Milestone 3: Design for Changeability & Interoperability

Released: Friday, Mar 1, 2024
Due: 11:59 pm Friday, Mar 22, 2024

Learning Objectives

- Redesign an existing system to improve its quality attributes.
- Extend an existing system to support additional features.
- Address issues of testability of external service dependencies using test double components and automated tests.
- Design an interface for interoperability with other systems.
- Design large-scale software systems 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. In the process, you will be asked to possibly redesign parts of your system for improved changeability and interoperability.

Task 1: Prototype Extension

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

Feature #1 - 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 a **test double component** that stands in for an external testing service. This test double component 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 03/05/2024, 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.

Feature #2 - Quarantine recommendations: Once the test result for a patient becomes available, your system should generate a quarantine recommendation (“No quarantine required” or “Quarantine for X number of days”) for the patient to be able to view. In practice, the

quarantine decision is made based on a number of factors that are determined by the health policy maker, such as the patient's test result, age, symptoms, vaccination history, and the density of infections within a geographical region. Like with the previous feature, for the purpose of this milestone, you should build a **test double component** that stands in for an external quarantine recommendation service. This test double component should take a patient's test result and recommend a quarantine if it is Positive. Adding this feature is likely to involve storing additional data about patients, including the range of dates recommended for a quarantine.

Redesign for changeability: As you extend your prototype with these two features, redesign the system to be ready for possible future changes. For example, consider the possibility that the system may one day be used not only for scheduling and processing testing appointments, but appointments for vaccination as well, or for supporting different types of stakeholders beside patients (more on this soon in Task 2). Look for opportunities to apply the principles and methods for changeability that you've learned in class. Evaluate the changeability of your redesign by analyzing the impact of possible changes on your system. In Task 3, you will be asked to include a discussion of your redesign and its changeability as part of the design report.

Staff testing: Like in M2, the course staff will directly interact with your application to test out the two new features. For the purpose of staff testing only, please configure the test double components so that test results and quarantine recommendations 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 type of stakeholders. There will be four shared services in total (one for each team), as follows:

- **Central patient database service:** Provides storage and retrieval of information about patients across multiple scheduling apps (including their latest test result, quarantine status, and vaccination history). Other apps and services will be able to upload and access information about patients through an API that is provided by this service.
- **Policy maker service**: Allows a policy maker (e.g., a government official) to modify the policy that is used to determine whether a patient should undergo a quarantine process. When deployed, this service will be used by the scheduling apps to determine whether a patient should be given a quarantine recommendation (and for how long).
- **Healthcare administrator service**: Supports tasks that are performed by a healthcare administrator to enter patients' test results and other medical information (such as pre-existing conditions that increase the risk of serious harm to a patient).
- **Public information service**: Allows the members of the public to view various statistics related to the pandemic, such as the number of known positive cases and the overall

trend in the past months. This service will be queried by the scheduling apps to access and display these statistics to the user at their request.

More details about these services will be provided in the design session on Monday, Mar 11 (see below).

API design activity (Mar 11, 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 **Monday, Mar 11, 2024** as a design session. **Every member of your team is expected to be present at this session**. Your team will document the outcome of this activity as an API documentation that describes (1) the list of interface functions (as HTTP request) and (2) parameters that are associated with each function, (3) semantics of the request, parameters, and response.

Tips for Cross-team Collaboration

To ensure that the API of your service will integrate with the APIs of other services and systems developed by other teams, **you should communicate with other teams** to avoid future integration issues and to make sure that the assumptions you are making about their services and systems are indeed correct. 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 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 two new features, 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. In particular, 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. Discuss at least **two possible changes** (e.g., a new feature) and how your updated design would support making those changes with small impact on the rest of the system.

3. 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 double components to test the two new features, including links to the test artifacts (e.g., parts of the source code that contains the test doubles).
3. **Shared service API documentation (1 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** (**3 pg max**): A document reflecting the (re)design decisions that you have made, as described above in Task 3.
5. **QA testing results (1 pg max)**: The results of your quality attribute test from milestone 2. If there are differences, describe possible reasons for the differences (for example, design decisions you made that might have improved / worsened the quality attribute).
6. **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 **120** points. For full points, we expect:

- **(30 pt)** A functional web application that extends the prototype from M2 with two additional features: test result reporting and quarantine recommendations.
- (**20 pt**) A documentation of your approach for implementing test doubles for the two features & corresponding automated tests.
- **(20 pt)** A documentation of an interface for the shared service that has been assigned to your team.
- **(30 pt)** A design reflection report with (i) a discussion of how the system design from M2 has been changed to support the two new features, with an updated component diagram (**10 pt**), (ii) a discussion of how the updated design support possible future changes with (**10 pt**), and (iii) a discussion of design decisions for the shared service interface and justifications for those decisions with respect to interoperability (**10 pt**).
- **(10 pt)** (i) The new results for your quality attribute tests from milestone 2 (if the changes made it impossible to run the tests and it is too much effort to fix your tests, “the tests are broken” is also an acceptable answer), (ii) possible explanations for the differences, if any (you do not have to identify the actual root cause, just describe plausible reasons connected to your changes)

- **(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).