CS5030 8 HOUR TAKE HOME EXAM

210016568

December 9, 2021

1 Software process

1.1 Question a

i. Three functional requirements for this system

ID	Functional requirement description	Priority	source	Dependencies
1	The system should get data from the existing software system	High	the data from existing software system	
2	The system should allow the frequency of patrols to be customised by the user	High	the new frequency from user's input	
3	The system should produce a weekly schedule of patrols for the area	High	work arrangements data from existing software system	The arrangements data depends on 1, the frequency of patrols depends on 2

ii. Three non-functional requirements for this system

1. Ease of use:
   The application should be able to be used by police officers with a certain level of education in a short period of time.
2. Available:
   - The system should be accessible over internet connection.
   - The system should be available 99.9% of the time.
3. Security:
   The system should be secured against attacks and data corruption by hardware authorization, users' authorization (like our university used, the Microsoft Authenticator).

1.2 Question b

Ethical concerns:

- Data security
  Because the data is stored in the server, so it should be transported by internet. Here is a high risk that users use this app in the unsafe networking environment (like cafe, restaurant), resulting the leak of information or data.
- Fairness of algorithms
  Because crime is different in every area, the algorithm may prefer to schedule patrols toward places with high crime rates and not in places with lower crime rates. This is unfair to areas with lower crime rates because criminals will not change when committing a crime because of the high or low crime rate here, but will abandon the crime because there is a police patrol here.

Further requirements or constraints:

- The system should encrypt the data on the device before send it to server.
- The system should allow administrators to set a minimum number of patrols per area for a certain length of time and schedule the patrol schedule according to this number.

1.3 Question c

The **modified waterfall model** is suitable for this system because it is a **critical system** where absence of patrols can lead to serious consequences threatening the public security. In addition to that, all requirements should be elicited in advance, and they are less likely to change over time. Furthermore, in case there is any modification on the system it should go through again through all software engineering processes.

1.4 Question d

Client-Server architecture style fits this system, all our data and processes should be centralized in one place with high security. All data should be kept private, with strictly controlled access. To support the availability of the system we should have many replications for the server with a regular backup.

The data interactions of this system and the existing system should happened on the server.

In this architecture, police staff still use the existing system, they through the server to get interaction with components. To be specific, superintendent may also patrol, so he will get linked with both two system. For getting the data from existing system, here is a directly link between existing system server and the DBMS of existing system, it can be called by the new system via "user–new system–new system server–existing system server–existing system–existing server DBMS".

The interaction between all entities related to the new system should be done using the server. The Figure 1 will illustrate how the system will work with users and other systems (next page).

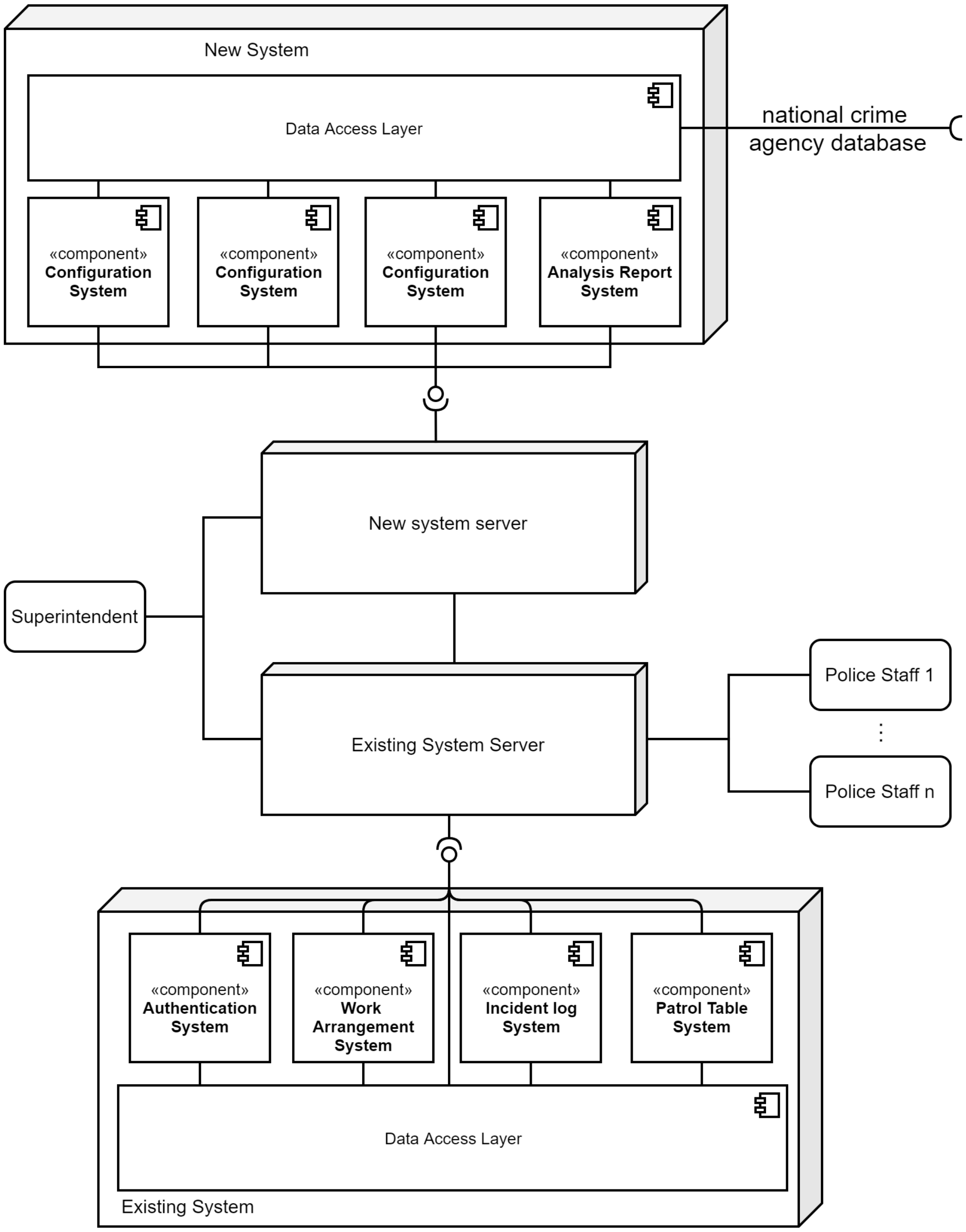

Figure 1: component diagram

2 Software maintenance, evolution and reuse

2.1 Question a

if we offer 2 different systems for the same purpose,

- Advantages
  1. Improve reliability
     When one of the 2 systems is unable to provide service due to unforeseen circumstances, users can still get the same service through the other system, so that can improve the overall reliability.
  2. Improve users experience
     if we only offer the web system, users need to take photo first, then open the web page, upload details and provisional identification, finally upload it. But when we offer 2, if users want to download app, it will be much easier to use, because the app can remember the provisional identification information for users, and when they upload, they can directly input the details of birds.
- Disadvantages
  1. Higher development cost
     when we offer 2 systems, they usually need different programming languages to develop, so the group need to hire more skilled programmers.
  2. Higher maintenance cost
     When making changes to a function, the development group need to do same thing on both 2 systems.

2.2 Question b

The components of existing system I identified are shown in Figure 2.

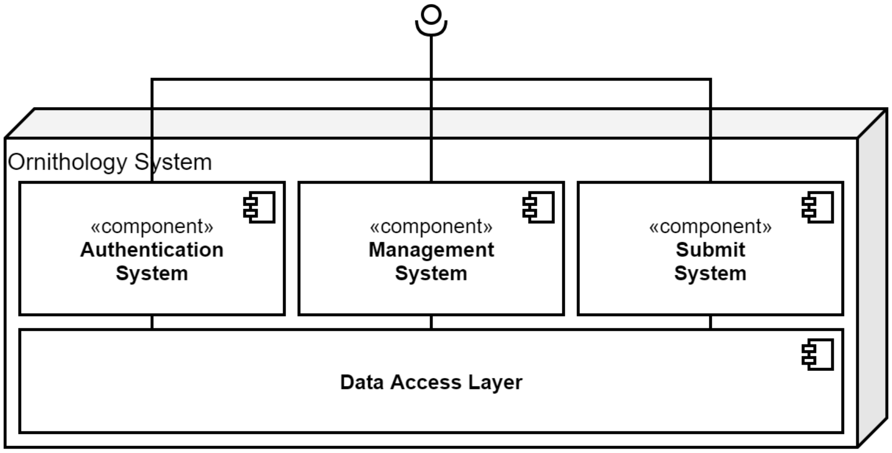

Figure 2: component diagram

Authentication System is used to verify users by email and name; Management System is used to allow Club officials to update database by the actions they took for confirming the sighting; Submit System is used for submitter to submit the sighting details; Data Access Layer is used to get information for other component from database.

All of those four component can be reused. Here I choose to illustrate "Submit system component" and "Management System component".

- Submit system component
  because the things submitter submitted are same in both two systems (characteristics of the bird, location, up to 3 photographs and a provisional identification), so this component can be reused.

- Management System component
  Club officials need to update database by the actions they took for confirming the sighting, in the new system, officials also need to do the same thing, so this component can be reused.

2.3 Question c

1. New requirements emerge when the software is used. e.g., a new function like use algorithm to automatically identify the category of birds.
2. New hardware or infrastructure is added to the system. e.g., to improve reliability, the association wanted to add a new server for distributed data storage.
3. Errors must be repaired. e.g., after releasing app, some users report that here are errors with some functions, i.e., the actual output is not equal to expected output.
4. The performance or reliability of the system may have to be improved. e.g., the software runs too slow to let users feel satisfied.

2.4 Question d

1. Endangered Bird Rescue
   - Authentication System component
     Users are able to verify their identity using their email and name before submitting the requests.
   - Submit System component
     Users are able to submit the endangered birds rescue request as same as original app, but the thing they will upload should be configurable. i.e., the users should upload some pictures of endangered birds, location, which kind of helping this organization should provide, etc.
   - Management System component
     The manager of the new system can set the status of request after doing some actions to rescue it. The function is as same as original application, but some details are needed to be configurable.
   - Data Access Layer
     The DBMS should be configurable to fit the name of entities and the structure of data.
   - Specialised application components
     The new app need to develop a new component for users to track the progress of rescuing.
2. Bird knowledge popularization App
   - Authentication System component
     Users are able to verify their identity using their email and name before commenting something in this app.
   - Data Access Layer
     The DBMS should be configurable to fit the name of entities and the structure of data.
   - Specialised application components
     - Comment System component
       Users are able to comment articles after authentication.
     - Management System component
       The manager of the app can delete and add some articles in this app.

2.5 Question e

- Budget and Cost
  Budget or cost should be a significant concern. Generally, creating new software is more expensive than refactoring, re-engineering, or maintaining existing one. The society should figure out if its budget is enough to create a new software, and can the additional expense of creating new software be compensated by its benefits compared to refactoring, re-engineering and maintaining the existing one.

- Undocumented business rules embedded in the existing system
  When developed the existing system, some important business rules may not be documented in the specification document, the programmer just write it as the process in software. so when replacing it by a new software, if the new software breaks those business rules, software request side may be subject to high risk. e.g., When deal with the database component, some business rules are potentially in the constraints (like CASCADE ON DELETE, CASCADE ON SET NULL ···).

- Lack of complete system specification
  There is rarely a complete specification of the legacy system. The original specification may have been lost. If a specification exists, it is unlikely that it incorporates details of all of the system changes that have been made. Therefore, there is no straightforward way of specifying a new system which is functionally identical to the system that is in use.

3 Software quality

3.1 Question a

1. Availability
   The systems should be available 24/7 to follow up any potential transfer or payment could happen, and the new function that maximize the interest customers can earn.

2. Reliability
   The system should be reliable in all cases, with back-up servers. When the main server is being attacked, the back-up servers can also deliver essential services like transfer or payment.

3. Safety
   The system should guarantee absolutely correct calculation of monetary amounts, accounts for the transfers in and transfers out.

4. Security
   The system should save the privacy of the bank system users by keeping the data accessible only for the authorized people, and it should be able to resist malicious intrusions.

5. Resilience
   The system should recover and continue to provide services in 1 hour after the main server hardware is damaged.

6. Repairability
   The system should be repaired well in 3 hours after the transfer or payment functions are failed.

7. Maintainability
   The system should be able to extend new functions, like it provides customers some investment strategies.

8. Error tolerance
   The system should double check the destination account number and name to avoid the input error when user input the wrong destination account number.

3.2 Question b

- The server is hacked by the attacker, and he changed or deleted data in database.
- The current rate was calculated incorrectly due to network delays.
- When the transfer was made, the money went to the wrong account.

3.3 Question c

- Regular backups and distributed storage can effectively improve the dependability of this system.
- consider the network delay when create the functions, and when use it, double check it.
- When design the system, the functions related to money should be withdrawable.

3.4 Question d

In this situation, new service needs to be tested its function, i.e., unit test, component test. Then the whole system needs to be tested for figuring out if the system can run well with the new component, i.e., system test.

- unit test
  Unit test focuses on testing the functionality of new function. It can reveal the defects in the component by test cases.
  The information we can get from this test should be:

   1. With the legal input, did the component output what we expected.
   2. with the illegal input, if the component will crash.

  In this case, this test will test whether the new function (automatically maximise the interest customers can earn) can run well and can it deal with the illegal input by some methods.

- component test
  Component test focuses on testing the function of component and the interaction of its subcomponents.
  The information we can get from this test should be:

  1. If component interface behaves according to its specification.
  2. If the subcomponents interact well.

  In this case, this test will test when this component for new function bases on other components as subcomponents, if it runs well. e.g., calculating the interest the customer earned bases on the log in function and the function which can get the rest of money in the account.

- system test
  System test focuses on testing the whole system. The information we can get from this test should be:

  1. If components are compatible
  2. If components interact correctly
  3. If there is a emergent behavior of this system

  In this case, this test will test if the whole system runs well, and whether there is any emergent behavior.