# Problem 3.3.2
# Public Health Emergency Apps

| DATE | PORTAL DAY | LESSON | TURN IN |
|---|---|---|---|
| **Friday 2026-12-10** | **Public-health design project** | **Lesson 3.3 Information Sharing** | **Pre-lab** |

> **Where this sits.** Day 1 of 2. Define and generate today; design, test, and defend Monday. Note that this is a Problem, not an Activity, and it is graded against a rubric.

## 01 Why today matters

This is the end-of-unit problem, and it is scored against a rubric you should read before you write anything.

Today you pick a real public health problem and write a problem statement sharp enough that someone could tell whether your app solved it.

## 02 What you already know

- You have practised user-centered design on a screen with requirements handed to you. Now nobody hands you the requirements.
- You have lived through an outbreak investigation, a field rescue, an emergency room at capacity, and a regional surge. Every one of them had a communication problem in it.
- You know how to keep events and layout separate on a sketch.

**WHAT THIS SHEET WILL NOT TELL YOU**

- This sheet does not give you an app idea. Picking the problem is the graded work.
- It does not tell you what a good problem is for your community. Research that.

## 03 Vocabulary

| TERM | WHAT IT MEANS, PLAINLY |
|---|---|
| **problem statement** | A short, precise description of the problem being solved. Good ones say who has it, what it is, where, when, how widespread, and how you know. |
| **design brief** | The short document that sets out the target user, the problem statement, the justification, and the general requirements before design starts. |
| **target consumer** | The specific end user the design is for. |
| **problem justification** | Why this problem is worth solving rather than another one. |
| **design requirements** | What the thing has to do, written broadly at this stage, before layout. |
| **user persona** | A short invented profile of a typical end user, written to keep the team designing for a person instead of an average. |

| | |
|---|---|
| **iterate** | To improve a design based on testing and feedback and then test it again. |
| **Morbidity and Mortality Weekly Report** | Written MMWR. The weekly public health report published by the CDC. A good source for what is actually happening right now. |
| **personally identifiable information** | Written PII. Anything that can identify one specific person: name, address, school, email, phone, photo, birth date, ID numbers. |
| **HIPAA** | The United States law that sets rules for who may see and share a person's health information. |

*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. Form teams as directed. Read the Problem 3.3.2 rubric together, out loud, before anything else. Everything you are graded on is in it, and reading it on Monday is too late.
2. Everyone starts a project record in their notebook. Every team member keeps their own record of the team's work at every phase. That is a rubric requirement, not a suggestion.
3. Go back through your unit notes as a team and list every point where communication failed, was slow, or was missing. The outbreak, the rescue, the emergency room, the surge. Every one of those had a moment where the right information did not reach the right person in time.
4. Decide what kind of app you are aiming at: detecting and tracing outbreaks, responding to emergencies, making response more efficient, improving communication and awareness, or another public health function.
5. Research public health concerns in Cleveland and in the wider world. Check whether an app already exists. If it does, find its gaps; improving something real is a stronger project than inventing something nobody needs. The MMWR is a good place to start.
6. Add every new idea to the list. Do not filter yet.
7. Now filter. As a team, first define what impact means to you: number of people helped, depth of help, or how badly the need is currently unmet. Then rate each idea against your own definition.
8. Pick your problem.
9. Draft the problem statement. Answer, in order: who has the problem, what exactly it is, where it exists, when and for how long, how widespread it is with a number if you can find one, and who says it is a problem.
10. Test your statement against three rules. It has to be short. It has to leave room for more than one solution. It must not have a solution hidden inside it. If your statement contains the word 'app', rewrite it.
11. Write your justification: why this problem and not the others on your list.
12. Name your target consumer as a specific person, not a category. Write a user persona: who they are, what they are doing when they open your app, what they need in that moment, and what would make them close it.
13. Write your design requirements as a bulleted list of what the app has to do. Big picture only. No layout yet, no screens yet, no colours.
14. Individually, each person brainstorms at least two different ideas for how the app could solve the problem. Individually first. Groups converge too early and lose the second-best idea before anyone hears it.
15. Come back together, share every idea, and test each against four questions: does it solve the stated problem, does it meet the requirements, can it be done as a simple app, and has thinking about it changed who the user is or what is required?
16. Choose the solution you will design. Write down why you chose it over the runner-up.

## 05 Safety

**READ EVERY LINE BEFORE YOU START**

- Screen and paper day. Nothing hands-on.
- No real personal data anywhere in this project. Not a classmate's name, not a family member's condition, not a real patient. Every persona is invented and every screenshot uses invented data.
- If your app would handle health information, you have to say in the design who may see what. A doctor seeing all patient records and a user seeing another user's records are not the same thing, and the second one is how apps end up in the news. Read the HIPAA resource before Monday if your design touches patient information.

## 06 Where students get stuck

- A problem statement with the solution buried in it. 'People need an app that tracks outbreaks' is not a problem, it is a product with a complaint attached. The problem is what is going wrong for whom, before anyone mentions software.
- A problem so big nothing could solve it. 'People get sick' is true and useless. Narrow it until you could tell whether you fixed it.
- Skipping the rubric. It is the actual specification for what you are being graded on. Teams that read it last always find something they had to do all along.
- Converging on the first idea somebody says out loud. That is why the individual brainstorm exists. Protect it.
- Choosing a problem because it is easy to draw. You are graded on the design reasoning, not on the drawing.
- Only one person keeping notes. Every member keeps their own record. This is in the rubric.

## 07 What you turn in

**PRE-LAB**

Pre-lab for Monday's build, checked before you leave. Your team's design brief on one page: target consumer, problem statement, justification, and a bulleted list of design requirements. Plus the chosen solution with one sentence on why it beat the runner-up, and your user persona. Every team member has this in their own notebook. No brief, no build Monday.

## 08 Check yourself

- [ ] Name three things a strong problem statement has to answer. (Vocabulary and step 9)
- [ ] Why must a problem statement not contain a solution? (Where students get stuck)
- [ ] What is PII, and give two examples that are not a person's name. (Vocabulary)