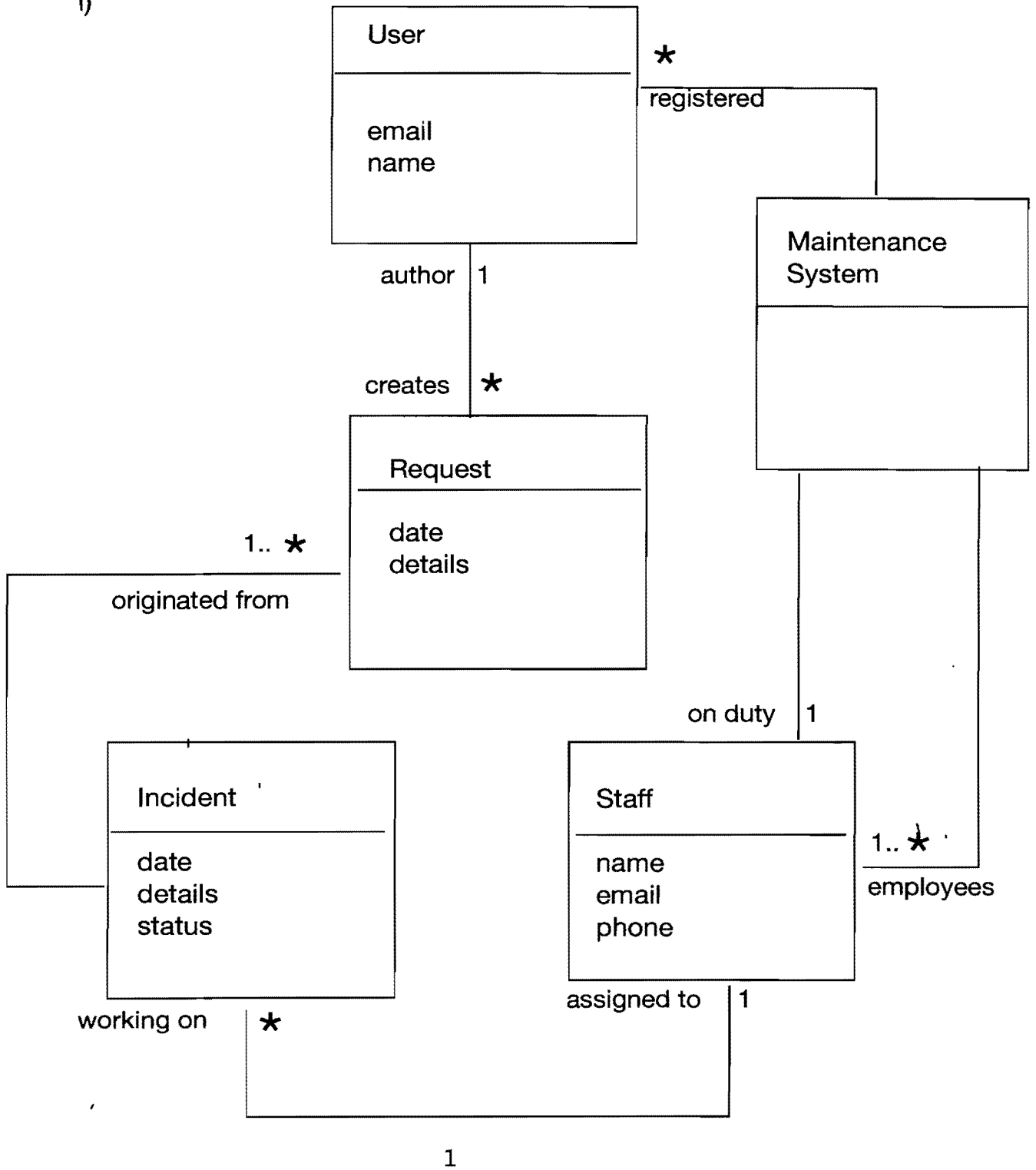


3F6 Software Engineering and Design 2013  
Solutions

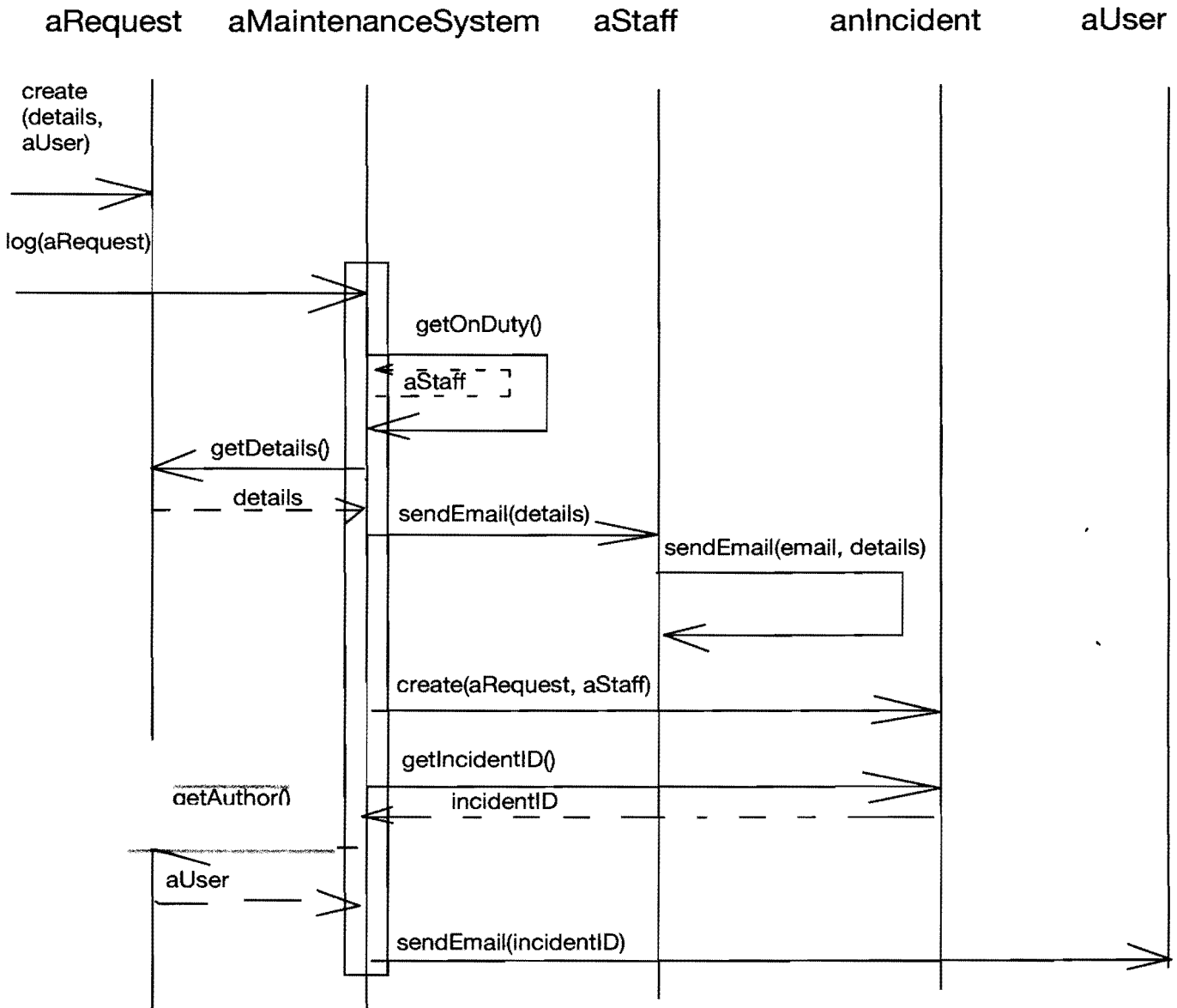
1. One of the possible solutions is presented below

a)

i)

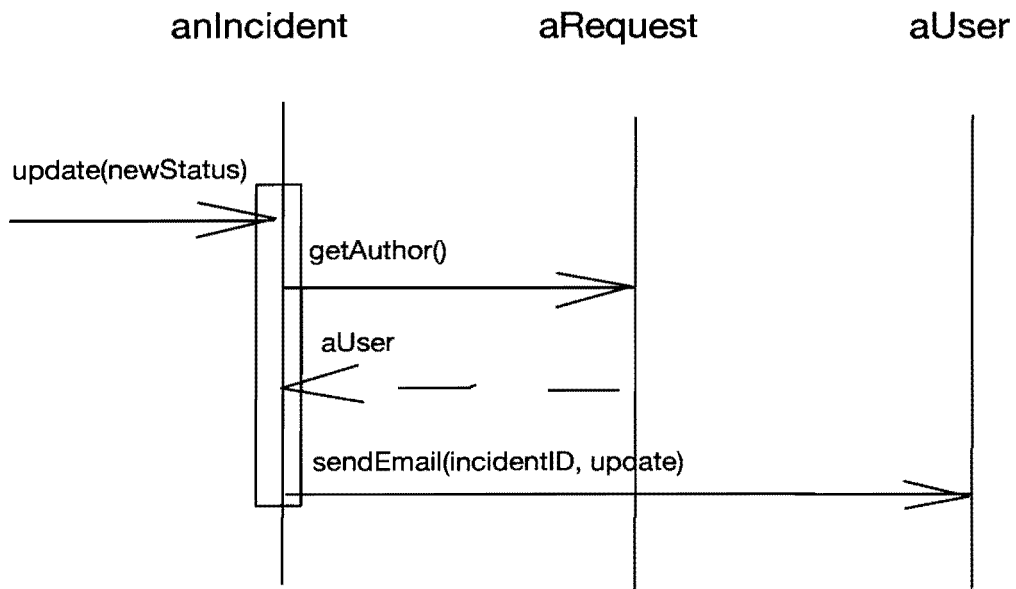


- 1.
  - a)
    - ii)



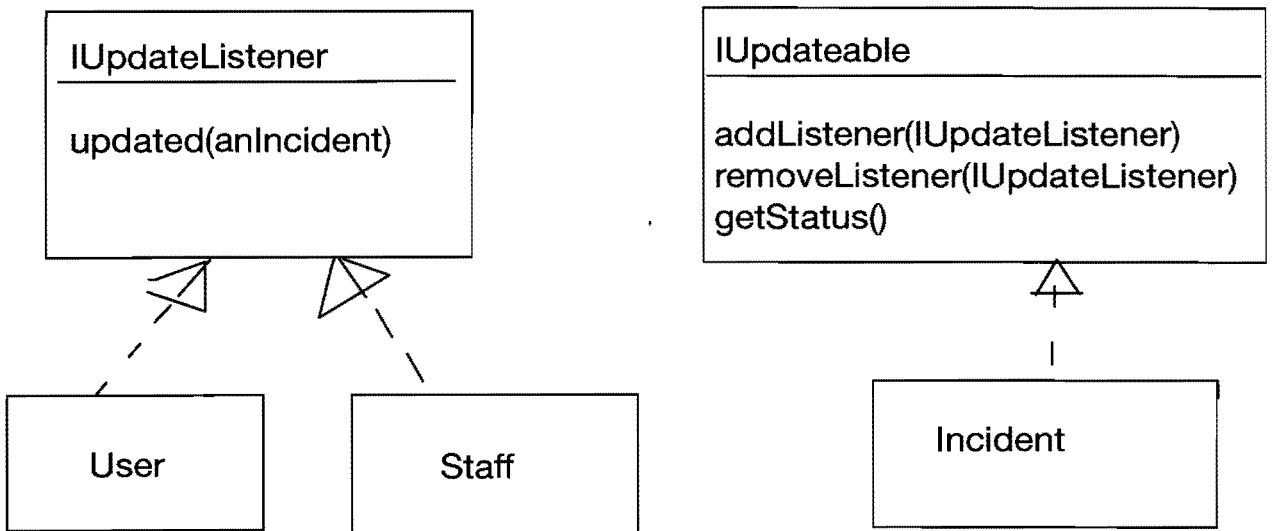
1.

b)



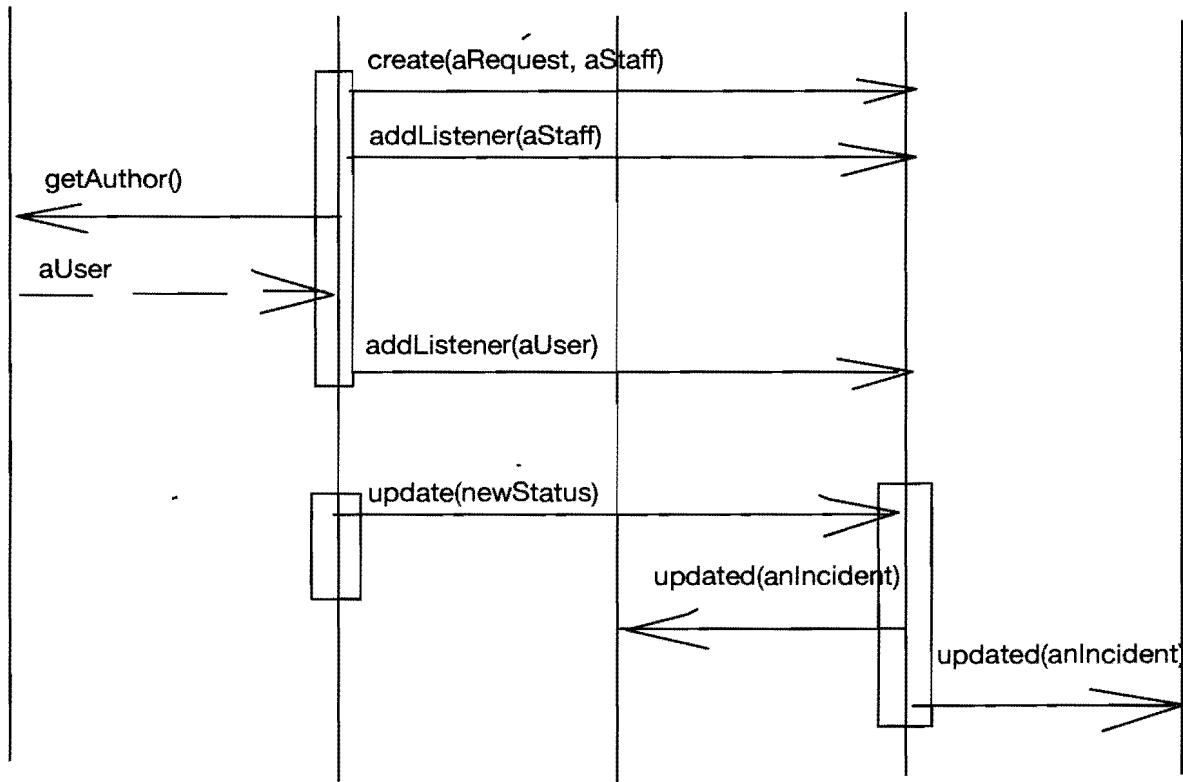
1.

c)



1.  
c) continue

aRequest    aMaintenanceSystem    aStaff    anIncident    aUser



2.

a) BOOKWORK

Applied to software applications, there are the following stages of user experience:

Discovery – a person is realising a need and finding the application to address it.

Installation – obtaining a copy of the application and setting it up.

First experience – “can I do what I downloaded it for or do I need to read instructions?”, being guided through the first use, building a conceptual model of the application.

Repeat experience – “I know what it can do, can I do what do most often in the quickest possible way?”

Error handling – what went wrong, how to recover.

Help reference – could facilitate the first use, help to discover advanced functionality or just serve as a reference for less common operations.

Maintenance – updates: notifications about availability, applying updates.

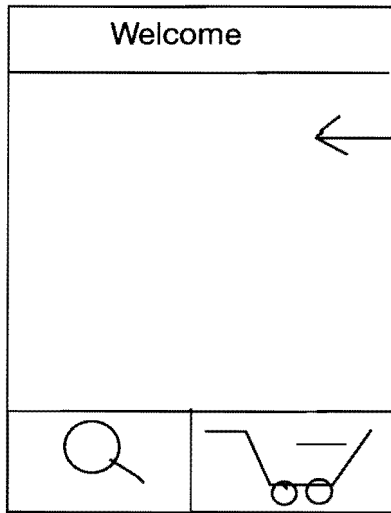
Support – if needed more help, user groups, support email/phone.

Feedback – providing a back channel, identifying issues and features requests.

2..a) Bookwork

b) One of many possible solutions is presented below

i)

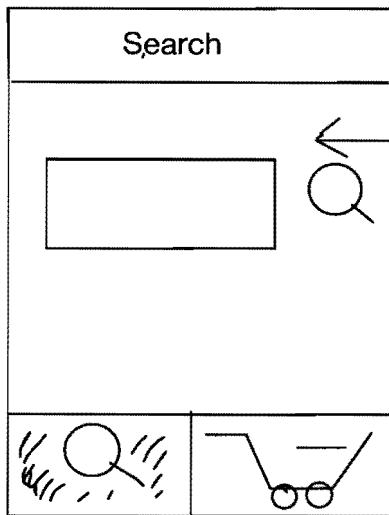


Search

Basket

← Welcome Message

Instruction to press Search to search for artists, tracks, albums



**Search selected**

Search screen

← Text field to enter the artist, track, album name to search

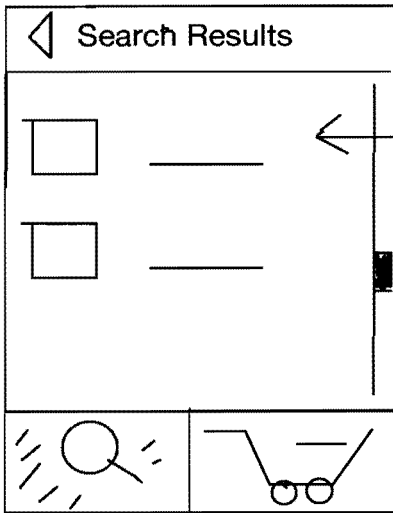
When selected the keyboard appears and the user can type

When the user presses enter the results are shown.

For all network connectivity dependent actions a progress indicator is shown and an error screen may appear if the connection failed.

2. b)  
i) continue 1

Back  
Button



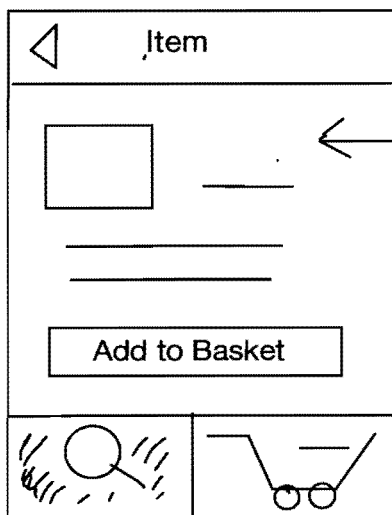
Search results appear as a list of items with pictures to illustrate

User can click on the picture or item to have more details displayed

Scroll view is enabled if required

**Search selected**

Back  
Button



Description of the item selected

Click on Add to Basket button to add to basket

**Search selected**

2.  
 b)  
 i) continue 2

Back Button

**Basket selected**

← List of items in the basket, including the name of the item, price per item and an option to delete

Back Button

**Basket selected**

← Instructions to enter the user name and passwords to checkout, followed by the fields for the username and password, followed by the Sign in button.

← Once checkout the item will be available to download to all user device.

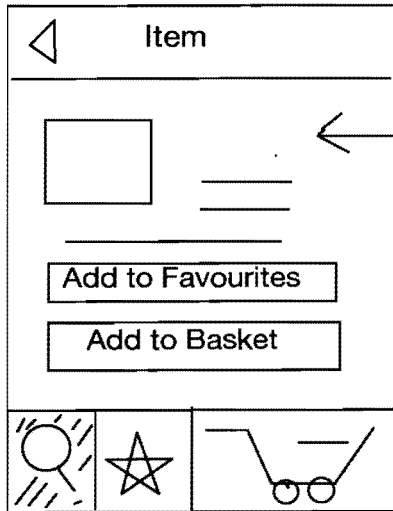
Optionally, once the item is purchased the button Add to Basket is no longer displayed.

For all network connectivity dependent actions a progress indicator is shown and an error screen may appear if the connection failed.



2.  
b)  
ii)

Back Button



Description of the item selected

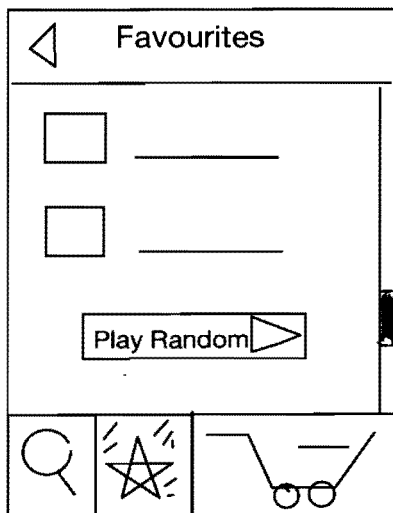
Introduce the button Add to Favourites on the Item page

Introduce the tab Favorites to enable Favourites functionality

Optionally, once the item is purchased the button Add to Basket is no longer displayed.

**Search selected**

Back Button



List of all items selected as Favourites with a picture, followed by the name of the item.

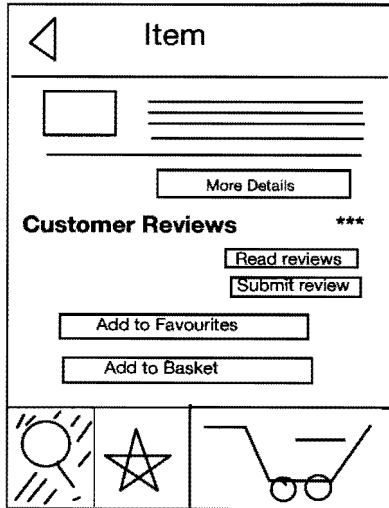
Scroll view is enabled if required.

Play random allows the user to listen to a random selection from the favourites. During playback the Item View is displayed with added buttons for skipping and pause.

**Favourites selected**

- 2.
- b)
- iii)

Back Button



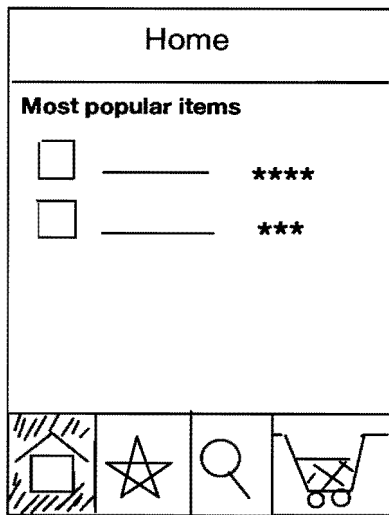
**Search selected**

Add Customer Reviews section to the item Description page.

Include summary of the current rating (stars) if already available, and buttons Read reviews if any available and Submit review.

If required, i.e. space is limited, include brief description of the item only followed by More details button.

Optionally, once the item is purchased the button Add to Basket is no longer displayed and instead button Play is present.

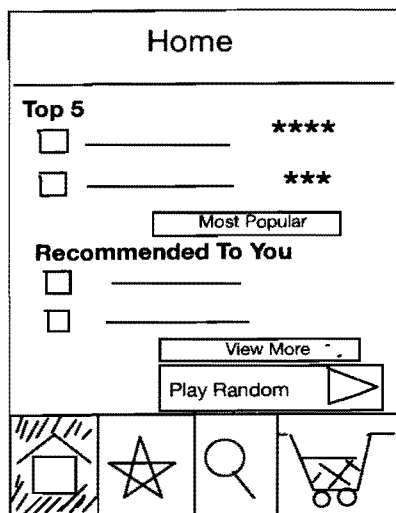


**Home tab added and is selected**

Add Home screen tab

Include list of most popular items on the Home screen, each item is followed by the rating and optionally the total number of reviews

2.  
b)  
iv)



**Home tab is selected**

Home screen

Add Recommended To You section with a selection of recommendations and View More recommendations button.

Limit Most Popular section to Top 5 and include Most Popular button to expand.

Add Button Play Random in the Recommended To You section which allows the user to listen to a random selection of recommended tracks. During playback the Item View is displayed with added buttons for skipping and pause.

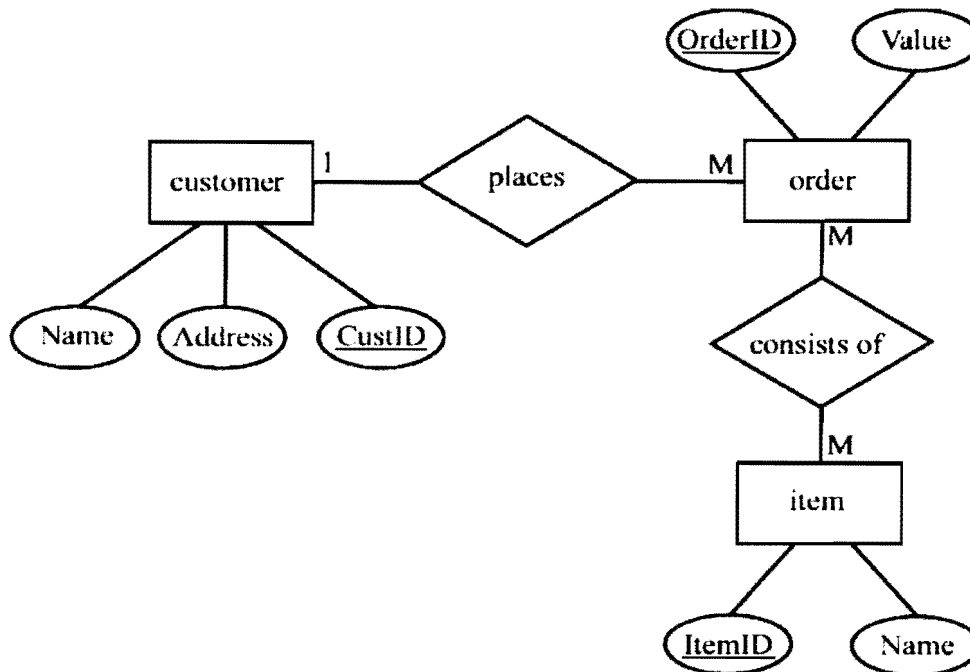
3 (a) BOOK WORK: Database normalisation is the process by which the *attributes* and *relations* of a database are organised to avoid logical inconsistencies arising when the database is used. Normalisation aims to minimise redundancy and dependency in the database thereby improving *consistency* (since inconsistencies are prevented by the structure of the database), *extensibility* (since changes to the database structure will only affect parts upon which they are logically dependent), and *efficiency* (since redundant data is not stored). It is important to note that selective denormalisation *may* help for performance reasons.

- (b) (i) The database design is poor for the following reasons:
- customer names are repeated many times (degeneracy/redundancy)
  - having an attribute for each item is a bad design choice since it is wasteful (often 0 are ordered), it makes it hard to design queries, and means that queries must be altered when new items are added to the database

The following additional attributes are required for processing orders:

- customer addresses - for delivery of the order
- a customer identification number --- as a unique reference number (and primary key)
- value - the cost of the order

The following ER diagram shows a redesign that addresses the problems mentioned above and includes the missing attributes:



(ii) There are five tables in the updated database:

CUSTOMER		
<u>CustID</u>	Name	Address

Attribute - primary key

Attribute\* - foreign key

ITEM	
<u>ItemID</u>	Name

ORDER	
<u>OrderID</u>	Value

ORDER-CUSTOMER	
<u>OrderID*</u>	<u>CustID*</u>

(from "places" relationship)

ORDER-ITEM		
<u>OrderID*</u>	<u>ItemID*</u>	Quantity

(from "consists of" relationship)

Orders can be fulfilled in three steps

pack: `SELECT ORDER-ITEM.OrderID, ITEM.Name, ORDER-ITEM.Quantity  
FROM ORDER-ITEM, ITEM WHERE ORDER-ITEM.ItemID = ITEM.ItemID  
AND OrderID="123"`

dispatch: `SELECT ORDER-CUSTOMER.OrderID, CUSTOMER.Name, CUSTOMER.Address  
FROM ORDER-CUSTOMER, CUSTOMER WHERE  
ORDER-CUSTOMER.CustomerID = CUSTOMER.CustomerID AND  
OrderID="123"`

charge: `SELECT ORDER-CUSTOMER.OrderID, CUSTOMER.Name, ORDER.Value  
FROM ORDER-CUSTOMER JOIN ORDER ON ORDER-CUSTOMER.OrderID = ORDER.OrderID  
JOIN CUSTOMER ON ORDER-CUSTOMER.CustomerID = CUSTOMER.CustomerID  
WHERE ORDER.OrderID = "123"`

(iii) The following extension can be used to store the tester's name and the items which were tested:

EMPLOYEE		
<u>EmployeeID</u>	Name	...

Attribute - primary key

Attribute\* - foreign key

ORDER-TESTER	
<u>OrderID*</u>	<u>EmployeeID*</u>

(iv) The following query will enable the company to see who (if anyone) inspected order 123:

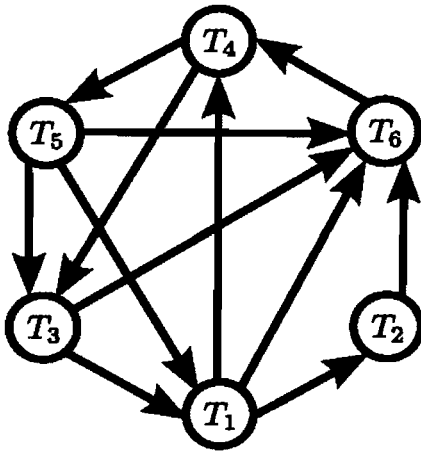
`SELECT ORDER-TESTER.OrderID, EMPLOYEE.Name FROM ORDER-TESTER, EMPLOYEE  
WHERE ORDER-TESTER.EmployeeID = EMPLOYEE.EmployeeID AND ORDER-TESTER.OrderID = "123"`

a)

2 (i) Multiple transactions can have read locks on the same resource, but a write lock is exclusive. Since there are multiple locks on resources  $A$ ,  $B$ ,  $C$  and  $D$  these must be read locks. Since two locks remain, they must both be write locks:  $T_6$  has a write lock on  $E$ , and  $T_4$  has a write lock on  $F$ .

4 (ii) BOOK WORK: A wait-for graph represents the waiting relationships between current transactions. The nodes represent transactions. A directed edge from node  $T_1$  to node  $T_2$  implies transaction  $T_1$  is waiting for transaction  $T_2$  to release a lock.

The wait-for graph for the scenario given in the question is:



$T_1 \rightarrow T_2, T_3, T_4, T_5, T_6$   
 $T_2 \rightarrow T_4$   
 $T_3 \rightarrow T_1, T_6$   
 $T_4 \rightarrow T_3, T_5$   
 $T_5 \rightarrow T_1, T_3, T_6$   
 $T_6 \rightarrow T_4$

6 (iii) The wait-for graph contains directed cycles and is therefore in a deadlock. All of the directed cycles pass through  $T_4$  and so aborting this single transaction resolves deadlock. The transactions then complete in this order:  $T_6, T_2, T_1, T_3, T_5$ .

4.  
b) 8

The key activities include

---

Setup the source code repository with branches for

- main development branch for new features
- maintenance releases - each bug fix release should be in a branch corresponding to the - released version of the software. Each bug fix should also be applied to the current main dev branch
- tax year related dev branch, can exist to as the branch from the main dev branch to implement and test changes for updated tax rates, merged with the main dev branch once the updated rates are coming in effect

Setup an automated build system (build on each check-in), build targets should include:

- development builds for active branches
- release builds for publishing

Setup automated testing for each of the build targets

Setup reporting system informing developers/managers if the build is broken (doesn't compile or fails testing)