Virtual Development Environment in a Box Milestone 1

Task Matrix

Task	Completion	Ian	Dylan	Todo
Compare and Select Technical Tools	100%	66%	33%	
"Hello World" Demos	100%	33%	66%	
Resolve Technical Challenges	100%	66%	33%	
Compare and Select Collaboration Tools	100%	55%	45%	
Requirement Document	100%	50%	50%	
Design Document	100%	10%	90%	
Test Plan	100%	90%	10%	

Compare and Select Tools

Selecting the technical tools for use

- **User Interface**
  CLI (Bash)
  GUI (Python)

- **Containerization**
  Qemu

- **Container Distribution**
  HTTP Web Server
  Local File Transfer

Hello World Demos

Creating basic demos for testing functionality

Containerization

Creating a basic container

Container Sharing

Sharing containers across computers
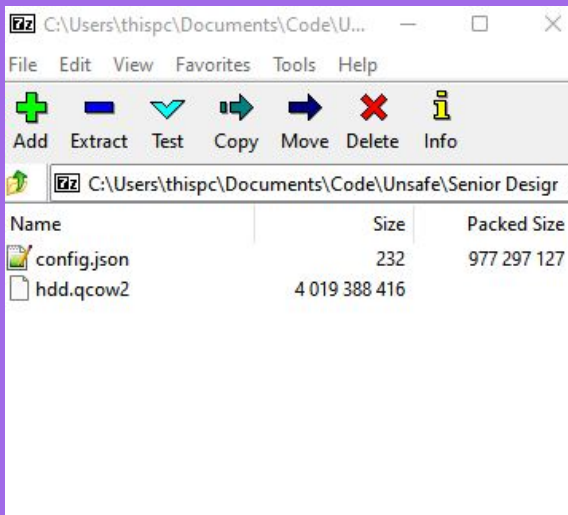
Developer Tools

Running developer tools in a container

Hello World Demos

Containerization

```
       ___                 root@debian
     (..| |                host    root@debian
     (<> |                 kernel  2.6.18-6-sparc32
    / __  \                uptime  3m
   ( /  \ /|               pkgs    237
   _/\ __)/_)              memory  6M / 468M
    \/-____\/

debian:~#
```

Container Sharing

```
7z  C:\Users\thispc\Documents\Code\U...        —    □    ✕
File  Edit  View  Favorites  Tools  Help

 Add   Extract   Test   Copy   Move   Delete   Info

 7z  C:\Users\thispc\Documents\Code\Unsafe\Senior Desigr ⌄

Name                          Size        Packed Size
config.json                    232         977 297 127
hdd.qcow2              4 019 388 416
```

Developer Tools

```
debian:~# cat hello.c
#include <stdio.h>
int main(int argc, char *argv[])
{
        puts("Hello, World!");
        return 0;
}
debian:~# gcc hello.c -o hello
debian:~# file hello
hello: ELF 32-bit MSB executable, SPARC, versi
on 1 (SYSV), for GNU/Linux 2.4.1, dynamically
linked (uses shared libs), for GNU/Linux 2.4.1
, not stripped
debian:~# ./hello
Hello, World!
debian:~#
```

Resolve Technical Challenges

Resolve technical issues that could occur

Learn Qemu

Know how to make containers with qemu

Compiler Theory

What needs to be in a container for Compiler Theory

Multi-Platform

Support Windows, MacOS, and Debian

Compare and Select Collab Tools

Select tools to be used for collaboration.

Software Development
Git
GitHub
Visual Studio Code
PyCharm

Documents
Google Docs
Microsoft Office
Latex

Communication
Discord
Microsoft Teams
Slack

Calendar
Trello
Jira

Documents

Requirements

- Created requirements for the following parts:
  - Container Manager
  - Container Storage
  - Container Creation
- Created functional and performance requirements

Design

- Created a system architecture diagram
- Created a mock-up of the command line interface
- Described the composition and functionality of containers
- Created a diagram pertaining to container distribution

Test

- Created tests for every requirement
- Created expected inputs and outputs for each test
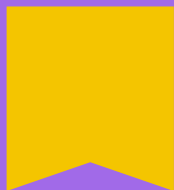- Tests involve multiple part of project working together
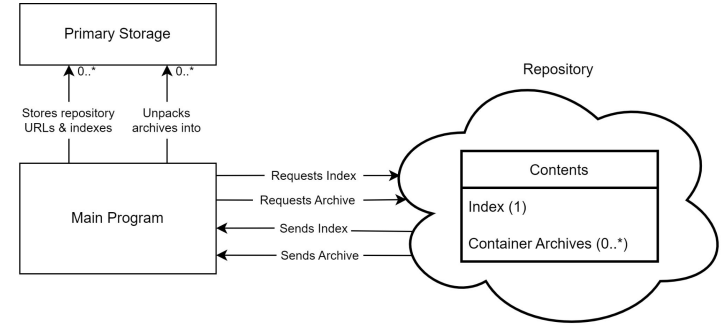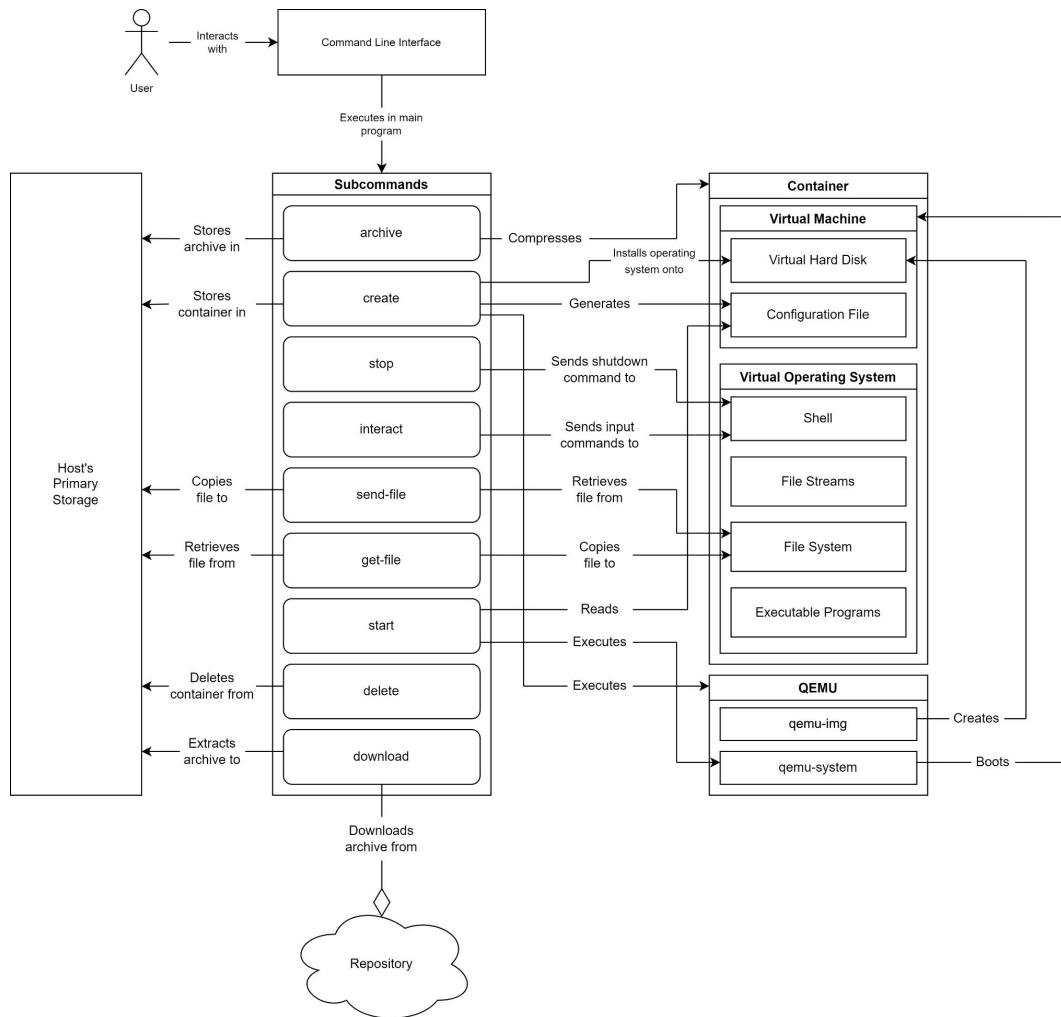
Next Milestone

- Basic Qemu Image for Compiler Theory
- Run Commands and Provide/Receive Standard Output/Input
- Import and Export Files from an Image

Questions?

Thank you!