SE 4458 Software Architecture & Design of Modern Large Scale Systems - Midterm 1

Group 1 – API Project for Mobile Provider Bill Payment System

Create an API project that will perform below requirements

In this fictitious system, clients want to do check their phone bills from their Mobile app, bank and pay via website. You are asked to create below APIs for Mobile Provider (i.e Turkcell)

Mobile Provider App	Parameters	API Response	Description
Query Bill	Subscriber No, Month	Bill Total, Paid Status	Limit call to 3 per subscriber per day
Query Bill Detailed	Subscriber No, Month	Bill Total, Bill Details	
Banking App			
Query Bill	Subscriber No	Bills NOT Paid, by month)	
Web Site			
Pay Bill	Subscriber No, Month	Payment Status (Successful, Error)	Marks bill as paid. If amount is not complete, will make sure remaining amount is saved. No real credit card payment work is needed
Admin - Add Bill	Subscriber No, Month	Transaction status	Adds a bill for a month for given subscriber
Admin - Add Bill – Batch	.csv file of Subscriber No, Months	Transaction status	Adds bills from a csv file

Other API requirements

Mobile Provider App	Authentication	Paging
Query Bill	YES	NO
Query Bill Detailed	YES	YES
Banking App		
Query Bill	YES	NO
Web Site		
Pay Bill	NO	NO
Admin - Add Bill	YES	NO

Students

ÜLKÜ BARTU SERBEST AYSİMA ADATEPE	MURAT HABİP OKAN MELİKE AYTAÇ

ELİF EMİNE GÜNAL IRMAK ARABACI DİLARA ACAR AHMET KEMAL BİLİCİLER YAĞMUR SABIRLI OZAN BÖCE	MELİSA DEMİRBAŞ PELİN DUMAN SELÇUK SUAT SAYIN DEFNE TEKYİĞİT LARA ÖZDUMAN DURU GENCAY

Group 2 – API project for a University Tuition Payment System

Create an API project that will perform below requirements

In a fictitious system, students want to do check their tuition fee status from Mobile app, bank and pay via online banking. You are asked to create below APIs for the university

University Mobile App	Parameters	API Response	Description
Query Tuition	Student No	Tuition Total, Balance	Returns tuition amount and current balance Limit call to 3 per student per day
Banking App			
Query Tuition	Student No	Tuition Total, Balance	Returns tuition amount and current balance
Pay Tuition	Student No, Term	Payment Status (Successful, Error)	Records payment for given team. If amount is not complete, will make sure remaining amount is saved. No real credit card payment work is needed
University Web Site - Admin			
Add Tuition	Student No, Term	Transaction status	Adds a tuition amount for given student term
Add Tuition – Batch	.csv files of Student No, Term	Transaction status	Adds a tuition amount from a .csv of student data
Unpaid Tuition Status	Term	List of students with unpaid tuition amounts	

Other API requirements

University Mobile App	Authentication	Paging
Query Tuition	NO	NO
Banking App		
Query Tuition	YES	NO
Pay Tuition	NO	NO

University Web Site - Admin		
Add Tuition	YES	NO
Unpaid Tuition Status	YES	YES

Students

SUDE KARAKAYA	BERKAY HEREK
DOĞUKAN YEŞİLKAYA	DENIZ YALIM YILMAZ
BAŞAR ÖZKAŞLI	DİLA GENÇAĞA
TEVFİK EFE AYDIN	MEHMET UTKU GÜNDOĞDU
KAAN YILMAZ	EMRE ŞENER
HÜSEYİN BALCI	CEMİL FAHRECİ
MELİSA ŞENER	ALİ HAKTAN SIĞIN
KEREM KOYUNCU	ESRA ECE GÜNGÜ
NURETTIN DEMIREL	

COMMON REQUIREMENTS

- You are only asked to develop APIs that will be test in their swaggers. NO FRONT END necessary
- Every student will do their own midterm, no groups
- All REST services must be versionable
- Services must support paging, authentication as described.
- You need to implement an API gateway and configure all apis in the gateway.
    - Rate limiting should be implemented in the API gateway . You can implement your own gateway or use services from Azure/AWS.
    - Logging should be done for at least following data. You can use services like Cloudwatch or create your own logging in your own gateway
        - **Request-level logs**
            - HTTP method (GET/POST/PUT/DELETE)
            - Full request path e.g., /api/v1/bills/1234?month=2024-10
            - Request timestamp
            - Source IP address
            - Headers received
            - Request size (bytes)
            - Whether authentication succeeded or failed
        - **Response-level logs**
            - Status code (200, 400, 401, 403, 500…)
            - Response latency (ms)
            - Mapping template failures
            - Response size (bytes)

- For authentication, JWT or Oauth can be implemented. Please check the examples from class. Authentication at API gateway level is nice-to-have but not required
- All APIs must have Swagger UI or document. Swagger should point to the **API Gateway invoke URL**
- You can choose any development environment you like as long as they support REST services.
- You can make assumptions as long as you document them
- create a data model and use a database service (Azure SQL, Progress, AWS Cognito) from any cloud service you like.
- For API hosting, use a cloud service like Azure or Render.com. Points will be deducted if you cant deploy your project to a hosting provider.

DELIVERABLES

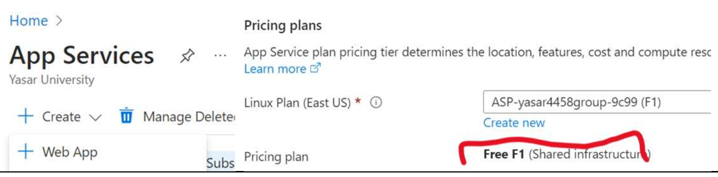
- A readme document in your github code repo that has
  - code link to source code of the project i.e github, bitbucket
  - your design, assumptions, and issues you encountered.
  - Data model (i.e an ER)
  - Include a link to a short video presenting your project

TodoApi WebApi	
TodoApi Code source	https://github.com/southriver/se4458-TodoApi
TodoApi Sample code deployed	https://yasar4458.azurewebsites.net/Swagger/index.html

Below is sample code from class	
.net	https://github.com/southriver/WebApplicationAPI https://github.com/southriver/se4458-TodoApi
Node.js	https://github.com/southriver/se4458-express
Flask	https://github.com/southriver/se4458-flask-api

Resources for creating REST services in different environments

- .NET - Sample webapi project in VSCode
  - https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-7.0&tabs=visual-studio-code
- Deploying to Azure App service via VS Code
  - https://davidgiard.com/deploying-a-web-app-to-azure-from-visual-studio-code
  - Make sure you choose F1 Free version in Azure for App Service that you will be creating
  - https://youtu.be/DUfPaY6FRII?si=X9pI0hhN209N3vwn



Pricing plans

App Service plan pricing tier determines the location, features, cost and compute resc
Learn more

Linux Plan (East US) * : ASP-yasar4458group-9c99 (F1)
Create new

Pricing plan: **Free F1** (Shared infrastructur

• PYTHON – Using flask   • https://dev.to/mursalfk/setup-flask-on-windows-system-using-vs-code-4p9j
• JAVA - Host a Spring Boot application   • https://www.baeldung.com/rest-with-spring-series   • https://javawhizz.com/2023/03/host-a-spring-boot-application-for-free-on-render