

Project and Testing Report

Project Overview

Objective

The primary goal of this project is to create a set of Python-based GUI tools for:

1. Managing Docker operations (e.g., creating Dockerfiles, building images, managing containers).
2. Searching, pulling, and managing Docker images.
3. Interactively creating and managing virtual machines using QEMU.

Modules Overview

1. Docker Management Tool (app.py)

This module provides an intuitive GUI for managing Docker resources. Key functionalities include:

- Creating Dockerfiles for Python or Java applications.
- Building Docker images.
- Managing Docker containers (creation, listing, and stopping).

2. Docker Image Search Tool (images.py)

This module allows users to:

- Search for Docker images on DockerHub.
- Search for locally stored Docker images.
- Pull images from DockerHub.

3. Virtual Machine Management Tool (VM.py)

This module provides a GUI for interactively creating and managing virtual machines using QEMU. Key functionalities include:

- Creating virtual disks.
- Launching VMs with specified configurations (name, memory, disk size).

Testing Report

Testing Environment

- **Operating System:** Windows 10/11
- **Python Version:** 3.9+

- **Docker Version:** 20.10+
- **QEMU Version:** 7.2.0+
- **Libraries Installed:** tkinter, docker, requests

Test Cases

1. Docker Management Tool

Test Case	Steps	Expected Result	Status
Create Dockerfile	Select directory, choose language, and confirm.	Dockerfile created successfully in the selected directory.	Passed
Build Docker Image	Select directory with Dockerfile, provide image name and tag.	Docker image built successfully.	Passed
Create Container	Provide image name and container name.	Container created and started successfully.	Passed
List Docker Images	Click the "List Docker Images" button.	All locally available Docker images are displayed.	Passed
List Running Containers	Click the "List Running Containers" button.	All active containers are listed with options to stop them.	Passed
Stop a Container	Provide the container ID to stop.	The container stops successfully, and a confirmation is displayed.	Passed

2. Docker Image Search Tool

Test Case	Steps	Expected Result	Status
Search on DockerHub	Enter image name and search.	Matching images on DockerHub are displayed.	Passed
Search Local Storage	Enter image name and search locally.	Matching local Docker images are displayed.	Passed
Pull Image from DockerHub	Enter image name and pull it.	Docker image is pulled successfully from DockerHub.	Passed

3. Virtual Machine Management Tool

Test Case	Steps	Expected Result	Status
Create Virtual Disk	Provide disk name and size, confirm creation.	Virtual disk is created in the working directory.	Passed

Test Case	Steps	Expected Result	Status
Launch VM with ISO	Provide VM details (name, memory, disk size, ISO path).	VM is launched successfully with the specified configuration.	Passed
Invalid ISO Path	Provide incorrect ISO file path.	Error message displayed, VM creation aborted.	Passed

Error Handling and Edge Cases

- **Invalid Inputs:**
 - Docker image names and tags are validated before proceeding with actions.
 - VM configurations are checked for completeness and validity.
- **Missing Dependencies:**
 - Error messages are displayed if Docker or QEMU is not installed.
- **Network Issues:**
 - DockerHub search or pull operations fail gracefully with appropriate error messages.

Performance Metrics

Metric	Observed Value
GUI Responsiveness	All operations responded within 1 second.
Docker Image Build Time	Dependent on image size, averaged 20 seconds for test images.
VM Creation and Boot Time	Virtual disk creation and VM boot took 1-2 minutes.

Conclusion

The tools developed are functional and provide a user-friendly interface for Docker and QEMU management. All test cases passed successfully, ensuring reliability and usability. Future enhancements could include:

- Adding support for more virtualization platforms.
- Expanding Docker operations (e.g., logging, advanced container configurations).
- Providing additional error diagnostics and logging mechanisms.