Inf2-SEPP 2020-21

Coursework 1

Capturing requirements for a COVID-19 shielding food box delivery system

1 Introduction

The aim of this coursework is to produce a requirements document for a simplified computer system for managing the delivery of food boxes to people that are shielding at home from COVID-19. Subsequent courseworks will require the software design, implementation and testing of this system. You will work as a group throughout this - and future - courseworks.

2 System Description

2.1 Background

During the COVID-19 pandemic, several groups from the population have been identified as being at greater risk. Individuals in those groups are more likely to be susceptible to adverse outcomes from contracting the disease due to various factors (e.g., age, medical conditions, etc.). The Scottish Government has put forth plans for a central, online food box order and delivery system, to help support these citizens during shielding. Shielding individuals will use the system to place food box orders with either catering companies or supermarkets, with these orders being completely free for them but only available once a week. You are part of the software development company which has been contracted to develop this system.

Shielding individuals will register and be uniquely identified using their Community Health Index (CHI) number. Once registered, they will have the option to request a food box from either a catering company or a supermarket, once a week only. In the former case (catering company), they can place their order directly on the system for one of a finite set of preassembled food boxes available. In the latter case (supermarket), they will be directed to a choice of supermarkets for browsing for the desired products and placing their order there. While supermarkets already provide a delivery service, a catering company might be required to use an external delivery service provider for dispatching the food boxes. Among other ways, individuals can access the system via SMS text messages. Lastly, other organisations might require access the system's order data, while maintaining confidentiality, to produce elaborate statistics and reports.

It is hoped that the system will enhance the response of the authorities during the pandemic and help improve the protection of vulnerable citizens.

2.2 Food box orders

First, shielding individuals register with their CHI number to the online system. During this registration process, the system communicates with Public Health Scotland's electronic record system (an external system) to check that the provided CHI number indeed belongs to a shielding individual, and if so retrieves their personal information (full name, address and phone number). Registration is refused if a person is not currently a shielding individual.

Once registered, the user has two options: (i) order a preassembled food box from a catering company based on their dietary restrictions (e.g., vegetarian, dairy-free, etc.), without needing to select a specific catering company or know its name (all such companies provide the same contents in their food boxes), (ii) request a box from a supermarket, in which case the system directs them to appropriate supermarkets for placing their order.

If choosing to order from a catering company, the system checks with the catering company stock levels and informs the user whether a food box respecting their dietary needs is available or not. If it is available, the user can optionally request to amend the box's contents. The system would then provide a listing of the box's contents that contains the name and quantity of each product, and the user is allowed to reduce the quantity of a product in it (only, not also increase it). This reduces wastefulness and allows any surplus to be available for other shielding individuals. For example, imagine that a senior citizen is accessing the system via SMS text messages; they might send "P3 0" which signifies the removal of the third product in the box. Without amending the food box, or after doing so and being satisfied with the box contents, the shielding individual can place the order for the selected food box selecting a date and time for delivery. This date and time selected by the user cannot be below the minimum estimated time for delivery for the catering company maintained by the system, which the catering company keeps updating on the system whenever it changes (e.g., if there are delays in delivery due to adverse weather).

Once the order is placed, the system issues a confirmation of the order. From then on, the individual can request on-demand updates about the status of the order (see Section 2.3) using its order number. They can also request to amend their already ordered food box (by only reducing product quantities, as above) as long as it has not been packed, and request to cancel it altogether if it has not been dispatched yet.

Moreover, they can select to be notified automatically of the status of their order.

We can assume that the system does not deal with maintaining the stock availability of food boxes and their containing products.

If choosing to make an order from supermarkets, the system uses information it holds about the individual's post code to suggest supermarkets which can deliver to their location, the normal time it takes for them to deliver, and provides links to the supermarkets' dedicated "food box" web pages. Therefore, the user can order directly from the supermarket and no longer use our system for this purpose. However, once the order has been placed, the supermarket system will notify our system about it, and later about each time the order changes status so that the delivery of the food box can be tracked. The user cannot enquire our system about this status (when enquiring about an order number which is not for a catering company food box, it will return an error message including the possibility that the order was not placed with it but with a supermarket).

2.3 Catering companies, supermarkets and order delivery

The system maintains a database of catering companies using their business names. It also maintains a list of default food boxes, one for each type of dietary restriction. Each box type contains the same types of products (but adapted in contents for the dietary restriction e.g., soy milk instead of dairy milk for vegans) in order to simplify the logistics of the whole operation during the pandemic. All catering companies delivery the same types of food boxes, and the use of a number of catering companies from different locations is only to ensure that the whole of Scotland is covered.

Once a food box order is received, or later edited by the user by amending its food box product quantities, the system notifies the catering company's system about it. Then, the catering company's system goes on to process the order, and notifies our system each time the status of the order changes. Once the order is to be delivered, things become more complicated as not all catering companies have an in-house delivery service. It will be up to the catering company to decide which delivery service to use to deliver each food box. The delivery service will become known to our system when its system starts providing updates about the delivery. If the delivery is carried out by the catering company itself, it will itself notify our system about the delivery status changes.

Anticipating demand, the system can also register supermarkets as available food providers using their business name, phone number, website URL and address. Supermarkets maintain their own websites which individuals can order the food boxes from and also handle their delivery themselves. The supermarket systems are outside our scope, however, for interoperability, our system maintains any food box order once notified by a supermarket system that it was placed, and allows the supermarket system to update an order's status as previously described.

2.4 Public Health Scotland

Since the numbers of shielding individuals is perpetually changing, Public Health Scotland (PHS) is responsible for keeping track and maintaining correct and updated patient records (e.g., their addresses). The system uses their CHI number to verify that they are indeed shielding and retrieve further information regarding their location from PHS's system.

2.5 Third party operators

As mentioned in Section 2.1, considering the critical role of the system in the community and the ever-changing conditions and restrictions, third party operators (e.g., local councils) will have access to food box order data so that they can produce reports and statistics over periods of time. An example of such an organisation is the Scottish Government which might require access to the system's order data to be able to oversee its operation and coordinate as policy and COVID-19 restrictions change.

2.6 Extensions

The above has just sketched out the basic functionality. 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.

The following are some potential extensions to the system:

- The Scottish Government requires the orders recorded by the system over a period of time, to calculate and obtain statistics on the number of food boxes delivered per post code. This will allow the generation of reports displaying the evolution of food box demand by shielding individuals in different areas, so that the service could be improved in the future (e.g., by setting up contracts with more local catering companies).
- Additionally, the Scottish Government requires the orders recorded by the system over a period of time, to calculate how many have been successfully delivered on time or not per catering company. This will allow the assessment of the level of service provided by the different catering companies, and motivate further contract extensions with them.

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

3 Your Software Engineering Tasks (worth 100% of the final mark for this coursework)

Your main task for this coursework is to create a requirements document for the software that organises and expands on the information presented in the previous sections.

For simplicity, you should model the system as a unified whole including both processes which take place on the system's main application (i.e., a web application backing the system's website), and on associated user applications (i.e., on a mobile phone). 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.

3.1 Identify stakeholders

Identify the stakeholders of the system and, for each stakeholder, describe how the system affects them. Focus on stakeholders particular to this system. Apart from the ones mentioned in the system description above, you may also consider others which would make sense given the context provided. There is no need here to cover stakeholders common to most software development projects (i.e., software architects, designers, developers and testers).

3.2 Describe system state

Describe in broad terms the nature of the state of the system. What information does the system need to keep on shielding individuals, food box orders, catering companies, supermarkets and delivery service providers? Is there any other information the system needs to record? What different statuses of boxes and orders should the system track? For example, it would be a good idea if the system records the quantities of products per box as modified by the user.

Organise the state description using an enumerated list and introduce multiple levels as needed. This description forms part of the characterization of the functional features of the system.

3.3 Describe use cases

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 7–9 use cases. The use cases you identify should include the following:

1. **Place order**
2. **Cancel order**
3. **Edit food box**

Keep the use cases high-level. They should be 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 *Place order* use case, produce a description using a template similar to that described in the *Tutorial 1* question sheet used for the Week 3 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 alternative scenarios.

3.4 Draw use case diagram

Draw a UML use case diagram summarising the use cases and the actor(s) 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

Describe non-functional requirements which are relevant for the food box system using two or more levels of enumerated lists. There are various general categories of non-functional requirements including Security, Performance, 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 Identify ambiguities, subtleties, incompleteness

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. Who would you approach and how to clarify these ambiguities?

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

3.7 The software development

For this section you should consider the assignment in terms of the type of system that is being developed, and the software process used to develop this system.

1. For the food box system that is being developed, and disregarding your tasks (i.e., the way you started work on it) in this assignment, would software project

[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/`.

engineering or software product engineering be a better choice? Please justify your answer, both in terms of the reasons for choosing it, as well as those for which the other option was less desirable.

2. Of the two types of software engineering processes that we have studied in this course, which of them have we been using so far in this assignment? Why do you think this is the case?

3. Would the other type of software development process have been better in this context? If so, why?

4 Your Professional Issues Tasks (only for formative assessment, not contributing to the mark for this coursework)

As a team, write a short essay (we will refer to them as 'blog posts') on each of the topics below.

Each blog post should be a maximum of 500 words (this amounts to about one side of A4), and we expect you to write at least 300 words.

This article gives good advice on how to structure a short essay:
`https://www.wikihow.com/Write-a-Short-Essay`

Remember that reading the first chapter of “A rulebook for arguments” should help here too.

Good essays will:

- Clearly choose a main position
- Clarify important details with reference to other sources
- Consider the terms used and define them where necessary
- Justify arguments with reference to course materials
- Anticipate and address counterarguments

How well you do these things will be the core criteria for marking in your blog posts, but this first round of blogs is not for credit; it is for formative purposes only. This allows you all to practice something Informatics students mostly haven't done for a while!

You will have two opportunities for formative feedback on the blog posts:

- Submitting any or both of your blog posts to the course blog, where they become visible to all your classmates. You will all then individually have around 4 days to provide comments on a good number (at least 3) of the other groups' blog posts. **Important!** You should prioritise picking blog posts with fewer reviews, and contribute a good amount of new comments to any existing reviews (i.e., not merely repeat what was said already). This feedback should be good quality- e.g., here is some guidance as to what this means: `https://lifehacker.com/the-six-qualities-of-good-feedback-1776302054`. Your contributions will be assessed based on a combination of quantity and quality, with a focus on quality. They may help round up your final mark for this course.

- After deadline 1 of this coursework, the markers will provide formative feedback on your blog posts as well.

Please see Section 9 on the deadlines.

4.1 Topics

Topic 1: "Discuss the extent to which the NHS data breach was a failure of professionalism"

The case study is available here: `https://www.bbc.co.uk/news/technology-44682369`

This topic is an opportunity to reflect on an event that has already happened, and so lends itself to exploring other sources, as well as reading and referencing others' thoughts on the topic.

Topic 2: "How might ACM Principles 2.1 - 2.3 apply in the development of your food box delivery system?"

This topic has a wider potential scope but don't try to cover every possibility. Focus on a few key points and explain to the reader why they are particularly important.

5 Self-assessment

For this section you should consider to what extent your have met the assessment criteria for this coursework. The main criteria are listed below, as separate for Software Engineering tasks and for Professional Issues tasks. For each criterion, write one paragraph discussing how well you think you have met that criterion and why (i.e. strengths and weaknesses of your solution). Moreover, we encourage you to reflect on things like: your experience with the approaches used, progress and work within your team, difficulties you encountered and how you addressed them, parts that you are particularly proud of and why, parts that you are not happy with and you could further improve, how you could improve them.

Important! You should make a real effort to be reflective, as well as honest, here. Please note that only making a statement about a criteria without a justification (e.g., "We identified stakeholders excellently well") is not acceptable and will result in no credit for this part.

After the first deadline, the markers will provide you with feedback on the same criteria to the one you have self-assessed on.

Moreover, we provide below a set of ratings which give you an understanding of your progress with this coursework. For the Software Engineering part, we keep these ratings informal so that we can provide you with an initial idea of how well you did after the first (formative) deadline. Please note that these ratings do not have a direct correspondence with the grade intervals of the Extended Common Marking Scheme[2] which we will use for your final mark, but they are still meant to be informative. For the Professional Issues part, as the first deadline is the only deadline, the ratings already correspond with the grade intervals of the Extended Common Marking Scheme, and we will provide you with an informative mark instead (again, for this first coursework this mark does not count).

5.1 Software Engineering Self-Assessment

The following are the criteria you need to self-assess on regarding the Software Engineering tasks by writing one paragraph for each:

1. Application of the notion of a stakeholder to identify stakeholders- both from the system description and that are relevant to the given case study

2. Identification of the system state through the interpretation of the system description and case study

3. Application of the notion of a use case to identify use cases from the system description

4. Application of other use case concepts and the use case template to describe the use cases

5. Application of the UML use case notation to represent use case diagrams for the identified use cases

6. Use of the concept of non-functional requirements to identify relevant, sometimes measurable, non-functional requirements from the case-study at hand

7. Identification of ambiguities and how they can be addressed and discussed (the latter using requirements engineering techniques)

8. Reflection on the applicability of the concepts of project and product engineering in the context of the given case study

9. Reflection on the applicability of different software development processes (in particular with regards to requirements engineering) in the context of the given case study

10. Consistency within the cw1 solution (level to which your solutions to different tasks are aligned between them)

11. Effort at this self-assessment

[2] `https://web.inf.ed.ac.uk/infweb/student-services/ito/students/common-marking-scheme`

For the formative assessment, you will receive a rating for the Software Engineering part amongst the ones provided below (this is only for your information, nothing to do here):

- **Worrying**: A solution where barely any of none of the SE concepts are correctly understood and identified from the case study. This feedback suggests that you should do a lot more work in studying the lectures, reading and practicing the tutorials, and then redo your solution to this coursework.
- **A lot to do**: A solution where only some of SE concepts are correctly understood and identified from the case study. This feedback suggests that you need to do more work in studying the lectures, reading and practicing the tutorials, and then re-consider your solution.
- **Getting there**: A solution where a good amount of SE concepts are correctly understood and identified from the case study. Additionally, the solution is mostly consistent, and the self-assessment is honest but only focuses on strengths and weaknesses.
- **Looking good!**: A solution where all SE concepts are correctly understood and identified from the case study. Moreover, the solution is consistent and the self-assessment is thorough and reflective, going beyond the discussion of strengths and weaknesses.

5.2 Professional Issues Self-Assessment

The following are the criteria you must self-assess on regarding the Professional Issues tasks by writing one paragraph for each:

1. Answers the question
2. Clarity
3. Appropriate structure
4. Supported by other sources
5. Quality of argument
6. Recognition of counterarguments / alternate views
7. Knowledge and understanding
8. Style and presentation
9. Effort at this self-assessment

For the only assessment for these tasks (formative one), you will receive a mark according to the following marking scheme and respecting the Extended Common Marking Scheme (this is only for your information, nothing to do here):

- Pass (40+): Essay attempts to address the provided question but is hard to understand and/or makes few clear points.
- Good (50+): Essay is understandable, clarifies key terms, and comes to one or more obvious conclusions. Some course materials and external sources are referenced.
- Very good (60+): Essay has a good structure, flowing between and building upon subsequent points. It identifies possible counter arguments or alternative views and integrates a variety of external sources. The self-assessment is honest, but not very reflective.
- Excellent (70+): Essay reads well and contains well-made arguments which pull together a variety of views and sources. The self-assessment shows extra effort in reflection.
- Excellent (80+): Marks in this range are uncommon. This essay draws the reader in and makes points beyond what would be expected of undergraduate students in Informaticss.

6 Some advice

6.1 Working online as a team

Teamwork is not easy, and this year it is made harder by the fact that we work remotely. However, you can turn this to your advantage if you use your experience as Informatics students, and make use of the wealth of software tools available to help you. Here are some that we would recommend:

1. Microsoft Teams (free through our university) for setting up a team with your colleague, setting up meetings in the calendar, video calls, the chat, editable file sharing, its Tasks By Planner and To Do to organise and split your work.
2. OneDrive or SharePoint under Office365 (free through our university) for storing documents, sharing and working collaboratively on them.
3. The GitHub (`https://github.com/`) online repository and version control system, which you will also use later in this course. Please be careful about access permissions (see subsection 6.3).
4. Trello (`https://trello.com/en`) as an excellent (and also free) alternative for splitting up work and recording progress on tasks. We will also use it later in our course.

5. Aww (`https://awwapp.com/`) as an online whiteboard where you can collaboratively sketch your ideas like you would do on paper.

You may want to mention what tools you used for your teamwork in your self-assessment from Section 5.

6.2 Asking questions

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

6.3 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`

Please note that we will run a plagiarism checker on your solutions.

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 like GitHub then you must set access permissions appropriately (permitting access only to yourself or your group).

7 Submission

7.1 For the Software Engineering Tasks in Section 3

Please submit a PDF (not a Word or OpenOffice 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 Inf2-SEPP course and go to “Coursework” - “SE Coursework Submission” - “Coursework 1”.

Submission is a two-step process: (i) upload the file, (ii) 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.

7.2 For the Professional Issues Tasks in Section 4

Please submit your blog posts for the markers to view as a PDF in 'Coursework" - "PI Coursework Submission" - 'Group Blog" by Deadline 1 (see Sections 4 and 9).

At the intermediate deadline for submitting blog posts for peer review (see Sections 4 and 9), please submit any blog posts that are ready and that you would like your classmates to provide feedback on under 'Coursework" - "PI Coursework Submission" - "Course blog". This will share these blog posts with all your classmates.

8 Deadlines

The deadlines for this coursework are as follows:

- **Deadline for requesting peer reviews on the Professional Issues blogs (formative): 23:59:59, Mon 25th January**

  This is your chance to submit any of your Professional Issues blog posts from Section 4 that you are ready with to the coursework blog where they are visible by all your other colleagues. Submissions after this deadline are not accepted. After this deadline, you will all be asked to provide feedback on other's groups' blog posts.

- **Returning peer reviews of PI blogs: 12:00, Fri 29th January**

  This is the time by which you should provide feedback to a good number (at least 3) of your colleagues' Professional Issues blog posts.

- **Deadline 1 (formative): 16:00, Mon 1st February**

  Please submit all parts of this coursework for formative marking at this deadline. You will get formative feedback back. Please note that this is the only deadline for submitting the Professional Issues blog posts to the markers. These blog posts will only get formative feedback for this coursework.

- **Deadline 2: 16:00, Mon 1st March**
  (compulsory, summative, marks which count!)

  For this final deadline, please consider the feedback received from the markers on the Software Engineering part of this coursework (only!) after the first deadline and re-submit this part, highlighting your changes to your solutions in green. Also, adjust your self-assessment justifications to explain how you have addressed marker comments, also as highlighted in green.

This coursework is worth 20% of the total coursework mark. The mark for this coursework only consists of the mark for the Software Engineering tasks, as the Professional Issues tasks will only receive formative feedback. This mark will become final after deadline 2 mentioned above.

Boris Mocialov, Chris Vasiladiotis, Cristina Alexandru, 2021.