Quality requirement scenarios

Design-time attributes

Modifiability

Scenario: Add a new conflict rule

Source	Timetable Committee
Stimulus	Defines new conflict rule
Artifact	Conflict Detector
Response	New conflict rule is added
Measurement	1 week development time

Scenario: A new type of scheduling conflict (exam overlap).
Solution: No change to architecture necessary because conflict detection is isolated in a dedicated component.

Testability

Scenario: Testing of new course statistics

Source	System tester
Stimulus	Tests new course statistics
Artifact	Statistics
Response	Prepare dummy data and expected results and verify correctness each new statistic
Measurement	100% coverage of new statistics in 1 week of testing

Scenario: System tester tests new course statistics.
Solution: Setup a new deployment diagram for testing.

Scenario: Testing of scheduling

Source	System Tester
Stimulus	Runs tests
Artifact	Scheduling Service
Response	Uses dummy data instead of database
Measurement	100% repeatable test results

Scenario: Scheduling and conflict logic are tested without using UI or database, tests are deterministic and fast.
Solution: No need to modify architecture as there as a separation between business logic and infrastructure.

Run-time attributes

Performance

Scenario: Mass schedule viewing

Source	Students
Stimulus	Thousands of requests to open schedule page at a 1s time frame
Artifact	Schedule Viewing Web App
Response	Sends read requests
Measurement	Latency up to 3 seconds

Scenario: When the timetable is published, thousands of students request schedules at the same time. The Schedule Viewer Service is replicated and stateless, so requests are distributed evenly.
Solution: Architecture update is not needed as horizontal scaling and load balancing are already implemented.

Scenario: Repeated statistics queries

Source	Statistics Web App
Stimulus	Sends request
Artifact	Statistics Web App
Response	Forwards statistics requests
Measurement	90% avg. speedup of repeated statistics queries

Scenario: Managers (a.k.a. someone who reviews statistics, teacher workloads, room utilization…) repeatedly request identical statistics (teacher workload, room utilization, historical schedules) and the Statistics Service returns the requested statistics

Solutions: Add Statistics Cache component between Statistics Service and Statistics DB. Enhance Statistics Service functionality to preserve one-hour-old request to database

Availability

Scenario: Notification Service Availability

Source	Scheduling Service
Stimulus	Unable to use for notification
Artifact	Notification Service
Response	Recover
Measurement	No downtime

Scenario: The system is using a Load Balancer when sending notifications to distribute load. When some container returns error while sending notification, we log that and retries it (the balancer will choose a different component).

Solution: Cloning of Notification Service and adding Load Balancer to prevent downtime.

Scalability

Scenario: Add new statistics category

Source	System developer
Stimulus	Need of adding a new statistics category
Artifact	Statistics services
Response	New statistics category is added to the statistics service
Measurement	The database and service is updated without a hitch

Scenario: The system developer needs to add a new statistics category.
Solution: Architecture is able to handle this, everything goes through the statistics service, which requests the data needed from the course service.

Scenario: Growing number of users for schedule viewing

Source	Schedule Viewing Web App Users
Stimulus	Increased current avg. user count
Artifact	Schedule Viewer Service
Response	Scale up
Measurement	No effect on performance and availability

Scenario: Growing number of users for schedule viewing.
Solution: Add Load Balancer from Statistics and Scheduling Service to Course Service in Production Deployment diagram.

Security

Scenario: Unauthorized course modification

Source	Student
Stimulus	Calls course registration API
Artifact	Course Service
Response	Validate token
Measurement	http 403, prevent unauthorized action

Scenario: A student attempts to access a teacher-only API. The request is rejected before any business logic executes. Response is http 403. With individual requests from web applications, adds additional checks in APIs it passes through to request validation

Solution: The architecture must be updated to bridge the Auth Service with the Scheduler API. This ensures that specific API calls related to teacher schedules and administrative tasks are strictly protected against unauthorized access.

Scenario: Stolen access token used by attacker

Source	Attacker
Stimulus	Reuses stolen token
Artifact	Backend Service
Response	Verifies token
Measurement	Prevent unauthorized action

Scenario: An attacker tries to reuse a stolen authentication token. Add timeout and cancellation for the token after logging out. The token is rejected and the attempt is logged to audit database.

Solutions: No change in the architecture, only send a request for token cancelment.