Oregon State University
Ecampus

CS361: Sprint 3 Assignment - Person Generator Project: Implementation Begins

Overview

Your client is eager to see the software working! They selected some functionality that, once implemented, will be a proof-of-concept.

- First, finish writing a formal use case for the functionality. This can help with planning implementation.
- Second, update your sequence diagram to match the use case. This can help with understanding how the parts of your program will interact.
- Third, update tasks in your task management system as needed to plan your work.
- Fourth, start programming! Implement the functionality your client requested.

Instructions

Create a PDF for the non-programming part of the assignment and a Python file for the programming part. No need to submit input/output files (we'll use our own).

1. Put your project name and your name at the top of the PDF
2. Submit your revised formal use case (the one you started during discussion)
    - Clearly indicate revisions you made
    - You must have made at least one revision
    - Clearly explain why you made the revision
3. Submit your revised UML sequence diagram (the one you started during discussion)
    - Clearly indicate revisions you made
    - You must have made at least one revision
    - Clearly explain why you made the revision
4. Update your tasks in your task management system (as needed)
    - Make sure each task a deadline and description
    - Label each task with the user story identifier provided by your client
    - Label each task "Sprint 3" (so that you can keep track of how much progress you made each Sprint)
    - Submit screenshot
5. Implement functionality requested by client (see below)

Sprint 3 Functionality Requested by Client

Make a Python 3 program that outputs a given number of street addresses for the selected state.

1. Make a Tkinter GUI that has...
   - A way to select a US state (only from the states in the dataset: `https://www.kaggle.com/openaddresses/openaddresses-us-west`)
   - A way to select the number of street addresses to select pseudo-randomly from the dataset
   - A button for generating the output
   - An area on the GUI that shows the output
2. Make it so the program can also receive CSV input when it starts
   - Required format for starting program: `python person-generator.py input.csv`
   - Required format for input.csv (with header): `input_state,input_number_to_generate`
3. Make it so the program creates output.csv when the button is pushed or when input.csv is sent (when program starts)
   - Required format for output.csv (with header): `input_state,input_number_to_generate,output_content_type,output_content_value`
   - output_content_type = street address
   - Output to same directory as program

Data must be from `https://www.kaggle.com/openaddresses/openaddresses-us-west`

Submission

Submit Python file and PDF via Canvas

Questions?

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