

# 17-723: Designing Large-scale Software Systems

Recitation: Attribute-driven design  
Feb 2, 2024

# Last Lecture

This is called **Attribute-Driven Design (ADD)**

## How to Make Design Decisions in a Large Design Space?

**Iteratively** improve your design:

1. **Select** a quality attribute to improve (iteration goal)
2. **Chose** one or more parts of the system to refine
3. **Find & sketch** candidate solution & describe design decisions
4. **Analyze** candidate solution for iteration goal and other quality attributes
5. **Iterate** if necessary

# This lecture:

Demonstration of Attribute Driven Design (ADD)

Designing an online Learning Management System (LMS) eg.  
Canvas, Blackboard



# Quality Attributes for an online LMS



# Quality Attributes for an online LMS

- Performance: responsiveness and fast loading
- Security: prevention of unauthorized access
- Scalability: handle growing number of courses and students
- Usability: easy navigation and access of learning materials
- Reliability: providing uninterrupted access
- others

# Step 1: Select a Quality Attribute

- **Performance: average response time for material access**
- Security: only students can access their portal
- Scalability: response time is not affected by increasing the number of courses and students
- Usability: satisfactory student experience accessing materials without prior training
- Reliability: providing uninterrupted access under normal operating conditions

Iteration goal:....?

# Step 1: Select a Quality Attribute

- **Performance: average response time for material access**
- Security: only students can access their portal
- Scalability: response time is not affected by increasing the number of courses and students
- Usability: satisfactory student experience accessing materials without prior training
- Reliability: providing uninterrupted access under normal operating conditions

Iteration goal: Reduce the average response time for users

## Step 2: Choose a part of the system to refine

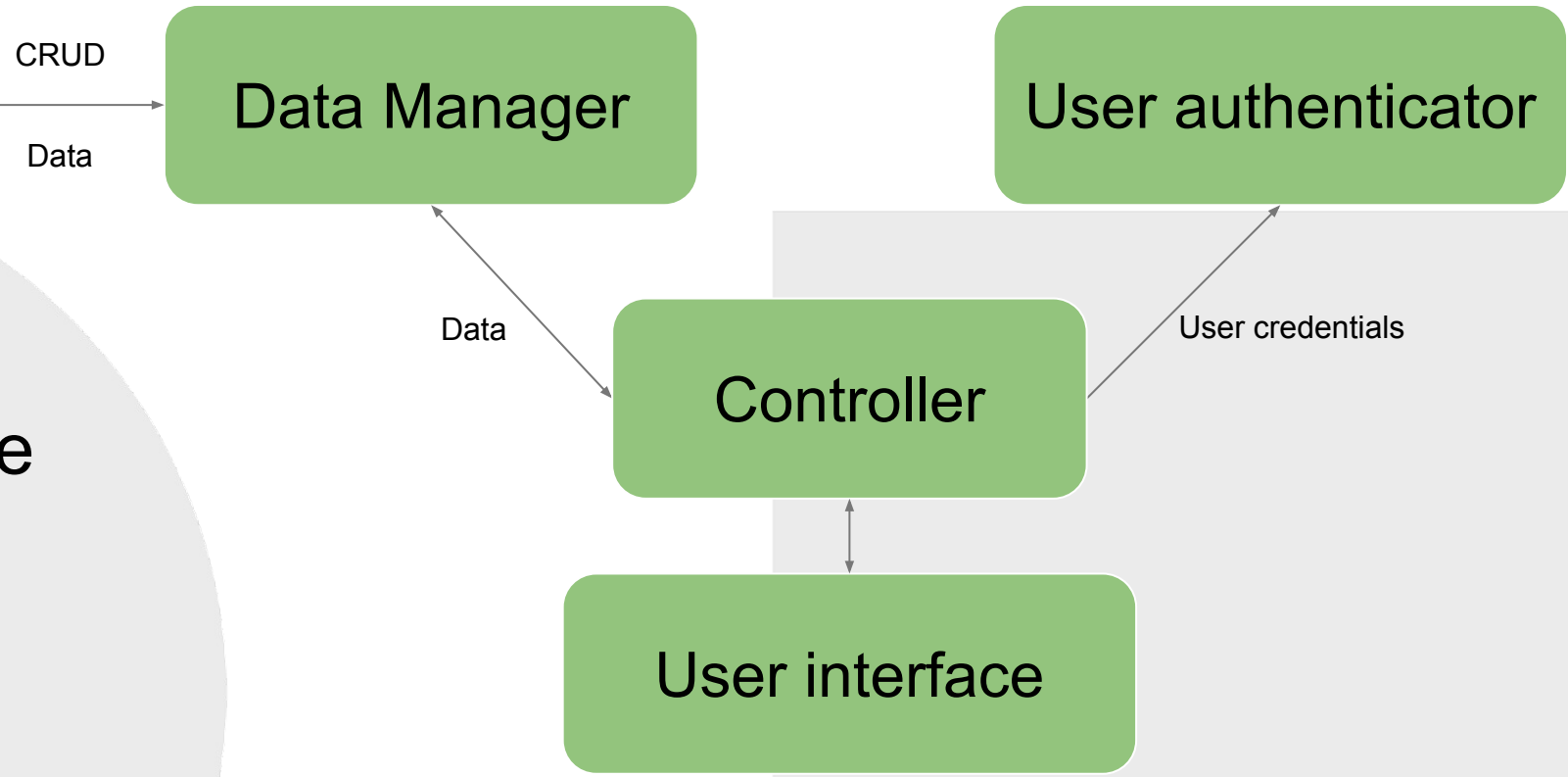
### System components:

- User Interface
- Data Manager
- Database
- User Authentication and Authorization





Database



## Step 2: Choose a part of the system to refine

### System components:

- User Interface
- **Data Manager**
- Database
- User Authentication and Authorization

## Step 3: Find candidate solution and describe design decisions

1. Query Optimization: Analyze slow queries and optimize them
2. Caching Mechanism: Choose caching strategy based on data access patterns and requirements
3. Load Balancing: Distribute incoming traffic and prevent bottlenecks

## Step 4: Analyze Candidate Solution for iteration goals and other QAs

Which other QAs might be affected by our candidate solution?

- ✓ **Performance**
- Security
- Scalability
- Usability
- Reliability

## Step 4: Analyze Candidate Solution for iteration goals and other QAs

Which other QAs might be affected by our candidate solution?

- ✓ **Performance**
- ❑ Security
- ✓ **Scalability**
- ❑ Usability
- ✓ **Reliability**

# Step 4: Analyze Candidate Solution for iteration goals and other QAs

Candidate Solution	Scalability	Reliability
Query Optimisation	Over optimization may limit scalability	Query times ↓ -> database contention ↓
Caching mechanism	Repeated data queries ↓ -> more users can be served	Stale data may reduce reliability
Load Balancing	Multiple servers can serve larger number of users	Data consistency issues may reduce reliability

## Step 5: Iterate if necessary

- Performance testing: Simulate scenarios to identify unforeseen situations
- User feedback: Gather user feedback in terms of system responsiveness

Based on results -> iterate!

# Exercise: ADD for another QA in our system!

QA: security!

- Performance: average response time for material access
- Security: only students can access their portal
- Scalability: response time is not affected by increasing the number of courses and students
- Usability: satisfactory student experience accessing materials without prior training
- Reliability: providing uninterrupted access under normal operating conditions



# Step 1: Select a Quality Attribute

- Performance: average response time for material access
- **Security: only students can access their portal**
- Scalability: response time is not affected by increasing the number of courses and students
- Usability: satisfactory student experience accessing materials without prior training
- Reliability: providing uninterrupted access under normal operating conditions

Iteration goal:....?

# Step 1: Select a Quality Attribute

- Performance: average response time for material access
- **Security: only students can access their portal**
- Scalability: response time is not affected by increasing the number of courses and students
- Usability: satisfactory student experience accessing materials without prior training
- Reliability: providing uninterrupted access under normal operating conditions

Iteration goal: Provide robust student authentication capabilities

## Step 2: Choose a part of the system to refine

### System components:

- User Interface
- Data Manager
- Database
- User Authentication and Authorization

## Step 2: Choose a part of the system to refine

### System components:

- User Interface
- Data Manager
- Database
- **User Authentication and Authorization**

## Step 3: Find candidate solution and describe design decisions

1. Multi Factor Authentication: Choose suitable MFA method (SMS codes, authenticator apps etc.) to enhance user security
2. Biometric Authentication: Use facial recognition/fingerprint recognition
3. SSO Authentication: Use institution authentication portal

## Step 4: Analyze Candidate Solution for iteration goals and other QAs

Which other QAs might be affected by our candidate solution?

- Performance
- Security**
- Scalability
- Usability
- Reliability

## Step 4: Analyze Candidate Solution for iteration goals and other QAs

Which other QAs might be affected by our candidate solution?

- Performance
- Security**
- Scalability**
- Usability
- Reliability**

## Step 4: Analyze Candidate Solution for iteration goals and other QAs

Candidate Solution	Scalability	Reliability
Multi Factor Auth	As number of users grow, generating codes/prompts might decrease scalability	if any one authentication is down, reliability decreases
Biometric Auth	Since authentication is done on user device, scalability remains tractable	False positives and false negatives might reduce reliability
SSO Auth	SSO systems are capable of handling large user requests so system scalability is un affected	SSO introduce a single point of access failure when downtime occurs hence reliability decreases



## Step 5: Iterate if necessary

- Performance testing: Simulate scenarios to identify unforeseen situations
- User feedback: Gather user feedback in terms of system responsiveness

## Step 5: Iterate if necessary

- **Performance testing: Simulate scenarios to identify unforeseen situations**
- User feedback: Gather user feedback in terms of system responsiveness

Exercise: Simulate some possible scenarios

Based on results -> iterate!

# Summary

## Demonstration of Attribute Driven Design (ADD)

- Selection of QA
- System component identification
- Propose Candidate solutions
- Analyze them with respect to other QAs
- Iterate based on feedback/scenarios