

Inf2-SEPP 2025-26

Coursework 1

Capturing Requirements for a University of Hindeburgh Events App

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 for University of Hindeburgh students.

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

Increasingly, university students report struggling with socializing and decreased general well-being. The University of Hindeburgh has made the decision to step in and encourage its students to go out and attend more social events, which can improve their well-being and mental health. The plan set forth by the university involves creating an app that allows students to find and book entertainment events, such as movie screenings, live shows, live music, escape rooms, sports events, and more in the city of Hindeburgh. You are an intern in AcmeCorp, the software company that was contracted to create this app. You work alongside other junior developers and few more experienced colleagues.

Entertainment providers - or for short EP's - such as museums, theatres, sports venues, cinemas, pubs, etc., can register onto the app by themselves. They do this by providing their name, business number, main contact email and description. To make sure they are a legitimate business, the system must verify them by checking with a list of valid EP's - this list is curated separately, and our system simply needs to query it to check if an EP is legitimate. If an EP is deemed illegitimate, the system disallows their registration.

Once registered, EP's can log in. Once logged in, they can create events, which involves providing information about them and about their performances. Events must have a title and a type (music, theatre, dance, movie, sports, games). They can be ticketed (i.e. have a limited number of tickets available at a certain price), or non-ticketed. Events can have multiple performances, each with potentially different performer(s), and taking place at different dates, times and venues. Venues may be indoors/outdoors, have a capacity (i.e. maximum attendees), and if indoors may or may not allow smoking. EPs can request to see all the performances (theirs and other EPs') for a certain date. They can also see full details for a particular performance, and edit any of their performances which have no bookings yet.

Individual students are pre-registered on the app with their university email address and university password. They can log in with them. Once logged in, they can request a list of (event) performances for a certain date, and view a specific performance in detail, just like the EPs. They can book a performance, choosing 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 student and a record for that booking. These records must include the student's name and contact information (email address, phone number), so that the University can keep track of the students who attended different events (for example, to contact them to advertise similar events).

Students can cancel bookings made on the app with a given booking number, provided the event in question is at least 24 hours away. EPs can cancel a performance on the app at any time, which will automatically request the payment system to refund students with bookings for it (if the booking was made on our system).

University admin staff are also pre-registered on the app (like students), and they can log into it. They can request the list of performances and view a specific performance in detail, just like students and EPs. They can additionally decide for the university to sponsor a future ticketed performance, providing the amount (£) by which the performance's ticket price would be reduced for students (up to the total ticket price). This will result in the ticket price being reduced by that amount for any future student bookings.

The external payment system handles notifying all parties involved of payments and reimbursements. Ticket payments are transferred to the university's bank accounts by this external payment system. From there, the university's financial team handles transfers to the entertainment providers' accounts, but this is also external to our system.

2.0.1 University Representative Interview

Much of the relevant information regarding the events app has been obtained from an interview between another member of your development team (Bobby) and a representative from the University of Hindeburg (Cristina). This interview meeting has been recorded and transcribed. You can watch the whole **interview video** at this this link

(available with closed captions). The **captions** (which you can use as a transcript) are available to download from this link.

As will be clarified in the tasks below, your team is tasked with detailing the requirements for the new system. You are expected to extract requirements from the above interview, and adhere to these requirements throughout the remainder of this coursework.

2.0.2 Student Representative Interview

In order to further clarify the requirements which concern students, your AcmeCorp managers suggested you interview a student representative. To this end, the lab demonstrators in your Inf2-SEPP labs can play the role of University of Hindeburg student reps, and you are encouraged to interview one of them. It would help if your team are all there together for this interview. Note that the lab demonstrator will ask for your team number. Plan your questions carefully – you will only have up to 5 minutes and you can only do one interview (the lab demonstrators will check).

In task 1 (section 3.1) you will be asked to report on ambiguities, questions and answers from your interview with the lab demonstrator playing the role of student rep, and any assumptions you needed to make for your solutions. Please note that the lab demonstrator may not be able to answer all your questions, and you should report on this and make any necessary assumptions deemed reasonable for your solutions to this coursework.

3 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 teamwork skills, and some credit will implicitly be given for them.

Include in your requirements document your solutions to the following tasks.

3.1 Task 1: Ambiguities and addressing them

The system description from section 2 and what you can learn from the university representative interview include some ambiguities. The purpose of this task is to discuss all ambiguities that you find while working on this coursework. Therefore, **as you go through subsequent tasks, make sure you return here to report on any emerging ambiguities.**

This task is concerned with those ambiguities that affect your team's work for this specific

coursework. You should **not** concern yourselves with other ambiguities (e.g. about parts of the system description that you are not focusing on, or that may only affect your work in subsequent courseworks). Therefore, **before starting work on this task, read the remainder of the tasks for your team size.**

To clarify any ambiguities, you may want to conduct **one** 5-minute interview with one of your lab demonstrators, as explained in section 2.0.2. This would help you achieve a higher mark for this task (see the marking scheme here) but, even if you do not do it, addressing this task is still required. Assume that the lab demonstrator is representing the views of students who would eventually use the event booking system, and can only see you once. If you discover a new relevant ambiguity after doing your interview with the lab demonstrator, simply write it down and state the assumption you made for it.

Write a bulleted list of ambiguities and how you addressed them. Make sure each item in the list contains (as concisely as possible):

- **What is the ambiguity?**
- **If you interviewed a lab demonstrator about this ambiguity, what was the question you asked them regarding it?**
- **If you interviewed a lab demonstrator about this ambiguity, what is the exact answer you were given?** Provide it here, even if it was 'I am not sure' or similar. This is important, as it allows the markers to check how much your solutions adhere to the interview answers.
- **If the ambiguity was not (fully) addressed by your lab demonstrator interview, or it was not relevant for students, which other stakeholders would you approach and how would you approach them (i.e. what requirements elicitation techniques would you use) to clarify it? In the unavailability of these stakeholders, what assumptions are you making for the purpose of this coursework?** Note that you must make a note of all assumptions that appear in the remainder of your submission.

3.2 Task 2: Use case diagram

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.

Regardless of group size, try to cover the entire system as described. Keep the 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.

3.3 Task 3: Use case descriptions

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

- 1) A full description using the template provided in the *Tutorial 2* question sheet used for the Week 3 tutorials for:
 - (a) the use cases that include the functionality of booking tickets and canceling a booking
 - (b) other of the more complex use cases you identified in Task 2 (numbers below).
- 2) A shorter description using a shorter 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:
 - (a) the use case that includes the functionality of creating an event.
 - (b) some of the simpler use cases you identified in Task 2 (numbers below).

Groups of 2: Write full descriptions for the use cases mentioned in 1a above. Write a shorter description for the use case mentioned in 2a.

Groups of 3: Write full descriptions for the use cases mentioned in 1a above, and for an additional use case as described in 1b. Write shorter descriptions for the use case mentioned in 2a, and for an additional use case as described in 2b.

Groups of 4: Write full descriptions for the use cases mentioned in 1a above, and for another 2 use cases as described in 1b. Write shorter descriptions for the use case mentioned in 2a, and for 2 other use cases as described in 2b.

Some guidelines:

- 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 2 solutions.
- Don't omit fields from the template unless the answer to them is 'None'.

3.4 Task 4: Non-functional requirements

State non-functional requirements **which can be attached to your use cases from task 3 above**, using the following format:

- **Use case name:** The name of the use case as in your previous task's solutions.
- **Non-functional requirement:** Use the carefully structured English format for requirements specification with its two verb alternatives shown to you in Lecture 4.

- **Non-functional requirement category:** Choose the appropriate one between Security, Performance, Privacy, Usability, Platform Compatibility, Availability, Accessibility, Interoperability, and Robustness.

Your non-functional requirements should be deduced from a mix of the system description, interview with the university representative and interview with the student representative, and/or be reasonable assumptions (make sure they are mentioned in Task 1 solutions!). They should be clearly connected to the use cases.

In what you write under 'Non-functional requirement', make sure that you try to always be specific enough such that the non-functional requirement is measurable. Sometimes this is not possible, but you should try to pick at least one non-functional requirement which can be made measurable.

Groups of 2: Identify 3 non-functional requirements falling under 2 categories.

Groups of 3: Identify 4 non-functional requirements falling under 3 categories.

Groups of 4: Identify 5 non-functional requirements falling under 4 categories.

3.5 Task 5: Reflection

This section asks you to reflect and self-assess your team's progress with this course-work. We think such reflection could have an impact on your learning and of your understanding/expectations of how you will be marked.

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 credit for this part.

Start by having a look at the following reflection model (adapted from the Integrated Reflective Cycle (Bassot, 2013)) that you are asked to 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. teamwork), 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](#).

3.5.1 Reflection on teamwork

Using the above reflection model, write one or two paragraphs summing up to *maximum 250 words* of reflection on your team's teamwork for this coursework. Focus 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 or online tools for managing your teamwork. You can use the reflection model repeatedly to describe how you improved your teamwork over time.

3.5.2 Reflection on the quality of your work

Using the same reflection model, write one or two paragraphs summing up to *maximum 350 words* of reflection on the quality of your work. Mention how well you think you tackled the work in the different tasks, including the interview with the lab demonstrator (Would you do anything differently if you were allowed to elicit requirements by approaching them again (thinking both in terms of elicitation technique but also information sought/points discussed)?). We recommend you have a look at the marking scheme from this link, making specific reference to parts of it, to help you structure this response, however touching on all marking criteria or marking yourself using it is not expected. You can use the reflection model repeatedly to describe how you improved your solutions over time.

3.5.3 Reflection on the approach taken

Finally, using the same reflection model, write one or two paragraphs summing up to *maximum 350 words* of reflection on whether and how the type of software development process that you used throughout this requirements engineering stage of development has helped (or not) you and your team in the context of developing this system. Make sure to mention the characteristics of the type of software development process you used, how it handles requirements engineering, and discuss how well this fitted aspects like your profile, your team, the nature of the system you are developing, and the type of engineering that was required here (project or product). In case you consider that the process type used has not been ideal, consider (as part of the 'Preparation' component of the reflection) what other process type you would use should you have the opportunity to re-initiate this work. What characteristics would make that alternative type more suitable in this context? Note: purely theoretical answers, i.e. that do not apply the reflective model and/or do not discuss concepts in relation to the current case study, will receive low marks.

3.6 Declaration of work

You are also required to declare the amount of work¹ that was carried out by each of your team members, so that we can adjust your individual mark accordingly. In particular, you are each expected to split up work *fairly*. This means each team member doing:

- **Teams of 2:** around half (50%) of the work
- **Teams of 3:** around a third (33%) of the work
- **Teams of 4:** around a quarter (25%) of the work

on this assignment (no matter how you split responsibilities).

For this task, prepare the following:

1. As a team, a **Team Work Declaration document** (see submission details in Section 5) and do either a) or b) from below:
 - a) If you consider *your work was split fairly according to the definition from the footer*, include in it the text "The work was split fairly between our team members". Then, each team member signs, *but only if they agree with the percentage next to their name* (see Terms below).
 - b) If not, write for each team member an estimate (in percentages) of how much work they have carried out. Then, each team member signs, *but only if they agree with the percentage next to their name* (see Terms below).

For the signature, a digital one or a picture of one taken with a mobile phone are both accepted. Alternatively, the document itself could be a picture of a handwritten document including the different team members' signatures.

2. Privately as an individual, fill in **the Individual Work Declaration Form from this link** to describe in your own words the work that you have put into this coursework. This will help us make a more informed decision regarding your mark.

Terms: You have 30 extra minutes to submit the two declarations. Failure to submit the Team Work Declaration document (even in the presence of submitted Individual Work Declaration Forms), or submitting it after the deadline + 30 minutes, will lead to our assumption that you have split the work fairly, and you will all receive the same mark without any objections possible. We will accept a Team Work Declaration document without all signatures included, as long as it is submitted within 30 minutes after the deadline. However, note that submitting it without all signatures will lead to a potentially difficult conversation between the course organiser and the team members. In the absence of any submitted Individual Work Declaration Forms, a fully signed Team

¹By "work" we mean a product of the effort and time spent throughout this coursework. Work contribution should thus not only include the contribution to the final submitted solutions, but also things like studying the theory, learning about teamwork, preparing the lab demonstrator interview, organising and attending team meetings, checking and providing feedback/improving others' solutions.

Work Declaration document option a) will result in the same mark for all the team members, while option b) will mean bringing the mark down for anybody with a lower percentage than expected, in accordance with the declared percentage (e.g. if a team member in a 4-person team did 15% of the work, this is 60% of the 25% they needed to do, so they will receive 60% of the team's mark). Individuals who have done more than their share will not have their mark increased according to the percentage, but may respect criteria for excellence and exceptionality for a very high mark (see marking scheme), and be considered for a small bonus (at the course organiser's discretion) after all courseworks. When Individual Work Declaration Forms are also submitted within 30 minutes after the deadline, they will also be used to decide on the individual marks, with the same considerations for students putting in more than their share of the work as above.

4 Some advice

4.1 Working as a team

To get organised and work effectively as a team, carry out the teamwork activities in meetings and labs, have a look at the **Teamwork Resources** from here, and follow any advice on teamwork from guest lectures. Mention what you have done and used for teamwork in Task 5. If you need advice on teamwork, or you have experienced difficulties with it which you have attempted to address for the past 7 days with no/little success, please ask for help on Piazza (tag with *teamwork*) or email your course organiser. She will discuss the matter privately with you and/or your peers, as appropriate.

4.2 Asking questions

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

In case of questions regarding the organisation of teamwork in this course, please consult the **Teamwork FAQ** document from this link . Use Piazza (tag with *teamwork*) or email your course organiser in case your question was not covered there.

4.3 Good Scholarly Practice

Please remember the university requirement as regards all assessed work, outlined at this link. **This coursework allows AI-Assisted Planning only** (see the same link).

We will run a plagiarism and Generative AI checker on your solutions, and report you in case we suspect academic misconduct.

You are also required to take reasonable measures to protect your assessed work from unauthorised access. E.g., if you put any such work on an online repository like GitHub then you must set access permissions to allow access only to you and your team.

5 Submission

Please separately submit the following three items:

1. A PDF (not a Word or OpenOffice document) of your team's requirements document. The document should be named **reportTeamX.pdf**, where you replace the X with your team number. Please **do NOT include the names and/or UUNs of the team members** within this document, so that we can mark anonymously. Only one of the team members needs to submit this document, and the last submission will be considered for marking if several team members submit.
2. A PDF (not a Word or OpenOffice document) of your 'Team Declaration of Work' document, entitled **DeclarationTeamX.pdf**, where you replace the X with your team number. This document will only be accessed by your lecturers. This submission should also only be made by one of the team members, but this team member can be different to the one who submitted the requirements document.
3. The responses to the **Individual Work Declaration Form** of each team member. These should be filled in and submitted privately by each individual.

How to Submit

Ensure you are logged into MyEd. Access the Learn page for the Inf2-SEPP course and go to "Assessment" - "Assignment 1". For your requirements document, use the "Submit Report" submission box. For your declaration of work, use the "Submit Declaration of Work" submission box. For your Individual Work Declarations, use this form.

Document 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 asejfa@ed.ac.uk attaching your assignment, and if possible a screenshot of any error message.

- 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 ask for help if you have a problem submitting.

6 Deadline

Please submit the report by **12:00, Wed 4th February 2026**. You are allowed to submit the team and individual Declarations of Work 30 minutes later.

This coursework is worth 15% of the total coursework mark. We estimate it should take each team member around 13 hours of work. Extension rule 2 from this link is used for this coursework- see details for group submissions. This coursework allows AI-Assisted Planning only, as described at this link.