The University of Western Ontario
Department of Computer Science
CS2212b 2015

Team Project
Stage 4 or Final Report

In this final stage, you will clean up your documentation to professional quality, create a video demo, and make a README.md for your GitHub page to orient other developers.

DUE: April 8th at Midnight

Github Wiki

Should be complete as of Stage 3

Documentation

Your code must be documented using JavaDoc-style comments for all classes and their methods. You must generate the JavaDoc html files for all members (not just public ones), add and commit them for your tagged submission.

In this stage you will be evaluated on the quality of the documentation.

Video Demonstration

You are to prepare a 2-4 minute narrated screencast demonstrating the various features of your application. This screencast is expected to be professionally narrated and edited, and should be of high video and audio quality. The intent is to have a professional video to which you can point employers for an example of your work.

- You **must** upload the video to YouTube. You may make the video an unlisted video and then provide us with the link. You **may not** make the video private, however, since we will not be able to access it.
- You **must** leave the video on YouTube for at least 4 weeks. Ideally, you would leave it up indefinitely.

- You do **not** have to personally identify yourselves in the video, but you should introduce yourself using the following language:
  - *We are members of team N in the second-year Introduction to Software Engineering course in the Department of Computer Science at Western University, in London, Ontario, Canada.*
- You should then briefly discuss the development process.
  - Coded in Java using ___ IDE(s)
  - Used git for version control, and GitHub to host the project
  - Maven for build automation
- You should then show the functionality of your project, starting with the base functionality, then running through each additional component you have completed. Show that data persists between runs.
- Your video **must** be recorded in HD resolution (at least 720p).
- Your video **must** be narrated and the audio **must** be of good quality.
- You are **not** expected to use a program like Adobe After Effects to do advanced post-processing, but you are expected to have a video free of errors.

Things that will lose you marks:

- Not being 2 minutes or exceeding 4 minutes in length.
- Maintaining a pace that is either too slow (boring) or too fast (confusing).
- Poor quality video/audio.
- Low audio levels or digital distortion caused by levels that are too high.
- Mistakes in your video that have not been edited out.
- Really obvious edits – when editing something out, you should do so in a manner that is transparent to the viewer.
- *Dead air* in the video (use transitions to “fast-forward" between the start and end of time-consuming tasks that are not relevant to the video).
- Using unprofessional transitions, such as *Blinds*, *Curtains*, etc.
- Sounding completely unscripted (e.g. lots of stuttering, “Uhh", “Um", etc.).
- Sounding completely scripted and unnatural (i.e. read, but don’t sound like you’re reading).
- Typing in gibberish or inappropriate/humourous/unprofessional data when performing your demo.
- Having all sorts of other programs open.
- Having an obnoxious wallpaper or names visible on your desktop, or a messy desktop. Show good computer hygiene.

Project README.md

You should generate a readme for new developers hoping to use or contribute to your project. A readme is intended to quickly orient other *developers* into your project. (Whereas your wiki was for more extensive information).

Successful projects have good READMEs! See GitHub repos: johndyer/mediaelement, Zulko/moviepy and Netflix/Hystrix for some examples.

- Title
- Synopsis
    - a short (1-2 sentences) description of what the program does
    - a short description of how the project is implemented. Include details: like that it connects to openweathermap.org's API (and include a link), any libraries used, etc.
- Install
    - How would one go about downloading, installing and using the software? (add a version of the jar to your repo)
- Build
    - How would one go about building the software?
    - List any dependencies/programs that are required
- Usage example
    - provide a link to the video you made showing the project in action
- Documentation
    - direct developers to the JavaDoc you generated