

## Project Plan

Advanced Development Factory: For Flutter/Dart

Development Security Operations Team

August 22, 2021

SWEN 670 9040 Software Engineering Project (2215) Summer 2021

Jeroen Soeurt,  
Michelle Monfort,  
Robert Wilson

## Revision History

[illegible]

# Advance Development Factory

Project name: Advance Development Factory

Date: June 6<sup>th</sup>, 2021

Project leader: Robert Wilson

Phase: Planning Cycle

For approval: Robert Wilson // Electronic Signature //

Date: June 6<sup>th</sup>, 2021

For approval: Mir Assadullah

Date:

---

## Table of Contents

1	Introduction: .....	6
2	General Information: .....	6
1.1	.....	6
1.2	.....	6
2.1	Situation sketch and problem definition of the project.....	6
2.2	Project assignment .....	6
2.3	Risk analysis .....	7
2.4	Organization of the project .....	7
3	Project Management Model .....	8
1.3	.....	8
3.1	Planning Cycle.....	8
3.2	Designing .....	8
3.3	Coding.....	9
3.4	Testing .....	10
3.5	Listening .....	12
4	Test Plan .....	13
1.4	.....	13
4.1	Build Pipeline: .....	13
4.1.1	(Charlie) Pipeline builds successfully .....	14
4.1.2	(Charlie) Pipeline Lint flutter code successfully .....	15
4.1.3	(Charlie) Pipeline Runs Unit Tests successfully .....	16

4.1.4	(Bravo) Pipeline builds successfully .....	17
4.1.5	(Bravo) Pipeline Lint flutter code successfully .....	18
4.1.6	(Bravo) Pipeline Runs Unit Tests successfully .....	19
4.1.7	(Amazing) Pipeline builds successfully .....	20
4.1.8	(Amazing) Pipeline Lint flutter code successfully .....	21
4.1.9	(Amazing) Pipeline Runs Unit Tests successfully .....	22
4.2	ADF: .....	23
4.2.1	ADF builds successfully .....	24
4.2.2	ADF can be accessed over RDP .....	24
4.2.3	User can log in.....	25
4.2.4	Network connection from within container works .....	25
4.2.5	Visual Studio Code starts .....	26
4.2.6	Git is available .....	27
4.2.7	Flutter/Dart is available .....	27
4.2.8	Emulator runs .....	28
4.2.9	Flutter/Dart app can run in emulator .....	28
4.3	CaPPMS:.....	28
4.3.1	Export Attribute Default is false.....	29
4.3.2	Column Header Attribute is true .....	29
4.3.3	Number Of Columns .....	29
4.3.4	Column Names .....	30
4.3.5	Verify Single Row .....	30
4.3.6	Delete File from file system .....	31
4.3.7	Save file to file system .....	31
4.3.8	Read stored file contents .....	31
4.3.9	Add Idea .....	32
4.3.10	Remove idea .....	32
4.3.11	Verify ProjectManagerService saved backing file .....	33
4.3.12	Update Idea.....	33
4.3.13	Export to PDF .....	33
4.3.14	Add Comment .....	34
4.3.15	Contact Initialization .....	34
4.3.16	Project List Does Not Load without Authentication .....	34
4.3.17	List of users does not load .....	35

4.3.18	A FAQ can be added .....	35
4.3.19	A Faq Reply can be added.....	36
4.3.20	The project idea page loads without authentication .....	37
4.3.21	FailedBlankSubmission .....	37
4.3.22	FileSizeUploadValidation .....	38
4.3.23	FileCountUploadValidation .....	39
4.3.24	Email Field Does not Validate .....	39
4.3.25	PhoneFieldDoesNotValidate .....	40
4.3.26	FristNameFieldDoesValidate49Characters .....	41
4.3.27	FristNameFieldDoesNotValidate51Characters .....	41
4.3.28	LastNameFieldDoesValidate49Characters.....	42
4.3.29	LastNameFieldDoesNotValidate51Characters.....	42
5	Summary.....	43
6	Review .....	44

## 1 Introduction:

University of Maryland Global Campus (UMGC) has been looking to solve a problem that spans between classes as more and more students move to remote learning. The resources that they have access to do not always permit a fully established development environment. For these reasons they are looking for a means to bring the fully established development environment to the student with minimal impact on the environment used by the student. The advanced development factory (ADF) will serve this purpose.

The Advanced Development Factory will give the student access to the development tools they need in order to provide a code, build, and test experience that any developer would expect to have. This means that ADF needs to be operating system agnostic and support a wide array of projects, frameworks, and languages. ADF will provide this flexibility to students and remove the barriers currently placed on them by their environments.

## 2 General Information:

### 2.1 Situation sketch and problem definition of the project

- UMGC is a global identity that provides higher level education to adults from many different backgrounds and has thousands of students across 90+ programs and specializations.
- The team supporting this effort is comprised of a diverse set of students that have varying life experiences and span nine time zones offering the widest range in support to ongoing commitments as well as offering dedicated time to this project.
- The team plans on continuing the development cycle started by previous classes and realizes the importance of completing this project.
- The team has a short time to access the current state of the project and determine the best path forward in order to meet the goals of the organization.
- The project will allow a common platform for developers to work on project despite constraints placed on their learning environments.

### 2.2 Project assignment

- There are two objectives that must be met by the DevSecOps team:
- The first objective is to support provide build, test, and pipeline support to three development teams.
- Identifying IDE for the teams to use, the project management system, and tracking operations is outside the scope of the DevSecOps team.
- Providing code reviews to prevent main branch contamination, establishing build quality and analysis checks, and maintaining the deployment pipeline are the main goals of the first objective.
- The second objective is to design and develop an environment that provides students access regardless of their personal environment scenario.

## 2.3 Risk analysis

- Time zone difference
  - The time zone difference of 9 hours cannot reasonably be reduced, as we all have other commitments in our regions (employment). We will work around this by scheduling meetings at night in Europe / in the morning on the west coast.
- Changing requirements
  - Looking at the previous GitHub repositories, it is clear that the programming language used over the past year has varied a lot. The solution for the ADF must be adjustable to future changes. We plan to use CLI tools, so that they can easily be replaced, and an IDE with support for more than one language.
- Short project timeline
  - It is a challenge to complete all this work in the duration of a single class. Furthermore, the pipeline is essential to maintaining the development teams code repositories. We will address this by first implementing the pipeline, and then working on the ADF.
- Limited experience
  - Two out of three members have limited experience with CI/CD. The risk is that the wrong decisions can be made early in the project, based on limited experience on knowledge. This issue is addressed by assigning the project manager position to the team member with the most experience and knowledge in this area.

## 2.4 Organization of the project

Multiple objectives will be concurrently supported during this class cycle. The first objective is to provide continuous integration and continuous deployment operations to three supported development team. The PM for Team Amazing is Shawn Kelly ([skelly46@student.umgc.edu](mailto:skelly46@student.umgc.edu)), the PM for team Bravo is Raul Benavides ([rbenavides8@student.umgc.edu](mailto:rbenavides8@student.umgc.edu)), and the PM for team Charlie is Michael Le ([skelly46@student.umgc.edu](mailto:skelly46@student.umgc.edu)). The class lead PM is Malcolm Freeney ([mfreeney@student.umgc.edu](mailto:mfreeney@student.umgc.edu)). The lead PM for the Development, Security, and Operations (DevSecOps) team is Robert Wilson ([rwilson127@student.umgc.edu](mailto:rwilson127@student.umgc.edu)).

The planning phase will end on June 8<sup>th</sup>, 2021 and the design phase will begin. The design phase poses to answer first what the pitfalls were of the previous efforts and the gap until the project reaches a sustainable point in its lifecycle. Once complete the team will be more able to determine the path forward in meeting the goal of a sustainable developer environment capable of being used by students globally.

The outputs of the design phase will set the tone for the rest of this development cycle. The largest amount of time is dedicated to the coding phase. The majority of allocated time is dedicated to this phase. Significant advancements in the current projected are projected due to the diversity of the team members. Each of the three team members represent expertise across the three primary operating system types: Windows, Linux, and MacOS.

This diversity can directly feed into the testing phase. The result of the combined system can then be tested thoroughly on each of the platforms with little to no troubles. This can validate from a high-level a limited number of support project types.

The listening phase will be accomplished by releasing the product to the mentors and sponsor of the overall project for use. The outputs will be used to generate a feedback document that can be used as input to the next cycle.

### 3 Project Management Model

The project has already started and so this is a continuation of work based on a previous development cycle. This requires an Agile approach and therefore a modified Extreme Programming method with Kanban boards, remote pair sessions, and code reviews will be used. This method allows for fast iterative cycles to deliver the end project without a high overhead on a small team. A Kanban automated board will be created in GitHub Project. This allows for integration with the repository, and is easily accessible by all team members using the GitHub web interface,

#### 3.1 Planning Cycle

The team conducted several meetings in order to sort out the best method to support continuous efforts across multiple fronts. The team also set up the initial repositories used by the supported three teams and produced simple how-to-videos in order to shorten the learning curve the supported teams may have.

#### 3.2 Designing

- Planned starting date: June 9<sup>th</sup>, 2021
- Planned ending date: June 20<sup>th</sup>, 2021

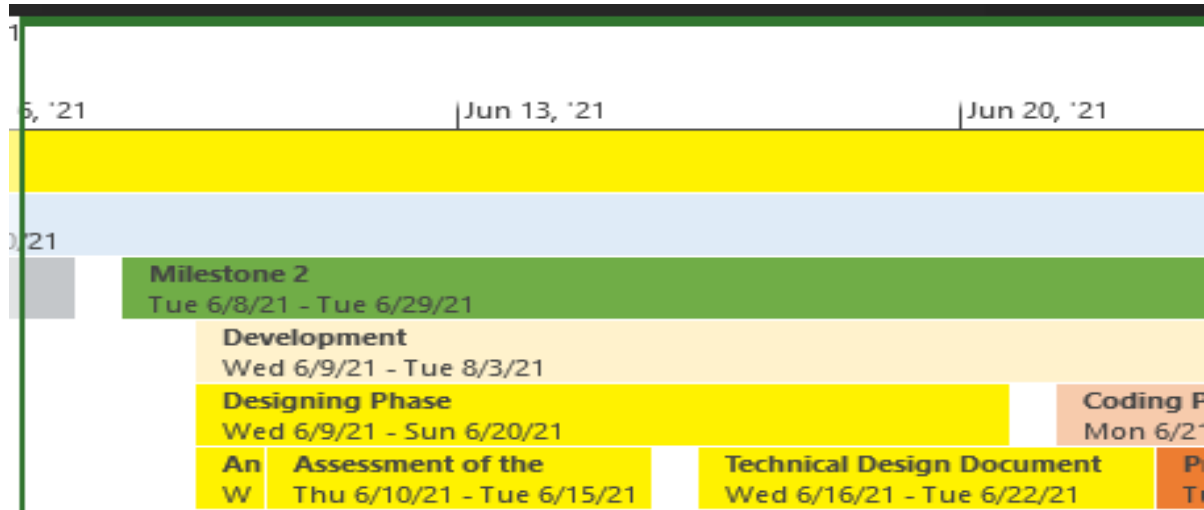
The outputs of this phase will lay a clear path of the objectives that are meant to be achieved during the coding phase. The major expected output of this phase is an analysis of the current state of the Advanced Developer Factory, an assessment of the functional gap between where the project is currently and the requirements of the sponsor, and the technical design document (TDD) that will feed the coding phase.

#### **Most important milestones:**

- Analysis of current state of the Advanced Developer Factory - June 13<sup>th</sup>, 2021
- Assessment of the functional gap - June 17<sup>th</sup>, 2021



Timeline:



Cost estimates:

Task Name	Cost
<b>Designing Phase</b>	<b>\$8,340.00</b>
Analysis of current ADF	\$540.00
Assessment of the functional gap	\$3,200.00
Technical Design Document	\$1,600.00

As this is an academic project that is part of the capstone course for the Masters in Information Technology with Software Engineering specialty at the University of Maryland Global Campus, the budget is set to \$0. The intent is to work within this budget in the following ways:

- Open-source software
- Freely available tools and services
  - Azure Trial allowance of \$200 and Microsoft Imagine Student credits
  - Google Trial allowance of \$300
- All team members are unpaid, as this is an academic project.

Internal information:

Microsoft Teams @DevSecOps Channel of United Global Master Coders will be the document repository for the outputs of the classroom milestone 2 which include the technical design document and Runbook.

External information:

No approvals are needed during this phase.

### 3.3 Coding

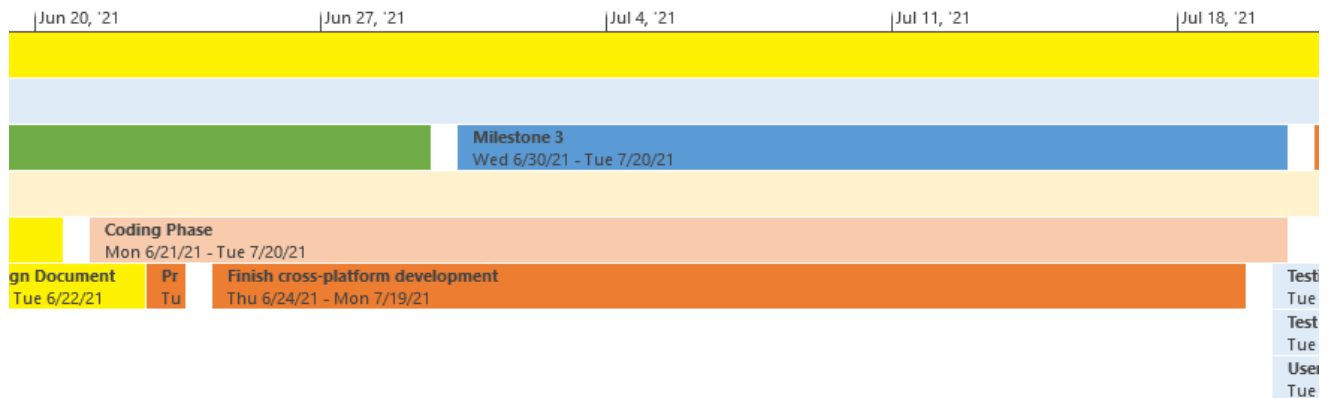
- Planned starting date: June 21<sup>st</sup>, 2021
- Planned ending date: July 20<sup>th</sup>, 2021

During this phase, the goal of the team is to fill 50% of the gap as identified during the analysis phase. It is fully expected that a complete solution is not reached but rather a simplified process that can start bridging the gap between operating systems to unified the development experience for all students.

Most important milestones:

- Start Development on cross-platform solution - June 21st, 2021
- Produce Runbook - June 23rd, 2021
- Finish cross-platform solution - July 19th, 2021

Timeline:



Cost estimates:

Task Name	Cost
<b>Coding Phase</b>	<b>\$61,664.79</b>
Start development on cross-platform solution	\$650.00
Produce Runbook	\$490.00
Finish cross-platform development	\$25,627.29

The budget for this project is set to \$0, as explained in 4.1.2.

Internal information:

The products of this phase represent two categories. The first category, code, will be stored in GitHub under the repo umgc/umgc.advance.development.factory. The second category, classwork, will be turned in via UMGC school website (LEO) as a group project.

External information:

No external requirements for this phase exist.

### 3.4 Testing

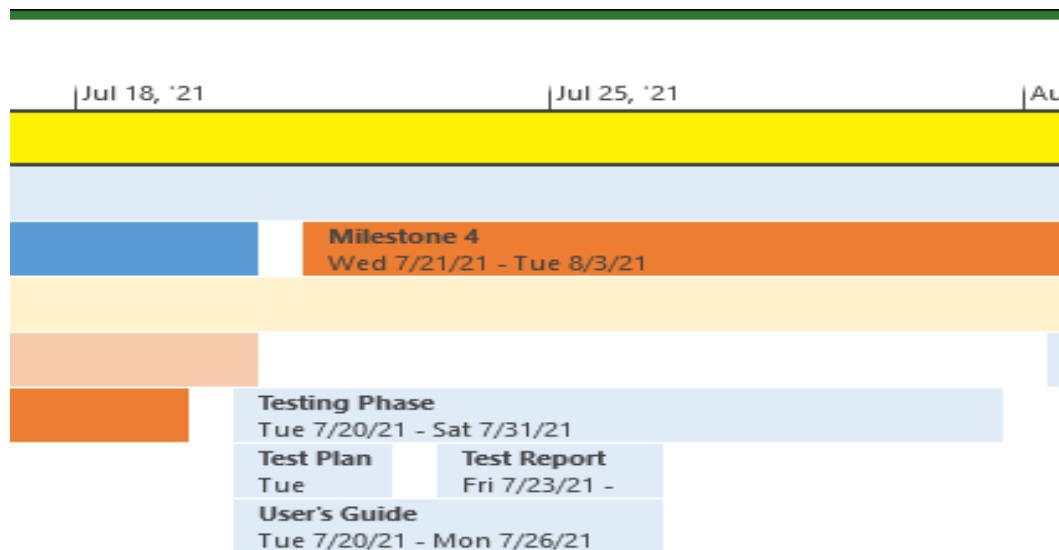
- Planned starting date: July 20<sup>th</sup>, 2021
- Planned ending date: July 31<sup>st</sup>, 2021

The major outputs from this phase will be the user's guide, test plan and the test report. This will validate as much as possible how ADF interacts with a limited number of project types across Windows, Linux, and MacOS host operating systems. Automated tests must be incorporated into the original plan in order to ensure the core functionality remains functional throughout the rest of the life cycle of the project.

Most important milestones:

- Test Plan completed - July 22nd, 2021
- User Guide - July 26th, 2021
- Test Report - July 26th, 2021

Timeline:



Cost estimates:

Task Name	Cost
<b>Testing Phase</b>	<b>\$17,713.50</b>
Test Plan	\$2,400.00
User's Guide	\$1,200.00
Test Report	\$600.00

The budget for this project is set to \$0, as explained in 4.1.2.

Internal information:

There are two categories of outputs for this phase. The first, code, are automated tests written to ensure high-level functionality of the developed system which will reside at GitHub under the repo: umgc/umgc.advance.development.factory. The second category, classwork, will be turned in via UMGC school webiste (LEO) as a group project.

External information:

No external requirements for this phase exist.

### 3.5 Listening

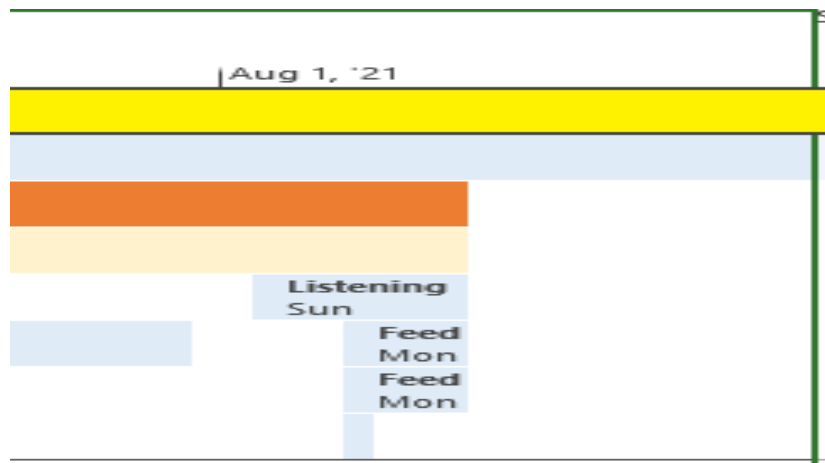
- Planned starting date: August 1<sup>st</sup>, 2021
- Planned ending date: August 3<sup>rd</sup>, 2021

The outputs of this phase will be aggregated into a handoff document for the next develop cycle. This output would contain information such as current state of the ADF system, compatibility assessment for each operating system, and any noted quarks or bugs found during the testing and listening phases.

Most important milestones:

- Feedback from mentors - August 2<sup>nd</sup>, 2021
- Feedback from sponsor - August 2<sup>nd</sup>, 2021
- Programmer's Guide Complete

Timeline:



Cost estimates:

Task Name	Cost
<b>Listening Phase</b>	<b>\$1,130.00</b>
Feedback from mentors	\$80.00
Feedback from sponsor	\$80.00
Programmer's Guide	\$720.00

The budget for this project is set to \$0, as explained in 4.1.2.

Internal information:

Outputs from this cycle will include the programmer's guide and future works.

External information:

Collection of feedback from mentors and sponsors of the ADF project will be vital in the hand-off between development cycles. As such there are no expected approvals required but rather the state of the project as seen from the mentor and sponsor point of view.

## 4 Test Plan

Humans are fallible, as is the code that is written to provide instruction to a computer system.

With this in mind we have to assume that any change to current code will break something as the current code is fallible. It is unlikely that enough test can be written that will simply ensure a product that does not falter. This is an un-realistic approach to testing. It also needs to be considered that the DevSecOps team is responsible for multiple projects within their own team and ensuring the build pipeline success for the use of development teams' collaboration.

The engineering team should be intimately familiar with the overall structure of ADF, but the actual usage of ADF is what needs to be tested. 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

### 4.1 Build Pipeline:

DevSecOps team is responsible for the development teams' pipelines. As such, any pipeline configuration change must be monitored for completion. The process of the pipeline must be considered to be faulty until completely validated against the contract with the development

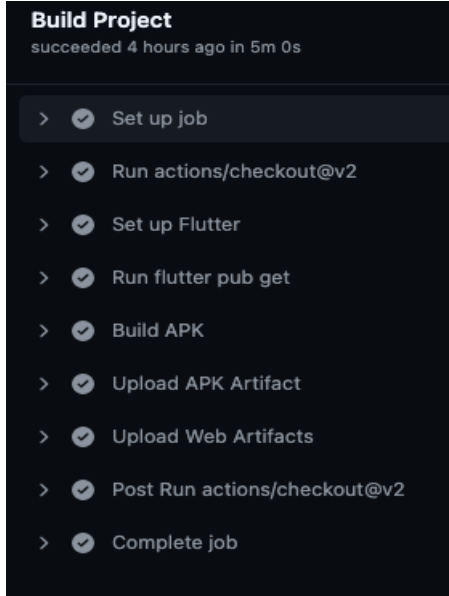
team. The basic elements of the pipeline must be able to analyze, build, test, and post artifacts from the build.

A configuration change to the pipeline could result in failures in any of these mention categories.

A validation of each category is the only way a change in configuration can be validated. It is equally important that the pipeline be validated occasionally after branch merge operations. A change in the development team's branch may affect how the pipeline conducts its own processes. For an example, if the development team chooses to change the base folder of their project, the pipeline may no longer build the project correctly.

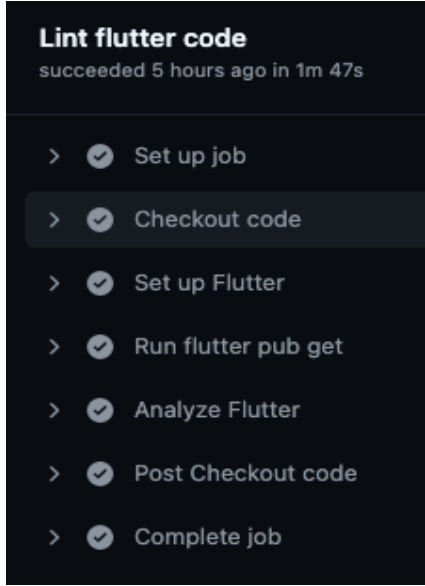
#### 4.1.1 (Charlie) Pipeline builds successfully

Item	Description
Title	Pipeline builds successfully
Description	Tests if the Pipeline builds successfully.
Input Data	See GitHub Actions Double click most recent workflow Double click build
Expected Result	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

Image	
-------	--

#### 4.1.2 (Charlie) Pipeline Lint flutter code successfully

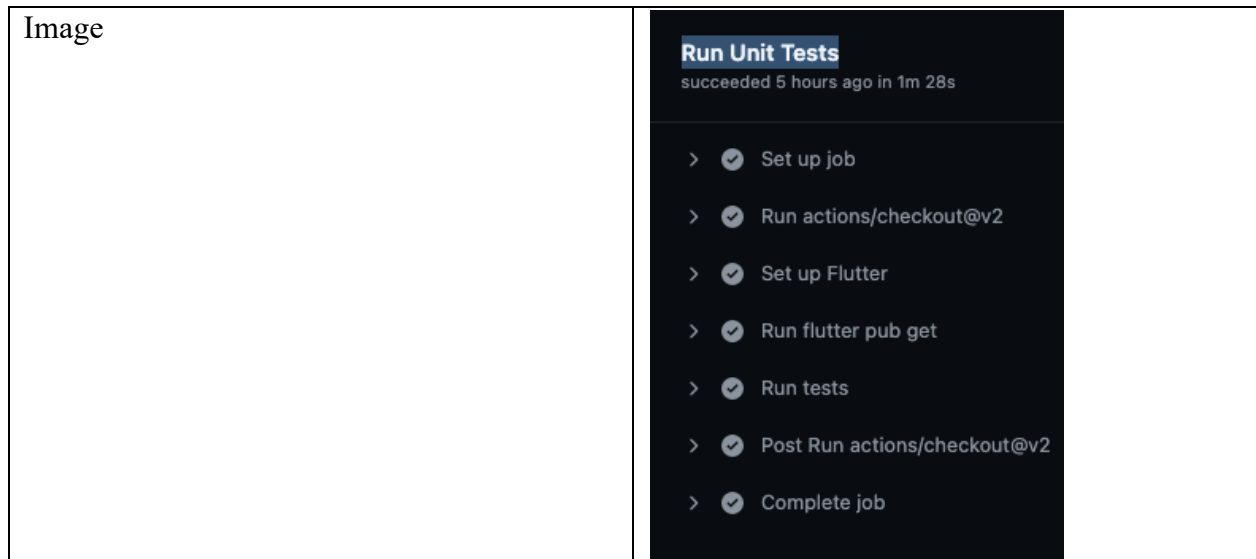
Item	Description
Title	Pipeline Lint flutter code successfully
Description	Tests if the Pipeline Lints flutter code successfully.
Input Data	See GitHub Actions Double click most recent workflow Double click Lint flutter code
Expected Result	Artifacts Produced during runtime: => Set up job => Checkout code => Set up Flutter => Run flutter pub get => Analyze Flutter => Post Checkout code => Complete job

Image	 <p><b>Lint flutter code</b> succeeded 5 hours ago in 1m 47s</p> <ul style="list-style-type: none"> <li>&gt; <input checked="" type="checkbox"/> Set up job</li> <li>&gt; <input checked="" type="checkbox"/> Checkout code</li> <li>&gt; <input checked="" type="checkbox"/> Set up Flutter</li> <li>&gt; <input checked="" type="checkbox"/> Run flutter pub get</li> <li>&gt; <input checked="" type="checkbox"/> Analyze Flutter</li> <li>&gt; <input checked="" type="checkbox"/> Post Checkout code</li> <li>&gt; <input checked="" type="checkbox"/> Complete job</li> </ul>
-------	---

#### 4.1.3 (Charlie) Pipeline Runs Unit Tests successfully

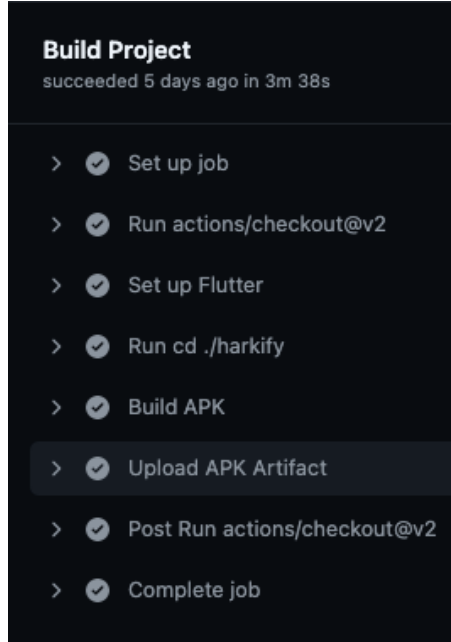
Item	Description
Title	Pipeline Runs Unit Tests Successfully
Description	Tests if the Pipeline Runs Unit Tests successfully.
Input Data	See GitHub Actions Double click most recent workflow Double click Run Unit Tests
Expected Result	Artifacts Produced during runtime: => Set up job => Run actions/checkout@v2 => Set up Flutter => Run flutter pub get => Run tests => Post Run actions/checkout@v2 => Complete job





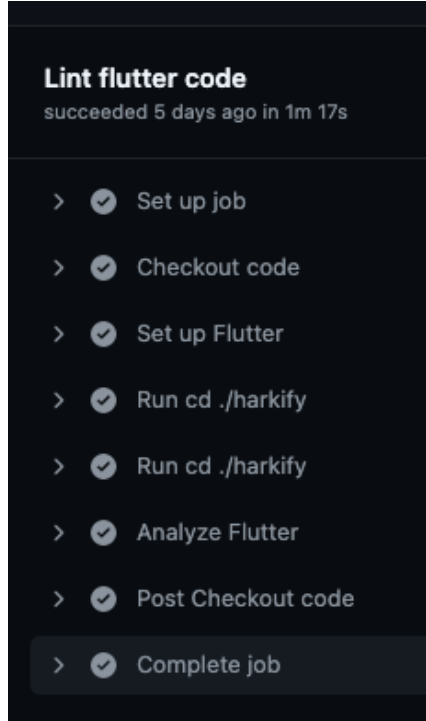
#### 4.1.4 (Bravo) Pipeline builds successfully

Item	Description
Title	Pipeline builds successfully
Description	Tests if the Pipeline builds successfully.
Input Data	See GitHub Actions Double click most recent workflow Double click build
Expected Result	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

Image	 <p><b>Build Project</b> succeeded 5 days ago in 3m 38s</p> <ul style="list-style-type: none"> <li>&gt; ✓ Set up job</li> <li>&gt; ✓ Run actions/checkout@v2</li> <li>&gt; ✓ Set up Flutter</li> <li>&gt; ✓ Run cd ./harkify</li> <li>&gt; ✓ Build APK</li> <li>&gt; ✓ Upload APK Artifact</li> <li>&gt; ✓ Post Run actions/checkout@v2</li> <li>&gt; ✓ Complete job</li> </ul>
-------	---

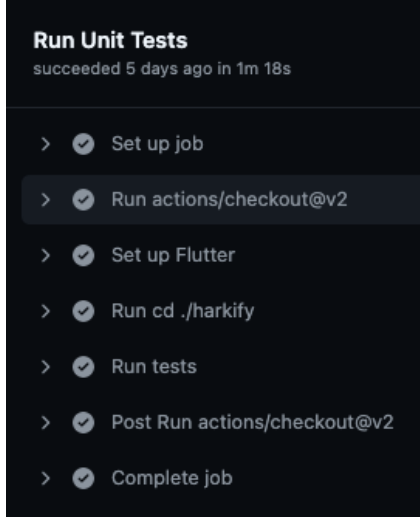
#### 4.1.5 (Bravo) Pipeline Lint flutter code successfully

Item	Description
Title	Pipeline Lint flutter code successfully
Description	Tests if the Pipeline Lints flutter code successfully.
Input Data	See GitHub Actions Double click most recent workflow Double click Lint flutter code
Expected Result	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

Image	 <p><b>Lint flutter code</b> succeeded 5 days ago in 1m 17s</p> <ul style="list-style-type: none"> <li>&gt; <input checked="" type="checkbox"/> Set up job</li> <li>&gt; <input checked="" type="checkbox"/> Checkout code</li> <li>&gt; <input checked="" type="checkbox"/> Set up Flutter</li> <li>&gt; <input checked="" type="checkbox"/> Run cd ./harkify</li> <li>&gt; <input checked="" type="checkbox"/> Run cd ./harkify</li> <li>&gt; <input checked="" type="checkbox"/> Analyze Flutter</li> <li>&gt; <input checked="" type="checkbox"/> Post Checkout code</li> <li>&gt; <input checked="" type="checkbox"/> Complete job</li> </ul>
-------	--

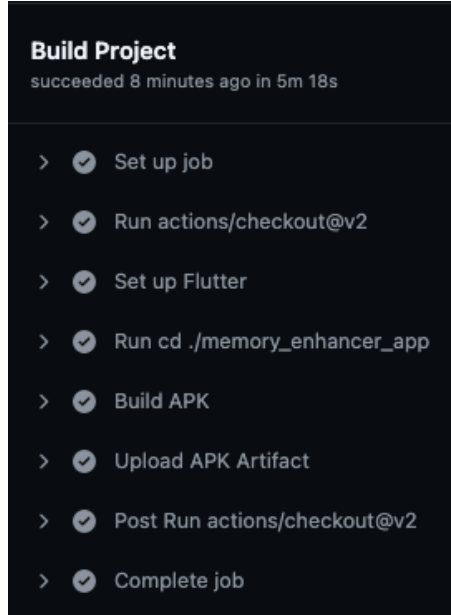
#### 4.1.6 (Bravo) Pipeline Runs Unit Tests successfully

Item	Description
Title	Pipeline Runs Unit Tests Successfully
Description	Tests if the Pipeline Runs Unit Tests successfully.
Input Data	See GitHub Actions Double click most recent workflow Double click Run Unit Tests
Expected Result	Artifacts Produced during runtime: => Set up job => Run actions/checkout@v2 => Set up Flutter => Run cd ./harikfy => Run tests => Post Run actions/checkout@v2 => Complete job

Image	
-------	--

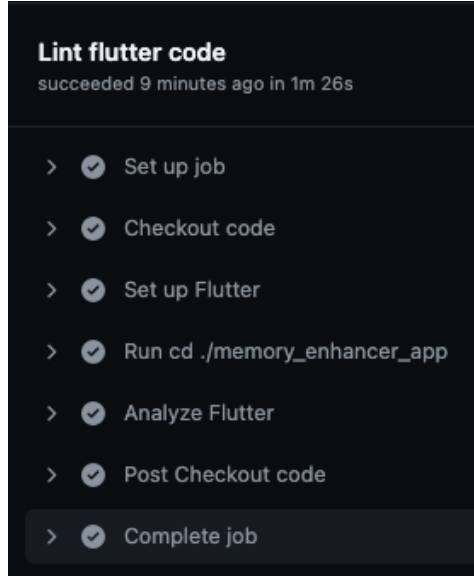
#### 4.1.7 (Amazing) Pipeline builds successfully

Item	Description
Title	Pipeline builds successfully
Description	Tests if the Pipeline builds successfully.
Input Data	See GitHub Actions Double click most recent workflow Double click build
Expected Result	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

Image	 <p><b>Build Project</b> succeeded 8 minutes ago in 5m 18s</p> <ul style="list-style-type: none"> <li>&gt; ✓ Set up job</li> <li>&gt; ✓ Run actions/checkout@v2</li> <li>&gt; ✓ Set up Flutter</li> <li>&gt; ✓ Run cd ./memory_enhancer_app</li> <li>&gt; ✓ Build APK</li> <li>&gt; ✓ Upload APK Artifact</li> <li>&gt; ✓ Post Run actions/checkout@v2</li> <li>&gt; ✓ Complete job</li> </ul>
-------	--

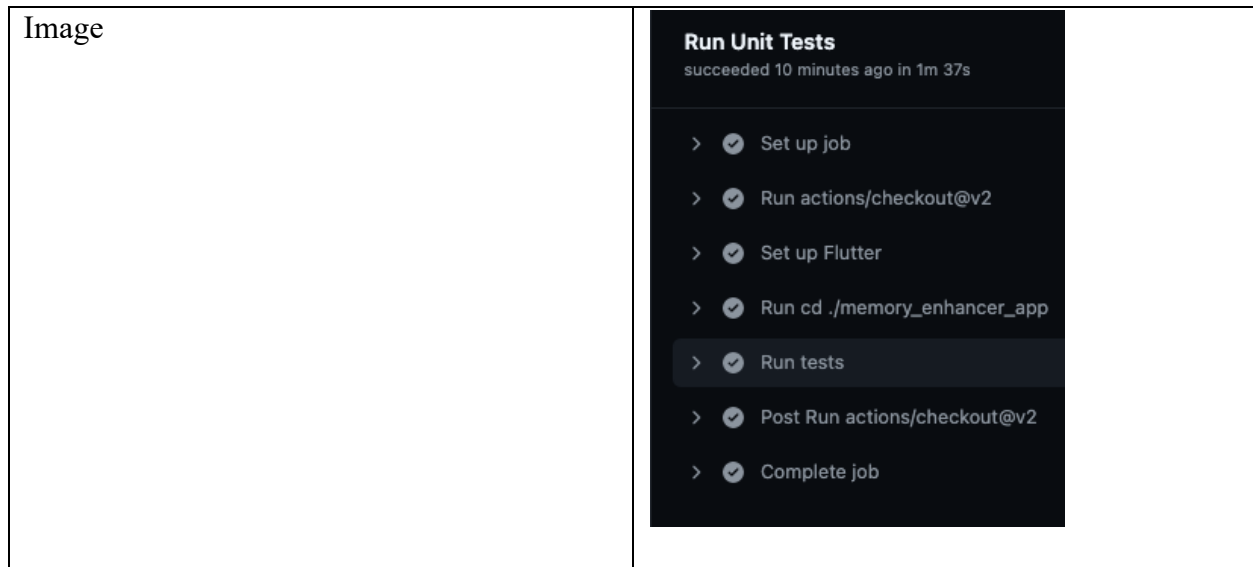
#### 4.1.8 (Amazing) Pipeline Lint flutter code successfully

Item	Description
Title	Pipeline Lint flutter code successfully
Description	Tests if the Pipeline Lints flutter code successfully.
Input Data	See GitHub Actions Double click most recent workflow Double click Lint flutter code
Expected Result	Artifacts Produced during runtime: => Set up job => Checkout code => Set up Flutter => Run cd ./memory_enhancer_app => Analyze Flutter => Post Checkout code => Complete job

Image	 <p><b>Lint flutter code</b> succeeded 9 minutes ago in 1m 26s</p> <ul style="list-style-type: none"> <li>&gt; ✓ Set up job</li> <li>&gt; ✓ Checkout code</li> <li>&gt; ✓ Set up Flutter</li> <li>&gt; ✓ Run cd ./memory_enhancer_app</li> <li>&gt; ✓ Analyze Flutter</li> <li>&gt; ✓ Post Checkout code</li> <li>&gt; ✓ Complete job</li> </ul>
-------	--

#### 4.1.9 (Amazing) Pipeline Runs Unit Tests successfully

Item	Description
Title	Pipeline Runs Unit Tests Successfully
Description	Tests if the Pipeline Runs Unit Tests successfully.
Input Data	See GitHub Actions Double click most recent workflow Double click Run Unit Tests
Expected Result	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



## 4.2 ADF:

The ADF project takes two files: Dockerfile and startup.sh file. Given that these two files are static but functional it is important to test the resulting combination of these two files. The startup.sh file is particularly touchy to change. Given that ADF is a Linux driven environment, if the startup.sh file is edited on a Windows system, the file may switch to using CRLF line endings. These line endings are not fully compatible with a Linux system and must LF line endings.

Because of this, the ADF system must be validated and verified after each configuration change. Validation is to ensure the ADF system meets the current demand of the courseware. The verification is to ensure ADF can execute the expected tasks of the current courseware. Some axis specifically tested are: Performance of the emulator on supported platforms, Access to data from remote sources, Platform agnostic capability (known issue: Docker doesn't support nested virtualization on Windows 10), and can ADF build a basic project in the desired language.

#### 4.2.1 ADF builds successfully

Item	Description
Title	ADF builds successfully
Description	Tests if the ADF image builds successfully.
Input Data	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"
Expected Result	Docker image build, and completes with:  => exporting to image  => => exporting layers  => => writing image sha256:XXXXXXXXXXXXXXXXXXXXX  => => naming to docker.io/library/adfsummer2021:latest

#### 4.2.2 ADF can be accessed over RDP

Item	Description
Title	ADF can be accessed over RDP
Description	Tests if the ADF is accessible over RDP.
Input Data	Run the following command, and try to connect to localhost:63389 using RDP.  docker run -dit -p 63389:3389 --rm --privileged adfsummer2021:latest



Expected Result	RDP client should connect and a login screen should show up.
-----------------	--

#### 4.2.3 User can log in

Item	Description
Title	User can log in
Description	Tests if the user can log in using the developer account.
Input Data	RDP is connected to the container, and shows the login screen. User enters:  Username: developer  Password: password
Expected Result	User should log in and desktop should be shown.

#### 4.2.4 Network connection from within container works

Item	Description
Title	Network connection from within container works.
Description	Tests if there is an active internet connection from within the container, given that there is an active internet connection on the host that runs the container.
Input Data	User open a new terminal window and issues the following commands:  sudo apt-get update  sudo apt-get install iputils-ping  ping google.com

Expected Result	<p>Google.com should respond, Ie:</p> <p>developer@3bcdd837d177:~\$ ping google.com</p> <p>PING google.com (142.250.184.238) 56(84) bytes of data.</p> <p>{returned ip address = 142.250.184.238 but ttl and response time may vary from tests to test}</p> <p>--- google.com ping statistics ---</p> <p>2 packets transmitted, 2 received, 0% packet loss, time 1001ms</p> <p>rtt min/avg/max/mdev = 5.308/5.776/6.244/0.468 ms</p>
-----------------	--

#### 4.2.5 Visual Studio Code starts

Item	Description
Title	Visual Studio Code starts
Description	Tests if VS Code starts
Input Data	User clicks on VS Code icon on desktop.
Expected Result	VS Code screen appears.

#### 4.2.6 Git is available

Item	Description
Title	Git is available
Description	Tests if Git is available from the command line.
Input Data	User opens a new terminal and issues the following command:  git --version
Expected Result	Git version appears, for example:  git version 2.25.1

#### 4.2.7 Flutter/Dart is available

Item	Description
Title	Flutter/Dart is available
Description	Tests if Flutter and Dart CLI tools are available from the command line.
Input Data	User opens a new terminal and issues the following command:  flutter --version
Expected Result	Flutter and Dart versions appears, for example:  Flutter 2.2.1 • channel stable • <a href="https://github.com/flutter/flutter.git">https://github.com/flutter/flutter.git</a> Framework • revision 02c026b03c (8 weeks ago) • 2021-05-27 12:24:44 -0700  Engine • revision 0fdb562ac8  Tools • Dart 2.13.1

#### 4.2.8 Emulator runs

Item	Description
Title	Android emulator runs
Description	Tests if the Android emulator runs
Input Data	User opens a new terminal and issues the following command:  flutter emulators --launch flutter_emulator
Expected Result	Emulator appears and starts Android.

#### 4.2.9 Flutter/Dart app can run in emulator

Item	Description
Title	Flutter/Dart app can run in emulator.
Description	Tests if a Flutter/Dart app can run in emulator.
Input Data	User opens a new terminal and issues the following command while the emulator is running:  flutter run
Expected Result	App builds and runs in the emulator

### 4.3 CaPPMS:

The CaPPMS website is the most interesting to test. This application will use dynamic functional unit tests in order to ensure as much functionality remains operational as possible. The goal would be to have at least 50% test coverage. The tests are expected to be boundary tests and ALAC tests. ALAC test are a natural selection as this is a customer facing website. ALAC tests are to be covered by validation tests by purposefully entering the wrong data which invokes validation to report the errors against fields. It is expected the system for humans will have

problems before the system for human is used by humans. Customers are expected to make mistakes as well and much of the website has been developed with this in mind.

#### 4.3.1 Export Attribute Default is false

Item	Description
Title	Export Attribute Default is false
Description	Test ensures the default value of the ExportAttribute class is set to false when placed on a property of an object.
Input Data	Any test class that uses the ExportAttribute without a parameter on a property within the object.
Expected Result	Attribute is set to false.

#### 4.3.2 Column Header Attribute is true

Item	Description
Title	Column Header Attribute is true
Description	Asserts that decoration of a property with the attribute ColumnHeaderAttribute returns true for the attribute
Input Data	Any test class that decorate a property with the ColumnHeaderAttribute.
Expected Result	ColumnHeader = true

#### 4.3.3 Number Of Columns

Item	Description
Title	Number Of Columns

Description	Test verifies that n number of columns are added to the table based on an enumerable object.
Input Data	An enumerable object that implements ColumnHeaderAttribute on at least two properties of the enumerated object.
Expected Result	N = table.HeaderRow.Count

#### 4.3.4 Column Names

Item	Description
Title	Column Names
Description	The names of expected n columns equals the names of n columns in the test table.
Input Data	An enumerable object that implements ColumnHeaderAttribute on at least two properties of the enumerated object.
Expected Result	Column[n].Name = table.HeaderRow[n].Value.ToString()

#### 4.3.5 Verify Single Row

Item	Description
Title	Verify Single Row
Description	At least the first row of the table matches an enumerable object with one item added to the collection.
Input Data	An enumerable object with at least one item in the collection with different values assigned to at least two properties.
Expected Result	N property values = table.rows[n][c].Value

#### 4.3.6 Delete File from file system

Item	Description
Title	Delete File from file system
Method	LocalProjectFileManager.DeleteAsync(string location)
Description	The system can delete a generated file.
Input Data	The system creates a file to be deleted.
Expected Result	The system receives an empty string after calling LocalProjectFileManager.DeleteAsync()

#### 4.3.7 Save file to file system

Item	Description
Title	Save file to file system
Method	LocalProjectFileManager.SaveAsync(stream, fileId, filename)
Description	Save Test generated file to file system
Input Data	System generated file can be saved with LocalFileProjectFileManager
Expected Result	File.Exists(location) = true

#### 4.3.8 Read stored file contents

Item	Description
Title	Read stored file contents
Method	LocalProjectFileManager.ReadAsync(string location)
Description	Reads the contents of a test generated file.
Input Data	Test generated file

Expected Result	File contents returned by ReadAsync matches generated data saved before read.
-----------------	---

#### 4.3.9 Add Idea

Item	Description
Title	Add Idea
Method	ProjectManagerService.AddAsync(ProjectInformation idea)
Description	Add a new instance of a ProjectInformation object
Input Data	A test generated instance of a ProjectInformation object
Expected Result	Return response from AddAsync is true and one title exists when checked using method ProjectManagerService.GetIdeaTitles()

#### 4.3.10 Remove idea

Item	Description
Title	Remove idea
Method	ProjectManagerService.RemoveAsync(ProjectInformation idea, IPrincipal principal)
Description	Test generates an idea, adds the idea to the system, verifies that the test has been added, removes the idea, and verifies that the test has been removed.
Input Data	Test generated idea
Expected Result	Verifies that added n items before verifying m items are removed



#### 4.3.11 Verify ProjectManagerService saved backing file

Item	Description
Title	Verify ProjectManagerService saved backing file
Description	The project manager saves the file upon receiving a change event. A test needs to be able to add an idea the verify that the change event fired and saved the file by manually loading the expected save file.
Input Data	N number of items added to the ProjectManagerService
Expected Result	N number of items retrieved manually

#### 4.3.12 Update Idea

Item	Description
Title	DidUpdateProjectManager
Description	After an idea is submitted to the manager can it be updated.
Input Data	An idea that has been added to the ProjectManagerService has an edit to the project title that includes a unique identifier.
Expected Result	N = \$"New Test Project Title-{Guid.NewGuid()}";

#### 4.3.13 Export to PDF

Item	Description
Title	DoesExportHashMatchExpected
Description	An exported pdf containing the a baseline project should match pre-computed hash.

Input Data	Base line Idea created from the CreateIdea method that can be used in all idea tests for creation.
Expected Result	Hash using SHA512 against known good Export

#### 4.3.14 Add Comment

Item	Description
Title	DoesAddComment
Description	Adds a comment and checks to see if comment has been added
Input Data	Adds \$"Test comment-{Guid.NewGuid()}";
Expected Result	N = \$"Test comment-{<guid created>}";

#### 4.3.15 Contact Initialization

Item	Description
Title	Contact Initialization
Description	Zero Properties are null and ContactID is not empty
Input Data	None
Expected Result	All Properties initialize to not null values and Guid is not empty

#### 4.3.16 Project List Does Not Load without Authentication

Item	Description
Name	ProjectListAuthenticationVerification
Description	The project list page does not display the project table
Requirements	The application is running in a hosted environment.

Prerequisites	The system has initialized bUnit for tests and has created a mock project manager service.
Steps	Navigate to /projectlist
Expected Output	<h3>You must be authorized to access this page.</h3>
Assumptions	Test harness is not authenticated with AAD.

#### 4.3.17 List of users does not load

Item	Description
Name	ProjectListAuthenticationVerification
Description	The project list page does not display the project table
Requirements	The application is running in a hosted environment.
Prerequisites	The system has initialized bUnit for tests and has created a mock project manager service.
Steps	Navigate to /users
Expected Output	<h3>You must be authorized to access this page.</h3>
Assumptions	Test harness is not authenticated with AAD.

#### 4.3.18 A FAQ can be added

Item	Description
Name	Add Faq
Description	A FAQ can be added to the system
Requirements	The system saves the fact and can be checked indepenantly of the FaqService. Can be verified via FaqService.

Prerequisites	The system has initialized bUnit for tests and has created a mock project manager service.
Steps	Faq Page initialized and a faq is added.
Expected Output	The added faq can be verified via the faq service and independently verified in the backing file
Assumptions	

#### 4.3.19 A Faq Reply can be added

Item	Description
Name	Add Reply to faq question
Description	A question can be replied to
Requirements	The system has a faq without a question.  The question gets replied to.  The question gets updated in backing service.
Prerequisites	The system has initialized bUnit for tests and has created a mock project manager service.
Steps	Faq Page initialized and a faq is added.  The previously added faq can be answered by authorized account holders
Expected Output	The updated faq can be verified via the faq service and independently verified in the backing file
Assumptions	Uses authenticated user with user role

#### 4.3.20 The project idea page loads without authentication

Item	Description
Name	Load Index page
Description	When the Index.razor component page is loaded the markup has load the submit idea form.
Requirements	The idea form contains at least the labels for first name, last name, email, phone, project title, project description, attachments, and website.
Prerequisites	The system has initialized bUnit for tests and has created a mock project manager service.
Steps	Navigate to /
Expected Output	The page contains the specified requirements as elements return by the system
Assumptions	No authentication is required

#### 4.3.21 FailedBlankSubmission

Item	Description
Name	FailedBlankSubmission
Description	Without inputting any data into the SubmitIdea component, a validation error is received upon clicking the submit button
Requirements	The application is running in a hosted environment.
Prerequisites	The system has initialized bUnit for tests and has created a mock project manager service.

Steps	Leaving all inputs blank, click submit
Expected Output	<p>Validation error displays with the following:</p> <ul style="list-style-type: none"> <li>• The ProjectTitle field is required.</li> <li>• The ProjectDescription field is required.</li> <li>• The FirstName field is required.</li> <li>• The LastName field is required.</li> <li>• The Email field is not a valid e-mail address.</li> <li>• The Phone field is not a valid phone number.</li> <li>• The SponsorFirstName field is required.</li> <li>• The SponsorLastName field is required.</li> <li>• The SponsorEmail field is not a valid e-mail address.</li> <li>• The SponsorPhone field is not a valid phone number.</li> </ul>
Assumptions	

#### 4.3.22 FileSizeUploadValidation

Item	Description
Name	FileSizeUploadValidation
Description	The file size will be validated upon upload.
Requirements	The application is running in a hosted environment.
Prerequisites	The system has initialized bUnit for tests and has created a mock project manager service.
Steps	Filling out First Name, Last Name, Email, Phone, Project Title, Project Description, and ensuring the Self Sponsored checkbox is check, attempt to upload a file larger than the maximum sized.
Expected Output	Validation error displays with the following:

	<ul style="list-style-type: none"> <li>• Max file size (10) exceeded on: testfile</li> </ul>
Assumptions	The maximum size is set in the appsetting.json file.

#### 4.3.23 FileCountUploadValidation

Item	Description
Name	FileCountUploadValidation
Description	The number of files will be validated upon upload.
Requirements	The application is running in a hosted environment.
Prerequisites	The system has initialized bUnit for tests and has created a mock project manager service.
Steps	Filling out First Name, Last Name, Email, Phone, Project Title, Project Description, and ensuring the Self Sponsored checkbox is check, attempt to upload a more files than allowed by the system.
Expected Output	Validation error displays with the following: <ul style="list-style-type: none"> <li>• Exceeded max number of files. Max:10.</li> </ul>
Assumptions	The maximum number of files is set in the appsetting.json file.

#### 4.3.24 Email Field Does not Validate

Item	Description
Name	EmailFieldDoesNotValidate

Description	Users may not enter a qualified email address. This should be validated before input is accepted.
Requirements	The application is running in a hosted environment.
Prerequisites	The system has initialized bUnit for tests and has created a mock project manager service.
Steps	Enter someemailaddress in the Email address input field
Expected Output	Validation error displays with the following: <ul style="list-style-type: none"> <li>• The Email field is not a valid e-mail address.</li> </ul>
Assumptions	

#### 4.3.25 PhoneFieldDoesNotValidate

Item	Description
Name	PhoneFieldDoesNotValidate
Description	Users may enter text for a phone number. We expect all numbers
Requirements	The application is running in a hosted environment.
Prerequisites	The system has initialized bUnit for tests and has created a mock project manager service.
Steps	Enter “Some Text” into the Phone input field.
Expected Output	Validation error displays with the following: <ul style="list-style-type: none"> <li>• The Phone field is not a valid phone number.</li> </ul>
Assumptions	



#### 4.3.26 FristNameFieldDoesValidate49Characters

Item	Description
Name	FristNameFieldDoesValidate49Characters
Description	User is limited to 50 characters for their first name
Requirements	The application is running in a hosted environment.
Prerequisites	The system has initialized bUnit for tests and has created a mock project manager service.
Steps	Enter 49 character into the First Name Field
Expected Output	There should be no First Name validation errors
Assumptions	Any other validation errors are ignored.

#### 4.3.27 FristNameFieldDoesNotValidate51Characters

Item	Description
Name	FristNameFieldDoesNotValidate51Characters
Description	User is limited to 50 characters for their first name
Requirements	The application is running in a hosted environment.
Prerequisites	The system has initialized bUnit for tests and has created a mock project manager service.
Steps	Enter 51 character into the First Name Field
Expected Output	There should be a First Name validation error: <ul style="list-style-type: none"><li>• First Name is too long.</li></ul>
Assumptions	Any other validation errors are ignored.

#### 4.3.28 LastNameFieldDoesValidate49Characters

Item	Description
Name	LastNameFieldDoesValidate49Characters
Description	User is limited to 50 characters for their first name
Requirements	The application is running in a hosted environment.
Prerequisites	The system has initialized bUnit for tests and has created a mock project manager service.
Steps	Enter 49 character into the Last Name Field
Expected Output	There should be no First Name validation errors
Assumptions	Any other validation errors are ignored.

#### 4.3.29 LastNameFieldDoesNotValidate51Characters

Item	Description
Name	LastNameFieldDoesNotValidate51Characters
Description	User is limited to 50 characters for their first name
Requirements	The system has initialized bUnit for tests and has created a mock project manager service.
Prerequisites	The test system is capable of automated selenium tests.
Steps	Enter 51 character into the Last Name Field
Expected Output	There should be a First Name validation error: <ul style="list-style-type: none"><li>• Last Name is too long.</li></ul>
Assumptions	Any other validation errors are ignored.

## 5 Summary

This is a relatively short-lived project that will use Kanban boards in order to maximize the small team and project cycle.

Most important milestones:

- Analysis of current state of the Advanced Developer Factory - June 13th, 2021
- Assessment of the functional gap - June 17th, 2021
- Technical Design Document Completed - June 20th, 2021
- Start Development on cross-platform solution - June 21st, 2021
- Produce Runbook - June 23rd, 2021
- Finish cross-platform solution - July 19th, 2021
- Test Plan completed - July 22nd, 2021
- User Guide - July 26th, 2021
- Test Report - July 26th, 2021
- Feedback from mentors - August 2nd, 2021
- Feedback from sponsor - August 2nd, 2021
- Programmer's Guide Complete

Activities in the follow-up phase:

Provide a list of the activities that must take place in order to achieve the milestones. Specify who will carry out these activities, when and by whom they will be approved (ultimate responsibility).

Timeline:



Cost estimates:

Task Name	Cost	Work
<b>Development</b>	<b>\$88,848.29</b>	<b>1,776.97 hrs</b>
<b>Designing Phase</b>	<b>\$8,340.00</b>	<b>166.8 hrs</b>
Analysis of current ADF	\$540.00	10.8 hrs
Assessment of the functional gap	\$3,200.00	64 hrs
Technical Design Document	\$1,600.00	32 hrs

<b>Coding Phase</b>	<b>\$61,664.79</b>	<b>1,233.3 hrs</b>
Start development on cross-platform solution	\$650.00	13 hrs
Produce Runbook	\$490.00	9.8 hrs
Finish cross-platform development	\$25,627.29	512.55 hrs
<b>Testing Phase</b>	<b>\$17,713.50</b>	<b>354.27 hrs</b>
Test Plan	\$2,400.00	48 hrs
User's Guide	\$1,200.00	24 hrs
Test Report	\$600.00	12 hrs
<b>Listening Phase</b>	<b>\$1,130.00</b>	<b>22.6 hrs</b>
Feedback from mentors	\$80.00	1.6 hrs
Feedback from sponsor	\$80.00	1.6 hrs
Programmer's Guide	\$720.00	14.4 hrs

## 6 Review

The final section of the project plan provides an overview of the costs and the timeline for the entire project. The below table does not include a resource use budget as the aim of this project is to keep overall compute cost to a minimum by taking advantage of free subscription services. This concept will be updated as the cycle continues.

<u>Cycle</u>	<u>Expected starting date</u>	<u>Expected date of completion</u>	<u>Total cost estimate</u>
Planning	May 29 <sup>th</sup> , 2021	June 8 <sup>th</sup> , 2021	\$7,050
Designing	June 9 <sup>th</sup> , 2021	June 20 <sup>th</sup> , 2021	\$8,340
Coding	June 21 <sup>st</sup> , 2021	July 20 <sup>th</sup> , 2021	\$61,664
Testing	July 20 <sup>th</sup> , 2021	July 31 <sup>st</sup> , 2021	\$17,713
Listening	August 1 <sup>st</sup> , 2021	August 3 <sup>rd</sup> , 2021	\$1,130
		Total:	