Summer 2021 – DevSecOps
ADF
Pipeline
CaPPMS

Test Report

Version 1.0

July 26, 2021

Jeroen Soeurt,
Michelle Monfort,
Robert Wilson

Table of Contents

1. Introduction

1.1 Purpose

This document highlights the test results of the test cases identified by the test plan. The test plan can be found in the Project Plan and the TDD. There are the major sections to the test plan as there are three major areas that the DevSecOps is responsible for. The three sections covered address the functional areas that represent build pipelines, Advance Development Factory (ADF), and Capstone Project Management System (CaPPMS).

Section two of this document will list the test cases for all functional areas. Build pipelines in 2.2, ADF will be listed in 2.3, and CaPPMS in 2.4. The results are broken down into individual sections of this document. Build pipelines in section 3, ADF results will be in section 4, and CaPPMS will be found in section 5. Section will be the last section of the document.

1.2 Scope

The document covers a wide area of technologies, platforms, and intent of end-users. With that each section has a very specific set of tests. For the build pipeline, the tests are centered around the accomplishment of the end results of the technology and not the technology itself. ADF tests functional requirements across three different platforms ensuring a similar experience for end-users regardless of their own experience. And CaPPMS, the website tests are written to test validation mechanisms of potential inputs of users. Dynamic tests have been written in other areas of CaPPMS to test hidden items of the system, items that a programmer might not think twice about changing but can have unexpected impact in the overall usability of the web application.

2. Testing

2.1 Test Strategy

The scope of tests is focused on customer usability. With a wide variety in skill and computer know-how our tests have to be able to identify who can and cannot be reached so that recommendations and possible alternatives can be assessed for the user's and programmer's guides. Here are the test axes that the DevSecOps team focused on: The premise of the tests will be conducted as functional tests that cover the following axes:

- Functionality
- External Interfaces
- Input Space
- Output Space
- Configuration elements
- Performance

The pipelines are the easiest to test and arguably the most important. The simple script ensures that the development team's project is analyzed, built, tested, and uploaded as artifacts for deployment cases. The simple configuration file can become completely unusable with a simple change by the user or an update to a dependency. As such, the pipeline needs to be monitored for correctness against the set criteria. One thing to note, the build pipeline sends notifications to the DevSecOps channel in MS Teams keeping the widest audience informed.

For ADF, the test must cover the use cases of three operating systems. Even though Docker is known to be a bridging solution, experience has given us the reality that it is not. The host has the final say in how compatible a docker image is. With this in mind the core functionality will be tested against MacOS, Linux, and Windows.

CaPPMS is meant to be used by the casual user. With this in mind, basic functionality and validation of inputs were tested. It is equally important the website stays simple and attempt to keep data secured and validated before making the record permanent.

2.2 Tests Conducted

Build Pipeline

Requirement #	Requirement
BR1	Charlie Pipeline builds successfully
BR2	Charlie Pipeline Lint flutter code successfully
BR3	Charlie Pipeline Runs Unit Tests successfully
BR4	Charlie Pipeline builds Artifact successfully
BR5	Charlie Artifact is downloadable
BR6	Bravo Pipeline builds successfully
BR7	Bravo Pipeline Lint flutter code successfully
BR8	Bravo Pipeline Runs Unit Tests successfully
BR9	Bravo Pipeline builds Artifact successfully
BR10	Bravo Artifact is downloadable
BR11	Amazing Pipeline builds successfully
BR12	Amazing Pipeline Lint flutter code successfully
BR13	Amazing Pipeline Runs Unit Tests successfully
BR14	Amazing Pipeline builds Artifact successfully
BR15	Amazing Artifact is downloadable

ADF

Requirement #	Requirement
AR1	The environment starts up successfully.
AR2	The environment is accessible over RDP.
AR3	The environment has access to the internet.
AR4	The environment provides VS Code.
AR5	The environment provides an Android emulator.
AR6	The environment provides a Git client.
AR7	The environment provides Flutter/Dart CLI tools.

CaPPMS

Requirement #	Requirement
CR5.1	The system shall be able to produce a table of represent data
CR5.2	The system shall maintain records in persistent storage
CR5.3	The system shall produce a PDF document that represents the project
CR5.4	The system shall limit authorization to managing the project list
CR5.5	The system shall limit authorization to managing the users access list
CR5.6	The system shall maintain records that represent frequently asked questions
CR5.7	The system shall maintain records that represent a list of projects

3. Function Test Execution Findings - Pipeline

All tests are functional manual tests. Manual steps are assessed to accommodate a lack of technology available to test these requirements. The project implements cutting-edge technology and the supporting elements for testing are not yet available.

Test ID #	Requirement #	Input	Expected Output
1	BR1	See GitHub Actions Under the left side filter through workflows Click the stable build Click most recent workflow Click Build Project	Artifacts Produced during runtime: => Set up job => Run actions/checkout@v2 => Set up Flutter => Run flutter pub get => Build APK => Upload APK Artifact => Upload Web Artifacts => Post Run actions/checkout@v2 => Complete job
2	BR2	See GitHub Actions Under the left side filter through workflows Click the stable build Click most recent workflow Click Lint flutter code	Artifacts Produced during runtime: => Set up job => Checkout code => Set up Flutter => Run flutter pub get => Analyze Flutter => Post Checkout code => Complete job
3	BR3	See GitHub Actions Under the left side filter through workflows Click the stable build Click most recent workflow Click Run Unit Tests	Artifacts Produced during runtime: => Set up job => Run actions/checkout@v2 => Set up Flutter => Run flutter pub get => Run tests => Post Run actions/checkout@v2 => Complete job

4	BR4	See GitHub Actions Under the left side filter through workflows Click the stable build Click most recent workflow Scroll to last component of page: Artifacts	**Artifacts** Produced during runtime: Name: mnemosyne-apk-release Size: 45.4 MB
5	BR5	See GitHub Actions Under the left side filter through workflows Click the stable build Click most recent workflow Scroll to last component of page: Artifacts Click Artifact name	Download Zip archive prompt Save As: mnemosyne-apk-release
6	BR6	See GitHub Actions Under the left side filter through workflows Click the stable build Click most recent workflow Click Build Project	Artifacts Produced during runtime: => Set up job => Run actions/checkout@v2 => Set up Flutter => Run cd ./harikfy => Build APK => Upload APK Artifact => Upload Web Artifacts => Post Run actions/checkout@v2 => Complete job
7	BR7	See GitHub Actions Double click most recent workflow Double click Lint flutter code. See GitHub Actions Under the left side filter through workflows Click the stable build Click most recent workflow Click Lint flutter code	Artifacts Produced during runtime: => Set up job => Checkout code => Set up Flutter => Run cd ./harikfy => Run cd ./harikfy => Analyze Flutter => Post Checkout code => Complete job
8	BR8	See GitHub Actions Under the left side filter through workflows Click the stable build	Artifacts Produced during runtime: => Set up job => Run actions/checkout@v2

		Click most recent workflow Click Run Unit Tests	=> Set up Flutter => Run cd ./harikfy => Run tests => Post Run actions/checkout@v2 => Complete job
9	BR9	See GitHub Actions Under the left side filter through workflows Click the stable build Click most recent workflow Scroll to last component of page: Artifacts	**Artifacts** Produced during runtime: Name: harkify-apk-release Size: 48.4 MB
10	BR10	See GitHub Actions Under the left side filter through workflows Click the stable build Click most recent workflow Scroll to last component of page: Artifacts Click Artifact name	Download Zip archive prompt Save As: harkify-apk-release
11	BR11	See GitHub Actions Under the left side filter through workflows Click the stable build Click most recent workflow Click Build Project	Artifacts Produced during runtime: => Set up job => Run actions/checkout@v2 => Set up Flutter => Run cd ./memory_enhancer_app => Build APK => Upload APK Artifact => Post Run actions/checkout@v2 => Complete job
12	BR12	See GitHub Actions Under the left side filter through workflows Click the stable build Click most recent workflow Click Lint flutter code	Artifacts Produced during runtime: => Set up job => Checkout code => Set up Flutter => Run cd ./memory_enhancer_app => Analyze Flutter => Post Checkout code => Complete job

13	BR13	See GitHub Actions Under the left side filter through workflows Click the stable build Click most recent workflow Click Run Unit Tests	Artifacts Produced during runtime: => Set up job => Run actions/checkout@v2 => Set up Flutter => Run cd ./memory_enhancer_app => Run tests => Post Run actions/checkout@v2 => Complete job
14	BR14	See GitHub Actions Under the left side filter through workflows Click the stable build Click most recent workflow Scroll to last component of page: Artifacts	**Artifacts** Produced during runtime: Name: memory_enhancer_app-apk-debug Size: 48.4 MB
15	BR15	See GitHub Actions Under the left side filter through workflows Click the stable build Click most recent workflow Scroll to last component of page: Artifacts Click Artifact name	Download Zip archive prompt Save As: memory_enhancer_app-apk-debug

No Significant Results.

4. Function Test Execution Findings – ADF

Tests Conducted

All tests are functional tests, as we are not aware of any automated testing tool that is capable of testing an operating system of an RDP connection running in a Docker container. If a HTML5-based RDP client is used, Selenium could be an option. This is a future possibility, but outside the scope of this current iteration.

Test ID	Requirements	Input	Expected Output
1	AR1	Run the following commands: git clone https://github.com/umgc/ADFSummer2021.git cd .\ADFSummer2021\ docker build --pull --rm -f "ADF/dockerfile" -t adfsummer2021:latest "ADF"	Docker image build, and completes with: => exporting to image => => exporting layers => => writing image sha256:XXXXXXXXXXXXXXXXXXX => => naming to docker.io/library/adfsummer2021:latest
2	AR2	Run the following command, and try to connect to localhost:63389 using RDP. docker run -dit -p 63389:3389 --rm -- privileged adfsummer2021:latest	RDP client should connect and a login screen should show up.

3	AR2	RDP is connected to the container, and shows the login screen. User enters: Username: developer Password: password	User should log in and desktop should be shown.
4	AR3	User open a new terminal window and issues the following commands: sudo apt-get update sudo apt-get install iputils-ping ping google.com	Google.com should respond, Ie: developer@3bcdd837d177:~$ ping google.com PING google.com (142.250.184.238) 56(84) bytes of data. {returned ip address = 142.250.184.238 but ttl and response time may vary from tests to test} --- google.com ping statistics --- 2 packets transmitted, 2 received, 0% packet loss, time 1001ms

			rtt min/avg/max/mdev = 5.308/5.776/6.244/0.468 ms
5	AR4	User clicks on VS Code icon on desktop.	VS Code screen appears.
6	AR6	User opens a new terminal and issues the following command: git --version	Git version appears, for example: git version 2.25.1
7	AR7	User opens a new terminal and issues the following command: flutter --version	Flutter and Dart versions appears, for example: Flutter 2.2.1 • channel stable • https://github.com/flutter/flutter.git Framework • revision 02c026b03c (8 weeks ago) • 2021-05-27 12:24:44 -0700 Engine • revision 0fdb562ac8 Tools • Dart 2.13.1

8	AR5	User opens a new terminal and issues the following command: flutter emulators --launch flutter_emulator	Emulator appears and starts Android.
9	AR5	User opens a new terminal and issues the following command while the emulator is running: flutter run	App builds and runs in the emulator

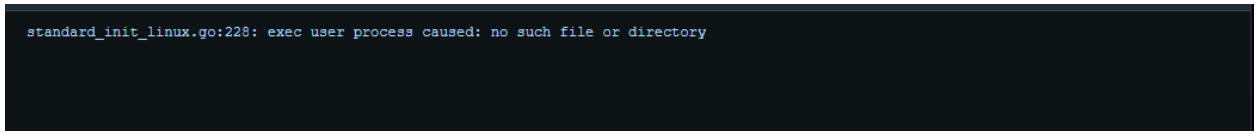
Test Case: 2
Failing Requirement: Run the following command, and try to connect to localhost:63389 using RDP.
docker run -dit -p 63389:3389 --rm --privileged adfsummer2021:latest

Expected Output: RDP client should connect and a login screen should show up.

Actual Output: None

Defect: RDP client does not show. Checked the Docker Dashboard and saw the container was not running. Attempted to start the container. Looks like the EOL needs to be set to LF. After fixing the EOL the remote session did not start but the container did.

Additional step: Use RDP tool to connect to localhost:63389
Screenshot of failure:



Requirement #	Requirement
#fail 1	The environment starts up successfully.
Pass 2	The environment is accessible over RDP.
Pass 3	The environment has access to the internet.
Pass 4	The environment provides VS Code.
#fail 5	The environment provides an Android emulator.
Pass 6	The environment provides a Git client.
Pass 7	The environment provides Flutter/Dart CLI tools.

Mac Results of ADF Test
Test Case: 1
Failing Requirement: The environment starts up successfully.
Expected Output: Docker image build, and completes with:

=> exporting to image

=> => exporting layers

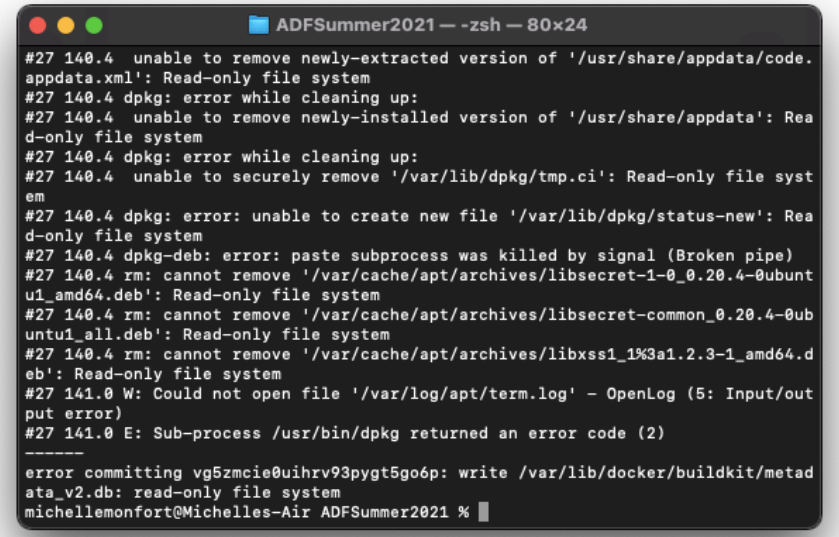
=> => writing image sha256:XXXXXXXXXXXXXXXXXXX

=> => naming to docker.io/library/adfsummer2021:latest
Actual Output: => [14/72] RUN apt-get install -y qemu qemu-kvm libvirt-daemon-system l 54.5s
=> [15/72] RUN apt-get install -y --no-install-recommends --allow-unaut 15.2s
=> [16/72] RUN sudo adduser xrdp ssl-cert 0.6s
=> [17/72] RUN apt-get install --yes --no-install-recommends openjdk-8- 78.0s
=> [18/72] RUN echo "ALL ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers 3.4s
=> [19/72] RUN groupadd --gid 1000 developer && useradd -s /bin/bash - 3.5s
=> [20/72] RUN usermod -aG sudo developer 3.8s
=> [21/72] WORKDIR /home/developer 0.3s
=> [22/72] RUN update-alternatives --set x-terminal-emulator /usr/bin/xf 1.9s

=> [23/72] RUN wget "https://go.microsoft.com/fwlink/?LinkID=760868" -O 25.0s
=> ERROR [24/72] RUN sudo apt-get install -y ./vscode.deb 141.4s

Defect: Docker does not build image
.= > ERROR [internal] load build definition from dockerfile
=> ERROR [internal] load .dockerignore

Additional step: Use RDP tool to connect to localhost:3389
Screenshot of failure:

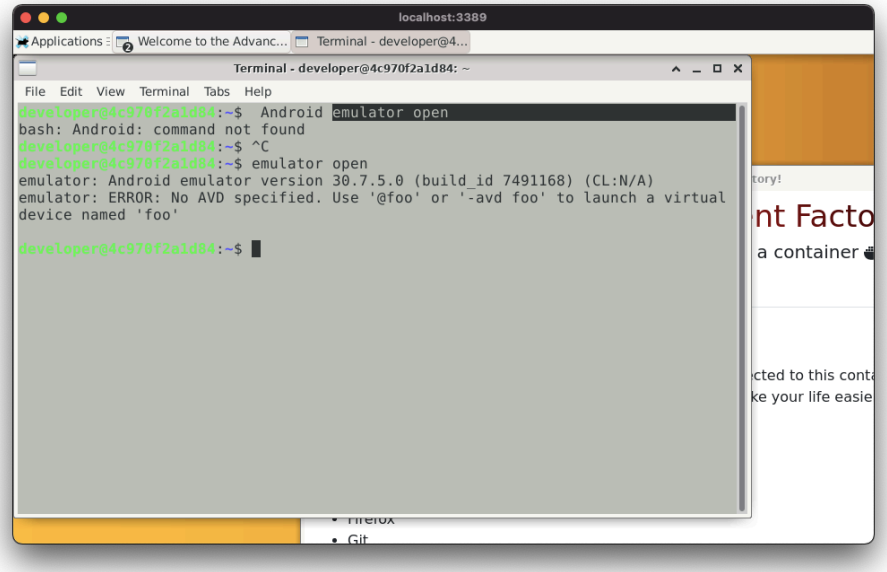


5	The environment provides an Android emulator.

Test Case: 5
Failing Requirement: 5 The environment provides an Android emulator.
Expected Output: The environment provides an Android emulator.

Actual Output: emulator: Android emulator version 30.7.5.0 (build_id 7491168) (CL:N/A)
emulator:
ERROR: No AVD specified. Use '@foo' or '-avd foo' to launch a virtual device named 'foo'

Defect: No android emulator found
Screenshot of failure:



5. Function Tests Execution Findings – CaPPMS

Tests Conducted

Test IDs marked with a CTU leading id are automated unit tests. Tests marked with a CT leading id were manual steps. The manual steps are assessed to be a gap in technology. This has happened to due to the cutting-edge technology the project was written in and the supporting elements still catching up to automate the rest of the workflows.

Test ID	Requirements	Input	Expected Output
CTU1	CR5.3	object with ExportAttribute	Attribute asserts false
CTU2	CR5.1,CR5.2,CR5.7	object with ColumnHeaderAttribute	Object asserts to having n column header
CTU3	CR5.1,CR5.2,CR5.7	object with ColumnHeaderAttribute	Object asserts to have expected column names
CTU4	CR5.1,CR5.2,CR5.3,CR5.7	ienumerable object with at least one value	n enumerated value equal expected r value
CTU5	CR5.2,CR5.7	user file to remove from system	file is removed
CTU6	CR5.2,CR5.7	user file to save to system	file is saved
CTU7	CR5.2,CR5.7	user file can be read from system	file can be read
CTU8	CR5.1,CR5.2,CR5.2,CR5.7	a test idea	assert the can be found in memory space
CTU9	CR5.1,CR5.2,CR5.2,CR5.7	a test idea	assert the idea can be found in memory space
CTU10	CR5.1,CR5.2,CR5.2,CR5.7	a test idea	assert the idea the backing physical file as the idea
CTU11	CR5.1,CR5.2,CR5.2,CR5.7	a test idea, then a change to the idea	assert the idea is verified to be updated
CT12	CR5.2,CR5.3,CR5.7	an idea to export that has comments	assert no comments are on the generated pdf, assert attached file can be downloaded, assert all other content matches idea, assert umgc header and footer present
CTU13	CR5.1,CR5.2,CR5.3,CR5.7	comment=$"Test comment-{Guid.NewGuid()}";	assert n idea has comment
CTU14	CR5.1,CR5.2,CR5.3,CR5.7	initializae a new instance of the Contact object	assert all properties are not null and ContactId is not Guid.Empty
CT15	CR5.5,CR5.7	user is not authenticated, projectlist page loads	page loads with "You must be authorized to access this page.
CT16	CR5.5,CR5.7	user is not authenticated, users page loads	page loads with "You must be authorized to access this page.
CTU17	CR5.1,CR5.2,CR5.6	faq is added to the system	assert the faq can be returned by the system
CTU18	CR5.1,CR5.2,CR5.6	faq is answered	assert the faq with answer can be return by the system

CTU19	CR5.7	operational site, loaded home page	page contains input for first name, last name, phone, email, project title, and project description
CTU20	CR5.7	no user inputs other than clicking the submit button	assert validation errors: • The ProjectTitle field is required.• The ProjectDescription field is required.• The FirstName field is required.• The LastName field is required.• The Email field is not a valid e-mail address.• The Phone field is not a valid phone number.• The SponsorFirstName field is required.• The SponsorLastName field is required.• The SponsorEmail field is not a valid e-mail address.• The SponsorPhone field is not a valid phone number.
CTU21	CR5.2,CR5.7	file larger than configuration=n	assert validation error:• Max file size (n) exceeded on: testfile
CTU22	CR5.2,CR5.7	file count more than configuration=n	assert validation error:• Exceeded max number of files. Max:n.
CTU23	CR5.7	fills out all fields while putting "someemailaddress" into the email field	assert validation error:• The Email field is not a valid e-mail address.
CTU24	CR5.7	fills out all fields while putting "Some Text" into the phone field	assert validation error:• The Phone field is not a valid phone number.
CTU25	CR5.7	fills out all fields while putting 49 characters into the first name field	assert no first name validation errors
CTU26	CR5.7	fills out all fields while putting 51 characters into the first name field	assert validation error:• First Name is too long.

CTU27	CR5.7	fills out all fields while putting 49 characters into the last name field	assert no last name validation errors
CTU28	CR5.7	fills out all fields while putting 51 characters into the last name field	assert validation error:• Last Name is too long.

No Significant Results.