4M21 Software Engineering and Design 2023/2024

Solutions

Version: EP3/POK1

Question 1

a)

The Hick-Hyman law does not apply as the task is not open-loop.

b)

The overall function is `Address Customer Call` (or similar). The overall function receives at least one signal, *Customer Call*, and outputs at least one signal *Action*. There are at least two key subfunctions. The first is `Propose Response` which receives a *Customer Call* and a *Task Knowledge* signal and in response emits a *Proposed Response* signal. The second subfunction is `Take Action` which receives a *Customer Call* and *Proposed Response* signal and emits an *Action*. Labels can vary as long as the semantics are clear.

c)

Mental workload The mental workload now involves both assessing proposed options and judging whether the call needs to be handled in-person in a short amount of time. This increases mental workload.

Situation awareness Emergent issues in customer support may be less apparent as many calls are likely to be routed to an automatic response mechanism.

Complacency The complacency risk is relatively low as the cost of not choosing an option is handling the call in-person.

Skill degradation Skills and knowledge required to handle routine customer calls may degrade over time as operators no longer handle such calls.

Automation reliability The success of the system is highly dependent on the ability of the system to propose viable options. A failure in doing so will erode operator trust in the system and make operators unlikely to choose system-proposed options.

Costs of decision and action outcomes There is default option of the operator manually handling the call, hence catastrophic outcomes can be avoided. The primary cost is in the operator routing calls to the wrong system-proposed responses.

d)

Skill-based behaviour means high automaticity and happen without conscious control. It is unlikely to be exhibited in this case as the operator must listen to the call and understand the nature of the call to make a decision. Rule-based behaviour means using stored mental procedures. The operator is highly likely to develop rule-based behaviour to relatively quickly route call of a similar nature to system-proposed responses with practice. Knowledge-based behaviour means the operator is faced with unfamiliar situations where the operator has not developed any rules or knowledge for how to control the system. The operator therefore has formulate an explicit goal goal and develop a concrete plan to achieve this goal, typically carried out in a trial-and-error fashion. This

behaviour is likely to emerge during the initial use phase of the system as the operator learns when the system proposes responses and situations where the system is unlikely to propose a valid response. This behaviour will also emerge when customer calls entail situations that have never, or rarely, been previously encountered.

e)

Different answers are possible as long as they are valid within the context of the question. Here are three examples.

1. The system must not present more than four options to the operator for a customer call. Verification method: inspection.
2. The system must provide valid keywords from customer conversations to explain to the operator the triggers of a system-generated response. Verification method: test.
3. When options are shown to the operator, at least one option must be suitable at least 96% of the time. Verification method: analysis.

Assessors' comments: This was a popular question. The question asked candidates to analyse the user interface of a semi-automated call service centre. Many candidates were unable to recognise that the task was closed-loop and hence Hick-Hyman would not apply. Most candidates could draw a function structure of the system and reason about user-centric and system-centric automation evaluation criteria for the task. Many candidates could reason about rule- and knowledge-based interaction but failed to realise that skill-based interaction did not apply for this task. Most candidates could propose requirements, but many proposed requirements were vague or not verifiable. Some candidates failed to suggest ways to verify their proposed requirements.

Question 2

a)

Utility is whether the goal is important—relevant functionality. Usability concerns whether users are able to actually use the service in such a way that they can achieve the utility.

b)

Usability is relational because it happens due to the interaction of users, tasks and systems. In this case the relational property relates three different stakeholders, two different procurement systems, and the task of performing procurement in such a way that both cost and procurement time are minimised.

Usability is emergent because usability only happens as a result of users actively using systems to achieve their goals. In this case, three different stakeholders use two different procurement systems to minimise cost and procurement time. It follows that in order to discuss usability we need to understand the users (the stakeholders) and the tasks.

c)

The acceptability of a system is whether users choose to use a system or not. Users may choose not to accept the new system as it is more difficult to use, hence incurring time costs. This is an example of practical acceptability. Users may also choose not to accept the new system as it changes how they interact with other stakeholders, such as perhaps their manager. This is an example of social acceptability. The new system may also not be considered by users as they are unable to carry out the functions they need within in. Perhaps because perceived necessary functionality is not accessible to them. This is an example of non-use.

d)

Heuristic evaluation is a form of usability inspection. It has high efficiency in detecting some usability problems, in particular those that can be attributed to the visible parts of the user interfaces. However, heuristic evaluation will exhibit high variability and false positives/false negatives as many problems cannot be detected reliably. In this case, appropriation strategies developed by the users in response to the new system will not be detected using such a method. Further, issues that relate to the social relationships, such as manager–engineer–account relationships will not be detected. Induced risk in the system, due to either system problems or users misunderstanding system, are also unlikely to be detected. Hence, a heuristic evaluation can be used to efficiently identify surface-level usability problems with the visible part of the user interface. However, understanding immediate and emerging process issues induced by the system will require an alternative evaluation method.

Assessors' comments: This was a less popular question. It assessed the understanding of utility, usability, and acceptability in the context of an industrial use case. Candidates were generally able to delineate utility and usability and reason around the three facets of acceptability (practical, social, and non-use). Candidates struggled to fully explain the benefits and non-benefits of heuristic evaluation, frequently omitting that such an evaluation approach would fail to capture risks and emergent interaction issues.

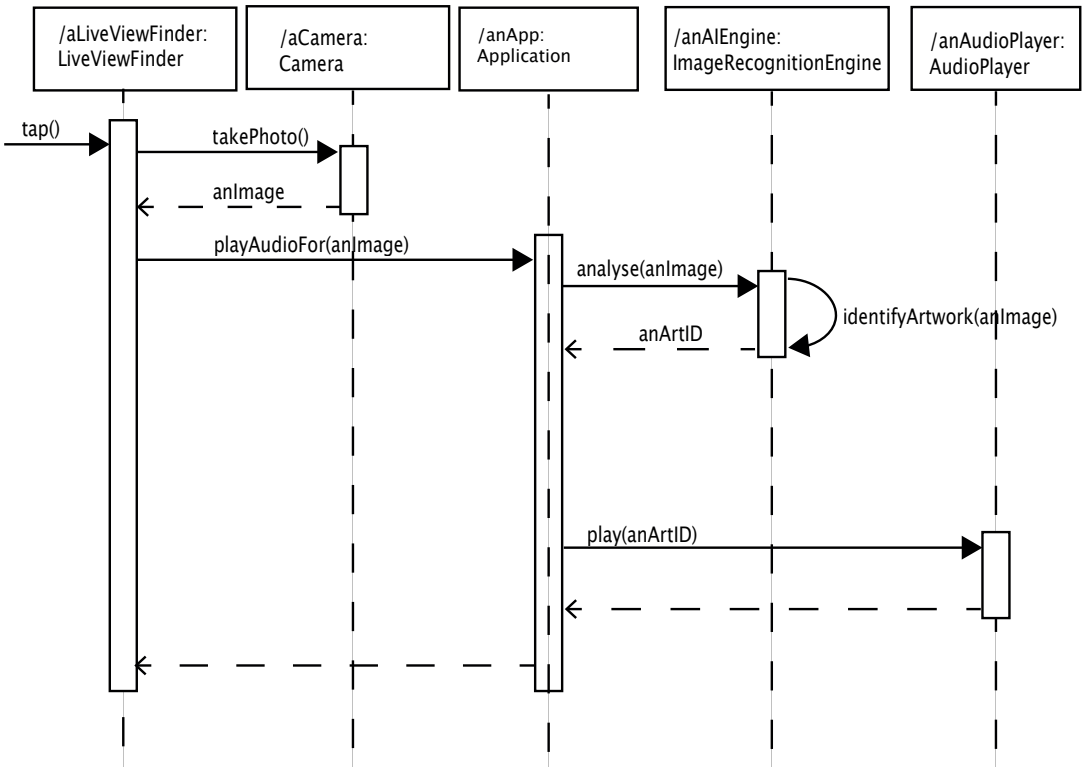
Question 3

a)

Class and sequence diagrams. Class diagrams describe the architecture of the system by specifying what classes are present in the system, and their relationships. Sequence diagrams show interactions between Objects.

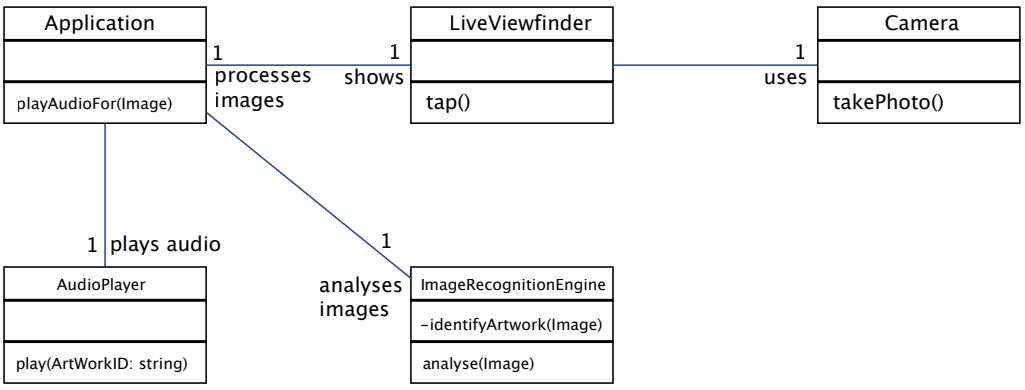
b) (i)

One of the possible solutions is presented below.



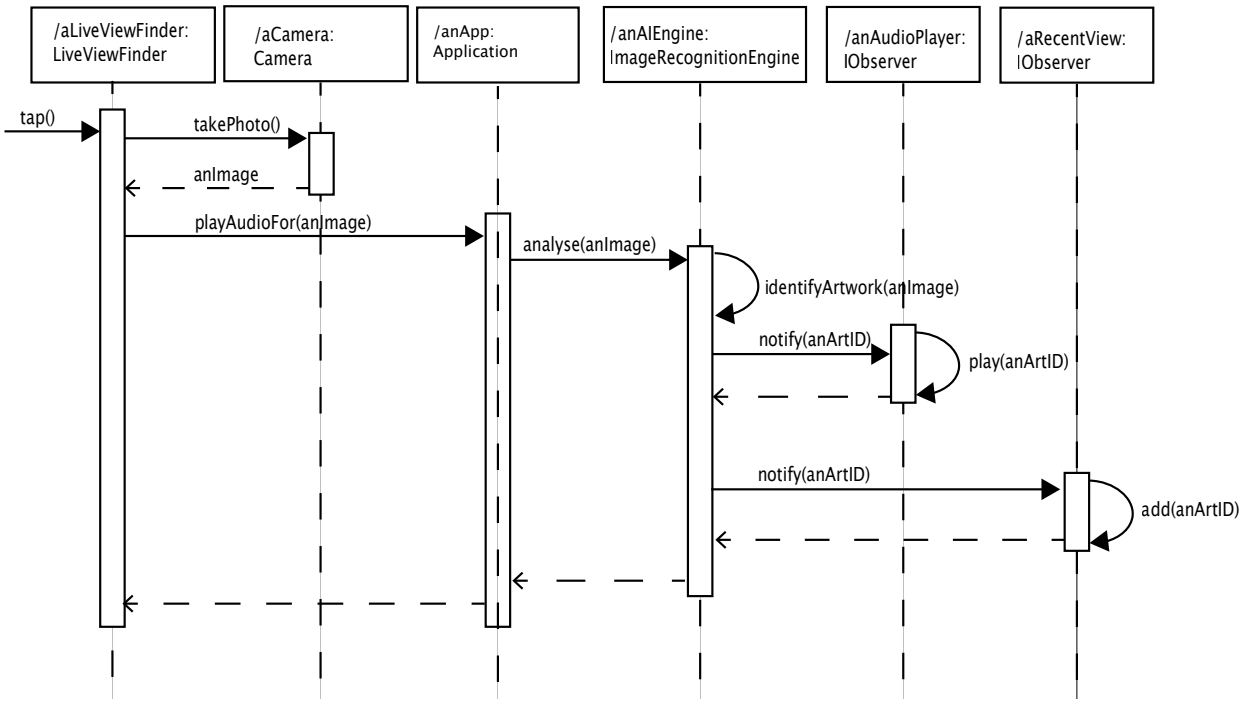
b) (ii)

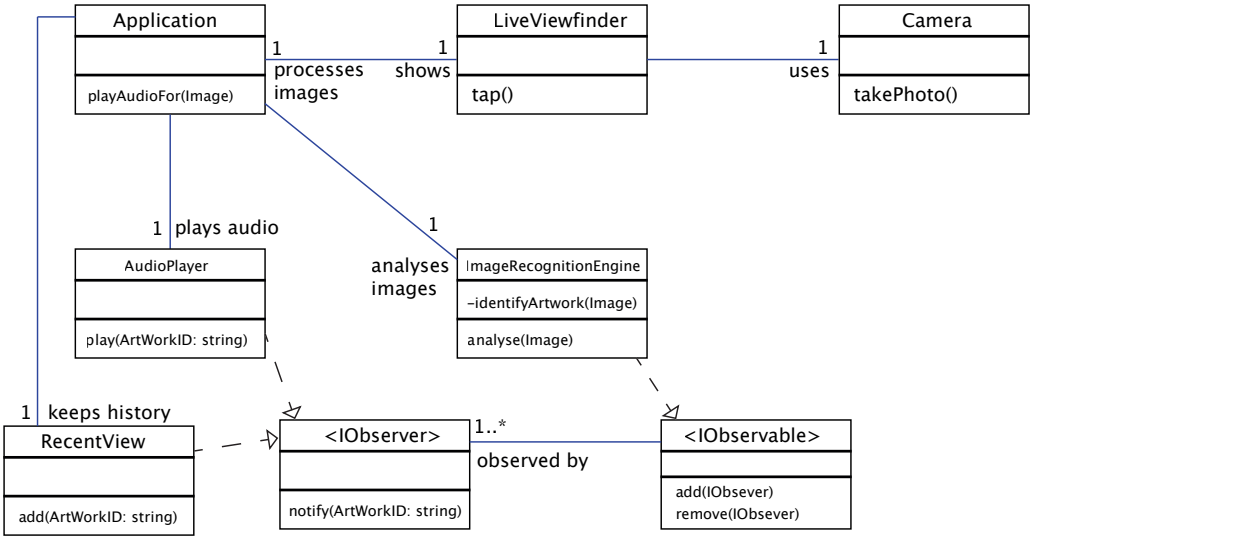
One of the possible solutions is presented below.



c)

One of the possible solutions is presented below.





Assessors' comments: The question was designed to test understanding of the key concepts of the object-oriented design such as polymorphism and inheritance; the ability to interpret the requirements and apply the main object-oriented design concepts in practice (in particular, principles of decoupling and abstraction), communicating the design through class and sequence diagrams; and the ability to extend the design by identifying a common design problem and figuring out a reliable solution to it in a form of an appropriate design pattern.

A popular question that candidates were able to answer reasonably well. Most students were

able to go through an independent design process successfully, identifying the key concepts and communicating the outcome clearly using the standard notation. Not everyone was careful when working with both class and sequence diagrams as often sequence diagram did not correspond to the class diagram. Some students were successful in identifying the observer pattern as the appropriate one. However, not everyone was able to apply the observer pattern correctly, there were some inconsistencies and inaccuracies in UML notation.

Question 4

a)

Patient safety Medical software is used by healthcare professionals to provide care for patience ⇒ the most important concern is patient safety.

Quality, accuracy and reliability Medical software is used to make critical (and often time critical) decisions about patient's health ⇒ quality (bugs, errors), accuracy and reliability can be a concern.

Security and data privacy Medical software may have access to sensitive patient data ⇒ security and data privacy may be a concern.

Usability Medical software is used by healthcare professionals who may be overloaded, under time pressure, under severe stress and may not be tech-savvy ⇒ usability and need for training might be a concern.

b)

In 80th, Therac-25 radiation therapy machine for treating cancer patients delivered massive overdoses of radiation, causing severe radiation burns and radiation poisoning in several patients. The main cause was poor software design, among others:

Insufficient testing Software released with a number of potentially fatal software bugs undetected. ⇒ rigorous (and in many cases) automated QA/testing procedures are well established and are facilitated by a number of advanced tools available on the market today.

Limited user interface Limited feedback and control for operators making it difficult to detect something is wrong and intervene. ⇒ the importance of user-centred design, the need to involve users from the very early stages and the value of user feedback are well understood today; usability studies are common.

No backup safety features by design No independent hardware failsafes or physical safeguards in case of software failure. ⇒ today stricter regulations and compliance requirements, increased scrutiny of medical software design process, increased emphasis on robust safety features and risk management, use of quality management systems (QMS).

c)(i)

This is the case of embedded software with strict safety and reliability requirements used in healthcare setting with specific regulations and compliance requirements; in general the requirements are well understood in this project and are unlikely to change significantly throughout development. The waterfall model structured approach is therefore a good fit.

The key stages of the waterfall model are analyses, design, implementation, testing, deployment and maintenance.

Waterfall model is well documented (starting with specification) approach is beneficial to ensure meeting the standards for compliance, quality assurance, risk management, project management and meeting regulatory requirements.

c)(ii)

Patient safety Conduct thorough risk analyses and implement risk management plan. Implement Quality Management System (QMS) to ensure relevant standards and regulations are met.

Quality, accuracy and reliability Select suitable software development model, implement rigorous software development practices at each stage of software development: consult key stakeholders at the stage of requirement gathering, follow principles of modularity and decoupling at the stage of design, follow coding standards and conduct code reviews at the implementation stage, define rigorous testing plan including unit, integration, validation and verification, resource exhaustion and recovery, performance testing, implement efficient version and configuration control to safely manage deployment and maintenance stages.

Security and data privacy Seek advice from security specialist, conduct security evaluation and testing.

Usability Get frequent feedback from healthcare professionals and patients' parents from early stages, conduct usability tests.

Assessors' comments: A reasonably straightforward question on software engineering methodologies and their application. Those who did attempt the question answered most parts well.

Most candidates successfully discussed key concerns related to the introduction of new medical software and identified Therac-25 failure as an example of medical software failure in the past, understood its causes and modern practices commonly adopted today to reduce the risk of this form of failure. Most candidates were able to propose a suitable software development model and justify their answer. Some candidates did not read the last part of the question carefully and some candidates did not provide direct answers to the questions, making a variety of generic statements centred mainly around generic testing instead. A few identified the type the software and suggested a suitable software development model, yet backtracked almost immediately and listed a lot of generic agile techniques.