CS4320 Software Engineering

Group 6

Sprint 2

Augur Libyear Metric Design Document

Ben Hudson, Aidan Henry, Andrew Gillis

20th April, 2021

Table of Contents

1. Overview

a. The Augur Libyear Metric Design Document will provide a detailed description of the incremental implementation of the Libyear metric, explaining the testing the project will undergo as well as diagrams displaying the specific functionality the project will provide. The goals and the methodology are comprehensive of the project, but the diagrams may not fully represent every activity implemented.

2. Sprint Goals

a. Sprint 1
    i. Fork Augur Github
    ii. Setup Azure Server with Augur Fork
    iii. Research
        1. Augur Architecture
        2. Possible Libyear Implementation
        3. Azure Server Setup
    iv. Created Requirements Document
b. Sprint 2
    i. Create Design Document
    ii. Create Test and Code Stubs
    iii. Create Skeleton Code Libyear_Worker
c. Sprint 3
    i. Implement Working Code
        1. Fill Worker Skeleton
        2. Configure Libyear to Work with Skeleton
        3. API Endpoint Setup
    ii. Update Requirements and Design if Needed
    iii. Create Video
d. Sprint 4
    i. Merge to Master
    ii. Issue Pull Request Back to Main Augur
    iii. Update Issues if Any
    iv. Update Requirements and Design if Needed

3. Testing Methodology

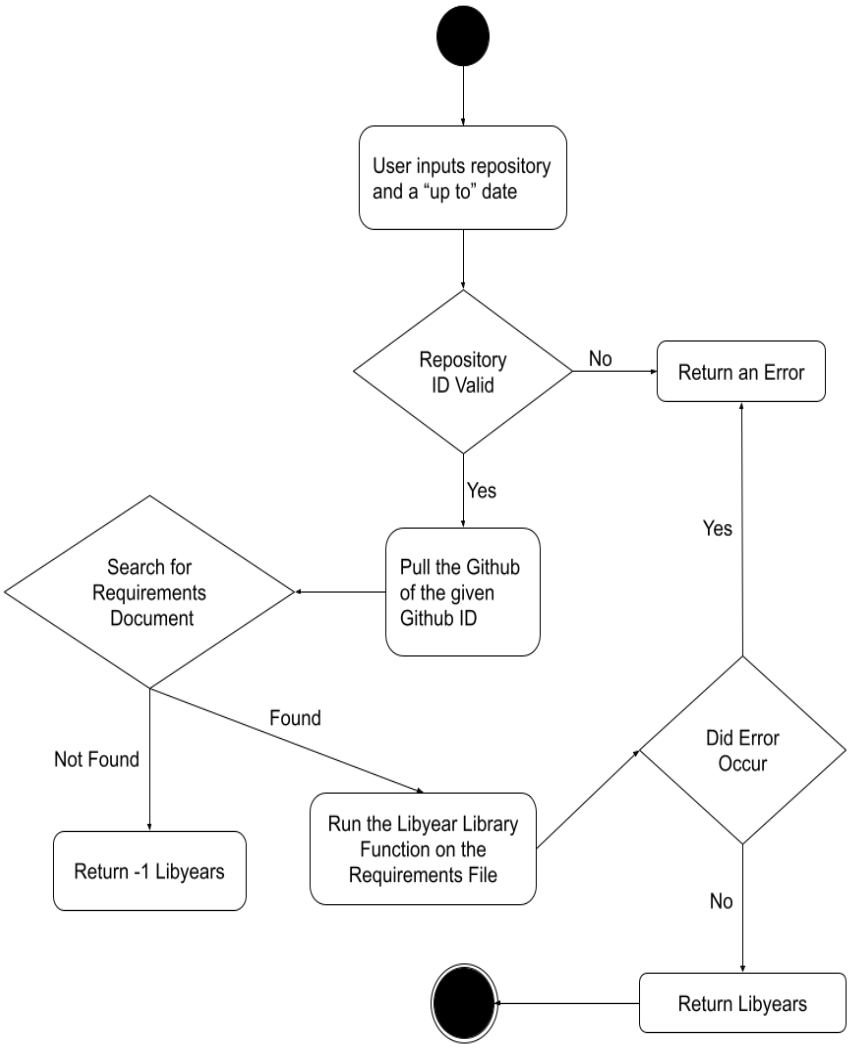
a. Testing System Overview
    i. The Libyear Metric project will be using the pytest library to run tests on the worker and the metric code.We will create a new github account to hold the 5 Dummy Github repositories. These Dummy Repositories will be used in the test functions to verify everything is working correctly. Each of them will have a different form of requirements specification to test the different

ways libyear can calculate given different methods of finding requirements manifests.

b. Testing Code Stubs
    i. test_getLibyears
    ii. test_getLibyearsOver

4. Activity Diagram #1

a. Label:Get Libyears
b. Description: Given a repository ID and a Date calculate the libyears up to that date.
c. Contributors: Andrew Gillis and Ben Hudson

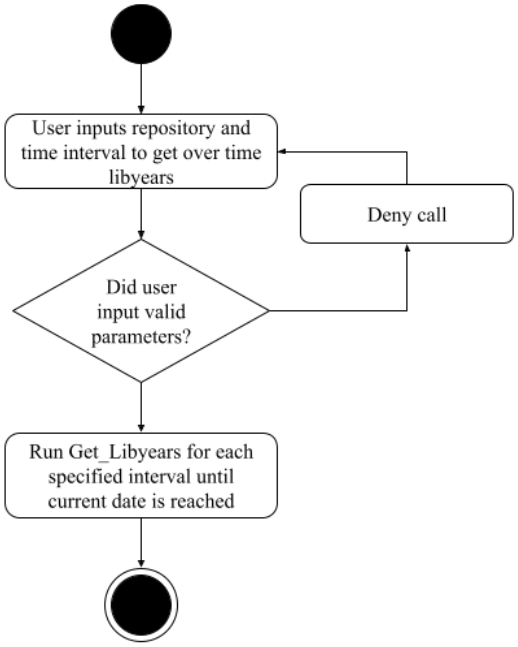


d.

5. Activity Diagram #2

a. Label: Get Libyears Over Time

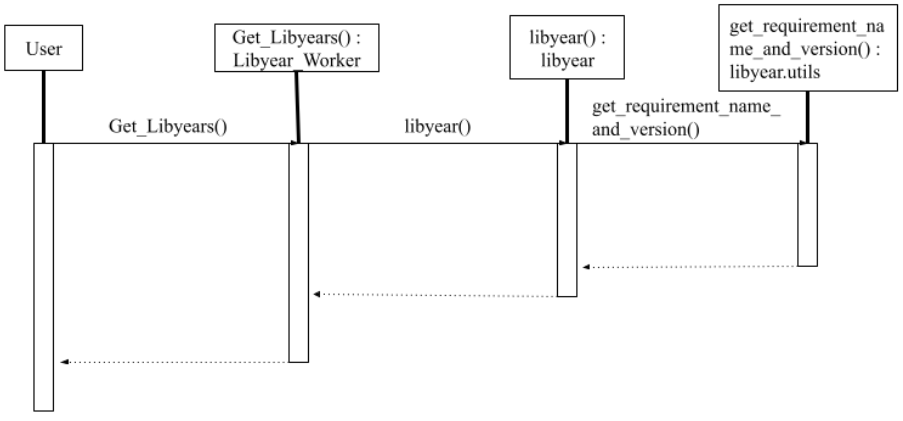
b. Description: Given a specific time interval (2 years, year, 6 mo, etc.), calls Get Libyears for each interval and calculates the libyears over time.
c. Contributors: Aidan Henry



d.

6. Sequence Diagram

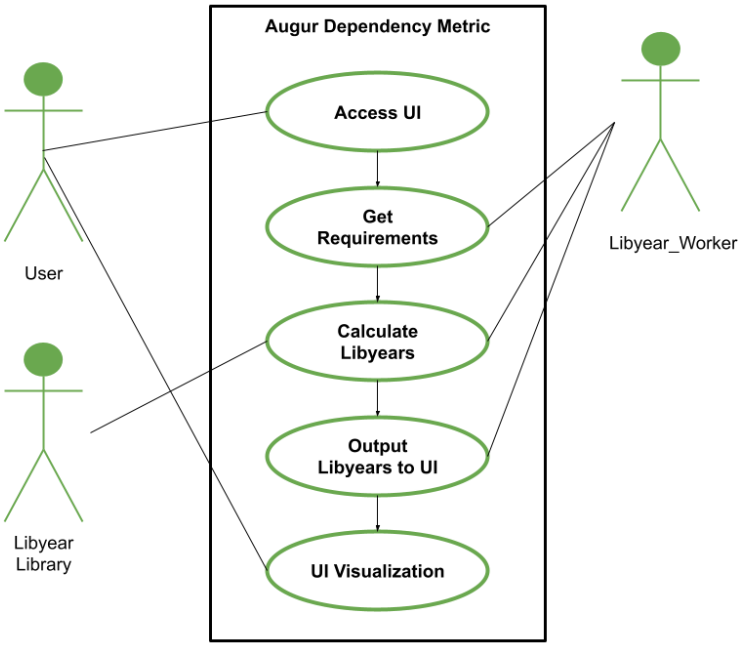
a. Label: Get_Libyears
b. Description: This diagram displays the system's sequence when Get_Libyears is called
c. Contributors: Aidan Henry



d.

7. Use Case Diagram

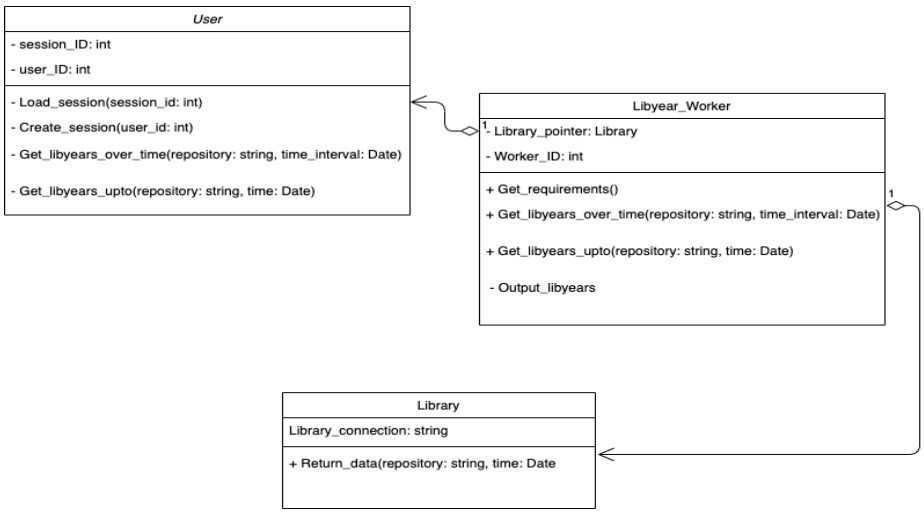
a. Label: Libyear Metric Overview
b. Description: A diagram of all stakeholders interacting.
c. Contributor: Aidan Henry



d.

8. Class Diagram

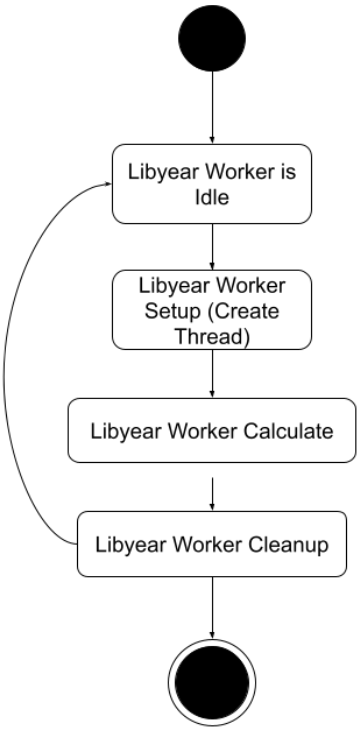
a. Label: Libyear Metric Classes
b. Description: Describes the classes involved in the Libyear Metric System
c. Contributors:



d.

9. State Diagram

a. Label: Libyear Worker State Diagram
b. Description: A very general overview of the Libyear Worker
c. Contributors: Andrew Gillis



d.

10. Team Reflection

During the completion of sprint 2, only a handful of activities posed much difficulties. Diagrams took a little longer than originally intended. Team meetings went flawlessly and communication was excellent from all parties. Diagrams may not fully represent all activities that we plan to implement.