Inf2C Software Engineering 2019-20

Coursework 1

Capturing requirements for a federalised bike rental system

1 Introduction

The aim of this coursework is to produce a requirements document for a simplified computer system for managing a bike rental system. Later courseworks will involve the design and implementation of this software.

2 System Description

2.1 Background

Cycling is a healthy and environmentally friendly way to travel around different sights in Scotland which is popular with both locals and tourists. Currently there are many different local bike rental shops in different locations across Scotland which provide bikes for hire. The Scottish tourism board has proposed an online federalised bike rental system, which allows potential customers to find and book bikes from different providers throughout Scotland. In contrast to centralised bike hire schemes, this will utilise existing providers (small local bike hire shops) to offer customers wide variety of types of bikes (e.g. road bikes, mountain bikes, hybrid bikes, ebikes, etc) at multiple locations across Scotland for multi-day rental. Customers requiring a given combination of bikes of different types for a given range of dates should be able to perform a search to get quotes from all providers that can satisfy their demands and place a booking after selecting the quote which best meets their needs (based on price, deposit amount, location etc). This scheme is expected to boost tourism, allowing visitors to easily find the provider who can best meet their needs, and to help the environment, allowing customers to cycle instead of drive.

2.2 Renting and Returning

Customers can use the online rental system to request a list of quotes from the different bike providers. To this end, they need to provide information on their rental needs (number and types of bikes, date range, location of hire), and the system will return a list of quotes from different bike providers. Each quote shows a bike provider who can offer the bikes requested on the desired dates, as well as the total price and deposit for renting the bikes. If no quote is available for the customer's date range, the system suggests any quotes for the same duration within 3 days before the start or after the end of the date range.

If the customer decides to book the bikes offered in one of the quotes, they must provide their personal information (first name, surname, address, post code, phone number), and their mode of collecting the bikes (in store collection or delivery). Bikes can be picked up in person from the bike provider's shop, or they can be delivered to the customer's address if this is near to the provider. Customers can return the bikes to the shop of the provider from which they were rented, or to the shop of one of the provider's partners (see Section 2.3).

Once bikes are booked, the customer will pay for them online. After payment, a confirmation is generated which includes the order number (unique for each order), order summary, deposit, total price, delivery and return information. Once booked, the bikes are also reserved for the customer on the required dates, and are unavailable to other customers on the same dates.

We can assume that the customers will always pick up and return all of the bikes in their order, using the mode of collection they chose when booking, and also that they will always return the bikes on time (before the rental expiry date). Bookings span entire calendar dates so it is not possible to place a booking for part of a day; collection and return may take place at any time within a bike provider's opening hours but the details of their timing do not need to be included in your model. When the customer picks up the bikes, they need to pay for their full deposit. If they wish their bikes to be delivered then a delivery driver will collect the bikes from the shop and drop them off at the customer's address at the start of their booking period whilst collecting their deposit. When the customer returns the bikes (to the provider or one of its partners) their deposit will always be returned (see Section 2.3).

2.3 Bike Providers

The online system contains a database of bike providers. A provider can register onto the system, by giving their name, shop address, shop postcode, phone number and opening hours. Once registered, the provider is able to register new bike types onto the system by providing each type's name (e.g. road bike, mountain bike, hybrid bike, ebike, etc), and the full replacement value for bikes of this type (for determining deposit amounts). Furthermore, they can add bikes of a certain type as part of their stock. The list of bike types is shared amongst different providers, however, each provider must set its own daily rental price for each type of bike included in its stock.

As well as setting their own daily price for each bike type, providers may also set their own deposit policies. The deposit amount is always a fixed percentage of the replacement value of the bike, the *deposit rate*, however, different providers may have different deposit rates.

When a customer picks a bike order, the provider or delivery driver registers the received deposit and updates the status of the bikes. When the customer returns the bikes directly to the provider from which they were rented, the provider returns the deposit and updates the status of the bikes to record their return.

Providers can also make partnership agreements with other providers, whereby each provider will accept the return of bikes rented from any of their partners. This means that a customer on a cycling holiday can, for example, hire bikes from a provider in Aberdeen and return them to a partner of that provider in Inverness. These partnerships can be registered in the system by indicating both of the providers entering into a partnership. Then, if a customer returns their bikes to one of these partner providers instead of their original provider, the partner returns the deposit and updates the system to notify the original provider that they have been returned. We assume that a provider's partners will use a delivery driver to return the bikes to the original provider overnight so they are available for rental on the next day, and that the status of the bikes will be tracked throughout this process.

2.4 Extensions

The above has just sketched out the basics. In real life many refinements to the system would be possible based on additional requirements of the client which may only become apparent after further discussion during the design phase, during the development of the system, or even, after it has been deployed.

The following are two potential extensions of the system:

- Providers may have more complex pricing schemes for bikes. For example, a provider may offer discounts taking into account the time of year, the length of the booking, or the number of bikes being booked. Moreover, different providers may have different pricing schemes as best fits their businesses.

- Providers may wish to take other factors into account when setting the deposit rate for a given bike, such as the age of the bike, or its condition. Moreover, for accounting reasons, different providers may use different rules for calculating deposit rates.

You do not need to include these in your requirements documents although you may be asked to develop these in future courseworks.

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, you should model the system as a unified whole including both processes which take place on the bike hire system's main website, and on associated apps that actors may use. The requirements should not be concerned with how these system components interact. You are also not required to model the details of any financial transactions between actors, however, the system may need to keep track of when payments have been made.

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, for each stakeholder, describe how the system affects them. Focus on stakeholders particular to this bike rental 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 customers, bikes, bookings, and providers? Is there any other information the system needs to record? What different statuses of bikes and bookings should the system track? For example, it would be a good idea if the system records whether or not a bike is currently in the shop.

Organise the state description using an enumerated list, and, as needed, introduce multiple levels of lists.

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. **Get quotes**
2. **Book quote**
3. **Record bike return to original provider**

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 *Book quote* 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 in the template if you feel it would help, and also omit fields when they are unnecessary.

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 of the interactions. In your descriptions, consider both the main success scenario, but also possible extensions.

3.4 Use case diagram (15%)

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

You may either draw this by hand and include a high-quality scan of your diagram in your report or use a software tool such as draw.io [1].

3.5 Describe non-functional requirements (10%)

Describe non-functional requirements using two or more levels of enumerated lists. There are many different general categories of non-functional requirements including Security, Performance, Quality, Privacy, Usability, Platform Compatibility, Availability, Accessibility, Interoperability, and Data Retention. Aim to identify at least 4 of these categories which could be relevant to this system, and for each, give some examples of non-functional requirements in this category which could apply to the system. You should aim to list at least 7 non-functional requirements in total. 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.

Identify some of the ambiguities you encountered and discuss some possible options of how these ambiguities might be resolved.

You may find it most convenient to discuss some of these ambiguities during your answers to other tasks, but do summarise these issues here.

[1] The use of `draw.io` to draw UML use case diagrams is explained in: `https://about.draw.io/uml-use-case-diagrams-with-draw-io/`.

3.7 Self-assessment (5%)

For this section you should consider to what extent your have met the assessment criteria for this coursework. The main criteria for each question are listed in list below. For each criteria you should indicate the mark which you believe your answer merits and justify your judgement.

You should format your answer as a two-level nested list (mirroring that below), providing your predicted mark, and a brief justification for it.

Q 3.1 Identify stakeholders **15%**

- Identify core stakeholders of the system 5%
- Identify additional stakeholders 5%
- Describe how the system affects each stakeholder 5%

Q 3.2 Describe system state **10%**

- Include state essential to the operation of the system 5%
- Include additional state mentioned in the description 5%

Q 3.3 Describe use cases **40%**

- Identify use cases 10%
- Describe use cases using the appropriate templates 30%

Q 3.4 Use case diagram **15%**

- Correctly use UML use case notation 5%
- Include key actors and use cases 5%
- Identify connections between actors and use cases 5%

Q 3.5 Describe non-functional requirements **10%**

- Identify non-functional requirements within the context of the system 7%
- Provide means for assessing non-functional requirements 3%

Q 3.6 Ambiguities and subtleties **5%**

- Identify some ambiguities in system description 3%
- Discuss potential options for resolution of ambiguities 2%

Q 3.7 Self-assessment **5%**

- Attempt a reflective self-assessment linked to the assessment criteria 5%

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 an online repository then you must set access permissions appropriately (permitting access only to yourself or your group).

6 Submission

Please submit 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**.

How to Submit

Ensure you are logged into MyEd. Access the Learn page for the Inf2C-SE course and go to "Coursework and labs" - "Coursework submission" - "Coursework 1".

Submission is a two-step process: upload the file, and then submit. This will submit the assignment and receipt will appear at the top of the screen meaning the submission has been successful. The unique id number which acts as proof of the submission will also be emailed to you. **Please check your email to ensure you have received confirmation of your submission**.

If you do have a problem submitting your assignment try these troubleshooting steps:

- If it will not upload, try logging out of Learn / MyEd completely and closing your browser. If possible try using a different browser.
- If you do not receive the expected confirmation of submission, try submitting again.

- If you cannot resubmit, contact the course organiser at Cristina.Alexandru@ed.ac.uk attaching your assignment, and if possible a screenshot of any error message which you may have.
- If you have a technical problem, contact the IS helpline (is.helpline@ed.ac.uk). Note the course name, type of computer, browser and connection you are using, and where possible take a screenshot of any error message you have.
- Always allow yourself time to contact helpline / your tutors if you have a problem submitting your assignment.

7 Deadlines

The three deadlines for the coursework are as follows.

- **Deadline 1 (formative): 16:00, Monday 14th Oct**

  A bonus of 2.5% of the final mark will be awarded to students who submit a **full attempt** at this coursework at the above deadline.
- **Deadline 2 (formative): 16:00, Monday 4th Nov**
- **Deadline 3: 16:00, Friday 29th Nov (compulsory, summative, marks which count!)**

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

Thomas Wright, Weili Fu, Cristina Alexandru 3rd October 2019.