Project Milestone Template

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: 11/1/2021

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.

Owen:

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 did eventually get the trustworthiness module to work but it consumed much of our time during the week. Our timeline is set back a bit.
- 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 be using google compute engine as our backend and set up all the microservices there as our first prototype instead of attempting to use multiple google cloud services.
- 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.

These are quite substantial updates but we believe that it is important to be transparent. In the coming weeks, we will look to add some of these features back into our design if there is time available.

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

Owen	• Worked on trustworthiness module, fixed authentication error that caused the tool to not run • Looked into git actions, changed Ryan's Pylint feature to only run on Python files • Set up Google compute engine VM and successfully ran trustworthiness module on it
Ryan	• Began work on Flask for interacting with the openAPI given by course staff. • Using flask to interact with the API and respond to HTTP requests.
Mohammed	• Began research and implementation of the GCP firestore and Cloud Storage our team initially planned to use. This has now been scrapped to use SQLalchemy library on python.

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

We are not currently on track. This will most likely set us back 1-2 weeks. This is why we are dropping the front-end website requirement and authentication features

Goal	**What was accomplished**	**Owner**	**Expected completion time (hrs)**	**Actual completion time (hrs)**
Start development on project 1 module to allow it to pass entire testing suite	• Fixed issue 1, "There is some sort of authentication error when we re-build and run the project again without modification. Every git API request returns a 401 response code, causing no meaningful data to be collected." • Fixed Issue 6, where	Owen	7	7

	the executable did not properly use the log file			
Work toward a very basic prototype	Set up a google compute engine instance and tested the trustworthiness module on it.	Owen	Not scheduled	1
	Work on Github action framework for testing Java functions. Modifying timeline to set up more test-driven development tools.	Owen	Not scheduled	1
Research on implementing Flask and Postman	We will no longer be utilizing Postman for offline verification and CI/CD. We will use GCP for this, and not worry about hosting it offline using this additional tool. Flask is currently in the works to interact with the given YAML file. We will use this file to guide our development.	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.	Ryan	3	1
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.	Ryan	3	2
Research on how to setup both databases (GCP firestore and GCP Cloud storage)	Our revised plan will no longer be using either of these databases but will now be using python's SQLalchemy library.	Mohammed	5	3

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

Yes, a few changes.

Owen: I will plan to spend an extra week (~10 hrs) working on the trustworthiness module. Based on new knowledge gained over the past week while working on the module and also information presented in class, I will adopt a test-driven development approach and spend the next week setting up unit tests and Github actions to continuously deploy the trustworthiness module. By week 5, we should have a good project framework, even though the trustworthiness module may not be completed yet. I will then plan to spend that extra week implementing the new feature alongside Mohammed and validating the design.

Mohammed: Due to our overall approach being modified to using a Fat VM approach instead of the majority of the GCP components we had in our initial plan, Our database will now be set up using python's SQLalchemy library. I will need to spend some time understanding this new database setup (~2 hours). Due to the language still being python, it should not cause a substantial deviation from our initial database approach.

Ryan: As a result of us focusing less on front end portions of the project, I will spend significantly less time on front end development than was originally planned. I instead will spend that time learning the GCP and configuring the compute engine to integrate with Flask and our Python-built back end.