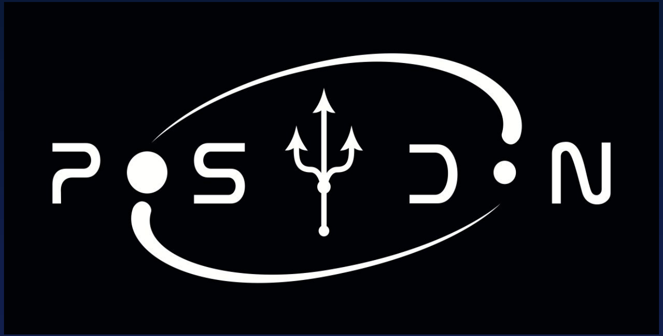
Detailed Design

By: Andrew Snyder, Alex Polston, Alek Norris,
Eamon Collins, James Byrd, Svyatoslav Varnitskyy

sdmay25-20

Context

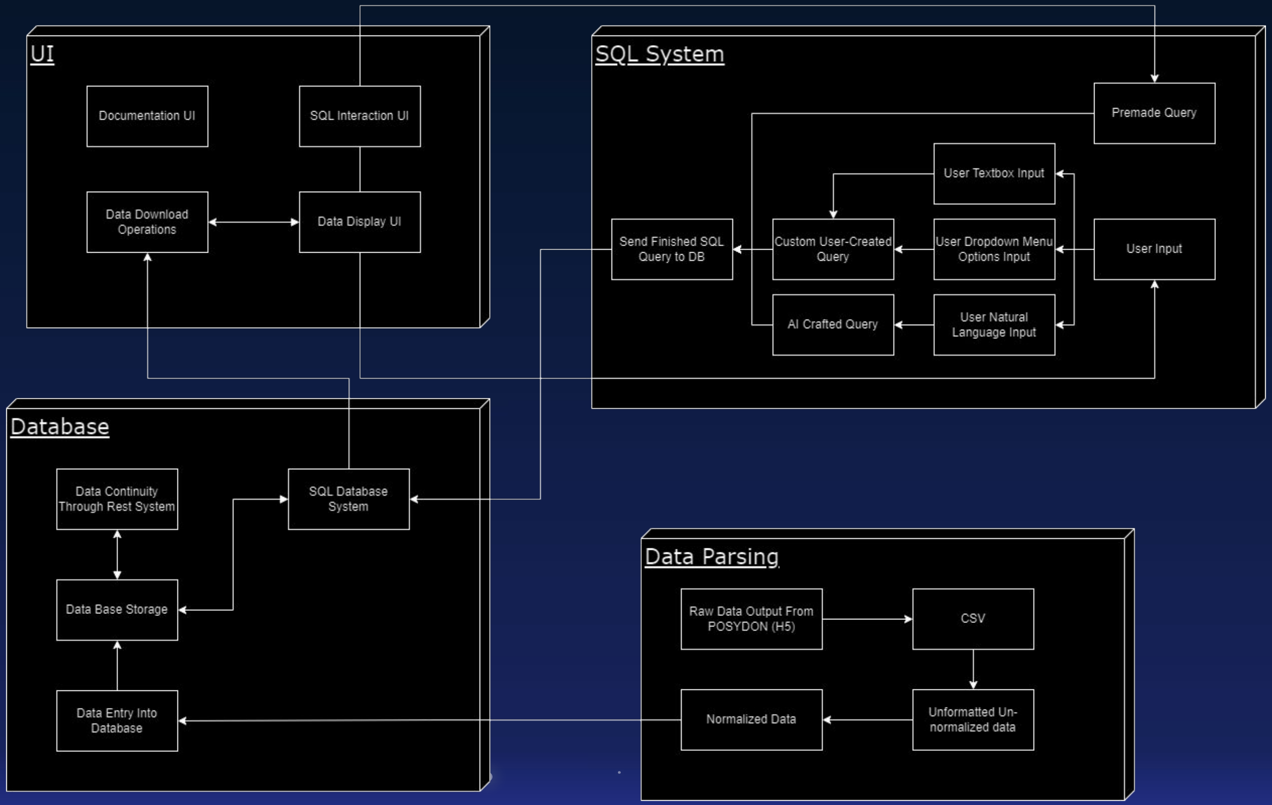
- POSYDON : POpulation SYnthesis with Detailed binary-evolution simulatiONs
- Developed by a collaborative group of scientists primarily at Northwestern University
- Simulates stellar evolution of binary stars
- Generates massive amounts of data as simulation output



Project Overview

- **Objective:** Develop a system to manage and analyze simulated binary star data
- **Key Features:**
  - Import multivariate time-series simulation data into relational database
  - Provide sample SQL queries
  - Enable custom queries through natural language processing
- **Deliverables:**
  - Relational database
  - User Interface for writing and viewing SQL queries
  - Sample SQL queries

Detailed Design



Detailed Design Explained

There are four main components of our design

- UI
  - SQL Interaction UI
  - Data Display UI
  - Result Data Download UI
  - Documentation UI
- Data Parsing
  - Take CSV Files from POSYDON
  - Normalize Data
  - Inject Data into Database
- Database
  - Postgresql database
  - Interacts with SQL
- SQL creation
  - Creation of SQL queries from
    - Natural language (AI processes)
    - Dropdown selection
    - Textbox

Functionality

- A user enters natural language into the user interface
- The frontend application sends the natural language to the backend
- The backend retrieves the database schema
- The backend sends the natural language string and the database schema to OpenAI via its API
- OpenAI responds with a relevant SQL query
- The backend validates the functionality of the query
- The backend returns the query to the user
- The user confirms the query and makes any desired changes before returning the query to the backend
- The backend queries the database using the specified query
- The backend returns the query resolution to the frontend
- The frontend downloads the resolution for later usage

Technology Considerations

PostgreSQL

- As a relational database, elements must follow strict guidelines
    - Ensures records follow patterns
    - May make certain records difficult to include in the database
- Supports custom data types

React

- Supports implementation of HTML directly inside of javascript
- Javascript is supported all major browsers
- Several team members have not utilized Javascript

Areas of Concern and Development

- Artificial Intelligence can misinterpret natural language and return an incorrect query
  - This requires our team to perform numerous tests to ensure correct interpretation of natural language
  - One method we have considered is having the AI translate natural language into a query then back into natural language

- Size of Database and Speed of the Returning the Query
  - This requires our team to perform tests and come up with a minimum specs requirement for intended use of the program and database.

Conclusions

- POSYDON is a project that collects data of binary star evolution through various simulations
  - Terabytes of data have been generated
  - Missing a tool to query data
- Our project aims to design a tool for managing this data in a database
- Features a user interface for writing queries and receiving data
- Features a backend for communicating with external APIs, the frontend, and the database
- Using AI may lead to poorly formatted SQL queries
  - Backend will validate queries before returning to the user