Activity 3.3.1
User-centered Design

DATE	PORTAL DAY	LESSON	TURN IN
Wednesday 2026-12-10	**Surge and usability notes**	**Lesson 3.3 Information Sharing**	**Notebook**

Where this sits. Single day. This is the practice run for the end-of-unit problem.

01 Why today matters

Every hard thing in this unit ran through an app, and an app that is confusing during an emergency is worse than no app at all.

Today you design one screen to a fixed list of user needs, which is the exact skill the end-of-unit problem is graded on.

02 What you already know

- You have used the Hospital Hub app for eight class days: patient details, maps, staff schedules, supply logs, triage files, and messages.
- From Unit 2 you know Cass Cassa and the glucose-monitoring app she and her family use.
- You know from the surge simulation that a decision made under time pressure needs information that is instantly readable.

WHAT THIS SHEET WILL NOT TELL YOU

- This sheet does not design the screen for you and it does not show you a finished answer to copy.
- It gives you the user needs. Turning those into a layout is the work, and there is more than one good answer.

03 Vocabulary

TERM	WHAT IT MEANS, PLAINLY
app	A software application built to run on a phone or tablet.
user-centered design	Designing around what the person using the thing actually needs to do, instead of around what is easy to build or nice to look at.
end user	The specific person who will actually use it. Not 'people'. A night-shift nurse with gloves on is a different end user from a worried parent at home.
user interface	Written UI. Everything the user can see and touch: buttons, icons, text boxes, images, menus.
event	Something that happens which the app has to respond to. A user tapping a button, or the device receiving a message.
event-driven programming	Building an app as a set of responses to events, rather than as a list of steps that runs start to finish.
design sketch	A drawing of a screen, by hand or on a computer, showing where each element sits.

wireframe	A plain sketch that shows layout and function with no colour or styling, so people react to how it works rather than how it looks.
critical feature	The thing the app absolutely has to do. If it does not do this, it has failed, whatever else it does.
navigation	How a user gets from one screen to another and back.
iterate	To change a design based on what testing or feedback told you, then test it again.

Every term on this list is flagged for the illustrated glossary, scoped to this activity only. Terms are not shared forward or backward between activities, because a definition from a later activity can hand you an answer you were supposed to derive.

04 How to do the work

1. With a partner, list the apps you actually use, what you use them for, which were most useful, and why. Then the sharper question: name two apps built for the same job where you preferred one, and say specifically what the better one did.
2. Answer in your notebook: during a public health crisis or a medical emergency, what does a user need? Do it twice, once for a medical professional and once for a member of the public. The lists should not be the same.
3. Read the Binh Mai career profile and note what an app developer's day actually involves.
4. Review the screens of the glucose-monitoring app from Unit 2 in the Gluc-O-Meter App Screens presentation.
5. For that app, write out the interactions using this pattern: 'When the user does ____, the app responds by ____.' Write at least six. Include at least one event that is not a user tapping something.
6. Write down what you think its critical feature is. Not the longest list of features. The one thing that, missing, makes the app pointless.
7. Read through the three-step design process the developers used: decide the features, sketch each screen, then map how the features on each screen work.
8. Now your turn. Users have asked for a sleep-tracking screen. The requirements are fixed: log sleep from the previous night, enter the times of falling asleep and waking, see the weekly average, switch to other days, and get to the other screens in the app.
9. Open the Mobile Device Template, or sketch on paper or in your notebook. Sketch the screen so that all five requirements are met.
10. Label every element on your sketch with the event it triggers. A button with no label next to it is a decoration, not a design.
11. Check yourself against the five requirements one at a time and point to where on your sketch each one is met. If you cannot point to it, it is not there.
12. Swap with a classmate. Do not explain your design first. Watch them try to read it and write down every place they hesitate. A hesitation is a design problem, not a user problem.
13. Describe your events to them, take their feedback, and revise. Write down what you changed and what made you change it.

05 Safety

READ EVERY LINE BEFORE YOU START

- Screen and paper day. Nothing hands-on.
- Do not put anyone's real health information, or anyone's name, into a sketch or a mock screen. Use invented data. This matters again on Friday when you design your own app, and it is a real rule in the real profession.

06 Where students get stuck

- Designing what looks good instead of what works. Colour, fonts, and icons are the last five percent. Layout and clarity are the other ninety-five. Spend your time where the value is.
- Putting everything on one screen. More on a screen means slower to read, and slower to read is worse in an emergency. Cutting is a design decision, and it is usually the right one.
- Designing for yourself. You are a teenager with good eyes, two free hands, and no crisis happening. Your user might be wearing gloves, in the dark, one-handed, panicking, or 70 years old.
- Skipping the event labels. The events are how the screen actually works. Without them you have drawn a picture of an app, not a design for one.
- Explaining your sketch before the reviewer looks at it. Your explanation patches every gap and hides exactly the problems you needed to find. Hand it over silently and watch.
- Missing a requirement and not noticing. There are five. Check them one at a time against the sketch, and point at each one.

07 What you turn in

NOTEBOOK

Notebook check. Four things: your two user-needs lists, professional and public, kept separate; at least six 'when the user does, the app responds by' statements for the glucose app plus your named critical feature; your labelled sleep-screen sketch with every element tied to an event; and a short note on what you changed after peer feedback and what made you change it.

08 Check yourself

- [ ] What does user-centered design mean, and how is it different from making an app look good? (Vocabulary and Where students get stuck)
- [ ] Give one example of an event that is not the user tapping something. (Vocabulary and step 5)
- [ ] Why do you hand a sketch to a reviewer without explaining it first? (Where students get stuck)