

DIT046 / DAT356 Requirements and User Experience Final Exam

April 13, 2022

Examiner/Contact Person

Jennifer Horkoff

Jennifer will come to the room to check for questions at roughly 15 and 17.

Alternatively Phone: 0733 050517

Authorized Aids

Textbook: Designing Interfaces, 2nd or 3rd Edition, By Jenifer Tidwell (Charles Brewer & Aynne Valencia)



Grading Scale for Exams

% Grade	Final Grade
0-49	Fail (U)
50-64	3
65-79	4
80-100	5

PLEASE OBSERVE THE FOLLOWING:

- All answers must be in English. Answers must be legible and readable.
- Sort the questions in order before handing them in.
- Put the number of the question on every paper.
- This exam has 6 pages.

Part 1: Multiple item and Short-Answer Questions

Question 1: Multiple Choice Questions (10 points)

On your paper, write the name of the question, then the letter(s) of your chosen answer(s), e.g., 1.1: c., 1.3: a, d, e. For each question, there may be more than one correct answer. For full marks, list all the correct answers. Each correct answer listed is +1 point, each incorrect answer is -1 point.

1. Which of these requirements captures design information and is therefore not well-formed?
Note: these requirements focus on a bike rental system. In this question there may be more than one correct answer, list them all.
 - (a) The system will allow users to rent a bicycle
 - (b) The system will use a payment gateway which connects to a merchant account
 - (c) The system will implement an event-based architecture to satisfy rental demands
 - (d) The bike lock will not be released until the payment transaction has completed successfully
 - (e) The system should support payment by common credit and debit cards
 - (f) The system will include a lock sensor to detect the presence of bikes

Answer:

b, c, f

2. Which of the following are desired qualities for user stories according to the INVEST acronym? One or more may be correct.
 - (a) Easy
 - (b) Estimable
 - (c) Interesting
 - (d) Independent
 - (e) Valuable
 - (f) Verifiable

Answer: b, d, e

3. Which of the following are positive user experience goals?
 - (a) Aesthetically pleasing
 - (b) Boring
 - (c) Annoying
 - (d) Provocative
 - (e) Frustrating

Answer: a, d

4. Which definitions of requirements terminology are correct? There may be more than one correct answer.
 - (a) Specification: things in the application domain that are true whether or not we build the system.
 - (b) Domain Assumptions: the software and hardware that solves some problem, meets some need.

- (c) Machine Domain: the behavior that a program needs in order to solve the problem.
- (d) Application Domain: the world, where people, organizations and problems live.
- (e) Requirements: things in the application domain we want to make true by building the system.

Answer: d, e

Question 2: Short-Answer Questions (20 points)

1. Name and briefly describe four creativity techniques. (4 marks)

Answer: There are so many of them, any of techniques listed here with a 1-sentence reasonable explanation would count: <http://becreative.city.ac.uk/>

2. Name and describe two approaches to prototyping and two types of prototypes. Why would you use or not use each of these approaches and types? Note, neither approaches or types are asking for specific prototyping software (e.g., Balsamiq, Figma are not the answers we are looking for). (8 marks)

Answer:

Approaches: Throwaway or evolutionary (evolving) prototypes

throwaway would be used when you want to explore ideas, wouldn't use it when you make a very detailed UI that is close to the real thing.

evolutionary would be used when you want to build on the UI and make it realistic, would not use when just exploring ideas, it can lead to a messy architecture and design if things are not well thought out.

Types: paper, digital, web-based

paper would be used to avoid using tools and giving the impression of a near complete prototype, not used when the UI is complicated or if your user tests are online

digital would be used when you want a dynamic prototype that starts to look like the real thing, you wouldn't use it when you want to avoid using extra software, or if you don't want to give the impression of a working system.

Web-based would be used when you want a dynamic prototype that starts to look like the real thing and when you possibly want an evolutionary prototype. You wouldn't use it when you want something quick to explore ideas.

3. In his book "the Design of Everyday Things", Norman describes seven fundamental principles of design. List and briefly describe two of them. (4 marks)

Answer:

1. Discoverability. It is possible to determine what actions are possible and the current state of the device. 2. Feedback. There is full and continuous information about the results of actions and the current state of the product or service. After an action has been executed, it is easy to determine the new state. 3. Conceptual model. The design projects all the information needed to create a good conceptual model of the system, leading to understanding and a feeling of control. The conceptual model enhances both discoverability and evaluation of results. 4. Affordances. The proper affordances exist to make the desired actions possible. 5. Signifiers. Effective use of signifiers ensures discoverability and that the feedback is well communicated and intelligible. 6. Mappings. The relationship between controls and their actions follows the principles of good mapping, enhanced as much as possible through spatial layout and temporal contiguity. 7. Constraints. Providing physical, logical, semantic, and cultural constraints guides actions and eases interpretation. [page 72]

4. Name and define two types of UI defects from the lectures (Hint: the lecture gave five types). (2 marks)

Answer: • 1. Missing functionality or bug: function the user wants is not there, or crashes, doesn't work • 2. Task Failure: user cannot figure out how to do a task in a fixed amount of time • 3. Annoying: user can do a task in a reasonable amount of time, but is annoyed • 4. Medium problem: user succeeds in task after a long time • 5. Minor

problem: user succeeds in task after a short amount of time (but it still took longer than expected)

5. Name two characteristics of UX design patterns (2 marks)

Answer:

- Concrete, not general
 - Not good design principles, like “Prevent errors,” “Create a strong visual hierarchy,” and “Don’t make the user think.” – Patterns are concrete enough to help fill the space between high-level general principles and the low-level “grammar” of user interface design
- Valid across different platforms and systems
 - The best patterns aren’t specific to a single platform or idiom
- Products, not processes
 - Unlike heuristics or user-centered design techniques, which usually advise on how to go about finding a solution to an engineering or design problem, patterns are possible solutions.
- Suggestions, not requirements
 - patterns are intended to be only suggestions; you can follow them or reject them, depending on your design context and user needs

Part 2: Domain Example and Long-Answer Questions

The remaining questions on the exam will relate to a problem in a domain, as described below. The scenario describes the situation today and the expectations for the new system. Focus your analysis and modeling on the envisioned (to be) system, but keep in mind the problems and requests with the as is situation, trying to avoid problems and satisfy user needs.

As is Situation: Tourist Website

City XYZ is a medium-sized city, but is becoming a trendy tourist destination. Right now, tourist attractions are listed on the XYZ city website, and each attraction has individual websites with information on what one can do, prices, opening and closing times, accessibility (for different abilities, strollers/prams for children). However, sometimes attractions are closed due to weather, COVID, holidays or other reasons. It is difficult for tourists to understand what they can do, given when they visit, and what their abilities and interest are. Often they go to a location and find out it is closed or is not accessible to them. Or they miss/forget about an attraction that was listed on the XYZ city site.

Desired to be Situation

XYZ City wants a tourist app/website that let users search for things to do the city. Users should be able to search by dates, times, type of attractions, accessibility. The app should show accurate information. Users should be able to save attractions they like and create an itinerary (schedule) for their visit. When they have attractions saved, if something changes in an attraction, e.g., new opening/closing times or constructions, the users should get a notification. Users should also be able to rate and add reviews to attractions, and read the ratings/reviews of others.

XYZ City managers should be able to see reports on which attractions are viewed most often and are added to itineraries. Attractions should be able to update their information in the app; however, updates must be approved by XYZ City managers before they appear in the app. XYZ City managers must also approve attraction review text before it appears in the app.

Because the app stores schedules, it should be secure so others cannot see the itineraries of users. There should be an offline mode to show the saved itinerary even if offline. When online, new searches should return information with 5 seconds.

The app/website should be both an app on a phone and something that one can login to in a browser on a computer. It should work on all Android/Apple phones within the last 5 years and all Windows/Mac browsers released within the last 5 years (e.g. Chrome, Edge). The system must be ready by July 2024.

Question 3: Context Diagram (10 points)

For the given case, draw a context diagram. Add the system actor, identify the relevant stakeholder/actor entities, and the relationships between the system and these entities, labelled with high-level inputs and outputs. Try to capture at least four actors (the system actor counts), and appropriate relationships. Supplement the diagram with text to explain any ambiguous or unclear parts of the model.

Question 4: Use Case Diagram (10 points)

For the given case case, draw a use case diagram. Capture the system and other relevant stakeholders. Identify the major use cases and relationships, both between the stakeholders and the use cases, and between use cases. Try to capture three actors, and appropriate use cases.

Question 5: Scenario (10 points)

Pick one use case (if you have not created a use case, create one now). For this use case, fill out Cockburn's Use Case template (obligatory template elements listed below). Flesh out the details of the process, including exception cases and error cases. Pick a use case with at least one extension and sub-variation. Consider prerequisites.

Cockburn's Use Case Template (reduced)

Use Case: < the name should be the goal as a short active verb phrase >

CHARACTERISTIC INFORMATION

Goal in Context: <a longer statement of the goal, if needed>

Scope: <what system is being considered black-box under design>

Preconditions: <what we expect is already the state of the world>

Success End Condition: <the state of the world upon successful completion>

Failed End Condition: <the state of the world if goal abandoned>

Primary Actor: <a role name for the primary actor, or description>

Trigger: <the action upon the system that starts the use case, may be time event>

MAIN SUCCESS SCENARIO

<put here the steps of the scenario from trigger to goal delivery, and any cleanup after>

<step #> <action description>

EXTENSIONS

<put here the extensions, one at a time, each referring to the step of the main scenario>

<step altered> <condition> : <action or sub.use case>

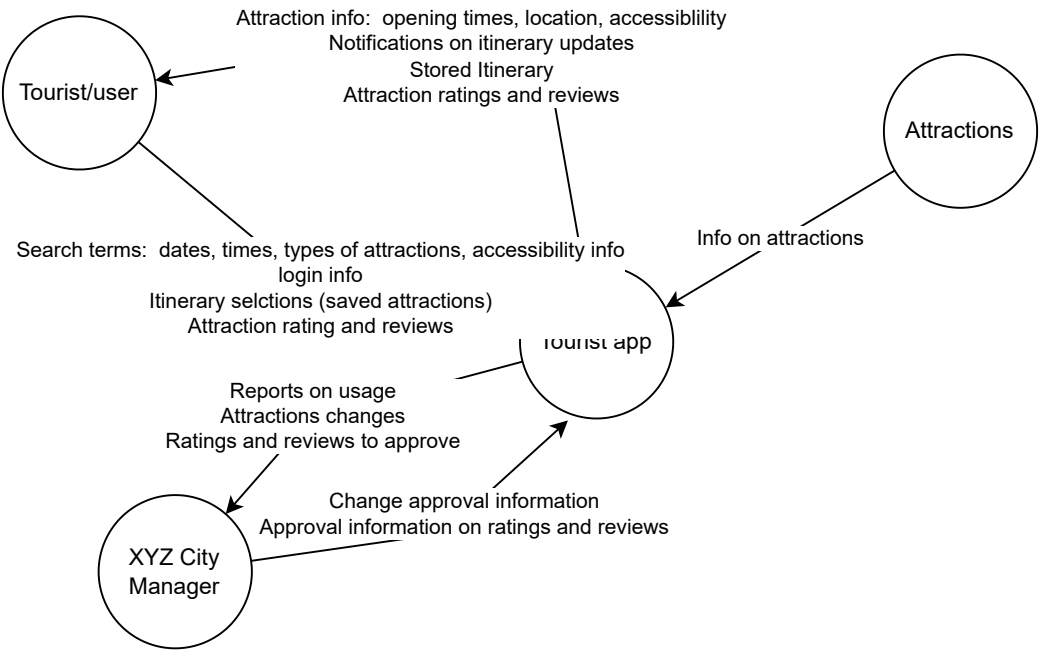
<step altered> <condition> : <action or sub.use case>

SUB-VARIATIONS

<put here the sub-variations that will cause eventual bifurcation in the scenario>

<step or variation # > <list of sub-variations>

<step or variation # > <list of sub-variations>



Question 6: Textual Requirements (15 points)

Record requirements for the case provided. List four functional requirements in SRS form, two non-functional requirements in SRS form, two domain assumptions, and two constraints. In addition, list five user stories. The user stories should be unique, i.e. not a repeat of one of the SRS requirements. Remember the desired characteristics of user stories and SRS requirements when recording your requirements.

Question 7: UI Design & Patterns (20 points)

Draw two UI screens for the given case implementation. Screens can include pop-up windows, just be clear how and why they appear. Each screen should use at least two different patterns from either Tidwell book, four different patterns in total. Each screen should implement at least two functional requirements (they do not have to be the requirements from question 6, just make it clear what requirements the screens implement). For each screen, write: the two requirements implemented (and how, if not obvious), the two patterns implemented, and why the patterns are a good choice for the screen and its functions.

Question 8: User Testing Tasks (5 points)

Come up with a list of five test tasks for the users to perform during usability tests for the two designed screens. Each screen should be tested by at least one task, for at least two tasks, indicate which of your two screens they test. The tasks should cover some of the main use cases or requirements of your system. Tasks should conform to the desirable qualities of user test tasks as discussed in the lecture.