Team Big Data

U-tification

High Level Design

Date: 10/5/2022

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: V4

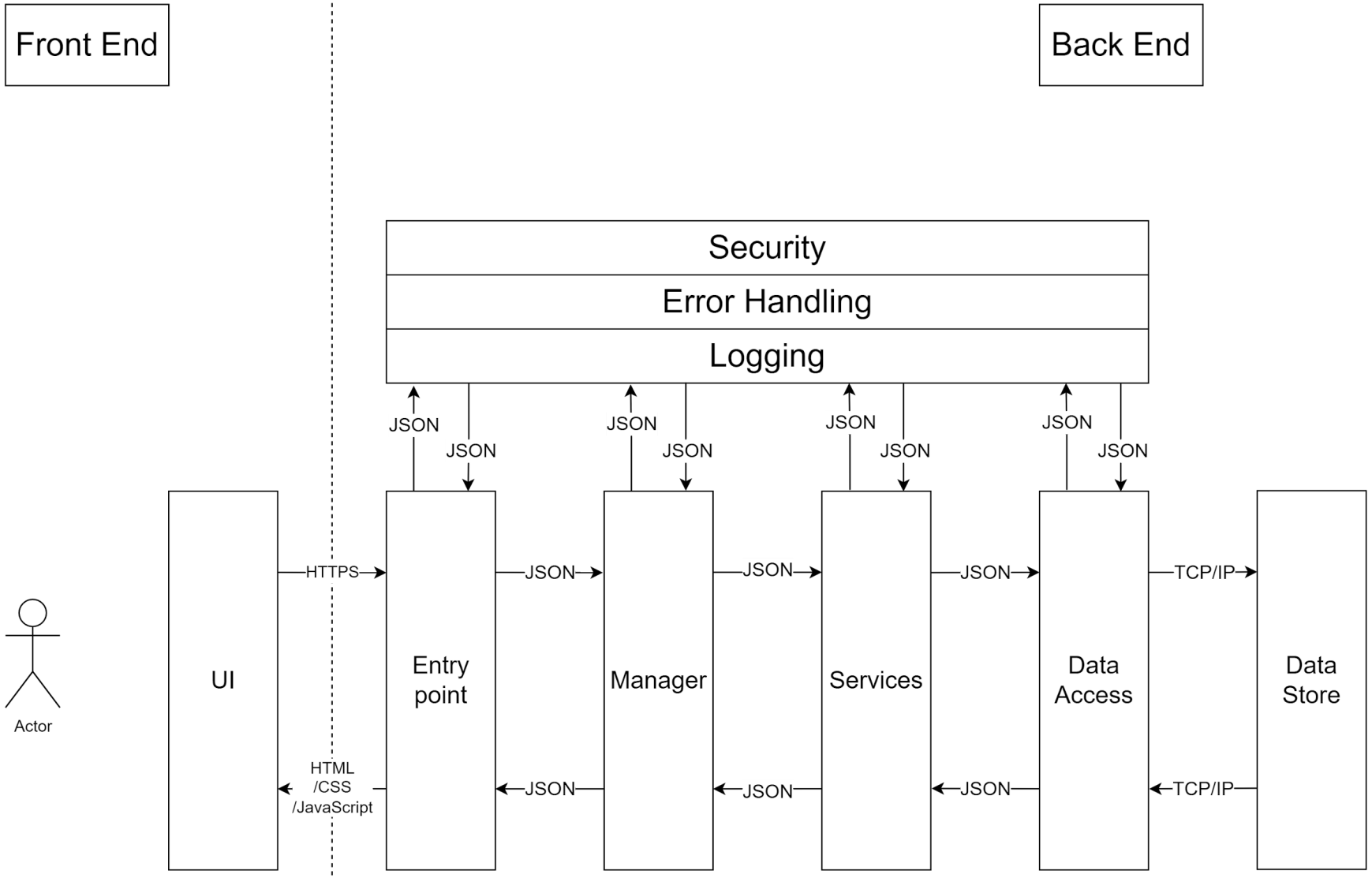
CHANGES

- Modified
    - Architecture Design
    - Message Format
    - Protocol
    - User Interface
    - Entry Point
    - Manger
    - Services
    - Data Access
    - Data Store
    - Reference: Vatanak Vong (Message Format)
- Added
    - Glossary: HTTPS, HTTP response status codes
    - Reference: HTTP response status codes
- Removed
    - Glossary: HTTP
    - Data Store Abstraction

Previous Versions: V3, V2, V1

Table of Contents

Architecture Design



Message Format

- Message format used will be ***JSON***
- The communication standard would be REST [1]

Protocol

- Use of ***HTTPS*** protocol

User Interface

- The architecture pattern we will use is ***Model-View-ViewModel*****.**

Entry Point

- **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.
- **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.

Manager

- **Security:**
    - Users will be prompted to authenticate themselves with the use of a login. When a user tries to access services without permission read or write permission, it will display a message “authorization error you do not have access”.
- **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”.

- **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.

Services

- **Security**
    - 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 to the next layer with a timestamp.

Data Access

- **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 with the data store layer.
- **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”.

- **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.

Data Store

- This is a Separated system and to communicate with the Data Access layer it will use TCP/IP.

Glossary

REST - Architecture that describes the flow of states to separated components in the system. [2]

HTTPS - Hypertext Transfer Protocol Secure. Allows web services to be securely distributed to users on a browser. [3]

JSON - An open standard file format that allows human-readable text to become a data object. [4]

Model-View-ViewModel - Is a software architectural pattern that facilitates the separation of the development of the graphical user interface. [5]

HTTP response status codes - HTTP response status codes indicate whether a specific HTTP request has been successfully completed. [6]

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.