Team Big Data

U-tification

High Level Design

Date: 10/14/2022

Team Leader:	Joseph Armas
Team Members:	Joshua Gherman
	Rhoy Oviedo
	Frank Curry
	Ghabrille Ampo
	David DeGirolamo

Git Repository: https://github.com/JosephArmas/cecs-491A-Team-Big-Data

Version History

Current Version: V7

CHANGES

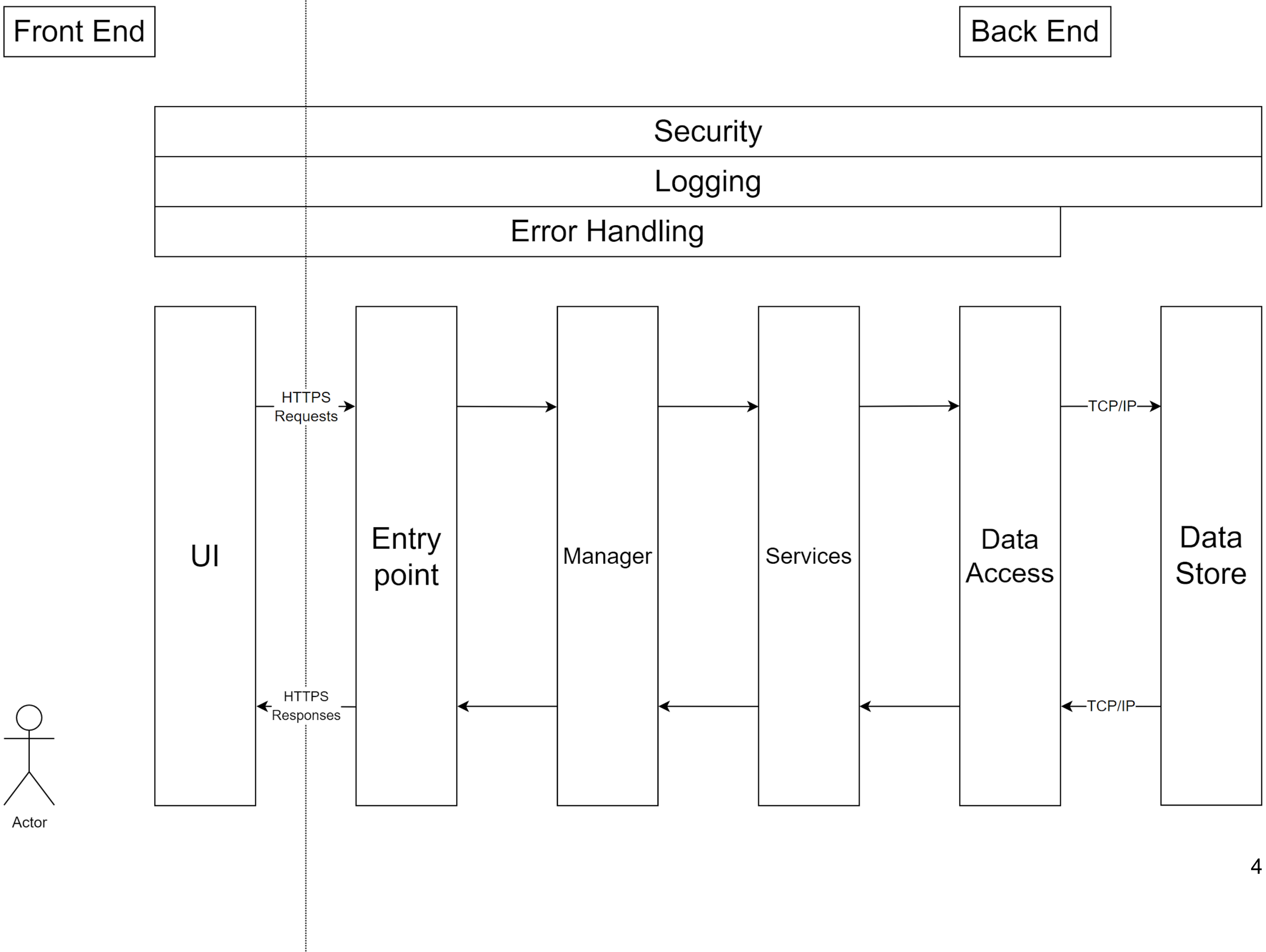
- Modified
    - Architecture Design
    - Services: security
    - Glossary

Previous Versions: V6,V5,V4, V3, V2, V1

Table of Contents

High Level Design: Page

Architecture Design



Architecture Overview

U-tification will be using a multi layered architecture. User interface will be on the front end that will have a ***Model-View-ViewModel*** architecture pattern and a ***microservices architecture*** for the backend.

Message Format

- Message format used will be ***JSON***
- The communication standard would be ***REST***

Protocol

- Use of ***HTTPS*** protocol

User Interface

Users will be interacting with the user interface and request through the use of the backend layers. The architecture pattern we will use is ***Model-View-ViewModel*.**

- **Security:**
  - Require authentication and authorization to perform actions
    - If user is not authorize display "permission denied"
    - If the user is not authenticated display "authentication failed"
- **Logging:**
  - Errors will be logged with a timestamp
  - Successful and failed authentications will be logged with timestamp
  - Successful and failed authorizations will be logged with timestamp

- **Error Handling:**
  - User input validation
    - When a user is prompted for an integer and instead the user inputs a string, display a message to the user "error, please re-enter number".

  - Failed authentication
    - When a user is prompted to be authenticated and does not have an existing account
    - Display a message “please try again or create an account to get started”
    - When a user is trying to access the manager without authorization a message will display “Access Denied”
  - No existing page
    - If a Users trys to visit a page that does not exist, display a message “No existing page, please return home”
  - Resources
    - If users are trying to reach U-tification, but resources are full and traffic is blocking to gather resources, display message “Resources are full, please come back later”

Entry Point

Entry point layer will handle the initialization of an application.

- **Security:**
  - Enforce a firewall with limitation and restrictions
- **Logging:**
  - All ***HTTPS response status codes*** will be logged with a timestamp.
  - Any failed UI communications will be logged with a timestamp.
  - Log failed communication to the next layer with a timestamp.
  - Errors will be logged with a timestamp.
- **Error Handling:**
  - If the entry point fails to initialize or HTTPS fails to communicate with the entry point, then no response with a blank screen will be returned.

Manager

Controls the flow and is the business layer of the web application.

- **Security:**
  - Users will be prompted to authenticate themselves with the use of a login. When a user tries to access services without read or write permission, it will display a message “authorization error you do not have access”.
- **Logging:**
  - Every time a user logs in or logs out, it will be logged with a timestamp.
  - Every time a user has a failed or successful authorization, it will be logged with a timestamp.
  - Every time a user fetches a service regardless if it fails or is successful, it will be logged with a timestamp.
  - Log failed communication to the next layer with a timestamp.
  - Errors will be logged with a timestamp
- **Error Handling:**
  - User input validation
    - When a user is prompted for an integer and instead the user inputs a string, display a message to the user “error, please re-enter number”.
  - Incorrect user credentials
    - Invalid credentials will display a message to the user “error, invalid credentials”.
  - Calling services failure
    - When manager layer fails in calling a service, display a message “service failed, please try again later”

Services

Microservices and web apis that are being used. We will be utilizing map service, uploading a file (.jpeg and .png) service, email service, search service, *CRUD* service, security service, cloud service, and storage service.

- **Security**
  - Authentication and authorization is required to use services. Admins are set with the permission to read and write access on this layer. Users are set with the permission to only have read access.
- **Error Handling:**
  - When a user tries to gain access to services and fails in the process, display a message to the user “failed to receive service, please try again later”.
- **Logging:**
  - Any services that failed to retrieve or use will be logged with a timestamp.
  - Log failed communication from Data Access layer and Manger Layer with a timestamp.
  - Creating, updating, reading, and deleting will be logged
  - Uploading a file will be logged
  - Utilization of email service will be logged
  - Security authorizations will be logged
  - Errors will be logged with a timestamp

Data Access

Anything that the user interacts with in the front end layer and is trying to access data will be handled through here. This layer interacts with the Data store layer through ***TCP/IP***.

- **Security:**
  - Users are required to authenticate and authorize through the use of a login. Users will only have read access to retrieve data from the data store. Admins will have the permissions of read and write access

- **Error Handling:**
    - If a non-admin user tries to directly access the Data Store, display a message to the user “permission denied”.
    - If Data Access fails to communicate with the Data Store through use of TCP/IP, display a message “Data Store server currently down”.
    - If trying to insert data into a storage that is full, display the message “Storage is full.”
    - If user is trying to receive data that does not exist, display a message “no existing data”
- **Logging:**
    - Anything created, updated, and deleted will be logged with a timestamp.
    - Any failed communications via TCP/IP from Data Access layer to Data Store layer will be logged with a timestamp.
    - Log failed communication to the next layer with a timestamp.
    - Errors will be logged with a timestamp

Data Store

The Data Access layer will communicate with the Data store though the use of TCP/IP. This is a Separated system and to communicate with the Data Access layer it will use TCP/IP. Utification will be using a relational database.

- **Security:**
    - Require authentication and authorization to mitigate any backdoor access to storage.
    - If a user tries to access data storage without permission, a message will be displayed “Access Denied”
- **Logging:**
    - Everytime a user attempts to access the data store will be logged with a timestamp
    - Any authentication or authorization will be logged with a timestamp
    - Any manual changes made in the datastore will be logged with a timestamp

Glossary

REST - A web service style that describes the flow between other web application components. [2]

HTTPS - Hypertext Transfer Protocol Secure. Allows secure communication over the internet to a user. [3]

JSON - A data format that allows human-readable text to be stored or transmit data objects consisting of serialized values.[4]

Model-View-ViewModel - Is a software architectural pattern implemented in the user interface. It allows separation, but also enables these three classes to communicate to eachother. [5]

HTTP response status codes - HTTP response status codes indicate whether a specific HTTP request has been successful or if the response has failed. [6]

Microservices architecture - Microservices breaks down an application to be stand alone services. For the functionality of the web application as a whole, services will call one another. [7]

TCP/IP - TCP/IP stands for Transmission Control Protocol/Internet Protocol and is a protocol used to communicate with other machines via internet [8]

CRUD - Represents Create, Read, Update, Delete. General functionality that is used in applications. [9]

Reference

[1] Vatanak Vong. {Message Format, Architecture Design}

[2] "Representational State Transfer." *Wikipedia*, Wikimedia Foundation, 2 Oct. 2022, \ en.wikipedia.org/wiki/Representational_state_transfer.

[3] "HTTPS." *Wikipedia*, Wikimedia Foundation, 28 Sept. 2022, https://en.wikipedia.org/wiki/HTTPS.

[4] "JSON." *Wikipedia*, Wikimedia Foundation, 27 Sept. 2022, en.wikipedia.org/wiki/JSON.

[5] "Model–View–Viewmodel." *Wikipedia*, Wikimedia Foundation, 9 Sept. 2022, https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel.

[6] "HTTP Response Status Codes - HTTP: MDN." *HTTP | MDN*, https://developer.mozilla.org/en-US/docs/Web/HTTP/Status.

[7] "What Is Microservices Architecture? | Google Cloud." *Google*, Google, https://cloud.google.com/learn/what-is-microservices-architecture.

[8] Shacklett, Mary E., et al. "TCP/IP: What Is TCP/IP and How Does It Work?" *SearchNetworking*, TechTarget, 13 July 2021, https://www.techtarget.com/searchnetworking/definition/TCP-IP.

[9] "What Is Crud?" *Codecademy*, https://www.codecademy.com/article/what-is-crud.