Project Milestone 3

As a Boilermaker pursuing academic excellence, we pledge to be honest and true in all that we do. Accountable together – We are Purdue.

Type or sign your names: Ryan Villarreal, Owen Prince, Mohammed Fashola
Write today's date: Nov 8, 2021

Assignment Goal

A weekly project update is a normal component of many engineering teams. Such an update has many uses. It helps you understand what you've actually accomplished and whether you are on track.

- It gives you a "paper" record of your accomplishments, to help justify your request for a raise or a bonus at the end of the quarter. It helps you track accomplishments that might not be captured by product-centric metrics (it is easy to measure "lines of code committed", but hard to measure "helped onboard the new intern").
- It informs other teammates or other teams about your areas of expertise, so they know whom to ask when they need help.
- It helps project managers see how things are going, re-prioritize the team's activities, and assign more personnel to shore up difficult areas.

This assignment does not imitate every aspect of such an update, but hopefully it gives you the flavour of the activity. Through the milestone assignments for this course, you have an opportunity to *report* and *reflect* on your team's progress to date.

Relevant Course Outcomes

A student who successfully completes this assignment will have demonstrated the ability to

- Outcome i:
  - Identify and follow an appropriate software engineering process for this context.
- Outcome iii:
  - Experience social aspects of software engineering (communication, teamwork).

Resources

Perhaps relevant are these ones:

- **Postmortems**
  - Postmortems at Google
  - Postmortems at Amazon

Assignment

In your project plan (Milestone 1), you submitted a design as well as a list of weekly milestones. In each intermediate milestone report, you will present:

1. "Substantial" updates to your design (I leave this definition to your engineering judgment).
2. A statement of the tasks your team accomplished by this date.
   a. How did you measure that they are accomplished?
   b. Who did the work? How long did they spend?

3. A comparison of what you accomplished vs. what you planned to accomplish. Are you on track? How were your time estimates?
4. Any changes in your planned timeline as a result of falling behind your initial plan.

If you *deviate substantially* from your timeline, consider attending one of the course staff office hours to discuss the deviation.

You can communicate this information using tables, charts, etc.

If you have working functionality that will add value to ACME Corp., feel free to include a screenshot of a demo. This will build trust with your customer, Sarah, and gives her the opportunity to propose tweaks ("requirements changes").

Grading

These assignments are worth relatively little in the overall Project grade (2.5% each).

This weighting reflects the relative *time* you should spend on the milestone report, but not the relative *importance* of the report. These documents are critical to help you understand whether you are on track to succeed.

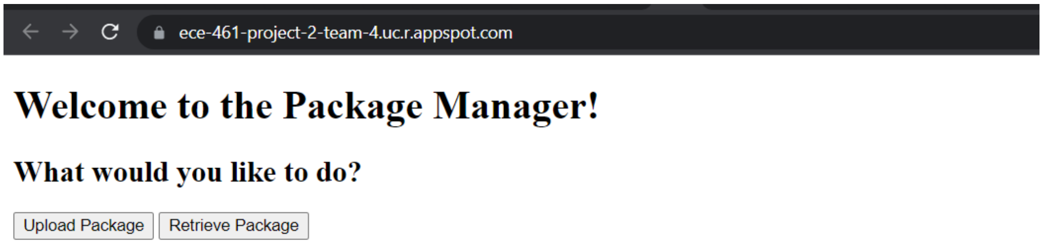
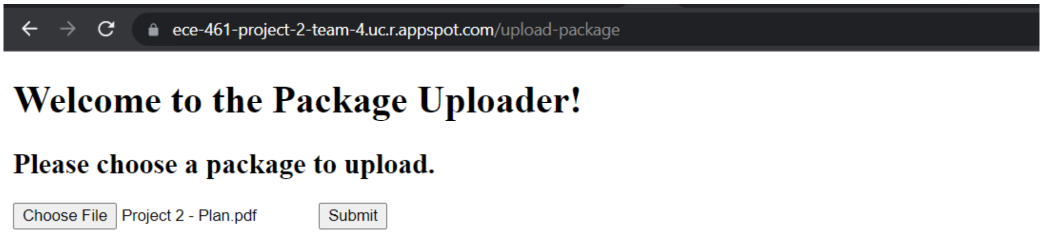
1. "Substantial" updates to your design (I leave this definition to your engineering judgment).

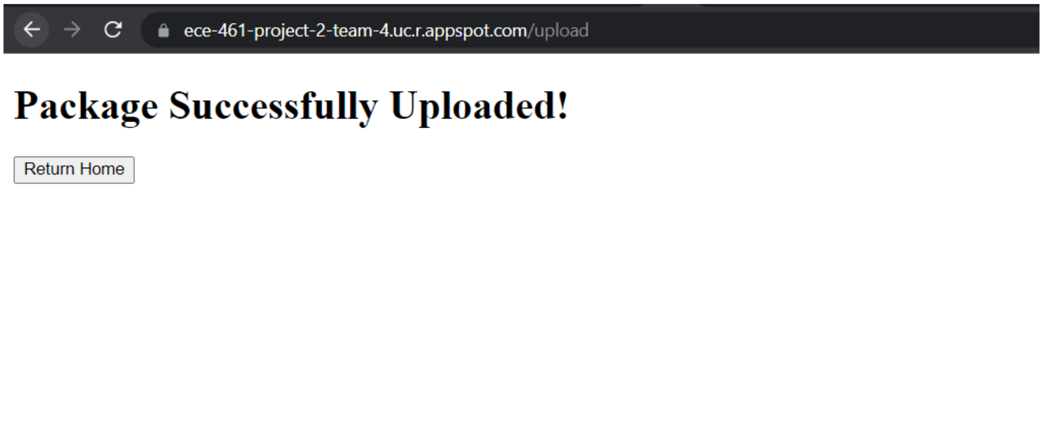
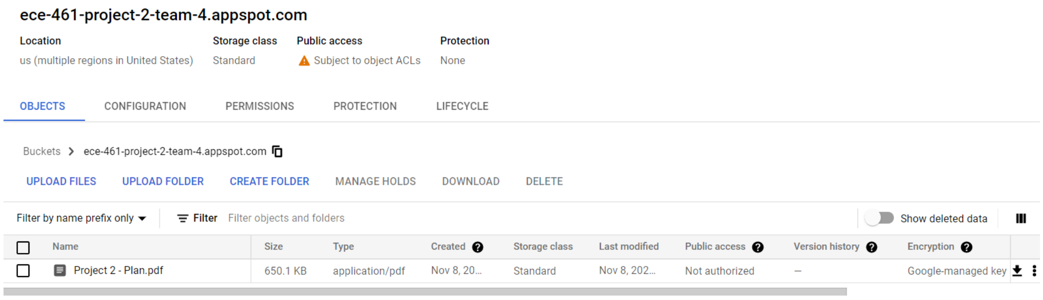
We have determined that we must revise our timeline due to the existing issues in the trustworthiness module and the ramp up time.

- We are changing our implementation from Google Compute Engine to Google App Engine. After consulting with an expert (TA) he recommended this platform for scalability purposes. Outside of this, there are no big updates.

- More granular test cases should be added to verify functionality. There is not a ton of modification necessary, but investing time today on a more robust framework will pay off.
- We will not be implementing authentication features in order to ensure delivery of the core functionality.
- Postman will not be utilized as we will host everything on GCP.
- We will utilize the given openAPI YAML file that was given to guide our development.
- We have decided to modify our database implementation from the GCP firestore and cloud storage to python's SQLalchemy library.

2. A statement of the tasks your team accomplished by this date.

Owen	● Worked on trustworthiness module: ○ Developed JSON parser to parse the package.json file of repositories ○ Developed class to calculate pinned dependency ratio score of a npmjs module and a test suite for it ○ Created Github action to run the test suite on pushes (not currently in our repository due to the issues)
Ryan	● A basic version of our program is up and running on the App Engine. This basic version allows for file uploading directly to Google Cloud Storage. Examples in screenshots below: **Homepage** ece-461-project-2-team-4.uc.r.appspot.com Welcome to the Package Manager! What would you like to do? **Package Uploader** ece-461-project-2-team-4.uc.r.appspot.com/upload-package Welcome to the Package Uploader! Please choose a package to upload. **Package Uploaded**

	 ece-461-project-2-team-4.uc.r.appspot.com/upload Package Successfully Uploaded! Return Home **File in Google Cloud Storage Bucket**  ece-461-project-2-team-4.appspot.com Location: us (multiple regions in United States) Storage class: Standard Public access: Subject to object ACLs Protection: None OBJECTS CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE Buckets > ece-461-project-2-team-4.appspot.com UPLOAD FILES UPLOAD FOLDER CREATE FOLDER MANAGE HOLDS DOWNLOAD DELETE Filter by name prefix only ▾ Filter Filter objects and folders Show deleted data □ Name | Size | Type | Created | Storage class | Last modified | Public access | Version history | Encryption □ Project 2 - Plan.pdf | 650.1 KB | application/pdf | Nov 8, 20... | Standard | Nov 8, 202... | Not authorized | – | Google-managed key
Mohammed	- This week I worked on understanding and utilizing Postgresql syntax and learning how to interact with it. - Created queries to interact with the examples I Setup - Next week, I will be integrating the database into our app engine to store our metric scores.

3. A comparison of what you accomplished vs. what you planned to accomplish. Are you on track? How were your time estimates?

We have recovered slightly from being behind last week. Dropping some tasks for front end and authentication allowed us to catch up a bit. Meeting with the TA's allowed us to have a better understanding of what we will need to budget more time towards.

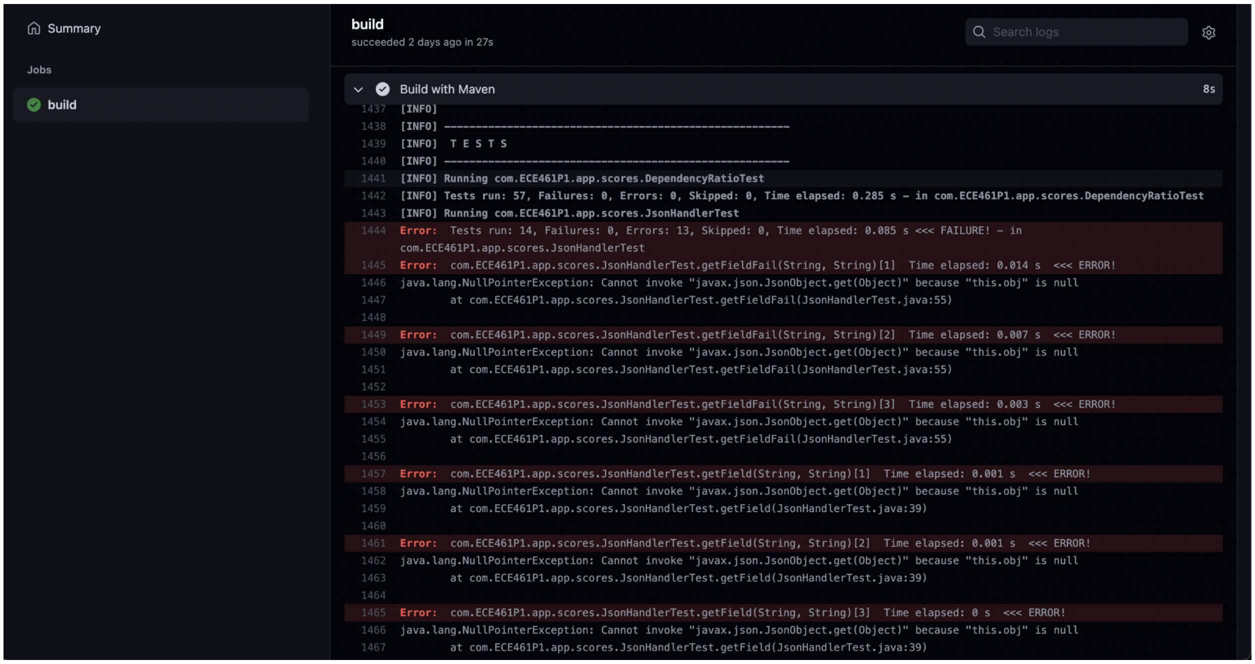
Goal	**What was accomplished**	**Owner**	**Expected completion time (hrs)**	**Actual completion time (hrs)**
Set up testing suite for	Testing framework and tests made for trustworthiness	Owen	3	4

project	module and its helper functions			
Set up github action to run tests	Set up a google compute engine instance and tested the trustworthiness module on it.	Owen	1	45 mins
Develop pinned dependency metric	Module can read a package.json file and correctly calculate a dependency ratio score for 9 different repos. It only fails 1 test case at the moment.	Owen	4	5
Research on implementing Flask	Flask is now working and deployed to the app engine. Next steps will involve figuring out how to unzip files given and how to return zipped files from the database.	Ryan	5	5
Research how to interact with GCP Storage Systems using Flask	GCP storage will be utilized to store persistent data. This is to ensure we don't lose any data in the chance our VM on the Cloud Compute Engine goes down. It has been set up to take in files sent by user on the App Engine.	Ryan	3	5
Deploying Flask builds to GCP App Engine	Deploying flask builds will be analogous to deploying any other Python project to GCP on our remote engine. This is currently up and running.	Ryan	3	4
Implemented and tested PostgreSQL	Testing Basic functionalities of PostgreSQL to understand how it will be implemented into our App engine to be used as a database to store out metric scores. This has not currently been integrated into our project because we require a JSON file containing our scores were we can read the data from. This JSON file will also be created in the coming week.	Mohammed	10	10

4. Any changes in your planned timeline as a result of falling behind your initial plan.

Yes, a few changes.

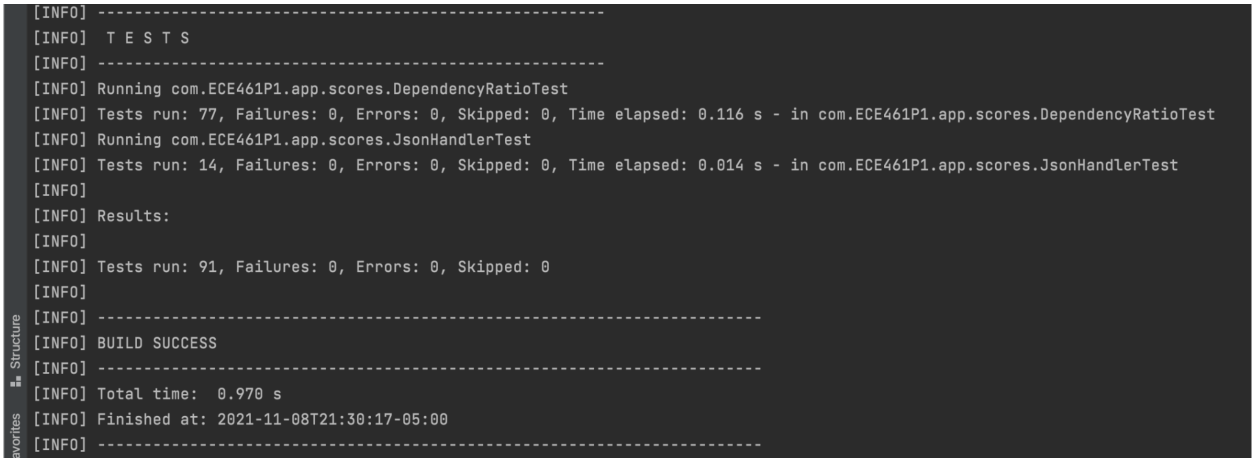
Owen: I followed the plan from last week. As a group, we learned that there was a large learning curve ahead of us, so we needed to use our time more efficiently, so we attended office hours. This definitely improved our efficiency this week. I adopted a test-driven development approach and spent the week setting up unit tests to continuously test the new features of the trustworthiness module. I am confident that the new feature of the trustworthiness module works. I spent a few minutes to develop a Github action to run unit tests on every push, but I did not integrate that into the repository due to the current limitations on Github actions. Instead, I tested them on a dummy repository.



build
succeeded 2 days ago in 27s

Build with Maven 8s

```
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running com.ECE461P1.app.scores.DependencyRatioTest
[INFO] Tests run: 57, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.285 s – in com.ECE461P1.app.scores.DependencyRatioTest
[INFO] Running com.ECE461P1.app.scores.JsonHandlerTest
Error:  Tests run: 14, Failures: 0, Errors: 13, Skipped: 0, Time elapsed: 0.085 s <<< FAILURE! – in
com.ECE461P1.app.scores.JsonHandlerTest
Error:  com.ECE461P1.app.scores.JsonHandlerTest.getFieldFail(String, String)[1]  Time elapsed: 0.014 s  <<< ERROR!
java.lang.NullPointerException: Cannot invoke "javax.json.JsonObject.get(Object)" because "this.obj" is null
        at com.ECE461P1.app.scores.JsonHandlerTest.getFieldFail(JsonHandlerTest.java:55)

Error:  com.ECE461P1.app.scores.JsonHandlerTest.getFieldFail(String, String)[2]  Time elapsed: 0.007 s  <<< ERROR!
java.lang.NullPointerException: Cannot invoke "javax.json.JsonObject.get(Object)" because "this.obj" is null
        at com.ECE461P1.app.scores.JsonHandlerTest.getFieldFail(JsonHandlerTest.java:55)

Error:  com.ECE461P1.app.scores.JsonHandlerTest.getFieldFail(String, String)[3]  Time elapsed: 0.003 s  <<< ERROR!
java.lang.NullPointerException: Cannot invoke "javax.json.JsonObject.get(Object)" because "this.obj" is null
        at com.ECE461P1.app.scores.JsonHandlerTest.getFieldFail(JsonHandlerTest.java:55)

Error:  com.ECE461P1.app.scores.JsonHandlerTest.getField(String, String)[1]  Time elapsed: 0.001 s  <<< ERROR!
java.lang.NullPointerException: Cannot invoke "javax.json.JsonObject.get(Object)" because "this.obj" is null
        at com.ECE461P1.app.scores.JsonHandlerTest.getField(JsonHandlerTest.java:39)

Error:  com.ECE461P1.app.scores.JsonHandlerTest.getField(String, String)[2]  Time elapsed: 0.001 s  <<< ERROR!
java.lang.NullPointerException: Cannot invoke "javax.json.JsonObject.get(Object)" because "this.obj" is null
        at com.ECE461P1.app.scores.JsonHandlerTest.getField(JsonHandlerTest.java:39)

Error:  com.ECE461P1.app.scores.JsonHandlerTest.getField(String, String)[3]  Time elapsed: 0 s  <<< ERROR!
java.lang.NullPointerException: Cannot invoke "javax.json.JsonObject.get(Object)" because "this.obj" is null
        at com.ECE461P1.app.scores.JsonHandlerTest.getField(JsonHandlerTest.java:39)
```



```
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running com.ECE461P1.app.scores.DependencyRatioTest
[INFO] Tests run: 77, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.116 s - in com.ECE461P1.app.scores.DependencyRatioTest
[INFO] Running com.ECE461P1.app.scores.JsonHandlerTest
[INFO] Tests run: 14, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.014 s - in com.ECE461P1.app.scores.JsonHandlerTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 91, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.970 s
[INFO] Finished at: 2021-11-08T21:30:17-05:00
[INFO] ------------------------------------------------------------------------
```

This next week will be focused on
1. Parsing JSON to get the URL from a package.json file,
2. Outputting scores as a JSON file,
3. Deploying to Google App Engine,
4. Integrating the new dependency method into the trustworthiness module.

If necessary, the integration of the dependency method will be pushed to next week.

Ryan: Although we were able to successfully get a build up and running utilizing Flask and App Engine, our project currently circumvents the use of the openAPI. This is in part due to Google Cloud Platform not supporting openAPI V3. We will investigate further how to implement the rest API with our project. One Python library of interest is the Flask-RESTful library that extends Flask to be more RESTful friendly. I will spend a significant amount of time this week getting this up and running, and speak to the TA's for guidance.