

CS Capstone Design

Technical Demo Grading Sheet (100 pts)

TEAM: Fossilized Containers

Overview: The main purpose of the “Technical Demos” is to very clearly communicate the extent to which the team has identified key challenges in the project, and has proven solutions to those challenges. Grading is based on how complete/accurate the list of challenges is, , and how convincingly and completely the given demos cover the given challenges.

This template is fleshed out by the team, approved by CS mentor, and brought to demo as a grading sheet.

Risky technical challenges

Based on our requirements acquisition work and current understanding of the problem and envisioned solution, the following are the key technical challenges that we will need to overcome in implementing our solution:

C1: Containerizing reconstruction model (RM) The team needs knowledge of Docker containers before they can help scientists build their own. We will build a container of the reconstruction model from the beginning. We will demonstrate capability by creating a container that shows the RM accurately.

C2: Installable CLI There needs to be an easy way that users can download and install the CLI. We will demonstrate capability by having the CLI easily be installable by users.

C3: Prompts for users in CLI Users are not meant to know how to create the Dockerfiles needed to run a container. This is where the team will have prompts that ask users questions about their mode. We will demonstrate capability by adding functionality to CLI that asks users questions and creates a Dockerfile

C4: Containerization through CLI Users are not meant to know how to create and run Docker containers. The team will essentially run all of the Docker commands for the user. We will demonstrate capability by adding functionality to the CLI that runs a container of a user’s RM

C5: Sending LiPD files through HTTP One of the main features is that users can send new data to their RMs. We will demonstrate capability by sending LiPD files to a local server from the CLI

C6: Receiving NetCDF files through HTTP One of the main features is that users will receive output files from the RMs in the form of NetCDF files. The users can run a command that looks at a server for new NetCDF files. We will demonstrate capability by receiving NetCDF files from a local server using the CLI

C7: Sample documentation on Github Pages Users will need to learn how to use the CLI easily and effectively. The team will provide high quality documentation about the CLI in a reliable and accessible way. We will demonstrate capability by

Challenges covered by demos:

In this section, we outline the demonstrations we have prepared, and exactly which of the challenge(s) each one of them proves a solution to.

Demonstration 1: Containerized Reconstruction Model

Challenges addressed: **C1 (Containerizing RM)**

Flight Plan: Step by step overview of demo

1. Downloading the LMR Turbo model
2. Download Ubuntu image from Dockerhub
3. Create Dockerfile for LMR Turbo model
4. Create the image of LMR Turbo
5. From the image, run the LMR Turbo model in a container
6. Check that the container is running properly (Docker containers command)
7. View the model
8. Exit the model
9. Stop and delete the container
10. Recreate the container from the already created image

Evaluation:

- ✓ Convincingly demoed each of listed challenges?
- ✓ Other evaluative comments:

Demonstration 2: Creating a Container with the CLI

Challenges addressed: **C2 (Install CLI), C3 (User Prompts in CLI), C4 (Containerization through CLI), C7 (Documentation on Github Pages)**

Flight Plan: Step by step overview of demo

1. Download and install the CLI
2. View the documentation on Github pages and see what steps you need to take
3. Run the CLI with the Dockerfile creation using the steps from the documentation
4. Step through the prompts and view the created Dockerfile
5. Run the CLI with the container creation by passing the newly created Dockerfile as specified in the documentation
6. Check if the container was created
7. View the newly created container
8. Stop and destroy the container

Evaluation:

- ✓ Convincingly demoed each of listed challenges?

- ✓ Other evaluative comments:

Demonstration 3: File Transfer

Challenges addressed: **C5 (Sending LiPD files), C6 (Receive NetCDF files)**

Flight Plan: Step by step overview of demo

1. Start a local server containing sample NetCDF files
2. Send LiPD files to the server using the CLI
3. Check that the LiPD files are in the server
4. Receive NetCDF files from the server using the CLI
5. Check that the files were received correctly from the server

Evaluation:

- ✓ Convincingly demoed each of listed challenges?
- ✓ Other evaluative comments:

Other challenges recognized by not addressed by demo:

If there were challenges you listed earlier that were *not* covered by a demo, list here. This will hopefully be a short list...but better to be clear about where you are. If you have items here, you could list (if applicable) any pending plans to reduce these risks.