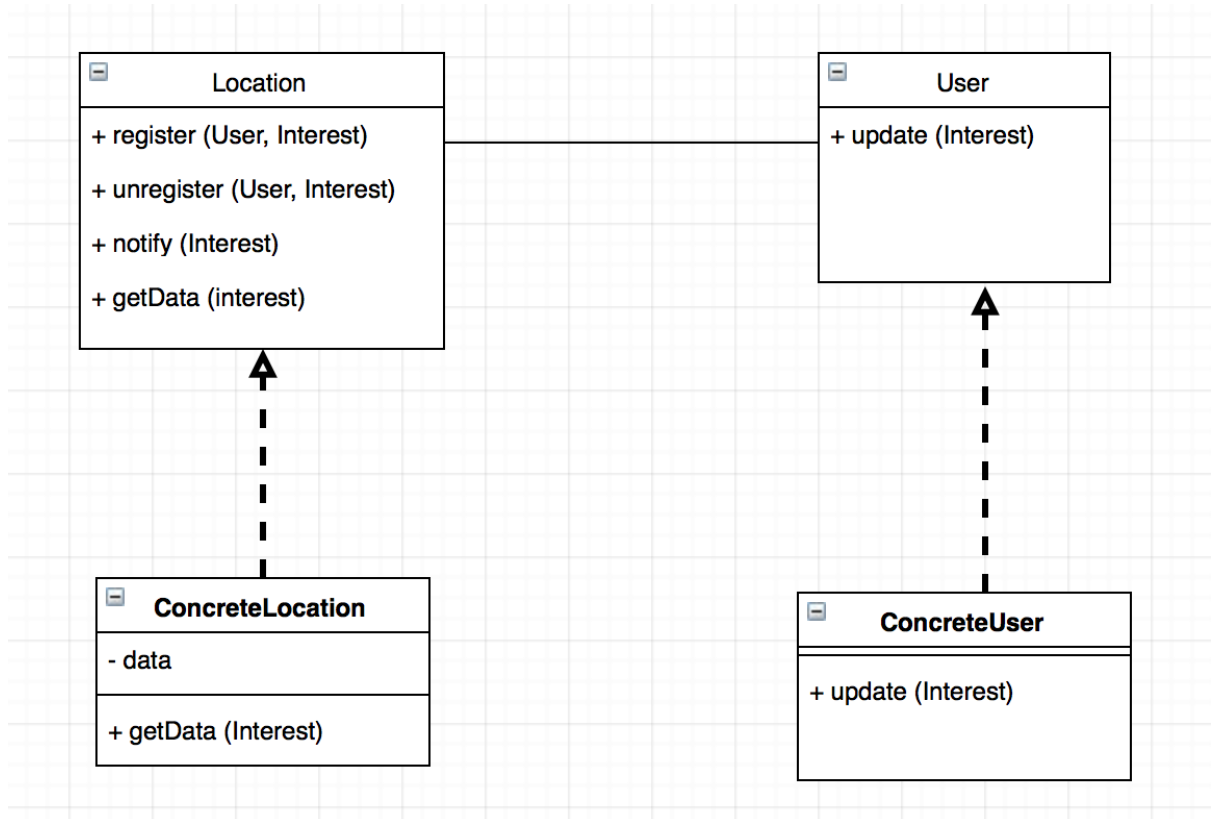
c) I would propose the use of Strategy Design + Factory Design to support this Non-functional requirement.

Note: I do not remember what I drew during the exams, or be sure that my answer will be correct.

Implement this Factory + Strategy design and extend it out on the UploadController class. By using this design, we essentially decouple the Machine Learning logic from the UploadController class, hence we can modify or add new algorithm without affecting the other classes.

Next, by using the Factory pattern, we can change the algorithm while the system is online, allowing for the modification and addition to be done on the fly.
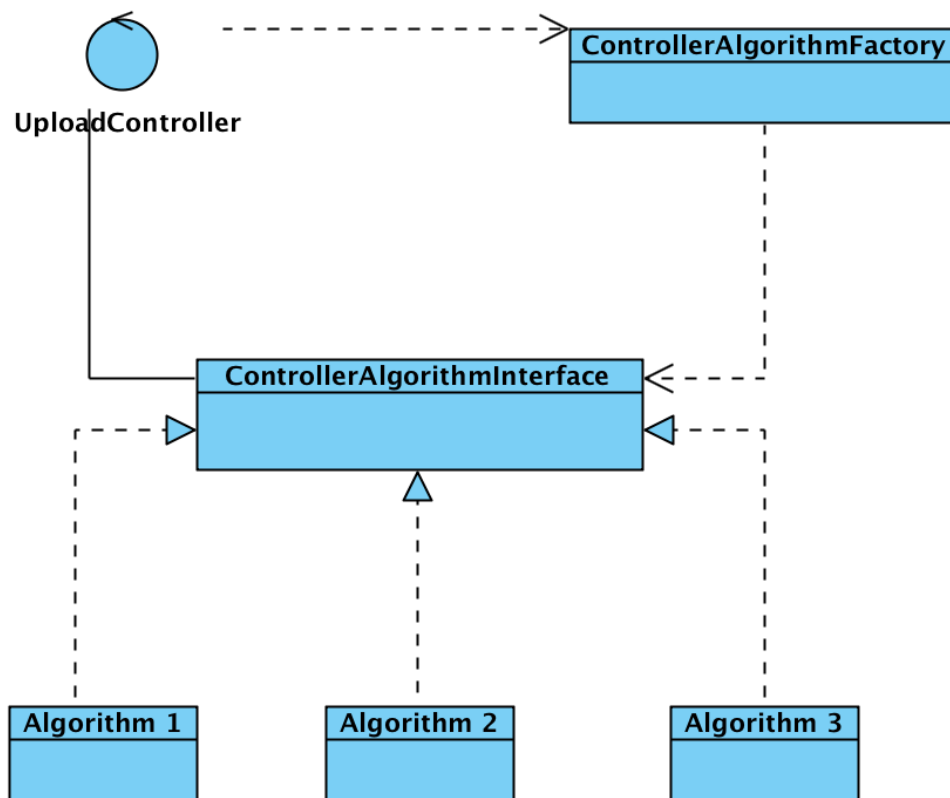
4.

  a)

    i)   Health Status would be discrete values, and valid equivalence classes would be 'Healthy Persons', 'Elderly, Pregnant women, Children' and 'Persons with chronic lung disease, heart disease'.

        PSI level equivalence classes would be:
            Invalid: PSI < 0
            Valid: 101 > PSI => 0 (No alert)
            Valid: PSI > 100 (Alert Sent)

        Assumption: The term 'testing the alert sending component' means to determine if an alert will be sent to the users, and not the type of alert which will sent. (Assumption based on the marks allocated, with respect to answers provided in previous PYP)

    ii) There is no boundary value for 'Health Status'.

PSI Level	Boundary Values
PSI < 0	-2, -1, 0
101 > PSI => 0	~~-1~~, 0, 1, 99, 100, ~~101~~
PSI > 100	~~100~~, 101, 102

    iii)

PSI Level	Health Status	Expected Result
1	Healthy Persons	No Alert, Valid
1	Elderly, Pregnant women, Children	No Alert, Valid
1	Persons with Chronic lung disease, heart disease	No Alert, Valid
1	Primary School Student	Invalid
101	Healthy Persons	Alert, Valid
101	Elderly, Pregnant women, Children	Alert, Valid
101	Persons with Chronic lung disease, heart disease	Alert, Valid
101	Primary School Student	Invalid
-2	Healthy Person	Invalid
-1	Healthy Person	Invalid
0	Healthy Person	No Alert, Valid
1	Healthy Person	No Alert, Valid
99	Healthy Person	No Alert, Valid
100	Healthy Person	No Alert, Valid
101	Healthy Person	Alert, Valid
102	Healthy Person	Alert, Valid

b)

i)
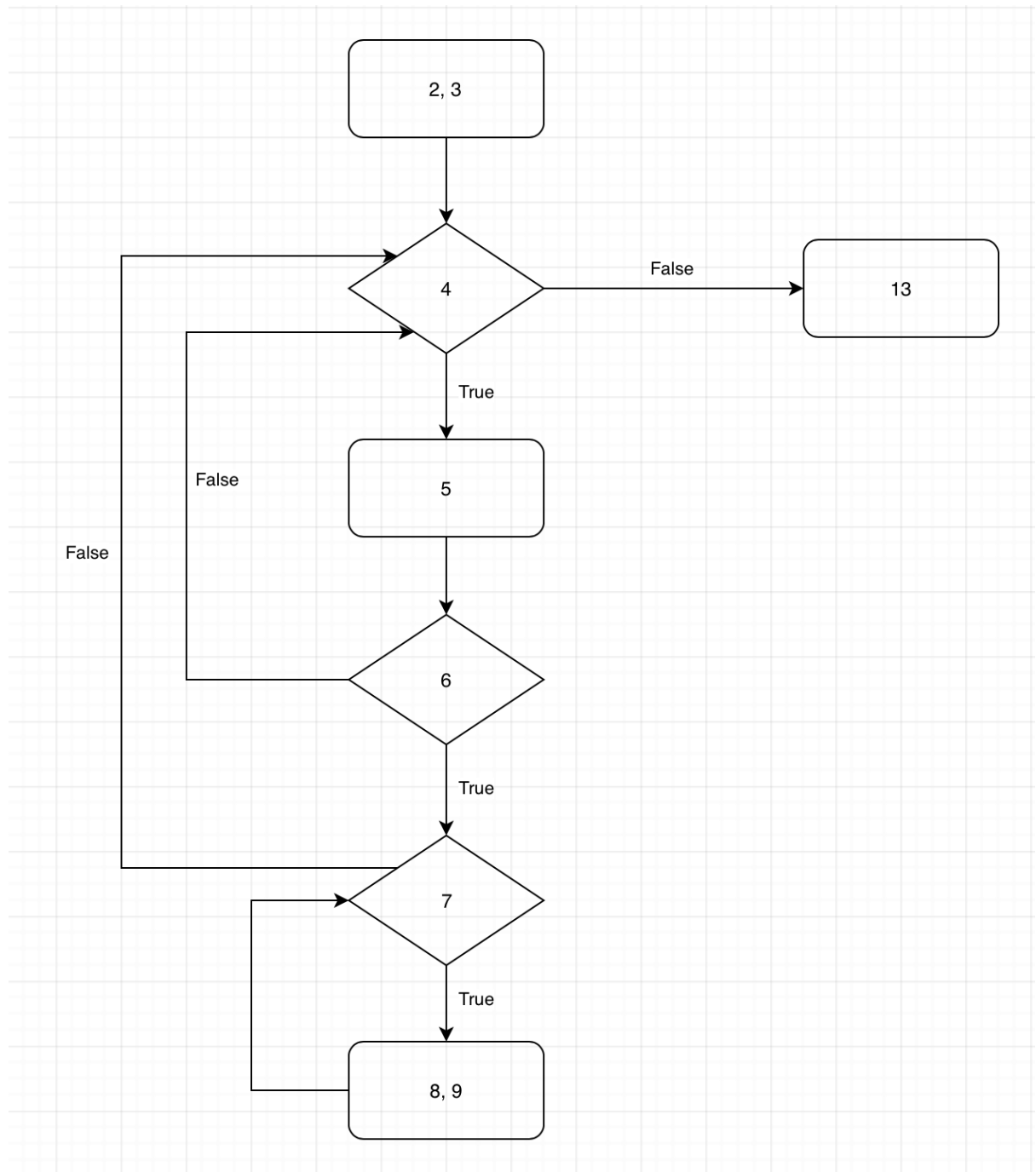
ii) No, testing of computeDistance() method is not required. Whitebox testing is to determine if the implementation of internal path and structure is sound, and to test every possible flow in the method.

It is acceptable to assume that any functions/methods within the unit which you're testing will be functional.

iii) CC = 4

(1) Basis Path

(a) 2, 3, 4, 13

(b) 2, 3, 4, 5, 6, 4, 13

(c) 2, 3, 4, 5, 6, 7, 4, 13

(d) 2, 3, 4, 5, 6, 7, 8, 9, 7, 4, 13

(2) Test Cases

(a) Cities.size() = 0

(b) Cities.size() = 1 (Singapore), computeDistance() = 11

(c) Cities.size() = 1 (Singapore), computeDistance() = 10, shortestDistance = 5

(d)   Cities.size() = 1 (Singapore), computeDistance() = 5, shortestDistance = 7

(3)   Real Execution Path (& Expected return value)

(a) 2, 3, 4, 13 (NULL)

(b) 2, 3, 4, 5, 6, 4, 13 (NULL)

(c) 2, 3, 4, 5, 6, 7, 4, 13 (NULL)

(d) 2, 3, 4, 5, 6, 7, 8, 9, 7, 4, 13 (Singapore)

For reporting of errors and errata, please visit pypdiscuss.appspot.com
Thank you and all the best for your exams! ☺