

Team Big Data

U-tification

High Level Design

Date: 10/11/2022

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: V5

CHANGES

- Modified
 - Architecture Design
- Added
 - Architecture Overview
 - User Interface: Logging and Error Handling
 - Entry Point: Security
 - Services: layer explanation
 - Manager: layer explanation
 - Data store: Logging, Security
 - Glossary: ASP.NET, TCP/IP
 - Reference: ASP.NET, TCP/IP

Previous Versions: V4, V3, V2, V1

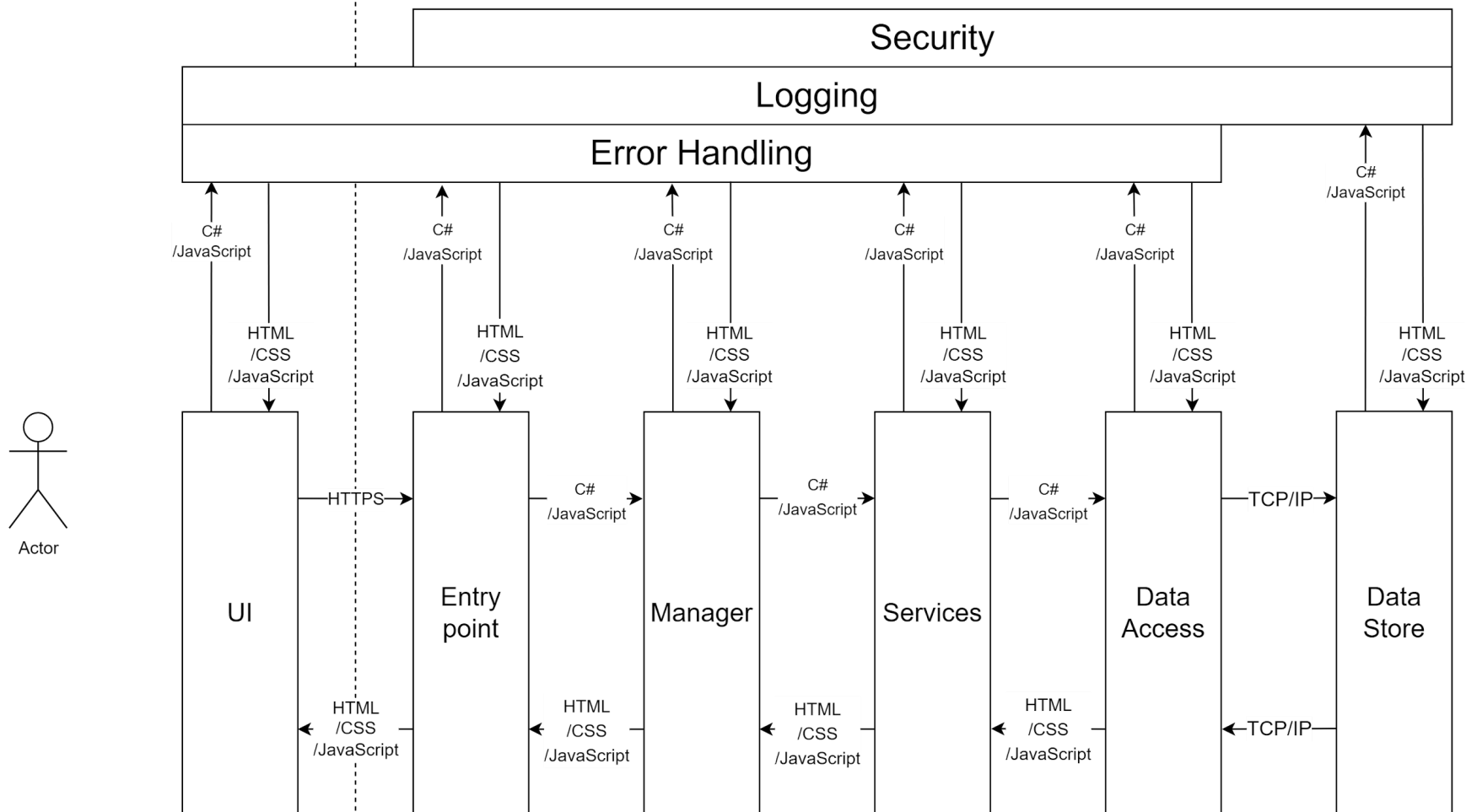
Table of Contents

High Level Design:	Page
Cover	1
Version History	2
Table of Contents	3
Architecture design	4
Architecture Overview	5
Message Format	5
Protocol	5
User Interface	5/6
Entry point	6
Manager	7
Services	7/8
Data Access	8
Data Store	9
Glossary	10
Reference	11

Architecture Design

Front End

Back End



Architecture Overview

U-tification will be using a multi layered architecture. User interface will be on the front end that the user interacts with and will be communicating with the entry point layer through HTTPS.

The entry point, manager, services and data access layer will be communicating with ASP.NET objects. From the Data access to the Datastore, it will be communicating with the server using TCP/IP. Error handling layers will be applied to the UI, Entry point, Manager, Services and Data access layers. Security layer will be applied to the Entry point, Manager, Services, Data Access and Data Store layers. Logging will be applied to all layers.

Message Format

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

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**.

- **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 tries 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”
- **Logging:**
 - Errors will be logged with a timestamp

Entry Point

Entry point layer will handle the initialization of an application. It is called from the UI layer using HTTPS and returns to the UI layer in the form of a page that consists of html, css, javascript.

- **Security:**
 - Prevent malware and bad actors from disrupting the system.
- **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.
 - If malware is caught, ban the bad actors.

Manager

Controls the flow and is the business layer of the web application. This layer requests for services that are in the service layer using ASP.NET and javascript, in return receives a page that consist of html, css, javascript.

- **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”.
- **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”.

Services

Microservices and web apis that are being used. It requests the Data Access layer using APS.NET and javascript. In return it gets a page that consists of html, css, javascript.

- **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.
 - 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. It return to the service layer with a page that consists of html,css and javascript.

- **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.

- Errors will be logged with a timestamp

Data Store

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

- **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 - 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]

ASP.NET - A platform for developing desktop, web, and cross-platform applications. The first version of the platform was published in 2002 and based on the .NET framework. The platform is based on [HTTP](#) protocol for requests and communication. [7]

TCP/IP - TCP/IP stands for Transmission Control Protocol/Internet Protocol and is a suite of communication protocols used to interconnect network devices on the internet. TCP/IP is also used as a communications protocol in a private computer network (an intranet or extranet). [8]

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] Vitaliy IlyukhaHead of TechnologyHead of Technology 5+ years. Vitaliy is taking
 technical ownership of projects including development, et al. “What Is the Difference
 between ASP.NET vs Asp.net Core?” *Jelvix*, 22 Feb. 2021,
 <https://jelvix.com/blog/asp-net-vs-asp-net-core>.
- [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>.