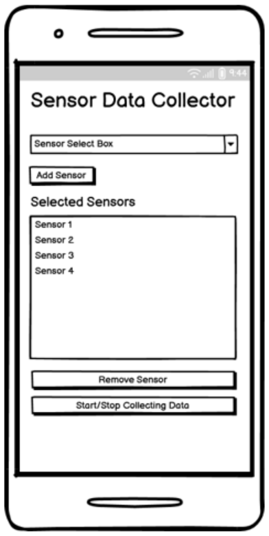
Module 4 Exercise: Cost/Time Estimation

- **Problem Question**: You have been approached to develop a new mobile software to collect data from sensors. The top of the application provides a select box with which the user could select a sensor. After selecting the sensor, the Add button will be pushed and the sensor is added to the list in the middle. The user could select several sensors. If the an added sensor is already in the list, the Add operation will be ignored. A selected sensor in the list could be marked. If the Remove button is pushed, the marked sensor will be removed from the list. If the start button is pushed, the application will start collecting data from the selected sensors. At the same time, the start button will change the label to Stop. While collecting data, adding or removing sensors is not possible. After pushing the Stop button, the collected data will be sent to a backend service.



- Major Estimation Workflows in Software Project Planning
  1. **Widget Points ➔ Function Points ➔ Expert Estimation**
     A. Workflow: Identify UI Elements (Widgets) ➔ Assign Points for Each Widget ➔ Calculate Function Points (FP = 2 × Widget Points) ➔ Estimate Hours per Function Point (5–10 hours) ➔ Multiply FP × Hours ➔ Total Effort ➔ Adjust for Risk Buffer (+10–30%) ➔ Calculate Cost based on Developer Rates
     B. Best for Settings:
        1) Small to Medium Apps (especially mobile apps, web apps, simple desktop apps)
        2) Projects where UI work dominates
        3) Agile development (short sprints, iterative cycles)
        4) Rapid prototyping scenarios
     C. Example: A mobile app to monitor sensors, a basic e-commerce frontend, a data entry admin dashboard.
  2. **COCOMO (Constructive Cost Model)**
     A. Workflow: Estimate Source Lines of Code (LOC) ➔ Convert to Thousands of Lines of Code (KLOC) ➔ Apply Basic Effort Formula (E = a × KLOC^b) ➔ Adjust Effort with Effort Multipliers ➔ Calculate Duration (D = 2.5 × E^0.38) ➔ Estimate Staff (P = E / D) ➔ Calculate Cost based on Monthly Salary

B. Best for Settings:
   1) Large Projects (>10,000 LOC)
   2) Enterprise software, backend systems, embedded systems
   3) Waterfall Model or heavily planned projects
   4) Formal documentation, testing, and project management overheads

C. Example: Developing an airline reservation system, a large hospital management backend, satellite control software.

3. **Expert Judgment / Analogy-Based Estimation**

   A. Workflow: Find Similar Past Projects ➔ Compare Scope, Complexity, Risk ➔ Adjust Based on Differences ➔ Estimate Time and Cost Based on Experience ➔ Add Buffer for Uncertainty

   B. Best for Settings:
      1) Startups or fast-moving teams
      2) Very novel projects where no historic data is available
      3) Teams with senior engineers with strong domain experience
      4) Highly agile environments

   C. Example: New AI prototypes, experimental mobile apps, quick MVPs for startups.

4. **Function Point Analysis (without Widgets)**

   A. Workflow: Identify Logical Inputs, Outputs, Inquiries, Files, Interfaces ➔ Assign Weights Based on Complexity (Simple/Avg/Complex) ➔ Calculate Unadjusted Function Points ➔ Adjust for Environmental Factors ➔ Estimate Time and Effort

   B. Best for Settings:
      1) Business applications (CRM systems, ERP modules)
      2) Middleware and integrations across databases
      3) Information-heavy applications
      4) Functionality-driven rather than UI-driven systems

   C. Example: An insurance claim management system, school registration portal.

5. **Story Points + Agile Velocity Estimation (Agile Methods)**

   A. Workflow: Break project into User Stories ➔ Assign Story Points to each story (relative size) ➔ Track Velocity (Points per Sprint) ➔ Estimate Number of Sprints Needed ➔ Calculate Total Time and Cost

   B. Best for Settings:
      1) Agile projects (Scrum, Kanban)
      2) Teams with established sprints and stable velocity
      3) Ongoing product development where features emerge over time

   C. Example: Adding new modules to a SaaS app over time, ongoing e-commerce enhancements.

6. **Comparison Table: Best Method by Situation**

Method	**Best When**	**Examples**
Widget ➔ Function ➔ Expert	Simple UI-heavy apps	Mobile apps, admin panels
COCOMO	Large, formal projects	ERP systems, control software
Expert Judgment	Unique/fast projects	Startup prototypes, AI research
Function Point Analysis	Functionality-driven systems	Business portals, internal tools
Agile Story Points	Continuous Agile dev	E-commerce sites, SaaS products

7. Important Rules of Thumb

   A. If your project is small, UI-driven, Widget Points are best.

   B. If your project is huge, structured, COCOMO is accurate but heavy.

    C. If your team is experienced, Expert Judgment is fast and practical.
    D. If your project is complex but business-driven, Function Points work.
    E. If you're working Agile, Story Points are mandatory.
8. Best Estimation Method by Project Size
    A. Small App (Few UI Screens) ➔ Widget Points + Expert Estimation
    B. Medium App (Business Logic Heavy) ➔ Function Points Analysis
    C. Large System (Formal, Big) ➔ COCOMO
    D. Agile Evolving Product ➔ Story Points Estimation
    E. Very Unique, No Similar Projects ➔ Expert Judgment

- Based on these rules, we will solve the provided cost estimation problem using the following methodology:

1. Use Expert Judgment (since it's a small clear project).
2. Use Widget Point Analysis (good because the app is UI-heavy, not complex algorithms).
3. Add Buffer for safety.
4. Estimate Cost based on developer hourly rate.

- Step 1: Widget Point Calculation:

UI Element	**Widget Type**	**Count**
Sensor select box	Input Widget (Combobox)	1
Add button	Input Widget (Pushbutton)	1
Remove button	Input Widget (Pushbutton)	1
Start/Stop button	Input Widget (Pushbutton)	1
Sensor list	Composite Widget (List/Table)	1

Function Points = 2 x Widget Points ➔ Function Points = 2 x 5 = **10**

- Step 2: Time Estimation Based on Function Points:

For small to medium mobile apps, a typical developer productivity is 5-10 hours per function point depending on design complexity, state management, and backend communication. Because the app is relatively simple, we estimate 6 hours per function point.

➔ Total time = 10 function points x 6 hours/function point = **60 hours**.

- Step 3: Cost Estimation:

Developer rates depend on experience. Online estimations using chatGPT suggest the following approximations:

Developer Level	**Rate ($/hour)**	**Total Cost**
Junior Developer	$30/hr	$1,800
Mid-level Developer	$50/hr	$3,000
Senior Developer	$80/hr	$4,800

- Step 4: Add Risk/Contingency Buffer:

Slides suggest adding **~20% buffer** for unexpected costs.

➔ Adjusted time = 60 hours X 1.2 = **72 hours**

Developer Level	**New Total Cost**
Junior	$2,160
Mid-level	$3,600
Senior	$5,760

- Final Professional Estimation given the following assumptions:
  1. Only basic backend connection (no complex server logic).
  2. Only simple UI (no animations, no complex state transitions).
  3. Targeting one platform first (Android or iOS).
  4. No hardware sensor drivers beyond basic API simulation.

Item	**Value**
Estimated time	60–72 hours
Estimated cost (Junior Dev)	$1,800–$2,160
Estimated cost (Mid-level Dev)	$3,000–$3,600
Estimated cost (Senior Dev)	$4,800–$5,760