# Design Document

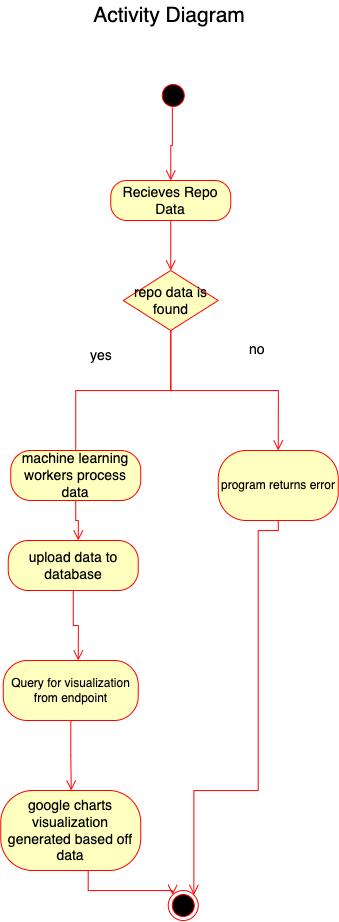
## Table of Contents

1. Activity Diagram (page 2)
2. Use Case Diagram (page 3)
3. Entity-Relationship Diagram (page 4)
4. Query Examples (page 6)

### 1. Activity Diagram

Title: **System processes data to produce endpoint**

Contributors (Roles): Jackson Haskamp (Design Collaborator), Kamden Riley (Design Collaborator)



Caption: Activity diagram referring to the use case, from the system's perspective of what happens during the use case, and how it flows to return the visualization.

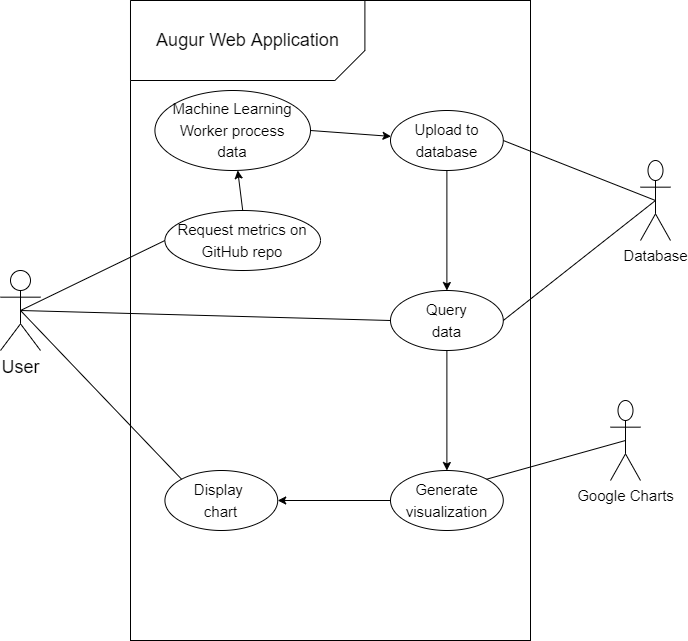
Reference(s):

* <https://creately.com/blog/diagrams/activity-diagram-tutorial/>
* “Module 5 – Design: UML and Data Models” class module. <https://github.com/MUSoftwareEngineering/CS-4320/blob/main/05-architecture-design/exercises/05.1-architecture-design.md>

### 2. Use Case Diagram

Title: **Pull Request Data Visualization Use Case**

Contributors (Roles): Tasha Ogoti (Design Collaborator), Erika Zhou (Peer Reviewer/Editor)



Caption: Use case for a user using the Augur web interface to generate a visualization based on machine learning output of GitHub repository pull request data.

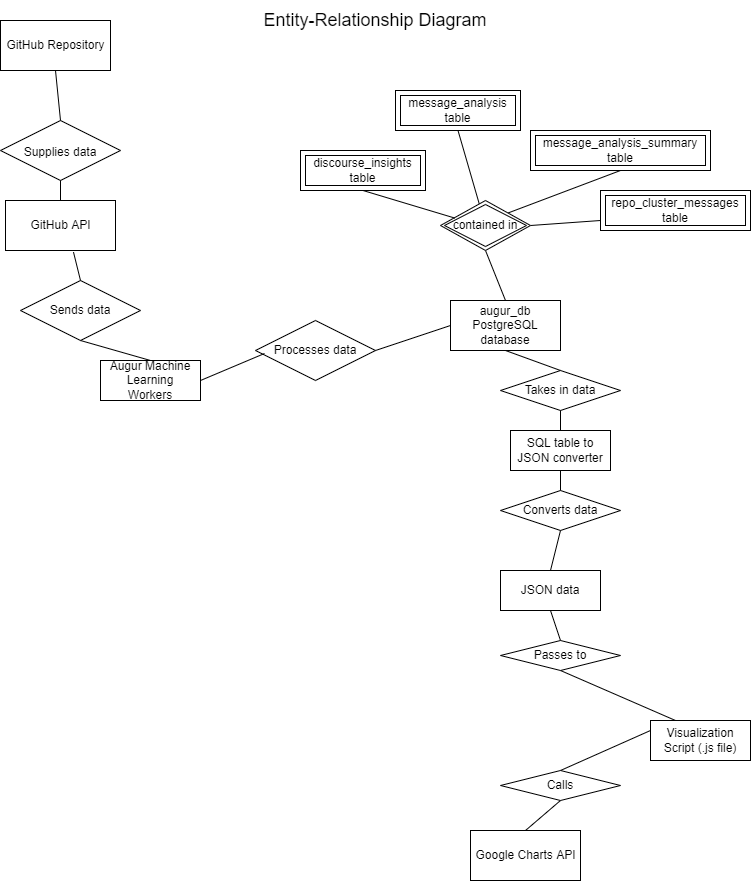
Reference(s):

* “Module 5 – Design: UML and Data Models” class module. <https://github.com/MUSoftwareEngineering/CS-4320/blob/main/05-architecture-design/exercises/05.1-architecture-design.md>

### 3. Entity-Relationship Diagram

Title: **Augur Machine Learning Visualization ER Diagram**

Contributors (Roles): Jackson Haskamp (Design Collaborator), Tasha Ogoti (Design Collaborator), Erika Zhou (Design Collaborator)



Caption: This ER diagram shows components of the Augur system that are relevant to the machine learning data visualizations.

Entities: GitHub Repository, GitHub API, Augur Machine Learning Workers, augur\_db PostgreSQL database, discourse\_insights table, message\_analysis table, message\_analysis\_summary table, repo\_cluster\_messages table, SQL table to JSON converter, JSON data, Visualization Script (.js file), Google Charts API

Reference(s):

* “Entity–relationship model” from Wikipedia   
   <https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model>
* “Module 5 – Design: UML and Data Models” diagram class slides
  + <https://github.com/MUSoftwareEngineering/CS-4320/blob/main/05-architecture-design/slides/Design_0_Diagrams.pdf>
* Augur Labs <https://www.augurlabs.io/>
  + Diagram of Augur architecture <https://www.augurlabs.io/wp-content/uploads/2022/03/augur-1.jpg>
* Entity-Relationship Diagram from Software Engineering Assignment 5-1

4. Query Examples

Example queries of what an Augur visualization from hitting the endpoint could look like, since our project has to do with visualizing machine learning data.

* Get grouped horizontal bar chart of average response times for closed pull requests.
  + API Reference: <https://oss-augur.readthedocs.io/en/main/rest-api/api.html#operation/Mean%20Response%20Times%20for%20Closed%20Pull%20Requests>
  + Visualization: <http://augur.chaoss.io/api/unstable/pull_request_reports/mean_response_times_for_PR/?repo_id=25440>
* “Average Commit Counts Per Year for All Pull Requests”
  + API Reference: <https://oss-augur.readthedocs.io/en/main/rest-api/api.html#operation/Average%20Commit%20Counts%20Per%20Year%20for%20All%20Pull%20Requests>
  + Visualization: <http://augur.chaoss.io/api/unstable/pull_request_reports/average_commits_per_PR/?repo_id=25440>

We’d be making Python-based API endpoint(s) so users can get a visualization like this for machine learning output data.

GSoC Idea: Machine Learning based Community Health and Communication #1637   
<https://github.com/chaoss/augur/issues/1637>