Inf2-SEPP 2021-22

Coursework 1

Capturing requirements for an events app during the Covid-19 pandemic

1 Introduction

The aim of this coursework is to produce a requirements document for a simplified app which encourages and facilitates the management and attendance of social events during the COVID-19 pandemic.

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

COVID-19 is still a major threat, and care must be taken to protect people at risk. However, isolation has resulted in many individuals' mental health being affected, and in some businesses struggling to stay afloat financially given the decrease in customer numbers. As such, the Scottish government has made the decision to step in and encourage citizens to go out and attend more social events which can improve their well-being, with the hope that this also helps sustain the economy by creating more cash flow between individuals, businesses, and workers at those business.

The plan set forth by the government involves creating an app that allows people to find and book entertainment events, such as movie screenings, live shows, live music, escape rooms, sports events, bowling, and more. You are part of the software company that has been contracted to create this app.

2.1 Managing and Booking Events

Entertainment providers (like tourist attractions, theatres, sports venues, cinemas, pubs, etc.) can register onto the app by themselves. Once registered, they can create events and provide information about them. This information must include: title; type (e.g. music, theatre, dance, movie, sports); any performer names; date(s), time(s) and venue(s); COVID-19 measures like any social distancing, air filtration, capacity limits, size of venue; whether it's ticketed or not; number of tickets available if ticketed; price of ticket if ticketed.

Individual consumers can also register on the app using a full name, email address, phone number and password, and log in with their email address and password. The government would like them to optionally be able to set up profiles that allow them to customise their various COVID-19 related preferences for events: whether they would like the events shown to them to comply to certain restrictions regarding social distancing, air filtration, capacity limits, size of venue, and/or the venue being outside.

Once logged in, consumers can ask for a list of available events for a certain date which respect their profile preferences (if any). They can then decide to book one ticketed event at a time (the non-ticketed ones do bookings). They can choose the number of tickets that they would like to book, as long as it does not exceed the number of tickets left available. The whole booking process is handled directly by our app, except for payment, which is handled by an external payment system. Successful payment on the payment system leads to our app issuing a unique booking number for the consumer and a record for that booking. These records must include the consumer's name and contact information (email address, phone number), so that the government can keep track of the potential spread of COVID-19 and inform the NHS and consumers (outside of the app) if needed.

Consumers should be able to cancel bookings made on our app with a given booking number, provided the event in question is at least 24 hours away. Companies should be able to cancel an event on our app at any time in response to changing government COVID-19 restrictions, which should automatically notify the consumers and request the payment system to refund them (if the booking was made on our system).

2.2 Government Sponsorship

The government would like to sponsor certain ticketed events by paying a percentage of the total price of the event's tickets in advance. This is meant to help the entertainment providers who would thus have a guaranteed cash flow, but also encourage consumers to attend these events by providing discounted tickets. For ease of use, the government would like there to be a system within the app where they can agree upon a deal with the entertainment provider. In particular, as the entertainment provider creates an event on the system with its original ticket value, they can request government sponsorship. The government can review such requests and accept them – specifying the percentage of

the original tickets' total price it can sponsor – or reject them. Consumers can then see such discounted events (and the discount percentage) in addition to regular events.

3 User Interviews

The app can be further extended in many different ways and has a potentially very large scope. In order to narrow down what specific features to add to the system, your team has interviewed some potential end-users of the app - people that would want to use it to find entertainment events. The team interviewed 100 such people. Each of them was asked what essential functionality the app should have such that they would consider using it, and what additional features they would like to see in it.

- 80 consumers requested that a map is provided for each event so they can get to it easier. Of them, 45 would also want a convenient way to see what public transport is available to get to the venue. The other 35 have cars or bikes, and would like to get directions to the location for their vehicle.
- 55 of the above 80 would also like to be able to specify preferences about whether they feel comfortable about travelling to the event by public transport or not (as they feel anxious because of COVID-19), what is their maximum wanted walking distance to the event, and what vehicles they currently have access to, such that they can get better filtered results when searching for events.
- 14 of the consumers would like to be able to also automatically book a taxi to the event through the app.
- 60 of the consumers (of which 15 wanted to see public transportation options to the venue) mentioned additional concerns about COVID-19 - they'd like to be able to know how crowded the venue where the event is held, and the transportation options to it, typically are.
- 20 of the consumers said they wanted a review system, where users can rate events after going to them and leave comments. When browsing possible events, the average rating should be readily visible, and users can look up comments from others. Also, 16 consumers said that the search results should be ordered by average rating, from highest to lowest.
- 13 of the consumers want the app to provide a weather forecast or a warning for bad weather if the event is outdoors.

Ultimately, the government wants consumers to use the app, so they are willing to support all reasonable feature proposals. In other words, for the rest of this part of the coursework, consider all the user-requested features above.

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

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

How you split the solving of each of the following tasks is up to you, but we recommend you at least work on the more complicated tasks as a team or sub-groups of the team. For the whole assessment for this course, it is very important that you practice your team work skills, and some credit will implicitly be given for them.

Include in your requirements your solutions to the following tasks.

4.1 Task 1: Stakeholders

Identify the stakeholders of the system and, for each stakeholder, describe how the system impacts them.

Groups of 2: Write down at least 6 stakeholders.

Groups of 3: Write down at least 8 stakeholders.

Groups of 4: Write down at least 10 stakeholders.

Regardless how many stakeholders you are providing, start with those that are mentioned in the system description and user interviews above (which may or may not already cover the minimum number of stakeholders you are requested). Then, make sure that you also add at least one additional stakeholder that would make sense in the context provided, but was not mentioned. There is no need here to cover stakeholders common to most software development projects (i.e., software architects, designers, developers and testers), and you will not receive credit if you do.

4.2 Task 2: Ambiguities and addressing them

The system description from the government and the interview responses omits many details, and may even be misleading on particular points. Note that for future courseworks, a lot of these issues will be resolved and clarified, but for this one, you need to identify them and discuss how you would handle them. As you go through this coursework, make sure you return to this task each time you find a new ambiguity and/or you need to make new assumptions. All of these should be discussed centrally in this task.

In this task, write a bulleted list of ambiguities you identify. Make sure each item in the list contains:

- Details of the ambiguity: what is unclear / ambiguous given the system description and interviews? What are different ways to address this in the system?

- Who to approach and how: If you wanted to clarify this ambiguity, who would you approach, and how would you try to clarify things (i.e. which requirement elicitation techniques are suitable)?
- Assumptions: If for any subsequent task you need to make assumptions regarding the identified ambiguity, clearly state these assumptions. Why did you choose to make these particular assumptions? Your answers to subsequent tasks should be consistent with your answer to this task.

We provide one example of an ambiguity with answers to the questions above to give you an idea of what we expect in each of your list items (not all your items need to be this long):

Example: "What information is needed from entertainment providers when they register on the system? Currently, the description doesn't specify any. This point should be discussed with both the providers and the government (as they may require certain company information) during interviews or facilitated meetings. For subsequent tasks, we assume only basic information is required, such as organisation name, main address, email, phone number, representative name(s), because this is usually the bare minimum contact information for a company."

4.3 Task 3: Use case diagram and use case prioritisation

Now consider the use cases for the system. For now, just think about them on a high level, without going into the details of each use case.

a) Draw a UML use case diagram showing visually the use cases that you have identified and the actor(s) (both primary and supporting ones) that 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.

   Your use case diagram must contain the use cases "Book event", "Create event" and "Cancel event" (note: the entire event and not a booking for it).

   Regardless of group size, try to cover the entire system as described. It is unlikely you will be able to do that with any less than 13 use cases (including the 3 from above). Keep these use cases as high level as possible, i.e. it is better to represent as a use case a large piece of functionality rather than its steps, unless those steps can be used independently or repeated in several contexts.

b) Under the diagram, take each of your identified use cases in turn and state whether in your view it should be *high-priority* (i.e. it is essential for the system's operation and should thus be within the first features to implement) or *low-priority* (i.e. it is an optional feature that isn't so important for the system's operation and main goal), and justify this decision briefly using all information that was given to you. Should this prioritisation be your decision?

4.4 Task 4: Use case descriptions

Describe the use cases you identified in the previous task in more detail. To this end, your solution should include:

a) A full description using the template provided in the *Tutorial 1* question sheet used for the Week 3 tutorials of the following use cases:

- "Book event"
- "Cancel event"
- For larger groups only, some more use cases - see below.

b) A shorter description using a simpler format with just the primary actor, supplementary actor(s) if relevant, and a maximum 6-sentence free-text summary of the use case and its interactions, for some of the simpler use cases. See exact number below. Regardless of group size, one of these simpler use cases should always be "Create Event".

Groups of 2: Write full descriptions for the 2 use cases mentioned in a) above. Write shorter descriptions as explained in b) for "Create event" and another 3 use cases, starting with any remaining high-priority ones.

Groups of 3: Write full descriptions for the 2 use cases mentioned in a) above plus another high-priority use case you identified in Task 3. Write shorter descriptions as explained in b) for "Create event" and another 4 use cases, starting with any remaining high-priority ones.

Groups of 4: Write full descriptions for the 2 use cases mentioned in a) above plus another 2 use cases, starting with any remaining high-priority ones you identified in Task 3. Write shorter descriptions as explained in b) for "Create event" and for another 5 use cases you, starting with any remaining high-priority ones.

Some guidelines:

- The use cases should be about the main interactions between actors external to the system and the system itself and should explain what is supposed to happen from the user's point of view. They should not be concerned with the details of user interface interactions or how the system handles everything underneath: you are not doing design at this stage.
- Feel free to add extra fields in the template if you feel it would help, but don't omit fields from it unless the answer to them is 'None'.
- In your descriptions, write both the main success scenario, but also any and all alternative scenarios.
- Make sure you reference use cases from within the descriptions where this would shorten the explanation, as exemplified in Tutorial 1 solutions.

4.5 Task 5: Non-functional requirements

Describe non-functional requirements which are relevant for the system. There are various general categories of non-functional requirements including Security, Performance, Privacy, Usability, Platform Compatibility, Availability, Accessibility, Interoperability, and Data Retention. Identify some 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. Write the requirements using the format for a requirements specification shown to you in Lecture 4.

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 (i.e. make these requirements measurable).

Groups of 2: Identify at least 2 categories and 5 non-functional requirements.

Groups of 3: Identify at least 3 categories and 6 non-functional requirements.

Groups of 4: Identify at least 4 categories and 7 non-functional requirements.

4.6 Task 6: 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. For each item below, we expect 2-4 sentences.

1. For the app that is being developed and your role within this development, and disregarding your tasks (i.e., the way you started work on it) in this assignment, would software project engineering or software product engineering be a better choice? Please justify your answer, both in terms of the reasons for choosing it, *and* 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 for requirements engineering in this assignment? Why do you think this is the case?

3. **Groups of 3 & 4:** Would the other type of software development process have been better in this context? Justify your reply.

4. **Groups of 4:** If we were instead using the other software development process, how would have requirements engineering been performed? Give some specific differences.

4.7 Task 7: Reflection

This section asks you to reflect and self-assess your team's progress with this coursework. This is a great opportunity to take the time to learn from your experience with this coursework, and thus develop your analytical thinking skills. It will also help you judge

your work before we do, by interacting with the marking scheme and thus gaining an understanding of how you are being marked.

Before you start doing any reflection, here is a useful model (adapted from the Integrated Reflective Cycle (Bassot, 2013)) that you could use to structure your reflection:

1. **The Experience**: Describe what you did, what you tried out.
2. **Reflection on Action**: What were the results? What went well? What didn't? Why?
3. **Theory**: What have you learned from this experience?
4. **Preparation**: What could you have done to make things better, according to the lessons learned? If you have the chance to do this again (e.g. team work), what will you do or try out next time to try to make things better?

You can see an example of this model being put to use at this link.

Write one paragraph with your reflection on each of the following topics (we expect between half a page and one page in total for both topics, with more effort gaining more credit):

1. Teamwork: here, reflect on things such as how you got organised, split up responsibilities between team members, communicated, and managed progress in working towards the deadline for this coursework. Make sure to mention and reflect on the use of and usefulness of any tools that you tried out in this process, e.g. physical tools or online tools for managing your team work.
2. The quality of your work: here, reflect on how well you think you tackled the work in the different tasks. We recommend you have a look at the marking scheme from this link to help you structure this response, however touching on all criteria from there, or marking yourself using it is not expected.

Important! You should make a real effort to be reflective, as well as honest, in this task. Please note that only making bold statements like "We did this excellently well", with no justification, and (even worse!) not being open to consider that there is always room for improvement, will result in very little, or even no, credit for this part.

5 Some advice

5.1 Working online as a team

Teamwork is not easy, and you may need to sometimes 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 colleagues, setting up meetings in the calendar, video calls, chat, editable file sharing, its Tasks By Planner and To Do to organise and split your work.
2. OneDrive (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 5.3).
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. Miro (`https://miro.com/online-whiteboard/`) 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 reflection from Task 7.

5.2 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, user interviews 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!

5.3 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).

6 Submission

Please submit a PDF (not a Word or OpenOffice document) of your requirements document. The document should be named **reportGroupX.pdf**, where you replace the

X with your group number. Please **do NOT include the names and/or UUNs of the team members** within this document, so that we can mark anonymously.

How to Submit

Ensure you are logged into MyEd. Access the Learn page for the Inf2-SEPP course and go to "Coursework" - "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 Deadline

Please submit all parts of this coursework for a final mark and feedback by **16:00, Thurs 10th February 2022**.

This coursework is worth 15% of the total coursework mark. The mark for this coursework only consists of the mark for the Software Engineering tasks outlined above. Professional Issues tasks will only be present for summative assessment in coursework 3.