

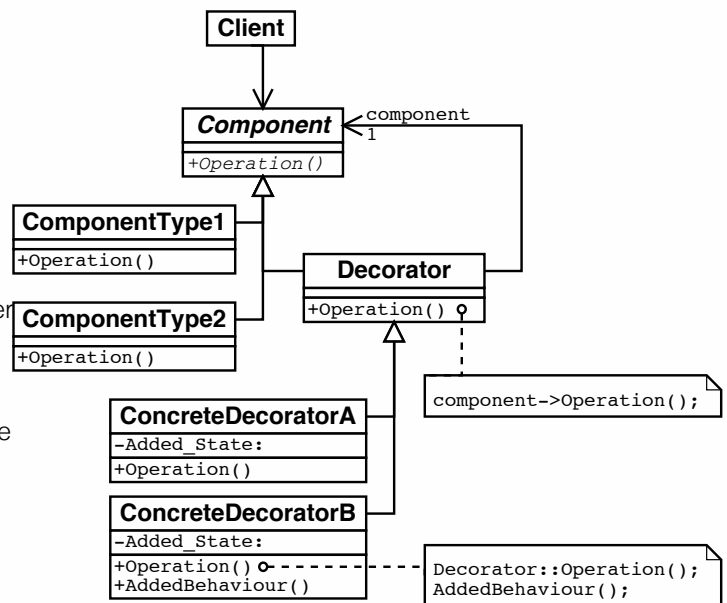
# 3F6 Software Engineering and Design: 2015/16 Solutions

Dr. Richard E. Turner and Dr. Elena Punskeya

19 February 2016

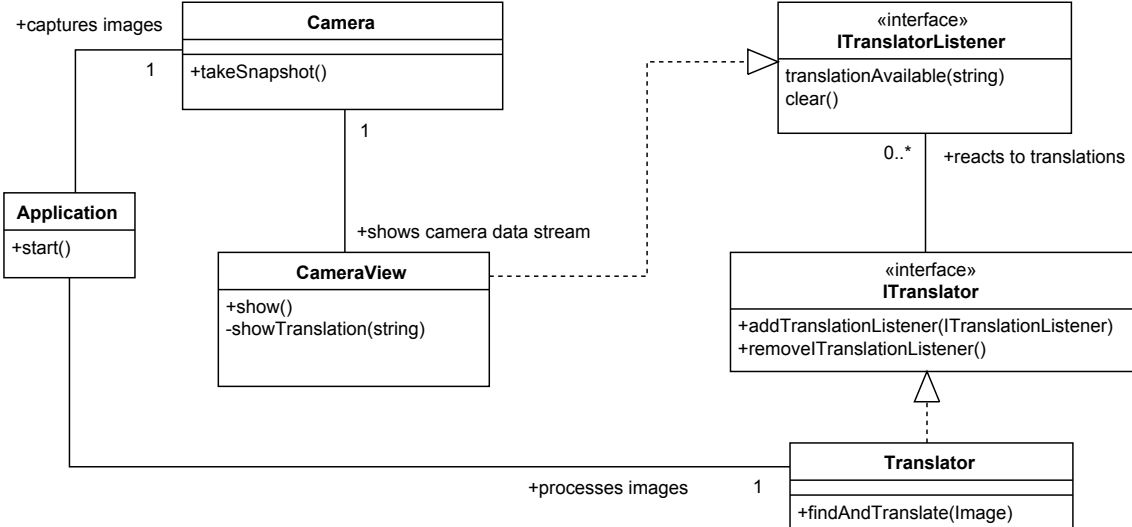
- **Decorator Pattern provides a way of adding optional functionality (“decoration”) to all classes in a hierarchy without changing the code for either the base class or any of the subclasses**

- Using this pattern, **multiple decorations** can be applied to an object, e.g. we can add a picture frame and scrollbars (in either order) to a picture in the drawing editor. If there are several different kinds of decoration that we want to be able to use, we can derive a number of classes from the Decorator class to handle these separate kinds of added functionality

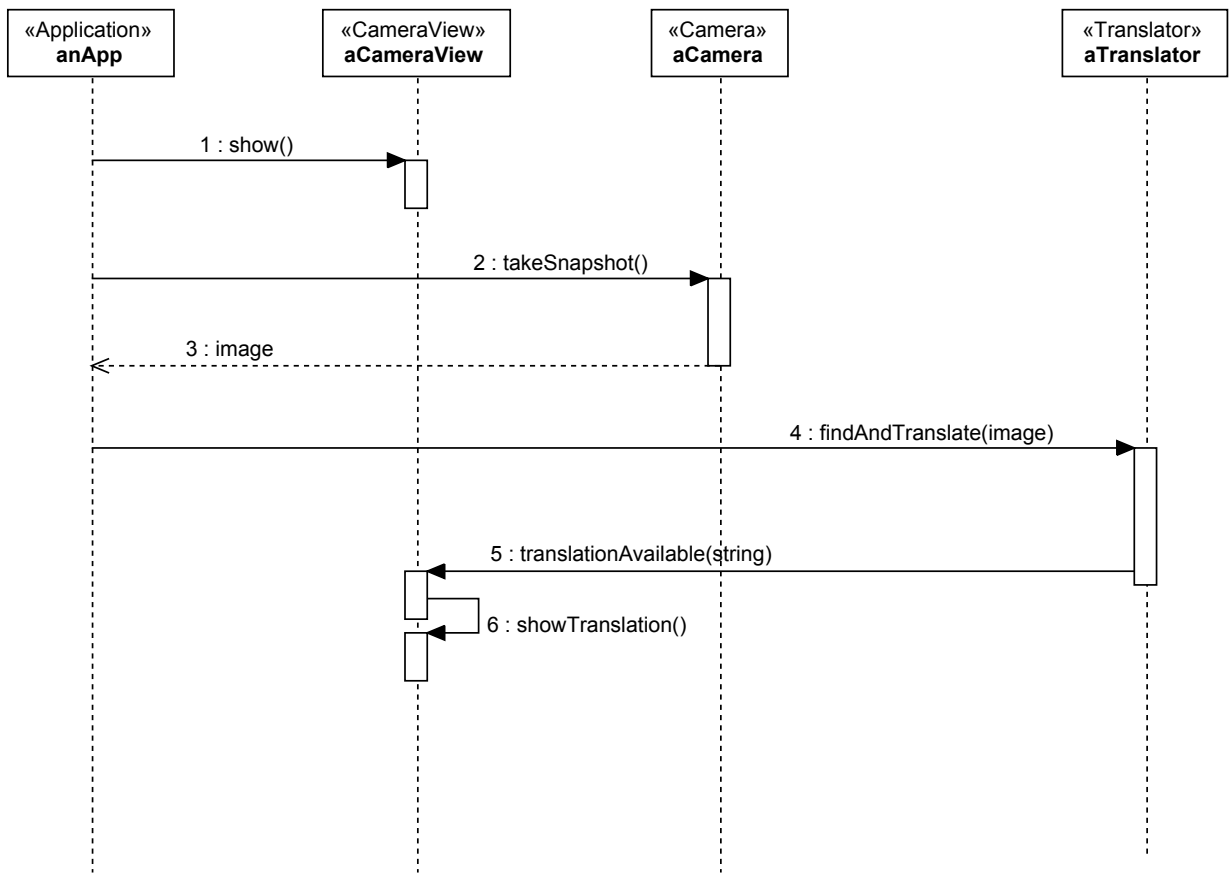


1b

(i) one of the possible solutions is presented below



(ii) one of the possible solutions is presented below



- **A use case describes a typical user activity from start to end. A good set of use cases motivates a design, and provides a basis for testing.**
- **A use case will consist of**
  - Definition of the user's goal
  - List of pre-conditions (e.g. information needed before the task can start)
  - Criteria for successful completion
  - List of main steps in the activity flow
  - Any extensions/alternatives to this use case
- **This is also called *activity based planning***

(a) Bookwork.

(b)

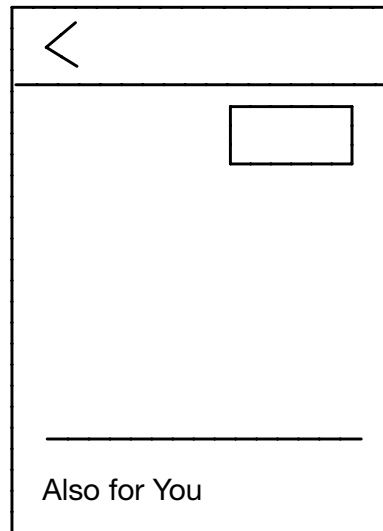
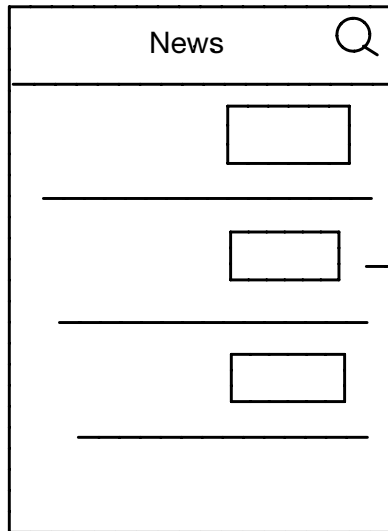
(i) One of the possible solutions is presented below.

Display short news summaries as a list with some photos if desired

Enable scrolling or introduce an alternative method to display a number of items

Group items by topics or otherwise

Check for news stories automatically and display the message "Checking for new stories" or slide down to download new stories



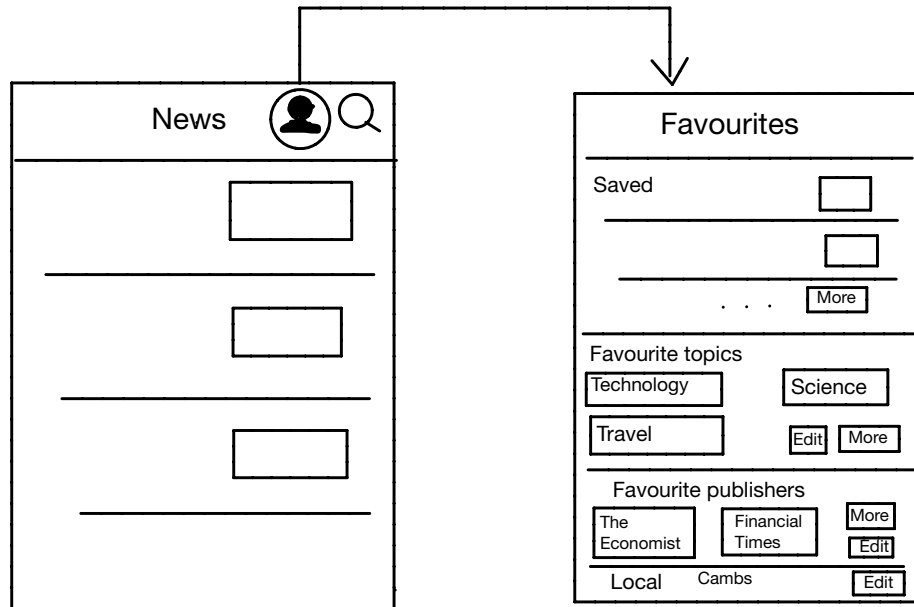
Add Search button, pressing Search takes the user to the Search page with some search suggestions or options to search different topics

Clicking on each item would take the user to the full story with pictures, videos, audio if applicable.

Suggestions of other stories that might be of interest are displayed at the bottom of the page

Swipe or use Back button to go back to the list of headlines

(ii) One of the possible solutions sketched below



Pressing on More in Saved takes the user to the page Saved, the layout of the page is similar to the layout of the main page with the list of the headlines. However, the user can delete the news items saved and there could be a shortcut (button) that allows the user to share the articles quickly without going to the end of each story.

Otherwise the button Share is introduced at the end of each story that allows the user to email, text, etc. the link to the article.

The button Edit in Favourite Topics and Favourite Publishers allows the user to add/delete different Topics/Publishers. Use gestures as appropriate to allow to delete items quicker.

The button More in Favourite Topics and Favourite Publishers takes the user to corresponding page where all Favourite Topics and Favourite Publishers correspondingly are displayed. There is an option to Edit and a list of suggestions of Topics or Publishers of interest could also be given.

The button Edit Location allows the users to edit their preferences for local news, location could be initially detected automatically.

If Saved, Favourite Topics and Favourite Publishers are empty useful tips how to Save news stories and specify Favourite Topics/Publishers could be presented correspondingly.

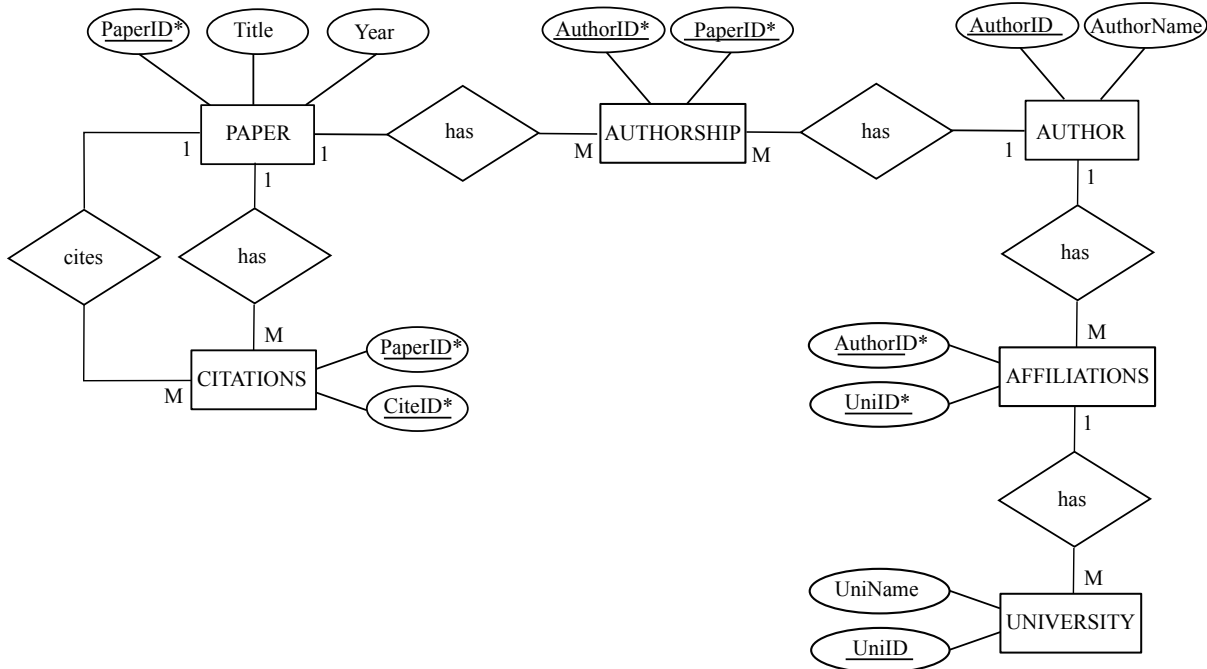
### (iii) Tablet optimisation

- larger screen and more space, heavier device - users hold tablets with two hands, bottom left and right are most reachable areas for users tap with a thumb - instead of list view, use similar to the print publication layout with the summaries on the left and on the right and top stories taking more space, make each story a reachable target for the fingers while still holding the tablet

- introduce different portrait and landscape layouts
- use larger and better quality pictures, and larger font



- 3 (a) “Evaluate the database design and suggest how to improve it. Draw an *Entity-Relationship Diagram* to illustrate your answer. [20%]”



- (b) “Show the updated design of the tables including any new Entities and Attributes added in the answer to part (a). Identify the Primary and Foreign keys used. [15%]”

The new tables are:

PAPER [PaperID, Title, Year]

primary key: PaperID

AUTHOR [AuthorID, AuthorName]

primary key: AuthorID

UNIVERSITY [UniID, UniName]

primary key: UniID

AFFILIATIONS [AuthorID, UniID]

primary key: (AuthorID, UniID) secondary key: AuthorID and UniID (referencing AUTHOR and UNIVERSITY)

AUTHORSHIP [AuthorID, PaperID]

primary key: (AuthorID, PaperID) foreign key: AuthorID and PaperID (referencing AUTHOR and PAPER)

CITATIONS [PaperID, CiteID]

primary key: (PaperID, CiteID) foreign key: PaperID and CiteID (both referencing PAPER PaperID)

- (c) “Design a query to return the titles of all the papers that Dirac has authored. Express your answer using relational algebra or SQL code and explain your solution. [20%]”

```
SELECT PAPER.PaperTitle FROM PAPER JOIN AUTHORSHIP ON PAPER.PaperID = AUTHORSHIP.PaperID JOIN AUTHOR ON AUTHORSHIP.AuthorID = AUTHOR.AuthorID WHERE AUTHOR.AuthorName = 'Dirac';
```

This query joins the tables PAPER to AUTHORSHIP (on paperID) and joins AUTHOR (on AUTHORSHIP authorID) and then selects out AuthorName=Dirac and projects onto the paper's title.

- (d) “Design a query to return the titles of all the papers cited by papers cited by Dirac. Design a suitable query and express your answer using relational algebra or SQL code. Explain your solution. [35%]”

```
SELECT DISTINCT Z.PaperTitle FROM CITATIONS AS X JOIN CITATIONS AS Y ON X.CiteID = Y.PaperID JOIN PAPER AS W ON X.PaperID = W.PaperID JOIN PAPER AS Z ON Z.PaperID = Y.CiteID JOIN AUTHORSHIP ON W.PaperID = AUTHORSHIP.PaperID JOIN AUTHOR ON AUTHOR.AuthorID = AUTHORSHIP.AuthorID WHERE AUTHOR.AuthorName = 'Dirac';
```

The key to formulating this query is to join CITATIONS to itself. This requires use of the rename operator. Once this is done, PAPER must be joined to each of the CITATIONS tables (this also requires the rename operator). The final steps essentially require what was needed for part (c) in order to select those papers written by Dirac and project onto the titles.

- (e) “The company wants to carry out many queries similar to the one in part (d). Describe features that could be added to the database to accelerate such queries and detail potential disadvantages they may have. [10%]”

Secondary indices could be added to the database. Four secondary indices could be added to PaperID and CiteID in CITATIONS, and also AuthorID in PAPER and AUTHOR. It might also be useful to add a secondary index to Name in AUTHOR. Adding these indices will accelerate queries, but they will slow down additions/modifications/deletions. This might be especially problematic for the CITATION and PAPER relations as new papers and citations will continually be added to it, potentially at a quick rate.

- 4 (a) “State the four ACID properties of transactions and explain why each is important. [20%]”

Atomicity: each transaction must be atomic, otherwise it would not be possible to recover cleanly from an aborted transaction.

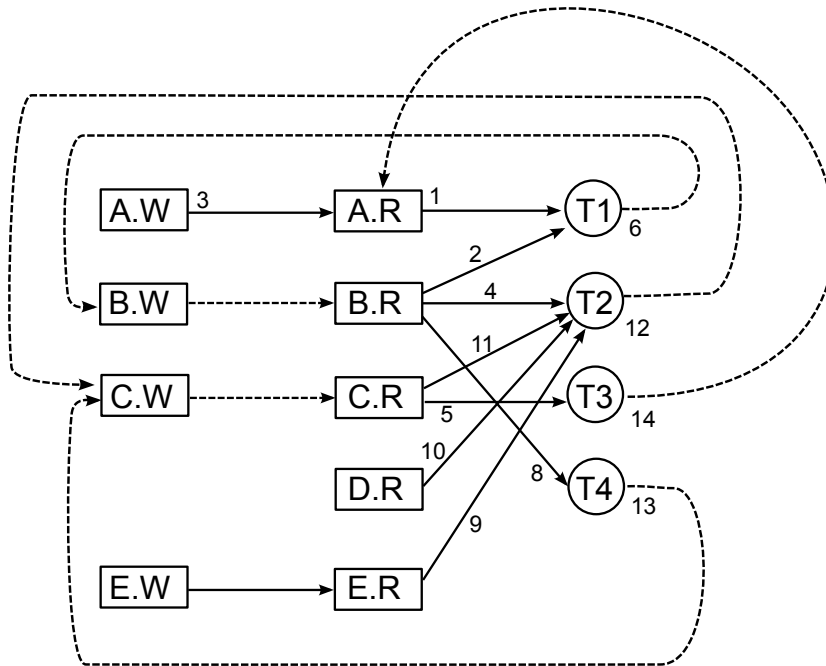
Consistency: transactions must leave the system in a consistent state. This is essential since any sequence of transactions can be aborted at any point.

Isolation: results of an incomplete transaction must not be visible to any other transaction. Otherwise, another transaction might see an inconsistent state and produce erroneous results.

Durability: the system must not fail between a transaction committing and the results of the transaction being recorded in the system state.

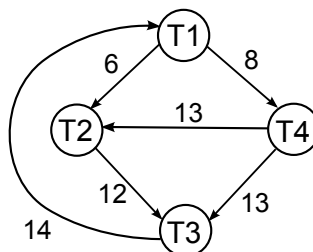
- b) “Draw a *resource allocation graph* for this sequence of transactions and hence determine the first point at which deadlock occurs. [40%]”

The resource allocation graph is shown below up to and including step 14 at which point deadlock occurs (**the definition for deadlock is: the first time when a directed cycle appears in the resource allocation or wait-for-graph**).



- c) “Draw the corresponding *wait-for-graph* at deadlock. [20%]”

The corresponding wait-graph is shown below.



- d) “Explain how a system can recover from deadlock and discuss the criteria for choosing a *victim*. What would be the best choice of victim in this case and in what order would the remaining transactions then complete? [20%]”

Recovery from deadlock involves:

- aborting a transaction which will break the deadlock loop (called the victim)
- rolling back to the last check point
- redoing the remaining transactions
- restarting the victim at some later point in time

The victim is usually selected using the following criteria:

- has not been running for a long time
- has made few updates
- is blocking multiple transactions

In this case, the choice lies between T1 and T3 (aborting T2 or T4 will not avoid the deadlock). T3 is the most sensible choice based on recency and fewer updates. T2 would complete first, followed by T4, and finally T1.

### **Principal Assessor's comments:**

#### Q1 Design Patterns

The least popular question and the worst answered question. The average was dragged down by a fairly large number of incomplete / partial attempts. In addition, a number of candidates could not reproduce the observer design pattern that was specifically asked for in part (b).

#### Q2 User Interfaces

A popular question that was reasonably well answered by most. Many candidates were able to identify and devise a sensible design for the mobile phone application's screens, but some missed discussing UI flow sufficiently. In addition some failed to identify sensible modifications for tablet devices.

#### Q3 Database design and queries.

This question was well very answered in the main. Many candidates failed to identify what attributes secondary indices should be applied to in part (e).

#### Q4 Concurrency control.

This question was very well answered in the main (there were a number of essentially perfect solutions) but a few candidates scored poorly. The candidates who struggled failed to identify when transactions would wait (and therefore stop acquiring locks) when constructing the resource allocation diagram.