

3F6 Software Engineering and Design: 2012 Solutions

Dr Elena Puns kaya

20 January 2012

1. (a) i. Bookwork. Classes are the key abstract concepts of the system that encapsulate data and behaviour of system objects, their relationships and interactions. Objects are instances of their respective classes that exist during execution of the software.
Benefits: managing complexity, in particular, the following can be mentioned
- allow us to describe complex systems through meaningful abstractions
 - support different level views of the system, including conceptual (high-level abstraction) and implementation (low-level abstraction) levels
 - allow all project members to have a consistent shared understanding of the system
- ii. Bookwork. Encapsulation is a mechanism of defining the functionality necessary for the objects to be used and hiding the details of how this functionality is implemented
Benefits: extensibility of the system, in particular, allows to minimise the effect of changing design decisions for certain parts of the system on the whole system
- iii. Bookwork. Inheritance is a relationship between classes where one (the subclass) extends the data and behaviour of another (the superclass).
Polymorphism is a mechanism for treating similar objects in the same way and requesting the same action from them, yet, allowing this action to be performed in different ways. Benefits: allows us to generalise operations on similar concepts (classes) yet support distinct data and behaviour where necessary.
- (b) i. The required class diagram is shown in Fig. 1 . Any reasonable naming conventions can be used.

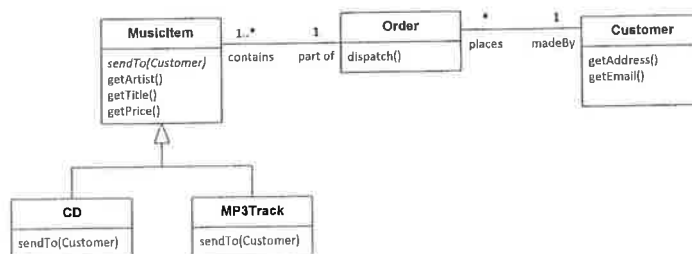


Figure 1:

- ii. The required sequence diagram is shown in Fig. 2 . Any reasonable naming conventions can be used.
- iii. Generalise all items sold as Product
- implement default `sendTo(Customer)` as shipped by post
 - implement `getPrice()` and `getTitle()` as they are common for all items

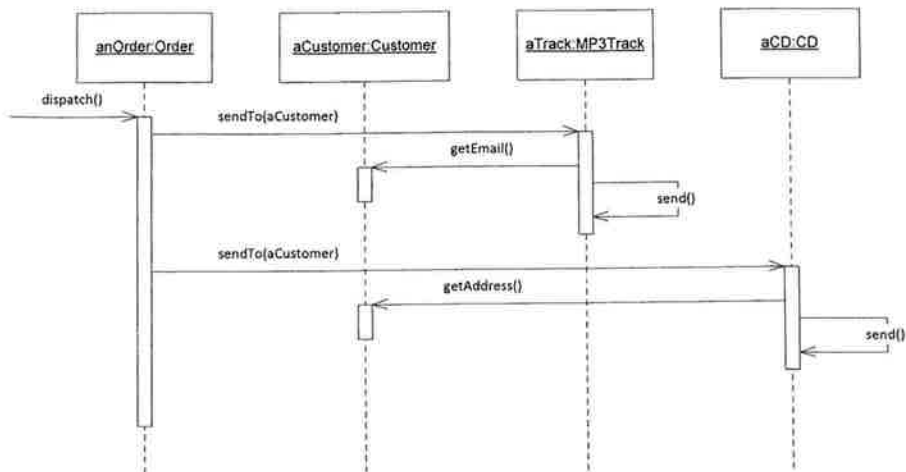


Figure 2:

- iv. Use Decorator Pattern for multiple items, say, `MultipleProduct`. `MultipleProduct.getPrice()` will calculate the total price based on the single item price `MultipleProduct.getTitle()`, `sendTo()` will call through to the original item `getTitle()`, `sendTo()`

The updated class diagram for (iii) and (iv) is presented in Fig. 3.

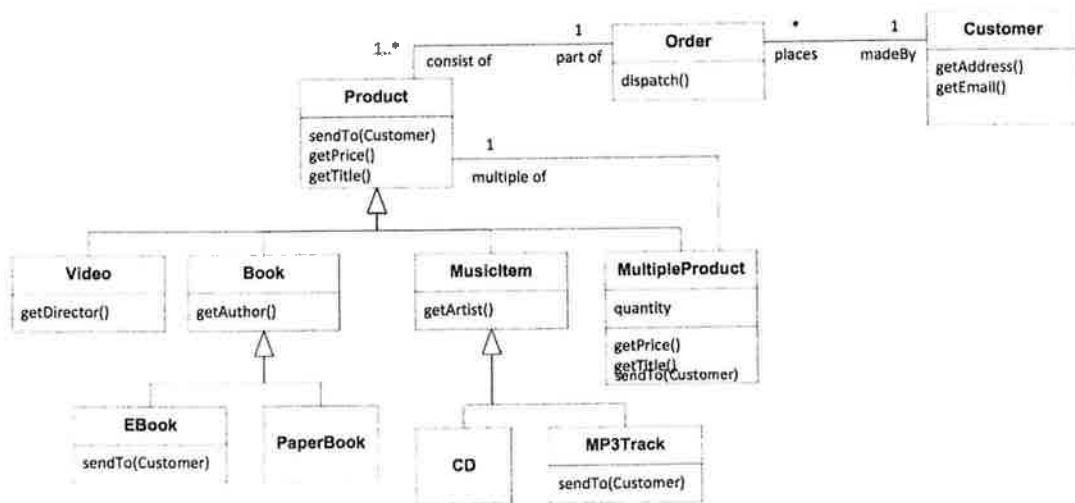


Figure 3:

2. (a) Bookwork. The following key processes must be mentioned: Analysis, Design, Implementation, Building, Testing, Deployment, Maintenance.
- (b) Bookwork.
- (c) i. • Why did the search traffic disappear? Answer: the search engine stopped indexing the site.

- Why did the search engine stop indexing the site? Answer: the web server configuration is incompatible with the search engine guidelines.
- Why is it incompatible (it worked well before)? Answer: a configuration change was made in a recent update that wasn't noticed.
- Why was it not noticed? Answer: the developer did not ask anyone to review the change and the automated testing did not pick it up.
- Why did not the developer ask for a review? Answer: the developer was not aware of the effect of this configuration parameter on the search engine indexing and considered it trivial.

Examples of the suggested corrective measures at each step:

- extend web server monitoring to check for any significant changes in the website appearance in search
 - add comments in the server configuration, highlighting the current value compatibility with indexing
 - implement alerts for each server configuration change to a few experienced members of the team
 - provide training to the developer on code reviews and extend the automated tests to check this configuration parameter
 - provide training to the developer on search engine indexing
- ii. The Testing Strategy should cover a number of areas such as
- Unit tests, Integration tests
 - Functional acceptance tests
 - Non-functional acceptance tests (e.g. scalability, security)
 - Usability testing, A/B testing

3. (a) Bookwork

(b) i. The required class diagram is shown in Fig. 4. Any reasonable naming conventions can be used.

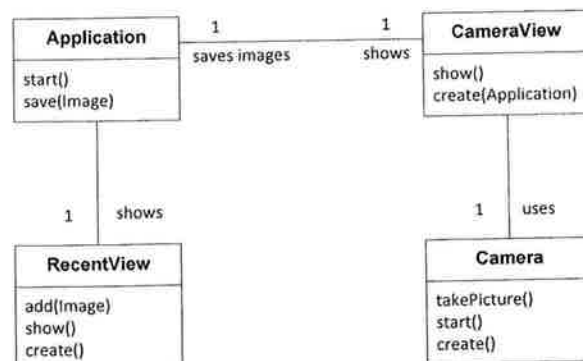


Figure 4:

ii-iii. The required class and sequence diagrams for the initialisation of the application and the reaction to the user tapping on the Camera View are shown in Fig. 5, 6 and 7. Any reasonable naming conventions can be used. Fig. 5 also illustrates extension of the diagram as a result of introducing class `CounterView`.

4. (a) Bookwork.

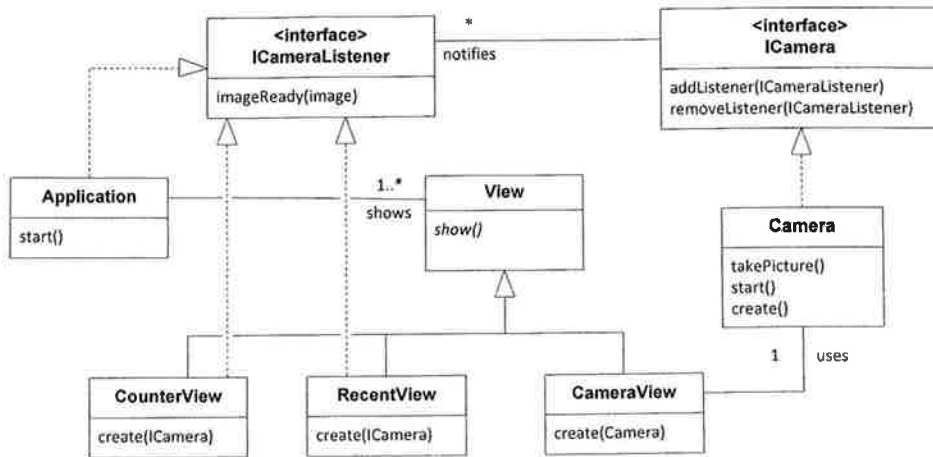


Figure 5:

- (b) i. Examples of the main screens that can be identified are presented in Fig. 8. Any reasonable naming conventions can be used.
- ii. Some of the use cases are presented in Fig. 9. Any reasonable naming conventions can be used.
- iii. The Usability Test should describe
- method
 - test setup: how the use cases are presented to the users (e.g. dummy web pages)
 - approach: qualitative interviews by user experience specialists
 - session time: e.g. 45 mins
 - target users sample: gender, age, level of experience with Internet banking etc.
 - scenarios: what specific actions the users are asked to perform
 - feedback
 - qualitative reviews (how did the user feel about using the product)
 - comparison to the current method
 - subjective ranking in pre-defined categories (e.g. easy to understand, innovative, secure)
 - observations made by user experience specialists during the scenarios (e.g. users entered half a code and then the code changed, requiring to start again)
-

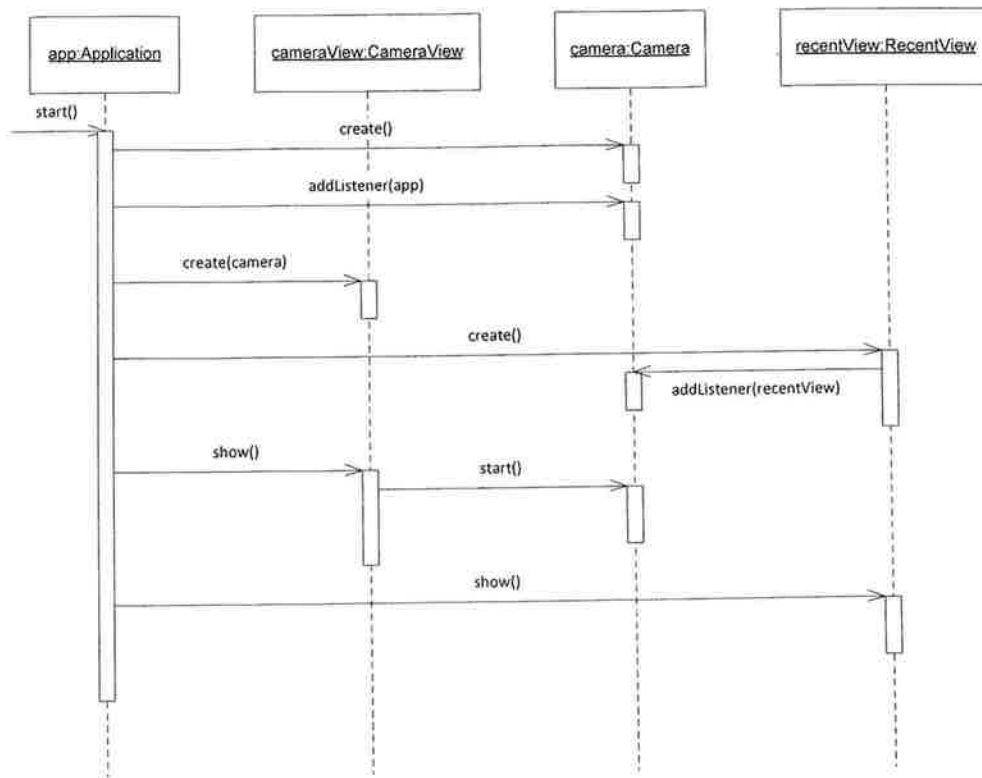


Figure 6:

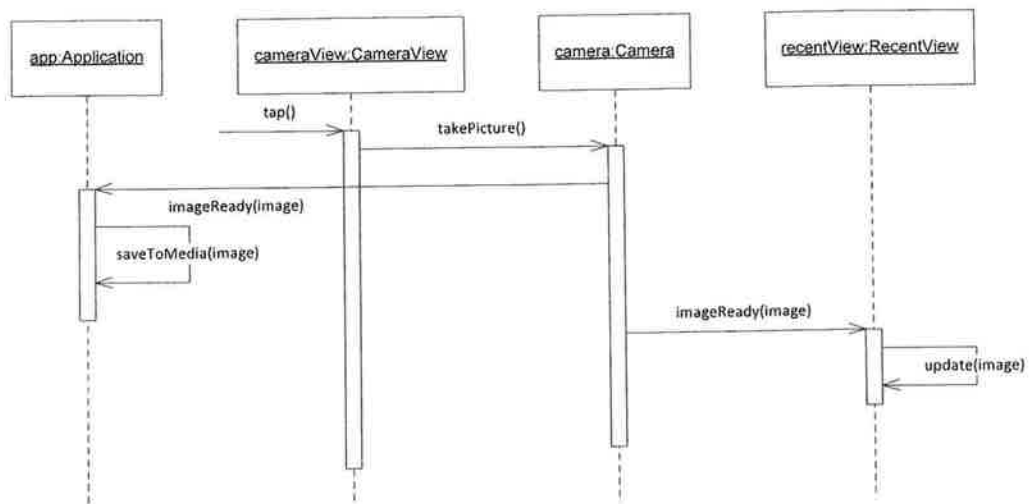


Figure 7:

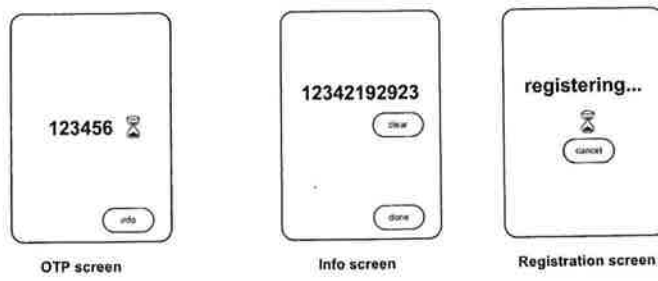


Figure 8:

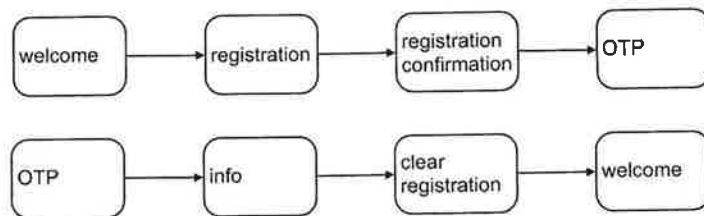


Figure 9: