

Milestone 3: Design for Interoperability

Released: Wednesday, Mar 5, 2025

Due: 11:59 pm Friday, Mar 21, 2025

Learning Objectives

- Extend an existing system to support additional features.
- Design interfaces for interoperability with other systems.
- Design a large-scale software system in multi-team settings.

Milestone Tasks

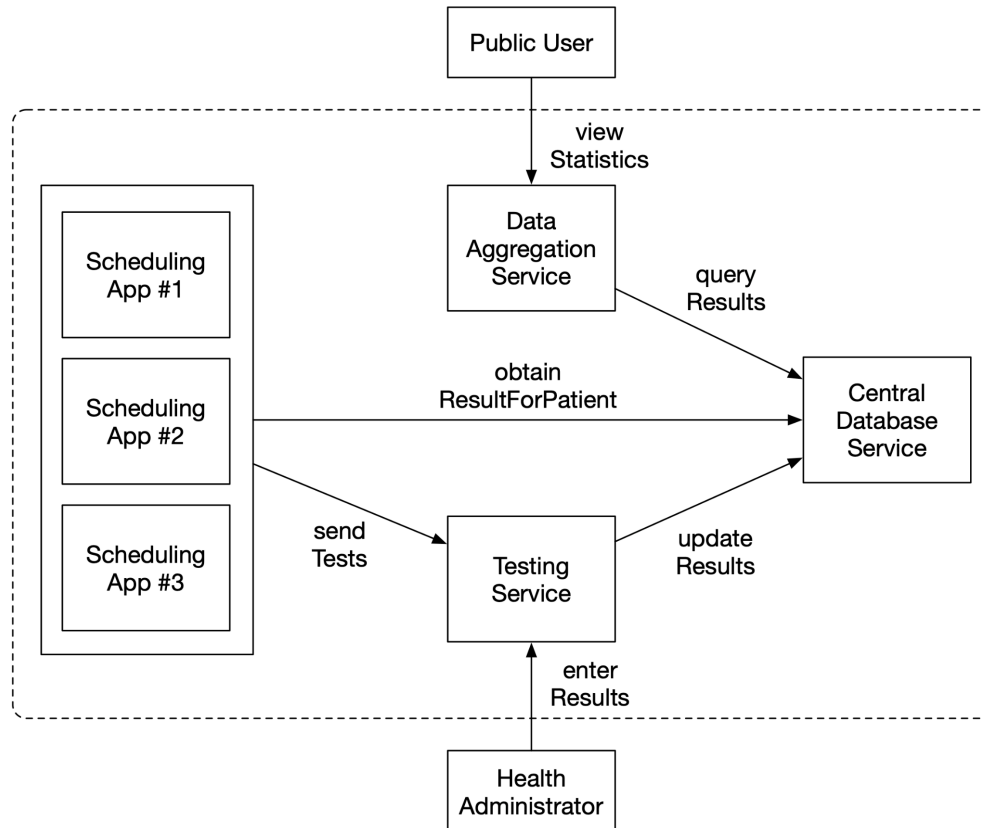
In this milestone, you will build on the prototype that you have developed in M2 to support additional features. In addition, you will start working with other teams in the class to design interfaces for a new service that will eventually be implemented and used by everyone else.

Task 1: Prototype Extension

You will extend the existing implementation from M2 to accommodate the following feature:

Feature: Test result reporting: Once a testing appointment has been fulfilled, your system should provide the user (i.e., a patient) with an ability to view the result of their test (Positive, Negative, or Pending). In reality, the test samples collected from the patients would be sent to a medical laboratory for processing and the results would be sent back after a few days. For the purpose of this milestone, you should build **test double components** that stand in for an external Testing Service and Central Database Service (which will eventually be built by the teams, as discussed below). The test double components should **randomly** produce, for each patient who has completed a testing appointment, an output that represents the hypothetical test result. Note that once a test result has been generated for a particular patient and an appointment, the test result should remain constant (i.e., if Alice tests Positive for a test done on 03/05/2025, then it should permanently be recorded in your system as Positive). Adding this feature is likely to involve storing additional data about patients, including test results and their corresponding dates. In your system, patients should be able to view test results for all their previous tests that have been reported back already.

Staff testing: Like in M2, the course staff will directly interact with your application to test out the new feature. For the purpose of staff testing only, please configure the test double component so that test results are generated immediately when the user makes a testing appointment.



Task 2: API Design for Interoperability

This part of the milestone will involve collaboration across multiple teams.

Starting this milestone, each team will be assigned the responsibility of designing (and eventually implementing) a service that will be deployed and shared by all teams. Each service will provide functionality to support a particular set of stakeholders and tasks. There will be three shared services (one for each team), as illustrated in the diagram above:

- **Testing Service:** Provides the service for scheduling apps to request test results. Each scheduling app sends information about patients and their testing appointments, which are first stored into the Central Database. Once a health administrator enters the test results, the Testing Service updates the corresponding entries with the test results.
- **Central Database Service:** Provides storage and retrieval of information about patients across multiple scheduling apps (including their up-to-date testing history). Other apps and services will be able to access information about patients through an API that is provided by this service.
- **Data Aggregation Service:** Allows the members of the public to view various statistics related to an on-going pandemic, such as the number of known positive cases and the overall trend over a user-specified period of time. To compute the statistics, this service relies on the information provided by the Central Database Service.

API design activity (Mar 12, 2024): In this milestone, your team will design (but not yet implement) an API for the assigned service. Since these services will be accessed by multiple applications, **interoperability** is a key quality attribute to be achieved. To facilitate the process of designing APIs together, we will allocate a lecture slot on **Wednesday, Mar 12, 2024** as a design session. **Every member of your team is expected to be present at this session.**

API documentation: Your team will document the outcome of this activity as an API documentation that describes (1) the list of interface functions, (2) input and output parameters that are associated with each function, and (3) a contract that describes the pre- and post-condition of each function. In addition, your documentation should include an ontology that describes the meaning of data elements that are passed as parameters, as well as the relationships between those elements.

Tips for Cross-team Collaboration

To ensure seamless integration of your service API with the services and scheduling apps developed by others, **cross-team communication will be crucial**. You can use Slack for cross-team communication and maintain a shared Google Docs document outlining all APIs and assumptions. We also recommend that each team picks one representative who is responsible for being an “**interface person**” of the team and responsible for answering questions from other teams. Tell the other teams who your interface person is so that they know who to contact. In your team contract, allocate a member to be responsible for cross-team communication tasks.

Remember that other teams might use your API and might need to write test double components for your API in this milestone. Please be courteous to them and **finalize your API design & documentation early** to give them time to design their system around your API.

Task 3: Design Reflections

In the process of extending your prototype with the new feature, it is likely that the design of your application will undergo some changes. Depending on how you designed your system in M1 & M2, these changes may be straightforward to make (e.g., by adding new components with no or minor changes to the existing ones) or disruptive (i.e., involves significantly changing the existing components). For this task, prepare a report that reflects on the design of your system has evolved to accommodate new features and possible future changes. The report should:

1. Describe the changes to the design of your system from M2. Include an updated component diagram that highlights those changes.
2. Describe design decisions that you have made for the API of the shared service in Task 2 and justify how those decisions support service interoperability.

Deliverables

Submit a report as a single PDF file to Gradescope that covers the following items in clearly labeled sections (ideally, each section should start on a new page). **Please correctly map the pages in the PDF to the corresponding sections.**

1. **Deployment (1 paragraph):** Include the URL for accessing the main frontend of your application.
2. **Testing report (1 pg max):** A description of how you used test doubles to test the new feature, including links to the test artifacts (e.g., parts of the source code that contains the test double).
3. **Shared service API documentation (2 pg max):** A documentation of the API for the shared service that has been assigned to your team, based on Task 2.
4. **Design reflection report (2 pg max):** A document reflecting on the (re)design decisions that you have made, as described above in Task 3.
5. **Team contract (1 pg max):** A documentation of (i) the division of development tasks among team members (including the interface person) and (ii) intermediate milestones, each with a target date and goals achieved.

Grading

This assignment is out of **110** points. For full points, we expect:

- **(30 pt)** A functional web application that extends the prototype from M2 with the new feature.
- **(20 pt)** A documentation of your approach to implementing the test double for the new feature & corresponding automated tests.
- **(30 pt)** A documentation of an interface for the shared service that has been assigned to your team, including (1) a contract for each function describing its expected behavior and **(20 pt)** and (2) an ontology for data elements that are passed as input and output parameters **(10 pt)**.
- **(20 pt)** A design reflection report with (i) a discussion of how the system design from M2 has been changed to support the new feature, with an updated component diagram **(10 pt)** and (ii) a discussion of design decisions for the shared service interface and justifications for those decisions with respect to interoperability **(10 pt)**.
- **(10 pt)** A team contract that describes (i) the division of tasks among team members and (ii) a set of intermediate milestones, their target date, and expected goals.
- **(5 pt)** Bonus social points (same as the previous milestones).