

CS361: Assignment 1: Microservices Warm-Up

# Overview

To demonstrate you can implement the microservices architecture, write software comprised of **three separate programs**:

1. A program that generates pseudo-random numbers (**PRNG Service**)
2. A program that, given a non-negative integer *i*, returns the *ith* image in a set (order doesn’t matter) (**Image Service**)
   * If *i* is *>*= the number of images, modulo *i* by the size of the image set
3. A user interface (**UI**) that either has a button or can receive a user command. When the button is pushed or the command is entered...
4. UI calls the PRNG Service
5. UI calls the Image Service using the pseudo-random number from the PRNG Service
6. UI displays the image (or a path to it)

The programs can be written in **any language(s)**.

Use **any set of images** (e.g., downloaded from [https://www.kaggle.com/).](http://www.kaggle.com/)) **Store images locally in a folder**; no API calls needed. No DB needed.

# How long will this assignment take?

It could take you **anywhere from 1 hour to 5+ hours** to complete this assignment depending on your comfort and familiarity with the programming language you choose to use.

# Requirements

* UI must either have a button (if UI is graphical) or be able to receive a user command (if UI is text-based)
* Each of the three programs must run in a **different process**
* Programs must **NOT call each other** directly (e.g., do not import one program into another)
* As the **communication pipe**, use text files as follows:

1. UI calls PRNG Service by writing the word "run" to prng-service.txt
2. PRNG Service reads prng-service.txt, erases it, and writes a pseudo-random number to it
3. UI reads prng-service.txt to get the pseudo-random number
4. UI writes the pseudo-random number to image-service.txt
5. Image Service reads image-service.txt, erases it, and writes an image path to it
6. UI reads image-service.txt then displays the image (or path) to the user

# Instructions

**PART 1: Plan**

After reading through the requirements and instructions, how long do you think it will take to complete the assignment and how will you go about it? Answer the following questions to start planning.

Complete each item below by replacing the highlighted text (**Usability note**: double-click the text to select it).

1. Which **programming language** will you use to complete this assignment?

|  |
| --- |
| *ProgrammingLanguage* |

1. **How familiar and comfortable** are you with this programming language? For example, when was the last time you used it?

|  |
| --- |
| *DescriptionOfComfortLevel* |

1. Based on the above and other relevant factors, **how long** do you think it’ll take to complete the assignment? Explain your answer. Track how long it actually takes to complete the assignment (you’ll be asked about this in Part 3).

|  |
| --- |
| *HoursEstimate* |

Now that you have a time estimate, I recommend you double it! That’ll help give you a time buffer in case you run into issues.

I also recommend scheduling when you’ll do the work. How many work sessions will you need? On which days will you work? How will this work fit around the other work you need to do?

**Tip**: One easy way to schedule your time is to create a text file that starts with today and proceeds chronologically. Example:

A Sublime Text screenshot with the following text.

March 6: Do the CS361 stuff you said you'd do

March 7: Respond to boss

March 7: Ask J to make a spreadsheet of authors and resources
 Skip research papers

March 7: Work on the START HERE page of OERCommons
 Schedue a second work session with J

March 7: Work on Week 1 CS565 Video

I like using Sublime Text for this because it starts up quickly and keeps my tabs open (including tabs of unsaved work) 🙂

**PART 2: Create Video**

* Create a **short video** (5 minutes or less) demonstrating you have satisfied the requirements.

**PART 3: Reflect**

Now that you’ve completed the assignment...

How long did it **actually** take you to complete the assignment?

|  |
| --- |
| *HoursActual* |

This reflection is meant to help you understand how accurate your time estimates tend to be so that you can factor that information into future time estimates. You won’t be asked to track your time spent on future assignment but you might find it helpful to do so on your own

# Submission

* Attach PDF or Word document. You MUST follow the instructions at **Modules > 'HOW TO: Attach a Document to "Text Entry" Field'.**
* Embed or link to **video**. You MUST follow the instructions at **Modules > “HOW TO: Create and Upload a Video”.**

# Grading

You are responsible for satisfying all criteria listed in the Canvas rubric for this assignment. You will be able to revise this assignment if you miss points.

# Questions?

Please ask via Ed so that others can benefit from the answers.