

ENGINEERING TRIPOS PART IIA

---

Tuesday 24 April 2012 9 to 10.30

---

Module 3F6

SOFTWARE ENGINEERING AND DESIGN

*Answer not more than **three** questions.*

*All questions carry the same number of marks.*

*The **approximate** percentage of marks allocated to each part of a question is indicated in the right margin.*

*There are no attachments.*

STATIONERY REQUIREMENTS

Single-sided script paper

SPECIAL REQUIREMENTS

Engineering Data Book

CUED approved calculator allowed

**You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator**

1 (a) Explain the following key concepts of an object oriented approach and describe the benefits they provide.

- (i) *Classes, Objects and Abstraction* [5%]
- (ii) *Encapsulation and Data Hiding* [5%]
- (iii) *Inheritance and Polymorphism* [5%]

(b) An online store offers music for sale both in physical (CDs) and electronic (MP3 files) form. A customer can buy CDs that are sent to their address as well as MP3 files that are emailed to them. A single order can contain a mixture of CDs and MP3 tracks.

- (i) Design an object oriented system that allows for the dispatching of items in an order. Draw a *Class Diagram* that describes the system. [25%]
- (ii) Draw a *Sequence Diagram* to demonstrate a scenario for dispatching an order that contains 1 CD and 1 MP3 track. [25%]
- (iii) Extend your design to support sales of other products such as books and videos, where books include paper books and ebooks, and videos include DVDs and Blu-ray movies. Show an updated *Class Diagram*. [20%]
- (iv) Update your *Class Diagram* further to allow customers to order multiple copies of each product. [15%]

- 2 (a) Describe the key processes in the software lifecycle. [15%]
- (b) Compare and contrast the *Waterfall Software Development* model and the *Extreme Programming (Agile)* approach. [25%]
- (c) A software engineering team is investigating why there is a decrease in the number of visitors from the Google search engine to their hotel reservation website. It looks like their website is no longer appearing in the search results. Further investigation shows that a recently joined developer made a small change in the web server configuration. This change is incompatible with Google's indexing policies, i.e. the search engine stopped indexing the hotel reservation website. The developer was unaware of the effect of this change.
- (i) Apply the *Five Whys Root Cause Analysis* technique to this case and propose suggestions to improve the software development process of the hotel reservation website team. [30%]
- (ii) Design a *Testing Strategy* for the hotel reservation website. Identify the main risks, the key test areas and specify the appropriate tests to mitigate these risks. [30%]

3 (a) Describe the difference between abstract classes and interfaces. [10%]

(b) A smartphone application for taking pictures provides a user with a *Camera View* and a *Recent View* as illustrated in Fig. 1. The *Recent View* shows the four most recent pictures taken. When the user taps on the *Camera View* the picture is taken and saved on the phone; additionally, the *Recent View* is updated with the new picture. The *Sequence Diagrams* in Fig. 2 and Fig. 3 show the initialisation of the application and the reaction to the user tapping on the *Camera View*.

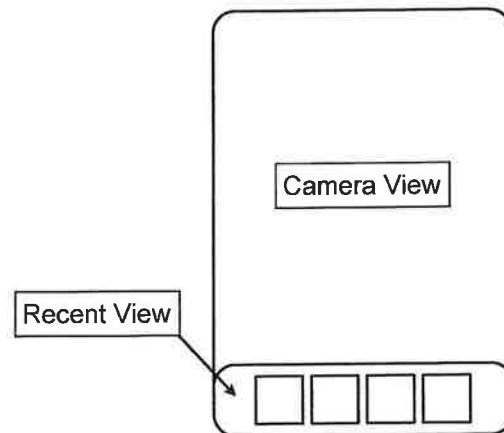


Fig. 1

(i) Using the *Sequence Diagram* and the provided description, draw a *Class Diagram* of the system, showing the classes used and their associations. [25%]

(ii) In the current design, the class *CameraView* has to maintain a reference to the class *Application* in order to save a captured image. Furthermore, the *save* method is also responsible for updating the class *RecentView*. This is conceptually incorrect and introduces tight coupling between these classes. Improve the design by using the *Observer Design Pattern* to allow the *CameraView* and *RecentView* to be notified of the captured image by the *Camera*. Draw updated *Class and Sequence Diagrams*. [40%]

(iii) Extend the design by adding a class *CounterView* to the application screen that will display the total number of pictures taken. Assume the *CounterView* is initialised with the number of existing pictures at the start of the application and then the counter is updated each time a picture is taken. [25%]

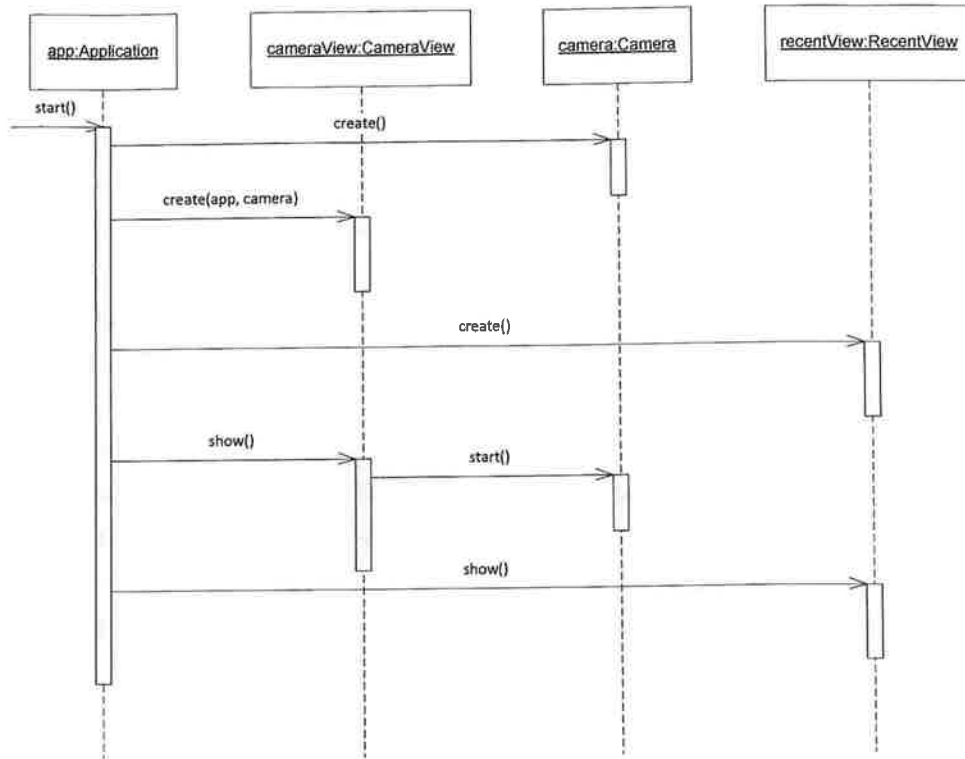


Fig. 2

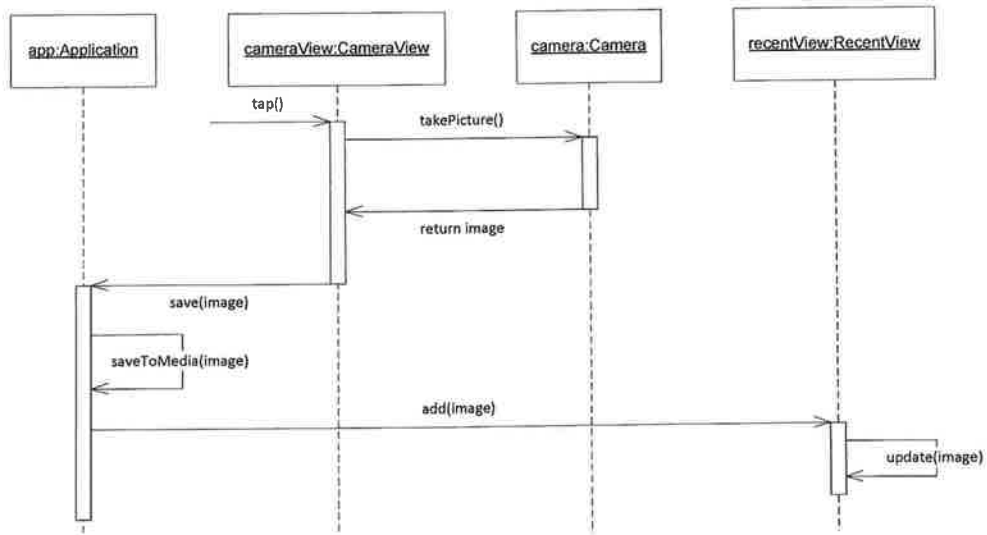


Fig. 3

- 4 (a) Describe the principals of a good *User Interface* (UI) design. [10%]

(b) In many Internet applications, a secure access is required. Some companies offer small portable devices called *One-Time Password* (OTP) tokens that are capable of generating a unique number that can be used as a secure password to access the service. The password is often valid for 30 seconds only. Each token is unique and generates distinctly unique passwords.

Design a *User Interface* (UI) for a software application for a smartphone that implements a time-based OTP generator to provide the following functionality. At first start, the application initialises itself by generating a random serial number and transmitting it to a remote website for registration. This serial number is used by the user to identify the OTP application to the customer support services in case of any problems. As a result, the user should be able to view the serial number when needed. The user should also be able to re-initialise the application, i.e. generate a new serial number. Once initialised, the application displays the OTP which is automatically updated every 30 seconds. The OTP should consist of 6 digits that can be entered by the user into the website to securely access the Internet service.

- (i) Identify the main screens required in the application and their contents, including any data and information that must be displayed and the main UI interaction elements such as buttons. Explain the purpose of each component and the design considerations. [40%]
- (ii) Identify the key use cases and show the corresponding UI screen flow. [30%]
- (iii) Design a *Usability Test* for the application. [20%]

**END OF PAPER**