




Grade item	weight	Grade	Range	Percentage	Feedback	Settings 
<input type="checkbox"/> Q1a Sequence Diagram	-	15.00	0–20	75.00 %	<div><i>Feedback comments</i></div> <p>This is a good realisation of the use case specification! The essential parts of the use case specifications - the main and the alternative ones - are captured well. You have used "nested" alt fragments competently with clearly defined guards. The loop fragments are used competently, too The diagram, however, is deficient is a number of aspects. For instance, it is unclear how the ShoppingList with a matching id is found. The search for the shopping list with a matching id should be done by looping through all lifelines :ShoppingList which reside in the local memory and comparing their id with the one provided by the user at the start of the use case. This loop, which is omitted, will either complete successfully - finding the right :ShoppingList or not and this will be the outcome of the loop, which will define the which of the operands of the respective alt fragment will be true and which - false. A similar deficiency is demonstrated with the search for a matching barcode. You start well by looping over the :FridgeItems, but if this search fails instead of creating a new new :FridgeItem (omitted!) you instantiate a new :Item lifeline and update its quantity with the quantity delivered in the package. There seems to be a confusion between :Item and :FridgeItem. :FridgeItem holds data on the food items held in the fridge, :Item is merely part of the ShoppingList, but may refer to food items which are not in the fridge. Therefore, :FridgeItem must be updated (or even instantiated, if no :FridgeItem with a matching barcode is found), not :Item (as is in your diagram). Destruction of :ShoppingList and all :Item objects is handled very well. The logic of the realization – 7/10 marks. Lack of details (on searches) and confusion between :FridgeItem and :Item (see the summary above). Use of combined fragments – 5/5 marks. Used extensively and very well . Instantiation of lifelines – 2/3 marks. A few instances are shown, but an essential instantiation of :FridgeItem is omitted. Destructions of :ShoppingList and :Item are OK. Communications with boundary classes – 1/2 mark. Robustness analysis rules are broken - message 7.1.5.3 is "impossible" as :Controller and :Item are not associated.</p>	-



Grade item	weight	Grade	Range	Percentage	Feedback	Settings
<input type="checkbox"/> Q1b Consistency between sequence and class diagram	-	4.00	0–5	80.00 %	<i>Feedback comments</i> Consistency with class diagram. 1/1 mark for using only lifelines defined in the class diagram. Very competent use of lifelines. Session is used and added to the class diagram. 1/1 mark for using only operations defined in the class diagrams. This seems OK. 0.5/1 mark for sending messages only between lifelines which refer to classes with associations. Robustness Analysis rules are violated (message 7.1.5.3) 1/1 mark for modelling the stated alternative flows. This is done well. 0.5/1 mark for not-deviating from the specification. This diagram deviates significantly from the provided specifications.	-
<input type="checkbox"/> Q2a Design Class Specifications	-	20.00	0–20	100.00 %	<i>Feedback comments</i> This is an excellent diagram, quite elaborate and well thought out. You refined the analysis class diagram extensively and seem to have thought out about the implementation of the Ordering subsystem. You have also considered introducing extra control classes to increase further cohesiveness. It is good to see that you thought about the access of object's private data by providing getters and setters. Well done! Specification of attributes – 3/3. All attributes are fully specified. Specification of methods - 15/ 15. Very well done. Constructors – 2/2 marks. Well thoight out. Even an example of a second constructor for the same class.	-
<input type="checkbox"/> Q2b Class Associations	-	7.00	0–10	70.00 %	<i>Feedback comments</i> Aggregation and composition – 4/7 marks. Most associations refined to compositions, which led to ShopingList being owned exclusively by two classes, same happend with Supermarket. This is a serious error. Also, you could have used a collection class to resolve the 1 - 1..* multiplicities between ShopingList and Item and use a combination of composition and aggregation to represent their relationship! By the way the association between them seem to be the wrong way round; ShopingList should be the 'Whole' and Item the 'part', a shoping list consists of one or many items. A couple of associations were not refined. Navigability - 3/3 mark. Navigability was specified for all associations.	-
<input type="checkbox"/> Q2c Interfaces	-	10.00	0–10	100.00 %	<i>Feedback comments</i> You use interfaces competently Implementation classes - 4/4 marks. Interfaces – 2/ 2 marks. Competent use of "provided interface" (i.e. a Facade) for the Ordering subsystem and of several other interfaces. Well done! Realisation relationship between interfaces and classes – 2/2 marks. This is OK. Dependence on interfaces – 2/2 marks. This is also OK.	-

Grade item	weight	Grade	Range	Percentage	Feedback	Settings
<input type="checkbox"/> Q2d Cohesion, Coupling, Patterns	-	9.00	0–10	90.00 %	<i>Feedback comments</i> Extensive use of interfaces reduces coupling between sub-systems. You have introduced extra control classes as collaborators to the Facade implementation class to achive better cohesion and reduce responsibilities of the entity classes. The only issue that would need to be revised further on the overall design is the associations. Try and use only one composition + aggregation(s), thus avoiding the common mistake of multiple compositions.	-
<input type="checkbox"/> Q3a State machines: States	-	9.00	0–10	90.00 %	<i>Feedback comments</i> The states are very well captured! But I ma not convinced the 'Created' state is necessary.	-
<input type="checkbox"/> Q3b State machines: Transitions	-	8.00	0–10	80.00 %	<i>Feedback comments</i> Transitions are clear. You use the methods of the class very competently, which is very good. The time event after(30 days)is captured, but the model is muddled. This event should be used properly, not used as a guard. Compare your solution with the model answer.	-
<input type="checkbox"/> Q3c State machines: Actions/Activities	-	3.50	0–5	70.00 %	<i>Feedback comments</i> Most of the entry actions are good. The do activity in the Archived state is spot on, the other ones need to be revised and most of them droppped.	-
Coursework Submission Area (2018-19)	-	-	0–100	-		-
CW submission area (2018-2019)- Extenuating Circumstances	-	-	0–100	-		-
Resit CW submission area (2018-2019) - Due in August 2019	-	-	0–100	-		-
20 Simple weighted mean of grades.	-	-	0–100	-		-

City, University of London is an independent member institution of the University of London. Established by Royal Charter in 1836, the University of London consists of 18 independent member institutions with outstanding global reputations and several prestigious central academic bodies and activities.