

*Cloud Networking and Service Provisioning. ENCS 691K. Fall 2015*

*Dr. Roch H. Glitho*

*By*

*Ali Sangari (Student ID: 26816304)*

TABLE OF CONTENTS

PART 1 – PROJECT SCOPE AND PLAN 7

[1 Introduction 8](#_Toc374312635)

[2 Project Description 8](#_Toc374312636)

[3 Goals and Constraints 9](#_Toc374312637)

[3.1 Functional Requirements 9](#_Toc374312638)

[3.1.1 Landing Page 9](#_Toc374312639)

[3.1.2 Login 9](#_Toc374312640)

[3.1.3 User Console 9](#_Toc374312641)

[3.1.4 Schedule Generator (for student user) 10](#_Toc374312642)

[3.1.5 Schedule Generator (for program director user) 10](#_Toc374312643)

[3.1.6 Registration 11](#_Toc374312644)

[3.1.7 Program Director Console 11](#_Toc374312645)

[3.1.8 Logout 11](#_Toc374312646)

[3.1.9 Use Case Model 12](#_Toc374312647)

[3.1.10 Use Cases 13](#_Toc374312648)

[3.2 Domain Model 38](#_Toc374312649)

[3.2.1 User 39](#_Toc374312650)

[3.2.2 Student 39](#_Toc374312651)

[3.2.3 Program Director 39](#_Toc374312652)

[3.2.4 Schedule Generator 39](#_Toc374312653)

[3.2.5 Preference 39](#_Toc374312654)

[3.2.6 Student Record 39](#_Toc374312655)

[3.2.7 Master Schedule 40](#_Toc374312656)

[3.2.8 Section 40](#_Toc374312657)

[3.2.9 Course 40](#_Toc374312658)

[3.2.10 Course Sequence 40](#_Toc374312659)

[3.2.11 Exception Request 40](#_Toc374312660)

[3.3 Constraints and Qualities 40](#_Toc374312661)

[3.3.1 Security 40](#_Toc374312662)

[3.3.2 Efficiency 41](#_Toc374312663)

[3.3.3 Maintainability 41](#_Toc374312664)

[3.3.4 Adaptability 41](#_Toc374312665)

[3.3.5 Usability 41](#_Toc374312666)

[3.3.6 Open Source 41](#_Toc374312667)

[3.3.7 Compatibility 41](#_Toc374312668)

[4 Resource Evaluation 42](#_Toc374312669)

[4.1 Human Resources 42](#_Toc374312670)

[4.2 Technical Resources 44](#_Toc374312671)

[5 Scoping 45](#_Toc374312672)

[6 Solution Sketch 46](#_Toc374312673)

[6.1 Architecture 46](#_Toc374312674)

[6.2 Technologies in Use 47](#_Toc374312675)

[6.2.1 Programing language 47](#_Toc374312676)

[6.2.2 Framework 48](#_Toc374312677)

[6.2.3 UI Design 48](#_Toc374312678)

[6.2.4 Database 48](#_Toc374312679)

[6.2.5 Version Control 49](#_Toc374312680)

[6.2.6 Cloud Hosting Platform 49](#_Toc374312681)

[6.2.7 Team Collaboration 49](#_Toc374312682)

[6.2.8 UML and Diagram Tools 49](#_Toc374312683)

[7 Plan 50](#_Toc374312684)

[7.1 Activities & Assignment 50](#_Toc374312685)

[7.2 Artifacts 53](#_Toc374312686)

[7.3 Project Estimates 54](#_Toc374312687)

[7.4 Schedule 54](#_Toc374312688)

[7.5 Risk 55](#_Toc374312689)

PART 2 - Project design 56

[8 Introduction to Design 57](#_Toc374312690)

[9 Architectural Design 57](#_Toc374312691)

[9.1 Architecture Diagram 57](#_Toc374312692)

[9.1.1 Logical View 57](#_Toc374312693)

[9.1.2 Physical View 59](#_Toc374312694)

[9.1.3 Development *View* 61](#_Toc374312695)

[9.1.4 Process View 64](#_Toc374312696)

[9.1.5 Scenarios 70](#_Toc374312698)

[9.2 Subsystem Interfaces Specifications 72](#_Toc374312699)

[9.2.1 ICourseManagement 72](#_Toc374312700)

[9.2.2 IUserManagement 76](#_Toc374312701)

[9.2.3 IStudentPreference 77](#_Toc374312702)

[10 Detailed Design 78](#_Toc374312703)

[10.1 Subsystems 78](#_Toc374312704)

[10.1.1 Detailed Design Diagram 78](#_Toc374312705)

[10.2 Unit Descriptions 81](#_Toc374312706)

[11 Dynamic Design Scenarios 86](#_Toc374312707)

[11.1 Use Case 8 – Drop Course 86](#_Toc374312708)

[11.1.1 Fully Dressed Use Case 86](#_Toc374312709)

[11.1.2 System sequence diagram 87](#_Toc374312710)

[11.1.3 Operating Contracts 88](#_Toc374312711)

[11.2 Use Case 24 – Record Waiting List 91](#_Toc374312712)

[11.2.1 Fully Dressed Use Case 91](#_Toc374312713)

[11.2.2 System sequence diagram 93](#_Toc374312714)

[11.2.3 Operating Contracts 94](#_Toc374312715)

[12 Rapid Prototyping and Risk 97](#_Toc374312716)

[12.1 Risk and Effect of Prototyping 97](#_Toc374312717)

[12.2 User Interface Mockup 98](#_Toc374312718)

[12.2.1 Login interface view mock-up 98](#_Toc374312719)

[12.2.2 User console menu mock-up 99](#_Toc374312720)

[12.2.3 Student record view mock-up 100](#_Toc374312721)

[12.2.4 Generate schedule view mock-up 101](#_Toc374312722)

[13 Cost Estimation 101](#_Toc374312723)

PART 3 - PROJECT TESTING AND DELIVERY 103

[14 Introduction 104](#_Toc374312724)

[15 Testing Report 105](#_Toc374312725)

[15.1 Test Coverage 105](#_Toc374312726)

[15.1.1 Tested Items 106](#_Toc374312727)

[15.1.2 Untested Items 109](#_Toc374312728)

[15.2 Test Cases and Test Results 109](#_Toc374312729)

[15.2.1 Unit Testing 109](#_Toc374312730)

[15.2.2 Requirements Testing 115](#_Toc374312731)

[15.2.3 Stress Testing 124](#_Toc374312732)

[15.2.4 Security Testing 126](#_Toc374312733)

[16 System Delivery 128](#_Toc374312734)

[16.1 Installation Manual 128](#_Toc374312735)

[16.1.1 Deployment Environment 128](#_Toc374312736)

[16.1.2 Software Packages 128](#_Toc374312737)

[16.1.3 Software Installation in windows OS 129](#_Toc374312738)

[16.1.4 Software installation in Linux based OS 130](#_Toc374312739)

[16.1.5 Downloading Source from Git Repository 130](#_Toc374312740)

[16.1.6 Deploying the Application 131](#_Toc374312741)

[16.1.7 Configuring MySQL database 132](#_Toc374312742)

[16.2 User Manual 134](#_Toc374312743)

[16.2.1 Introduction 134](#_Toc374312744)

[16.2.2 Login Screen 134](#_Toc374312745)

[16.2.3 Main Screen 135](#_Toc374312746)

[16.2.4 Logout 150](#_Toc374312747)

[17 Final Cost Estimate 151](#_Toc374312748)

[18 References 153](#_Toc374312749)

[Appendix 1. Project Schedule – Gantt Chart 154](#_Toc374312750)

[Appendix 2. Extraction script 156](#_Toc374312751)

# Project Overview

Aim of this project was to develop an auction system to be deployed on a cloud platform.

Required functionalities of the auction system are as follows,

## User Registeration

Any new user must b able to register and use the service. A user record includes *Name*, *username*, and *password*.

## User login

To be able to use the system a visitor must have a valid record in the system and also be logged in to the system.

## User Account Termination

A user must be able to terminate his/her account given that he/she is not currently engaged in an auction. Engaging in an auction is defined as either having a product in the action that other user are bidding for or currently having the highest bid on at least one product.

## Addition of New Products

Once a user is logged in, he/she must be able to add new products to his/her profile.Fields that are included in a Product object include, *Name*, and *Price*.

## Placing a Product in Auction

At any time after a product has been added to the user profile it can be placed in auction, given that it isn’t already in auction.

A product remains in auction for a priod of x minutes (currently configured for on minute),to receive bids. At the end of one minute the product will be sold to the highest bidder, hence will be removed from its’s original owners’ account to the new owner.

In the event that a product does not receive any bids during the allocated time slot, it will be renewed for another block of time, until it is sold.

## Bidding on Products that are placed in auction

Once a product is in auction, it can receive bids from any user who is logged in to the system. Every bid automatically raises the price by 5 units. The initial bid will match the product’s base price to form the highest current bid.

# Provisioned Cloud Services and Service Provides

In this project two cloud solution providers that I provisioned and deploy the application on were Amazon Web Services (AWS) and Microsoft Azure.

## Amazun Web Services

Amazon provides a FreeTier service which includes all of its services free for one year.

These services includes Elastic Computing 2 (EC2) which is a Virtual Machine (VM) with different Operating Systems to choose from. Other services include Simple Storage System (S3), Amazon DynamoDB, Amazon IoT, RDS, etc.

During this project, I tested a few of the above services, but the main service that was used to deploy and run the code in the end was Amazon EC2.

## Microsoft Azure

Microsoft Azure provides free service for users of DreamSpark. DreamSpark is service or account that Microsoft has built for students many Mocrosoft products and services during their studies for free. The management console of Azure is completely different from that of AWS. It looks more appealing and less overwhelming. All services are pre-pakaged, menu driven, and there is lesser amount of configuration required to run a service. On the other handazure provides less freebies for development outside of Microsoft’s development eco-system.

# Challenges in Setting up Amazon EC2

The following are some of the challenges during the set up of the system to be able to run my project.

## Setting up Security

Figuring out how to configure security to be able to access my application remotely (through a browser) took the most time. I realize many people had a similar issue the first time they tried to use EC2. The process is simple enough, but jut not very obvius to the first time user.

## Remote Connection to the VM

The OS image that I installed on the VM was Windows Server so my remote connection to VM was thrugh Remote Desktop. Amazon generates a key file that helps discover the password for the administrator account.

## Installation of Required Application

After connecting to the VM instance, I needed to configure the system to run my application. Since my code was written in Java, I installed and configured Java 1.8 and Tomcat 9. No surprises here!

## Firewall Configuration

A bit of Firewall configuration was needed to allow incomming and outgoing requests and responses on port 8080 (default port set on tomcat) that I wanted to use to access the application.

# Challenges in Setting up Microsoft Azure

The free service has many limitations compared to Amazon Web Services. These limitations include application containers/servers, databases (SQL and NoSQL). Understandabily, Azure, based on DreamSpark, imposes fewer limitations on Microsoft Ecosystem users. Tomcat, a NoSQL database, and many other services were not considered part of DreamSpark, therefore they would become paid service. For every paid service Microsoft provides one month free trial but it is one twelvth of what Amazon offers.

# Auction Service

The project was developed in Java using Jersey framework develop this web application I took the path of providing

# Object Relationships