

# Inf2C Software Engineering 2018-19

## Coursework 1

### Capturing requirements for an auction house system

#### 1 Introduction

The aim of this coursework is to produce a requirements document for a simplified computer system for managing auctions at an auction house. Later courseworks will involve the design and implementation of this software.

#### 2 System Description

##### 2.1 Background

For many years, the Auld Reekie Auction House has held auctions at its premises in Edinburgh. Its auctions have been organised in a traditional way, with all bidders gathering in a room with an auctioneer, and buyers bidding on *lots* (collections of objects being sold together) by raising paddles with bidder identification numbers. It wishes to continue with auctions of lots being run on its premises by an auctioneer and with buyers gathering at the auction house. But it wishes to switch to bids being submitted from electronic devices (e.g. smartphones or personal computers) and to allow at least some buyers to be off-site. A computer-based system is proposed to support this. The following subsections describe how the system is expected to operate.

##### 2.2 Buying

Before the auction of a lot, buyers can browse a description of the lot in an online catalogue of all lots for sale and, if they are on the auction house premises, physically view and inspect the lot. If a buyer is interested in perhaps bidding on a lot, they must first note their interest in the lot with the system. They then will receive notifications when the auctioneer opens bidding on the lot and when any other buyer makes a bid.

A buyer makes a bid by tapping or clicking a button on the screen of their device. They can make one of two kinds of bid:

1. An *incremental bid* where they bid a *bid increment* above the current bid price. The bid increment is a fixed amount set by the auction house for all auctions.
2. A *jump bid* where they also explicitly specify a bid amount at least that of an incremental bid.

At some point, when it seems that no more bids are to be made, the auctioneer declares the auction closed with the tap of a hammer on a block and the lot is sold to the highest bidder. The auction system informs all buyers who have noted interest in the auction of the auction's closure, and confirms with them the final agreed purchase price, the *hammer price*. The system then automatically collects payment from the buyer of the hammer price plus a *buyer's premium*, some fixed percentage (commonly in the range 15% - 25%) of the hammer price.

Buyers must register with the system before they can note interest in lots and make bids. Registration includes providing personal and bank account details and giving the auction house authorisation to collect payments for lots purchased.

## 2.3 Selling

When a seller wishes to sell a collection of objects as a lot, it deposits the objects on the auction house premises. Auction house staff work with the seller on producing a description of the lot using both text and photographs. Staff enter this description into the system. This description is used in the online catalogue for the auction of the lot. Staff also include high and low estimates of the hammer price in the description. The low estimate is used by the auctioneer to help decide the price at which to open bidding.

The seller of a lot sets a *reserve price* that is kept secret. If the hammer price is not at least this reserve price, the sale does not go through.

After a lot is sold and payment has been collected from the buyer, the system pays the seller the hammer price less a *seller's commission* (maybe 10 - 15% of the hammer price).

Sellers are required to register with the system before they can offer lots for sale. When registering the seller provides some personal information and details of a bank account into which they would like payments to be made.

## 2.4 Further details

- The auction house employs multiple auctioneers and auctions of lots can be carried out in parallel.
- The online catalogue can be viewed by any member of the public. Viewing is not limited to buyers or sellers.

## 2.5 Extensions

The above has just sketched out the basics. In real life many refinements would be possible and desirable. A few are sketched below. Unless otherwise specified, you should not cover

these in the requirements document you produce for this coursework. Some of these or others may be included in later courseworks.

- Auctions of lots are grouped into *auction sessions* whose dates and times are announced in advance. Often each session is focussed on a particular category of objects, e.g. furniture or rare books. The session that a lot is entered in might not be determined at the point the lot is handed over to the auction house.
- Auctioneers typically issue a *final warning* to bidders just before they close an auction in order to prompt any final bids.
- The auction house might stream live audio or video feeds of auctions in order that off-site buyers can be more fully immersed in the auction process.

## 3 Your job

Your job for this coursework is to create a requirements document for the software that organises and expands on the information presented above.

For simplicity consider the system being designed as encompassing both some main computer system on the premises or in the cloud and also the devices of the various actors. The requirements should not be concerned with how these system components interact.

Include in your requirements document the information asked for in the following tasks. The percentages after each subsection title show the weights of the subsections in the coursework marking scheme.

### 3.1 Identify stakeholders (15%)

Identify the stakeholders of the system and the value or benefit each seeks from it. Focus on stakeholders particular to this auction system. There is no need here to cover stakeholders common to most software development projects – software architects, designers, coders and testers, for example.

### 3.2 Describe system state (10%)

Describe in broad terms the nature of the state of the system. What information does the system need to keep on buyers, sellers and lots? What different statuses of a lot should the system track? For example, it would be a good idea if the system records whether or not a lot has been auctioned.

Organise the state description using an enumerated list, and, as needed, introduce multiple levels of lists. Later on, this categorisation of the system state can be referred to when describing the preconditions and guarantees of use cases.

This description forms part of the description of the functional features of the system.

### 3.3 Describe use-cases (40%)

The provided description can suggest different numbers of use cases, depending on the size and level of abstraction of each use case.

For this coursework identify 7–10 use cases. The use cases you identify should include the following:

1. **Register Buyer**
2. **Note Interest in Lot**
3. **Bid on Lot**
4. **Close Lot Auction.** This includes the notification of relevant parties and execution of appropriate financial transactions.

Keep the use cases high level: they are about the main interactions between actors external to the system and the system itself, they should not be concerned with all the details of user interface interactions: you are not doing design at this stage.

For the *Close Lot Auction* use case, produce a description using a template similar to that described in the *Tutorial 1* question sheet used for the Week 4 tutorials. Feel free to add extra fields if you feel it would help, and also omit fields when they are unnecessary. In the main success scenario, focus on when the sale goes through, but also include an extension scenario for when the hammer price is lower than the reserve price.

Describe one other use case with a reasonably full template. For the rest of the use-cases, use simpler format with just the primary actor, supplementary actor(s) if relevant, and a 2–5 line free-text description.

### 3.4 Use case diagram (10%)

Draw a UML use-case diagram summarising the use cases and the actor or actors each is associated with.

### 3.5 Describe non-functional requirements (20%)

Describe non-functional requirements using two or more levels of enumerated lists. General categories of non-functional requirements include security, usability, performance and reliability. Aim to include at least two non-functional requirements in each of these categories. In at least a few of these requirements, add enough concrete details that someone reading the requirements would have some idea of how to measure them and assess whether they are being met.

### 3.6 Ambiguities, subtleties, incompletenesses (5%)

The provided system description above omits many details. Perhaps it is even misleading on particular points. Some of these issues might be resolved at a later design stage. But some might best be discussed with stakeholders at this requirements gathering stage.

Make a note of a few such issues you have noticed, when appropriate making suggestions of obvious ways you think these might be fixed. You may find it most convenient to include these suggestions in other tasks, but do summarise these suggestions here.

## 4 Asking Questions

Please ask questions on the class discussion forum if you are unclear about any aspect of the system description or about what exactly you need to do. For this coursework tag your questions using the *cw1* folder. As questions and answers build up on the forum, remember to check over the existing questions first: maybe your question has already been answered!

## 5 Good Scholarly Practice

Please remember the University requirement as regards all assessed work. Details about this can be found at:

<http://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct>

Furthermore, you are required to take reasonable measures to protect your assessed work from unauthorised access. For example, if you put any such work on a public repository then you must set access permissions appropriately (generally permitting access only to yourself, or your group in the case of group practicals).

## 6 Submission

Please submit two files

1. A PDF (not a Word or Open Office document) of your requirements document. The document should be named **report.pdf** and should include **a title page with names and UUNs of the team members**.
2. a text file named **team.txt** with only the UUNs of the team members (one UUN on each line) as shown,

```
s1234567  
s7891234
```

### How to Submit

**Only one member of each coursework group should make a submission.** If both members accidentally submit, please alert the course organiser so confusion during marking is avoided.

Ensure you are logged onto a DICE computer and are at a shell prompt in a terminal window. Place your *report.pdf* requirements document and your *team.txt* file in the same directory, ensure this directory is set as your current directory (i.e. `cd` to it), and submit your work using the command:

```
submit inf2c-se cw1 report.pdf team.txt
```

This coursework is due at

**16:00 on Friday 19th October**

The coursework is worth 20% of the total coursework mark and 8% of the overall course mark.

Paul Jackson, 10th October 2018.