CyberFirst Academy 2022

A person holding a microphone

Description automatically generated with low confidence

Software Development

**Student overview**

Table of Contents

[Introduction 2](#_Toc700689561)

[Outline and rationale 3](#_Toc302749498)

[How this will work 3](#_Toc852983136)

[Project Background 3](#_Toc210489693)

[Task 1: Core requirements 4](#_Toc1126997333)

[Task 2: User interface requirements 4](#_Toc1969310480)

[Task 3: Data issues to identify and report 4](#_Toc1198895876)

[Task 4: Coding standards and documentation 5](#_Toc1472853532)

[Technical information 5](#_Toc675168068)

[File format 5](#_Toc842762693)

[Batches 6](#_Toc2009649047)

[Data Range 6](#_Toc1244301751)

[Filename convention 6](#_Toc283956149)

[Additional information 6](#_Toc1313208508)

[Agile development suggestions 7](#_Toc271749228)

[Sprints (and sprint planning) 7](#_Toc642301442)

[Pair programming 7](#_Toc1063788283)

[Stand-ups 7](#_Toc357406313)

[Iteration review 7](#_Toc2017344765)

[Closing thought… 7](#_Toc1636575231)

[Support available during the project 8](#_Toc716307076)

[Breakdown of support opportunities 8](#_Toc1358634926)

[Panel presentation 9](#_Toc1501627258)

[Evidence to include 9](#_Toc1464100154)

[What you will be asked to complete at project's end 9](#_Toc868353688)

[Final thoughts 10](#_Toc2044303971)

## Introduction

*Welcome to the Software Development component of the 2022 CyberFirst Academy!*

Unlike earlier years where the aim has been to provide programming knowledge and hone practical programming skills, we decided to expand our ambitions and build a project which contains a typical Junior Developer-level coding challenge that you might encounter in the workplace.

Fortunately, you will not need to solve this independently!

You will form part of a 3- or 4-person Software Development "cell" who will work collectively, under the direction of a designated Project Lead, to produce a working solution by a set deadline. But remember, e*veryone* will need to contribute *something*.

## Outline and rationale

Why the change? *Good question!*

Despite the Software Development component receiving great feedback during the last few years, we felt it was time to step-up the challenge and rewards on offer.

Primarily this is due to the need to demonstrate the vital significance of Software Development in a problem-solving oriented workplace.

Secondly, this is to encourage everyone to engage with Software Development as a potential career pathway. With the rise of consumer-based mobile applications, web applications, microservices, and greater shifts to cloud technologies and virtualised infrastructure, there has never been a need for more code-aware (and capable) members of a workforce.

*We sincerely believe that coding is for everyone.*

And, if you don't already agree with that, we will want you to believe that too by the project's end.

## How this will work

We will form you into teams, headed by a Project Lead who has identified themselves as willing and capable to fulfil that role.

We will set you a Project task, then support you throughout this project's lifetime with a variety of scheduled sessions, exercises, workshops, contact methods, resources and tools that allow you to solve the set problem *together*.

At the project's end we will be interested in not only *how well* you solve the problem, but also *how* you solve the problem.

We will ask you to reflect on new skills learned, softer skills (communication, cooperation etc) that may have improved and whether you feel capable (and interested) in following this type of career pathway.

**Let's start by giving you some information on the project…**

## Project Background

A University has a trusted scientific partner for medical research. The University Medical school needs to be able to download pharmaceutical trial data from this partner in a secure and reliable fashion.

Although the scientific partner intends to move to a RESTful web-based API in the near future they are currently using a simple FTP service, despite the accepted risks and known vulnerabilities.

## Task 1: Core requirements

It is envisaged that a client-side application is needed that allows the staff in the Medical school to securely download data in both a scheduled (cron/scheduled task) and interactive manner, e.g., requesting data for a specific date (the default being "today").

Data must be validated (see [Task 3](#_Task_3:_Data)) before it is accepted, copied for investigation, and then securely archived in a logical directory hierarchy.

It is vitally important that there are no accidentally duplicated data sets as this could create bias problems during analysis, therefore a tracking mechanism must be used to keep each set unique.

## Task 2: User interface requirements

The application should have command line options (for automated scheduling) and a user-friendly interactive front-end (to launch manual requests). Both aspects should follow best practice principles in UI (user interface)/UX (user experience).

## Task 3: Data issues to identify and report

Previous manual data downloads have revealed problems with the downloaded csv files.

This has led to a suspicion that some data files are manually collated and edited before being uploaded to the FTP. Consequently, there has been some communication regarding the variable quality of the data.

Common issues identified include:

* Malformed files (causing difficulties when importing into other applications, e.g., spreadsheets, databases, cloud-based machine learning etc.)
* Duplicated batch\_ids (see [Technical Information](#_Technical_information))
* Missing headers or misspelt/incorrect headers, e.g. "batch" rather than "batch\_id"
* Missing columns on a row
* Invalid entries, e.g., reading values of 10 or greater
* Empty files (there are no "nil" returns)
* Incorrectly formatted filenames.

Files which contain these issues must be identified and logged using an appropriate technique; it is intended that these logs can be ingested by another application for monitoring and reporting purposes – this is currently outside development scope, however.

## Task 4: Coding standards and documentation

All code should be written and documented to appropriate standards using the selected programming language(s)

The precise method of coding is selected by the team and can contains components written in different languages, operating in concert. For example – PowerShell for FTP interaction, Python for data file processing etc.

It is critical that all code is held securely within a suitably secure remote repository that utilizes a popular VCS (version control system).

## Technical information

### File format

Trial data is exported from proprietary medical devices as CSV files which generally follow rfc4180 guidelines, cf. <https://tools.ietf.org/html/rfc4180>

**For example, in raw form:**

"batch\_id","timestamp","reading1","reading2","reading3","reading4","reading5","reading6","reading7","reading8","reading9","reading10"

187,"20:12:29",6.058,7.461,5.152,3.326,1.335,9.755,9.308,0.956,8.984,3.792

107,"20:12:29",5.44,2.228,6.581,7.723,6.53,7.702,5.751,5.448,2.532,4.585

171,"20:12:29",2.542,1.507,6.663,8.522,6.426,6.842,6.648,1.573,8.623,5.644

154,"20:12:29",8.025,2.678,6.618,6.641,4.096,2.961,4.771,3.243,8.225,9.766

65,"20:12:29",5.107,1.712,2.221,6.011,9.437,3.658,0.96,0.09,1.893,9.767

63,"20:12:29",5.479,4.965,7.766,8.875,8.85,4.994,5.863,1.428,2.574,4.946

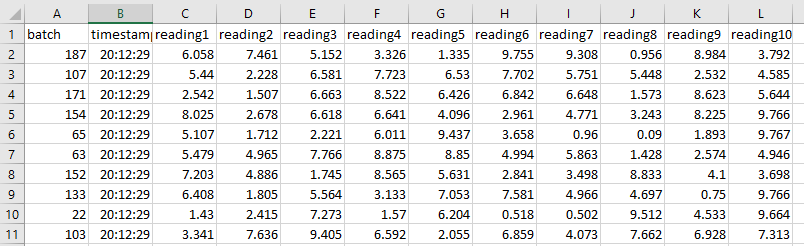
152,"20:12:29",7.203,4.886,1.745,8.565,5.631,2.841,3.498,8.833,4.1,3.698

133,"20:12:29",6.408,1.805,5.564,3.133,7.053,7.581,4.966,4.697,0.75,9.766

22,"20:12:29",1.43,2.415,7.273,1.57,6.204,0.518,0.502,9.512,4.533,9.664

103,"20:12:29",3.341,7.636,9.405,6.592,2.055,6.859,4.073,7.662,6.928,7.313

**and as viewed in Microsoft Excel:**



### Batches

Each discrete batch is identified by a **batch\_id** (field 1) and this value should be unique (though unsorted) in *each* .csv file; duplications would indicate a faulty .csv file and should be logged and excluded from further study.

Batches within a single file may be exported within the same second or over multiple seconds, depending on throughput of the scientific partner's proprietary devices, i.e. multiple entries for the same timestamp are not considered unusual.

### Data Range

Each batch has 10 associated readings.

All 10 readings should be represented as floating point numbers formatted up to three decimal places with no value *exceeding* 9.9 (this would be considered invalid data).

### Filename convention

Filenames for trial data exports are created using the following naming convention:

MED\_DATA\_YYYYMMDDHHMMSS.csv

For example:

MED\_DATA\_20210505201229.csv

Would represent a trial data .csv file created at 8:12:29 pm on 05/05/2021.

**Note**. The inclusion of the timestamp allows multiple .CSV files to be exported from the proprietary devices in a single day.

## Additional information

* Target users are technically proficient and are able to use the operating system's command prompt/shell but are not considered to be software developers or network engineers.
* Although the reference platform is considered to be Microsoft Windows 10, the solution should be platform independent, i.e., it should work unmodified on a Microsoft Windows client or a Linux distribution or an Apple Mac.
* Target executables are not mandatory, i.e., interpreted scripts are acceptable.
* Best reasonable efforts should be made to make the code resilient both in terms of *execution*, i.e. using effective error handling to increase robustness of solution and *use*, by sanitising all user inputs to prevent malicious access or modification of data and/or code.

## Agile development suggestions

Remember, this project isn't intended as an individual effort – all members of your development team should contribute *meaningfully* to the presented solution.

As such it is wise to review and reflect upon standard agile development practices which you may find useful during the project period. Key aspects include:

### Sprints (and sprint planning)

During a particular time window (e.g. typically lasting 1 or – more common - 2 weeks), the items from the project backlog that are identified as highest priority and needed to be completed. These items are usually tracked on an agile board (manual or electronic) in various stages, e.g., in minimalist terms: "open", "in progress", "done". The latter, after testing, of course. 😉

A completed sprint represents the next iteration of a product; this may well contain features which are only partially implemented.

### Pair programming

A technique where two programmers work at one workstation (ideally), or remotely using screen sharing, to develop code collectively. Each programmer takes turns to code or review/give feedback, swapping roles frequently. Not only can it assist problem-solving, but it is also a good way to exchange ideas, knowledge and techniques.

Moreover, estimates say that code developed in this way reduces errors in a codebase by up to 60%.

### Stand-ups

A quick 10–15-minute meeting where each individual's status on their respective programming tasks within the current sprint can be discussed. Coverage typically includes: what's been completed, what's still being worked on, and (crucially): Are they are any impediments (slowing) or blockers (halting) meaningful progress?

### Iteration review

A time to demonstrate progress which has been achieved by the team at the end of a sprint. Typically these are done to receive feedback from the project stakeholders.

### Closing thought…

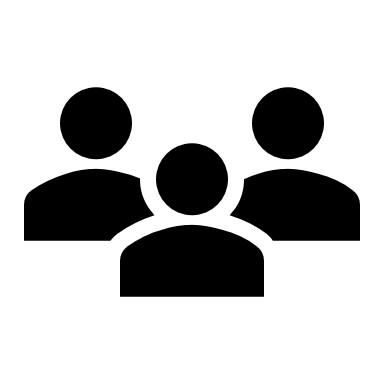
It is worthwhile remembering that knowing a programming language and understanding its syntax, core features, popular libraries, strengths, and weaknesses is only a part of becoming a software developer.

*Communication skills and contribution to an agile workflow are just as important.*

## Support available during the project

Your coding development project runs in parallel to your CyberFirst Academy Programme. Various resources are available to assist your completion of the task.

The following diagram demonstrates the general flow through the designated weeks:



Introduction to Python

Introduction to PowerShell

Introduction to C

Introduction to GIT

Project Workshop #1

Project Workshop #2

Project Panel Presentation

Project Feedback

Self-assessment (post-project)

Self-assessment (pre-project)

Pre-programme

Team formation

Some scheduled project time in the next few weeks

## Breakdown of support opportunities

The software development project is purposively designed to be tackled using any of the three targeted programming languages: C, Python or PowerShell.

Each language presents its own idiosyncrasies, advantages, and challenges!

We know from experience that practical levels of software development vary greatly amongst a single cohort. As such, prior to the start of the academy a series of questions were asked which encouraged you to rate your own development skills in these languages, use of version control systems and other project-related criteria.

Those who have been identified as potentially benefitting from software support should attend any of the three-day workshops we've organised.

Each day will provide a working knowledge of the programming languages selected. Remember, these days are designed to introduce the languages and their core features. You'll need to dedicate some of your own time to practice and fine-tune your skills as the slide-based materials and exercises will cover much more than we can explore interactively in a single day.

It is anticipated that a team should be able to select the language which it feels is (a) most appropriate to the type of solution they want to develop and (b) most appropriate to the skill levels and preferences of the development team. That said, it is possible to build a solution using different language components, e.g. one script getting the data files, another executable validating them etc.

Trainer-led sessions for each language are:

* Introduction to Python
* Introduction to PowerShell
* Introduction to C

In addition, we have created a self-study programme for working with version control systems.

* Introduction to Git

This can be studied *at any time* but it is a good idea to review it early in your project's lifecycle so you can contribute fully (and correctly) to your team's project codebase.

A fourth day is also provided, geared to providing project support. Additional "twilight" workshops will be scheduled which provide optional drop-in support with a Programming Specialist:

* Project workshop #1
* Project workshop #2

These optional sessions will allow you to discuss problem solving, potential solutions and code issues (impediments or blockers) with your Trainers during the lifetime of the project.

Of course, Trainers and Course Ambassadors will also monitor communication channels such as Microsoft Teams to answer specific queries as and when needed.

## Panel presentation

You may have the option to present your completed solution to the panel during a designated timeslot in the target week and receiving feedback. This is likely to be 19th August.

For those who do not present, could you submit slides explaining your solution.

Presentations should be 10 minutes long.

## Evidence to include

The project's assessment criteria are based on nine key elements that focus on a combination of technical and soft skills:

1. Code artefacts
2. Code production
3. User Experience (UX) Design
4. User Interface (UI) Design
5. Problem solving
6. Communication
7. Cooperation
8. Time management
9. Team leadership

Feedback will be provided on a group basis after the completion of the Project Panel Presentation.

## What you will be asked to complete at project's end

Two items, essentially.

1. We will ask you to reflect and produce a reflective "What I have learned" statement (max. 250 words or ½ side of A4 page, single spaced).

You should include changes to your Software Development proficiency and also targeted soft skills.

1. We will also ask you to revisit the Software Development-related questions which were asked at the start of the academy.

## Final thoughts

Please don't treat this like an academic project you might complete in college or university.

Although the technical outcomes are obviously important and assessment is informative, it is more useful as a vehicle to self-assess your preferences, willingness to learn, problem solve and work together. These are all personal qualities the client highly values.