SP-2 DataMining & AI Software Design Document

4803 - 03

Fall Semester 2024

Professor Perry

8/29/24

Tanner Velzy

Project Lead/Documentation

Andujar Brutus

Development

Name	Role	Cell Phone / Alt Email
Tanner Velzy	Team Leader/Documentation	404-405-3524 Tdv1201@gmail.com
Andujar Brutus	Developer	470-476-4473 andujarbrutus@gmail.com
Sharon Perry	Project Owner/Advisor	770-329-3895 Sperry46@kennesaw.edu

Contents

1. Introduction and Overview

In this System Design Document (SDD), we will describe the design of our project's systems. This SDD will provide a high-level overview of the design and architecture of our system as well as our approach to this project. This will also explain why we plan to use these designs and approaches.

2. Design Considerations

2.1. Assumptions and Dependencies

- Client requires a working PC with an internet connection
- Proper authorization for Azure and GitHub
- Application and AI require an informative dataset
- Depends on Azure functioning correctly
- Cloud software must be online, and user must be connected to internet to allow the program to access the dataset
- The Lambda server keeps up with requests and data
- The CICD Pipeline functions correctly and without issue

2.2. General Constraints

- Dataset accuracy
    - This entire project depends on the dataset being accurate, which we will do our best to circumvent by using a trusted source.
- Cloud software uptime and downtime
    - As it is an online service, our storage is based on the uptime and downtime of our provider.
- Python library limitations
    - The limitations are based on the version of the libraries themselves depending on the version of Python we use.
- Azure free cloud software storage provided
    - Due to being a free version of the cloud software, the storage might be limited. Has only a limited number of credit hours before asking you to begin paying for their service.
- Effectiveness of the automated testing
- Budget
    - We will be using all free options with any software we decide to use, which might limit the capabilities of what we might be able to do with paid subscriptions.
- Lambda Server limits

    - Has a limited amount of processing power compared to cloud software.
- CICD pipeline constraints

As it is running constantly, resource management will be a big part. This would mean either watching over the pipeline or automating it completely.

2.3. Development Methods

We will be using Scrum of which we will be using Continuous Integration & Continuous Delivery (CICD). We are using Scrum because it is a framework for managing work that will allow us to work together efficiently and focus on a common goal. We decided to use CICD because it will allow us to keep up with the version control of the project as we develop it. This will help automate testing, training and publishing while ensuring the group always has the same version.

3. Architectural Strategies

- Automated Deployment
    - We will be designing the architecture API using an IDE known as PyCharm. We will then use the IDE/Terminal to connect to a version control repository that is connected to GitHub to allow us to use CICD. Using GitHub actions, we will create a CICD pipeline to use with Azure. This pipeline will be responsible for testing and training of the application and AI.
- Azure
    - A cloud software with a free version that we will be using to act as a data warehouse for the dataset we decide to use for our application and AI. Our second choice was Google Cloud because it is convenient, but Azure was more lucrative for job opportunities as more companies use it compared to Google Cloud.
- GitHub
    - It is a developer platform that will allow us to store and share code which we will be using for version control and to store the code we have made in a way that is easily accessible by the group. GitHub would also allow us to make the most of CICD due to its usability with group projects and documentation, however, due to budget constraints, we will not be able to utilize CICD and instead must base it on Azure instead.
    - Github Pages will also be used to host our frontend. It is a free static website hosting service created from Github repositories. With this, we can easily and for free host our website. This will allow for us to integrate our frontend with our pipeline and allow for the pipeline to automatically update the frontend along with the rest of the project.
- PyTorch
    - A machine learning library that will help train the AI and create the architecture for the AI. We were also considering using TensorFlow as it is

used for big projects in companies, but PyTorch was more suited for smaller projects like ours.

- Python
    - Programming language that is better suited for our machine learning objectives.
- Terrorism Dataset
    - This dataset is taken from the Kaggle website which will detail terrorism-related crimes that have happened between 1970 to 2017. It has 135 columns of data; however, we will be using only columns of data that are the most important to display and for our AI to digest.
- PowerBI
    - An interactive data visualization software product that allows us to more easily display necessary information from the data mining side of our project.
- OneLake
    - A data lake that will act as our data repository where we retrieve data for Microsoft Fabric to use.
- Microsoft Fabric
    - An end-to-end analytics and data platform designed for enterprises that require a unified solution. We will be utilizing this for the data lakehouse and OneLake.
- IntelliJ
    - By JetBrains, this functions as our IDE where we do most of the coding before implementing into the pipeline.

4. System Architecture

There are five subsystems in our system architecture: the Cloud, the Application, the AI, the Website, and the Bridge. The Cloud will store the dataset we have given it so it may be accessed more easily by the Application. The Application will download the dataset from the cloud and process it into various categories and patterns. The AI will then take this data that the Application has processed and draw conclusions from the data. Both the Application and AI will submit their findings to the Website where it will all be displayed in an easily readable format for the average viewer. And finally, the Bridge connects the backend of the systems to the frontend as the frontend will request data using the Bridge.

5. Detailed System Design

5.1. Classification

- The Cloud - Subsystem
- The Application - Backend
- The AI - Module

- The Website – Frontend
- The Bridge – Server

5.2. Definition

- The Cloud – The main goal of the Cloud is to store our dataset that the application and AI will use.
- The Application – Will communicate between the cloud and backend as well as gather and evaluate the data for use by both the AI and Website.
- The AI – The AI will process the dataset provided by the Application and correlate any patterns from the data before outputting them into the Website
- The Website – The dedicated frontend that will display all the outputs from the Application and AI.
- The Bridge – This acts as the bridge between the frontend and the backend, or everything and the Website.

5.3. Constraints

- The Cloud – A potential constraint for the Cloud is the fact we are using a free license that could limit the processing power available to us.
- The Application – Potentially, there could be a compatibility between the Application and the Cloud that could prevent us from even using the dataset.
- The AI – Depends on the Application for its data so it will not even get the dataset if the Application doesn't either. There is also the possibility of the data being wrong which is more important for the AI to have correct data as it is making assumptions and correlations on that data.
- The Website – Depending on how the data from the Application and AI is given, the Website might have issues formatting the data in an easily readable format.
- The Bridge – Constrained by the load the server can take and how many calls it is able to process.

5.4. Resources

- The Cloud – Has usable storage of 50 MB which should store the dataset we provide it.
- The Application – Utilizes the Cloud's data warehouse to retrieve the dataset that it will use. The Application will also be utilizing Python's libraries.
- The Bridge – Utilizes a server with the capability of transferring requests.

5.5. Interface/Exports

- The Cloud – Provides a data warehouse and technical dashboard for our machine learning needs. Azure services provide various machine learning tools and can be classified as a subsystem to our project.

- The Application – Provides communication between the frontend and backend while also processing the dataset it is given for display and use. This will use a module API that will communicate between our subsystems.
- The AI – Uses created model and dataset to provide time series analysis and anomaly detection using Azure Studio. This will provide a collection of files that will be exported to the Website for viewing.
- The Website – Displays the information created by the Application and AI for the clients to read in an easily accessible and readable format. This is a function that will allow us to display all the information about the dataset.
- The Bridge – Our server, which is on the lambda workstation as an API module, will take various GET calls from the frontend which it will process through the modules before giving that GET call to the backend at which point the frontend will receive the data it requires.

6. Glossary

CICD – Continuous Integration & Continuous Delivery; an automated process used by software development teams to streamline development, testing, and delivery of products.

Scrum – A collaboration framework used in software development and other industries that helps group's structure and manage their work.

Cloud Software – Software that is based in the cloud which in turn operates over the internet rather than a defined physical space, allowing people to store and access data from anywhere with an internet connection.

IDE – Integrated Development Environment; a software that provides development tools and a GUI for programmers.

Data Lakehouse – creates a single platform by combining key benefits of data lakes and data warehouses.

Data Lakes – Large repositories of raw data in its original form.

Data Warehouses – organized sets of structured data.

Lambda Workstation – serverless computing service

7. Bibliography

Sanchez, Sandro. "Continuous Integration and Continuous Deployment (CI/CD) in Azure with Github Actions." *Medium*, Medium, 26 Nov. 2023, medium.com/@sandropucp/continuous-integration-and-continuous-deployment-ci-cd-in-azure-with-github-actions-bf501531dd7d