TEAM 7- Architecture Document
VCU Registration System

Project Overview

The VCU Registration System is a web based application that supports student and instructor workflows. Students search for courses and enroll in sections. Instructors view schedules and access class rosters. The system uses a three tier architecture deployed in AWS.

Cloud Service Provider

- Provider
- Amazon Web Services

Justification

AWS provides scalable infrastructure, flexible pricing, and strong integration between compute, networking, and database services. The free tier supports development. Auto Scaling and Load Balancing support future growth.

AWS Services Used

- EC2 for hosting the Node application server
- RDS for hosting the MySQL database
- S3 for backups and static file storage
- VPC for network isolation
- Elastic Load Balancer for traffic distribution
- CloudWatch for monitoring and logging

Application Design

Programming Language

• JavaScript

Runtime Environment

• Node.js

API Design

• RESTful API

Application Framework

• Frontend built with HTML, CSS, and JavaScript
• Backend built with Express.js

Architecture Model

• Three tier architecture

Presentation Layer

• User interface built with HTML, CSS, JavaScript

Application Layer

• Node.js with Express handles routing and business logic

Data Layer

• MySQL database hosted on Amazon RDS

Operating System and Virtual Servers

Operating System

• Amazon Linux 2023

Reason

• Optimized for AWS
• No licensing cost
• Compatible with Node and MySQL

Instance Configuration

Development

• EC2 t3.micro
• 2 vCPU
• 1 GB RAM

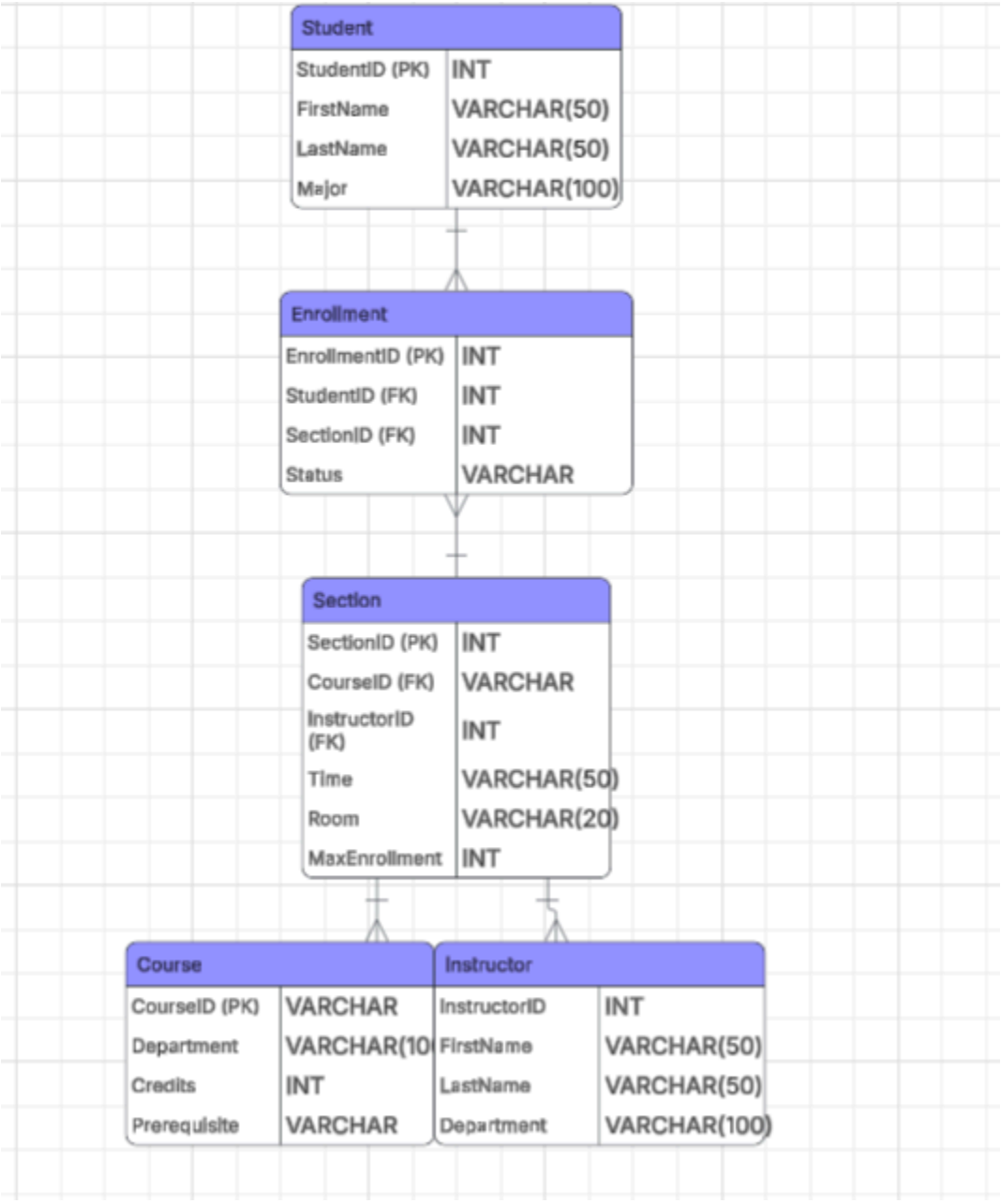
Production

• EC2 t3.small
• 2 vCPU
• 2 GB RAM

Storage

• 20 GB General Purpose SSD EBS per instance

Database Design



The database includes Users, Courses, and Registrations tables.

Users and Courses each have a primary key. Registrations contains foreign keys linking Users and Courses.

One user can register for many courses. One course can have many users. This design reduces duplication and keeps data organized.

Network Architecture and Design

VPC

• CIDR block 10.0.0.0 slash 16

Subnets

• Two public subnets in separate availability zones
• Two private subnets in separate availability zones

Public Subnets

• Host Elastic Load Balancer
• Host Bastion host for SSH access

Private Subnets

• Host EC2 application servers
• Host RDS database

High Availability

• Application servers deployed across two availability zones
• RDS configured for Multi AZ deployment

Security Groups

Load Balancer Security Group

Ingress

• Port 80 from internet
• Port 443 from internet

Egress

• Port 80 and 443 to application servers

Application Server Security Group

Ingress

• Port 80 from Load Balancer
• Port 443 from Load Balancer
• Port 22 SSH from Bastion host
• ICMP enabled for ping testing

Egress

• All outbound traffic allowed

Database Security Group

Ingress

• Port 3306 from Application Security Group only

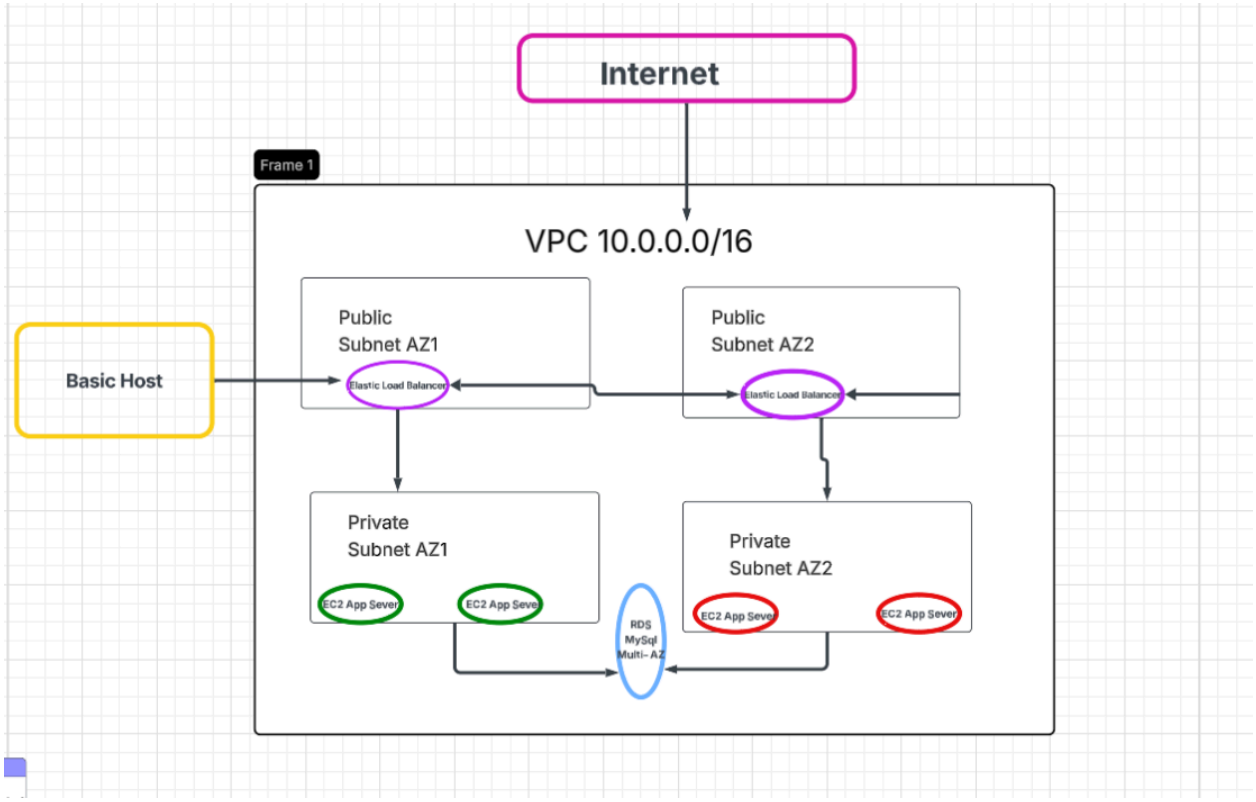
Egress

• Restricted to VPC internal traffic

Port Configuration

• HTTP uses port 80
• HTTPS uses port 443
• MySQL uses port 3306
• SSH uses port 22

Data Visualization Tool



Selected Tool

• Microsoft Power BI

Justification

• Interactive dashboards
• Strong visualization library
• Built in AI insights
• Natural language query support
• Team familiarity

- Free desktop version for development
- Scalable enterprise licensing

Use Cases

- Enrollment trend analysis
- Course demand reporting
- Instructor workload tracking

Testing and Quality Assurance

Unit Testing

- Jest used for backend testing
- Test each API endpoint independently
- Validate business logic functions

Integration Testing

- Test API connection to MySQL database
- Use Postman to validate request and response behavior

End to End Testing

- Simulate student workflow from login to enrollment
- Simulate instructor workflow from login to roster view

Code Review

- All changes submitted through pull requests
- Peer review required before merge into main branch

Monitoring

• CloudWatch monitors CPU, memory, and logs

Authentication and Authorization

System Access

• System open for development and testing

Roles

• Student
• Instructor
• Administrator

• Authentication implementation is out of scope for this phase

9. Team Contribution Summary

Tas Chowdhury

• Cloud Architect
• Application Developer
• Selected AWS services
• Designed system architecture
• Defined backend structure and API design

Omar Aziz

• Documentation Support
 • Formatted architecture document
 • Reviewed content for clarity
 • Assisted with final submission compilation

Viana Coles

• Network Engineer
 • Designed VPC layout
 • Defined public and private subnets
 • Configured security group rules

Christian Taylor

• QA Analyst
 • Developed testing strategy
 • Defined unit, integration, and end to end testing plan
 • Reviewed validation processes

Project Management Responsibilities

• Responsibilities shared across the team
 • Coordinated deadlines collectively
 • Reviewed document before submission
 • Managed GitHub updates collaboratively

Collaboration Summary

• Divided responsibilities based on strengths
 • Conducted group review before submission
 • Refined network and database sections after discussion
 • Approved final version as a team