SOFTENG 325 Assignment 2

Name: Aiden Burgess

UPI: abur970

Quality Attribute Scenarios

1. Availability

Portion of Scenario	Concrete Value
Source	Random event
Stimulus	Single processor failure
Artifact	Server
Environment	Normal working conditions
Response	Notify operator
Response Measure	No loss of service (100% availability)

2. Security

Portion of Scenario	Concrete Value
Source	Unknown external source
Stimulus	Unauthorized attempt to view exam information
Artifact	Website
Environment	Online website
Response	Refuse access to information
Response Measure	Detect unauthorized access after 3 requests

3. Usability

Portion of Scenario	Concrete Value
Source	Student
Stimulus	Attempting login
Artifact	Login page of website
Environment	Online website at runtime
Response	Login page should make it clear where to input username and password
Response Measure	#Successful Logins/#Attempts is 99%

4. Performance

Portion of Scenario	Concrete Value
Source	Student
Stimulus	Watching live session
Artifact	Server and Website
Environment	Normal working conditions
Response	Stream video and audio to student
Response Measure	Average latency of connection is less than 300ms

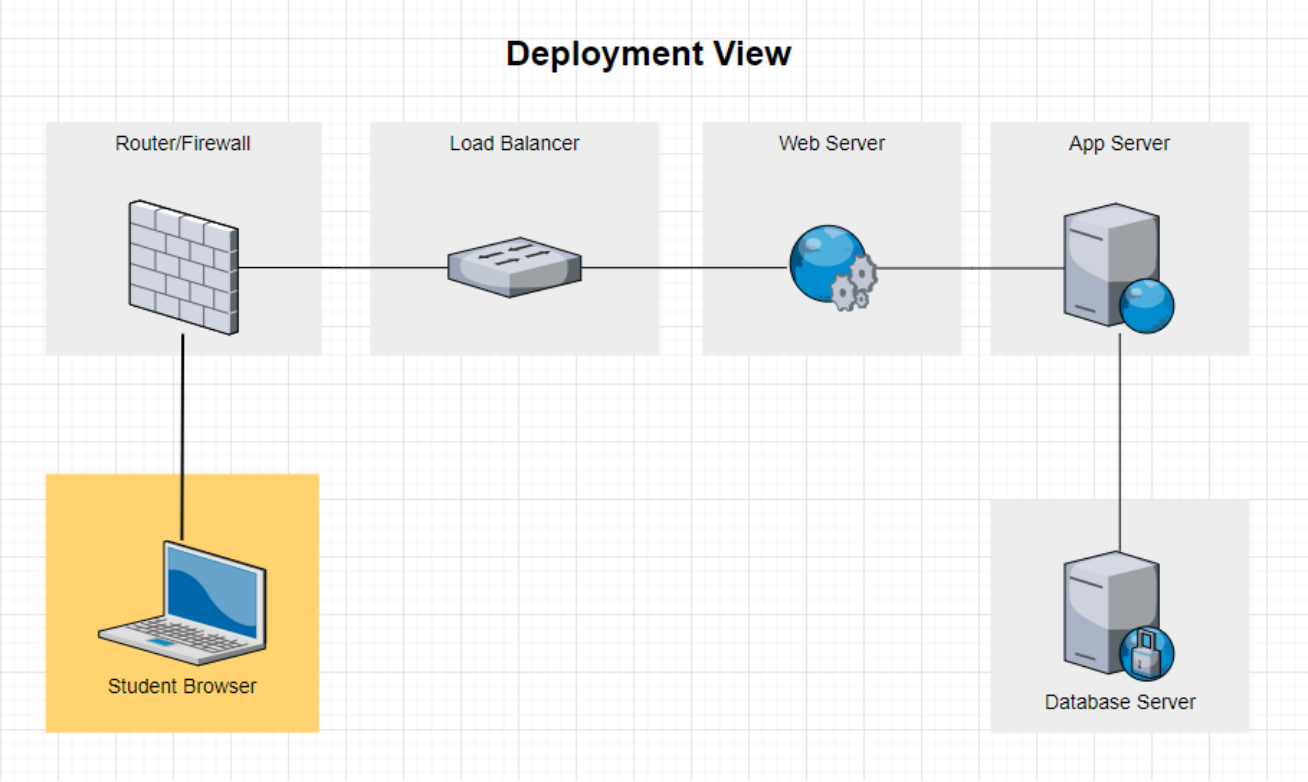
5. Interoperability

Portion of Scenario	Concrete Value
Source	UoA login service
Stimulus	User attempting login with university account

Portion of Scenario	Concrete Value
Artifact	Login system of server
Environment	System wishing to interoperate known prior to runtime
Response	Request is appropriately accepted and user is logged in
Response Measure	99% successful logins with UoA login service

Architecture Design

Deployment View

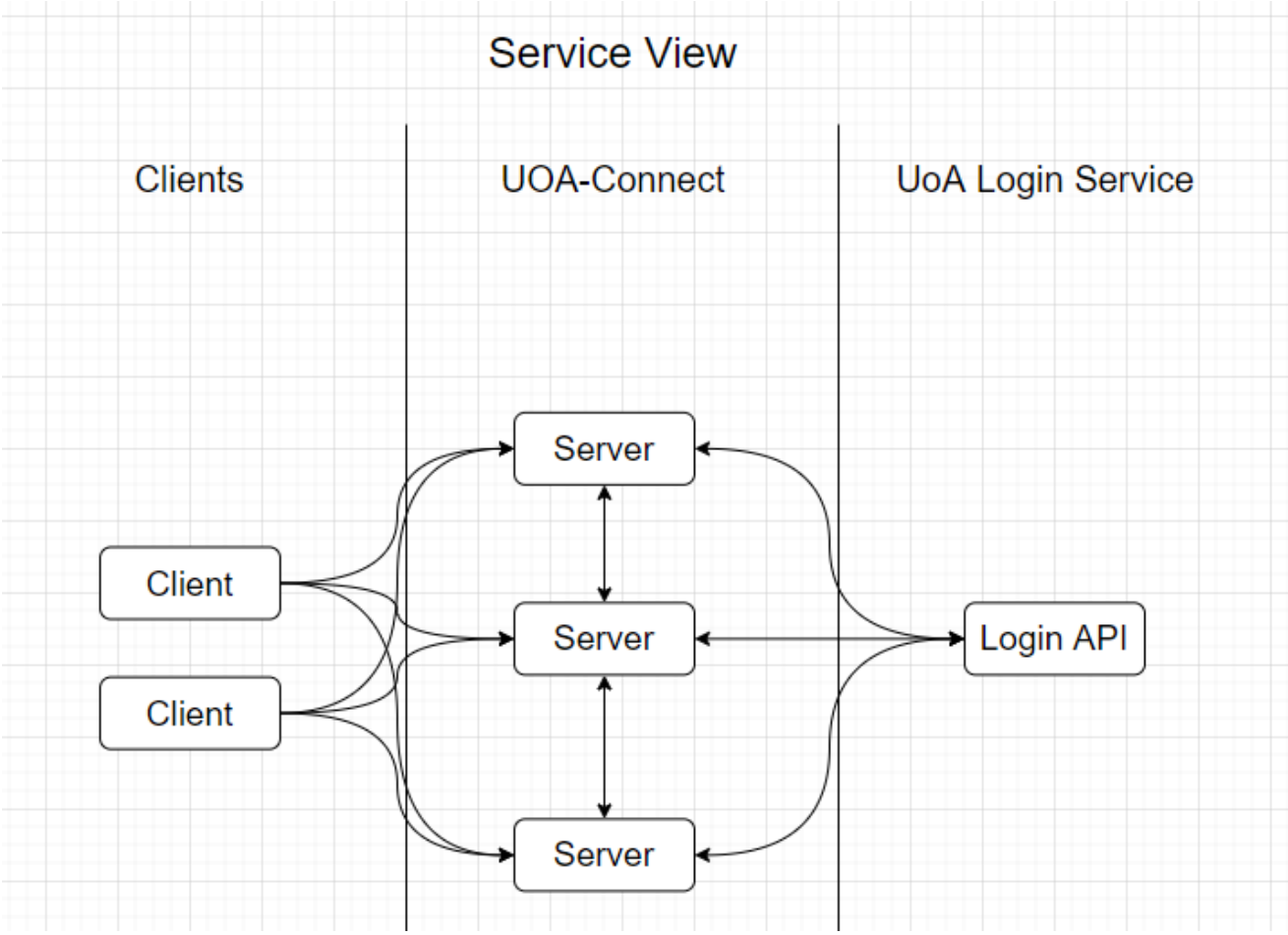


In order to meet the **security** quality attribute scenario, a way of detecting and blocking access to unauthorized external sources is needed. Therefore, the first point of contact for any source attempting to access secured information is a firewall. The trade-off with using this tactic comes at the cost of performance, which is slowed as the connection must pass through these checks.

A web server by itself is not enough to log information about login request success rates, which is required to fulfil the **usability** requirement. The tactic used to supply this functionality is the introduction of an application server which logs information about the web server and serves as a way to debug any usability issues with the entire website (including the login page). Unfortunately, this logging takes time and reduces performance.

In order to achieve the **performance** metric specified, the deployment view shows that each component of the backend system is separated into its own server. This allows each server to be allocated more resources and optimised, which improves the speed of processing and retrieval, thereby helping to achieve the 300ms latency scenario.

Service View



The **availability** quality attribute specifies that there should be no loss of service even if a server processor fails randomly. To achieve this measure, each client is connected to multiple servers, which prevents any single server failure resulting in loss of availability. Clearly, this comes at a large performance cost, so there must be a tradeoff between how many servers each client is connected to and how much protection our system requires from processor failures.

Another tactic used in the service view is the connection to the login service from all servers. Rather than a single point of failure within the UOA-Connect service, each server communicates with the login service API individually. This helps achieve the **interoperability** measure as the communication between the login service is more likely to be valid.