Faculty of Engineering and Applied Science

ENGR 4941U Capstone Systems Design for ECSE II

Design and Development of …

R4: Acceptance Testing and Results Report

Team Members

Umar Malik - 100705241

Taha Akhtar - 100704774

Rushi Amin - 100701929

Lefrancois Valenski - 100700573

Jayash Singh - 100695587

Faculty Advisor: Dr. Akramul Azim

Capstone Coordinator (Fall 2021 & Winter 2022): Dr. Q. Mahmoud

Table of Contents

Generate a table of contents…

List of Tables

List of Figures

In this report you need to demonstrate how well a working product or simulation meets the requirements of the project through a set of tests (functional, performance, regression). Some of the tests were defined in Report#1 but should have been refined in Report#2 and Report#3.

1. Acceptability Planning (26%)

In this section you should discuss your acceptance testing approach, acceptance testing process, procedure or steps involved, the tools and the material that will be used for testing, software and hardware, and acceptance testing criteria.

Our acceptance testing approach mainly involved verifying acceptance tests passed based on specific criteria that apply to it. User scenarios were created in order to properly test the passing criteria of the acceptance tests. Each of the acceptance tests were written based on the specific stakeholder requirements that were given to us at the beginning of the project. These were essentially key features that were outlined to be implemented within the project.

The steps that were involved in acceptance testing were to first look at the requirements that were given to us by the stakeholder. These requirements were turned into tests that could be tested through the framework. After having a defined list of the acceptance tests, a method of testing each acceptance test was outlined to be able to assign a pass or fail. Once the test was completed, comments were given to expand on the results to further understand any details that are necessary to the test.

The tools involved with running the acceptance tests were the CARLA test bench and the Jenkins pipeline that was constructed. Each of these individual acceptance tests could be correctly tested using the components that our group has constructed over the past few weeks. Some of these tests involve pushing changes to a project in order to trigger a build, performing an action on the simulator, and testing various components of the pipeline such as Git and Jenkins.

Acceptance criteria for the acceptance tests are important in order to categorize the tests correctly. There are many possibilities for testing criteria, so it is important to assign the criteria that is most relevant to the test case. Some of the possible testing criteria are usability, performance, upgradability, and scalability. Usability is relevant to how easily usable the specific feature is to use. Performance is used to test how well the specific feature performs on the system. Upgradability can be used to define the ability of the feature to be upgraded or changed in the future. The final testing criteria is scalability, which is used to test if the feature is easily able to adapt to different situations while still maintaining performance.

2. Acceptability Tests (27%)

In this section, provide a list of user scenarios and their related tests (In a table or as you see best presented). These should match your original agreed upon stakeholder requirements.

Requirement/Scenario	Implementation	Related Tests
Integration tests run on new code push or update, from 1 or many developers.	GitHub hook triggers Jenkins job containing integration test. This hook is activated with any code push to the repository.	1, 2, 3
Add a new hardware component to the test bench and implement functionality within CARLA	Added a gear shifter to the test bench and mapped inputs of the physical gear shifter to CARLA controls.	4
The system must have a physical infotainment unit that can communicate with the rest of the system	Added an Android 10 based infotainment unit to the testbench.	5, 6
The system still functions as intended once a new test is added to the framework.	Added a new test to the framework and Jenkins build test.	7
The system has a frontend UI that connects to the framework that provides information about Jenkins builds.	Connected an ELK stack service, DataDogto the Jenkins server and created a dashboards for relevant information and logs.	8, 9
Users have login access to the frontend UI	DataDog provides features to add users to the workspace with specific permissions. Authentication is taken care of by DataDog.	10
Test suite includes tests involving infotainment unit	Various tests were written that use ADB as a communication interface to allow for tests with the	11

	infotainment unit.	
Test suite includes tests involving CARLA and physical components	Tests involving CARLA and physical components such as the gear shifter are currently in progress	12

Test #	Title	Purpose	Pre-Conditions	Inputs/Outputs	Process
1	Test GitHub Hook Polling	To test if GitHub hook is communicating with Jenkins server. i.e a push to the repository triggers the build	Configuration of Github Webhook already completed	Input is a commit to a project. Output is a build trigger.	A push will be made to the application repository which will trigger the Jenkins build.
2	Test GitHub Hook Trigger	To test if GitHub hook executes the correctly corresponding job.	Configuration of Github Webhook already completed	Input is a commit from a specific app repository. Output is a build trigger of a specific project.	A push will be made to two different applications and the corresponding Jenkins build will be viewed.
3	Test GitHub Hook Trigger with multiple developers	To test if GitHub hook schedules builds correctly with multiple developers pushing changes	Configuration of Github Webhook already completed	Input is a commit from multiple app repositories in a similar time frame. Output is a build trigger done in the corresponding order.	A push will be made by multiple developers to the same application to see if the Jenkins build queue runs in the corresponding order.

4	Test Gear shifter inputs	To test if the gear shifter inputs are correctly recognized by CARLA	The gear shifter is connected to Carla and is working as intended.	Input is an activating the gear shifter. Output is the car going into drive/park/reverse correspondingly.	An actor will put the gear shift into drive from park and from drive into reverse to see the effects on CARLA.
5	Test Infotainment power	To test if the infotainment unit is functional	The infotainment system is connected to power and to the PC	Input is the infotainment screen is being clicked. Output is that an action occurs on the screen.	An actor will click on the screen to start an app and wait to see if the app activates and works.
6	Test infotainment communication	To test if the infotainment communication bridge (ADB) is functional.	Connect the infotainment system to the PC using a USB cable.	Input is an ADB command. Output is a response from ADB.	An actor uses the adb devices command in the command line and receives an output of the connected device.
7	Test new test addition	To test if the rest of the test suite functions as intended when a new test is added to it.	Existing tests in the framework.	Input is a new test in the framework. Output is the result from the test.	An actor adds a new test to the test pipeline and then activates the build to check if it fails.

8	Test connection of frontend UI	To test if the ELK stack front end is connected to the Jenkins server and pulls information	Frontend UI is established and is connected to Jenkins.	Input is login to the UI. Output is seeing the previous build results.	An actor logs into the frontend UI and checks to see if the previous build results are present in the dashboard.
9	Test realtime updating of Frontend UI	To test if the dashboard created on the ELK stack is updated when required.	ELK Stack is already established and configured	Input: Jenkin build change i.e. pass, fail, started, waiting, stopped etc. Output: Visual update on dashboard	An actor starts a build on Jenkins server and checks to see if the dashboard is updated with this information
10	Test user login and permission on Frontend UI	To test if different users can login to the frontend UI and only see what they are permitted to see	Frontend UI is established and is connected to Jenkins.	Input user login. Output is the user dashboard.	Multiple actors try logging into their dashboard and it is determined if the correct security permissions apply to each user.
11	Test infotainment unit test suite	To test if the test suite includes tests involving the infotainment unit and ADB and they function as intended	Test suite is established and configured in Jenkins server	Input: test suite Output: pass/fail from tests involving infotainment unit	An actor runs a build and checks the output for each of the tests involve the infotainment unit

12	Test CARLA test suite	To test if the test suite contains tests which involve CARLA and physical components and they function as intended	Test suite is established and configured in Jenkins server	Input: test suite Output: pass/fail from tests involving CARLA system	An actor runs a build and checks the output for each of the tests involving the CARLA system.

Also, in this section provide a list of tests (in a table or as you see best presented) that you are performing to the stakeholders (faculty advisor or client) and their relationship to the scenarios listed in section 1. Another way of doing this is to describe the scenarios followed by a list of tests in table form. A test should have a title, a purpose, pre-conditions, inputs, outputs, and a process.

3. Test Results (20%)

In this section provide a table that lists the results of the execution of the tests. This should include a date, pass or fail, and comments.

Test #	Date	Pass	Fail	Comments

4. Issues and Defects (27%)

Provide the process for recording and tracking issues and defects found during acceptance testing, and a summary of improvements to the product extracted from the tests along with action items. Note that at this late date in the project some defects might not be achievable.

5. Contribution Matrix

The contribution matrix provides details on how each member has contributed to the content of the report both in writing and material. For example, the list of tasks and contributors are shown in Table 5.1.

Table 5.1: Contribution matrix

Tasks	**People**		
Task 1	Name1	Name2	…others
	X (%percentage)		X (%percentage)
		X (%percentage)	

References

List of references should be formatted using one of the referencing styles: APA or MLA or **IEEE**. All references should be cited within the report.

Signatures

This document should be signed by all the group members and faculty advisor at the end of the acceptance testing meeting.

Position	**Name**	**Signature**
Student	Jayash Singh	
Student	Umar Malik	
Student	Lefrancois Valenski	
Student	Rushi Amin	
Student	Taha Akhtar	
Capstone Advisor	Dr. Akramul Azim	Akramul Azim