By William Adoukonou

Object oriented Programming Assignment 1

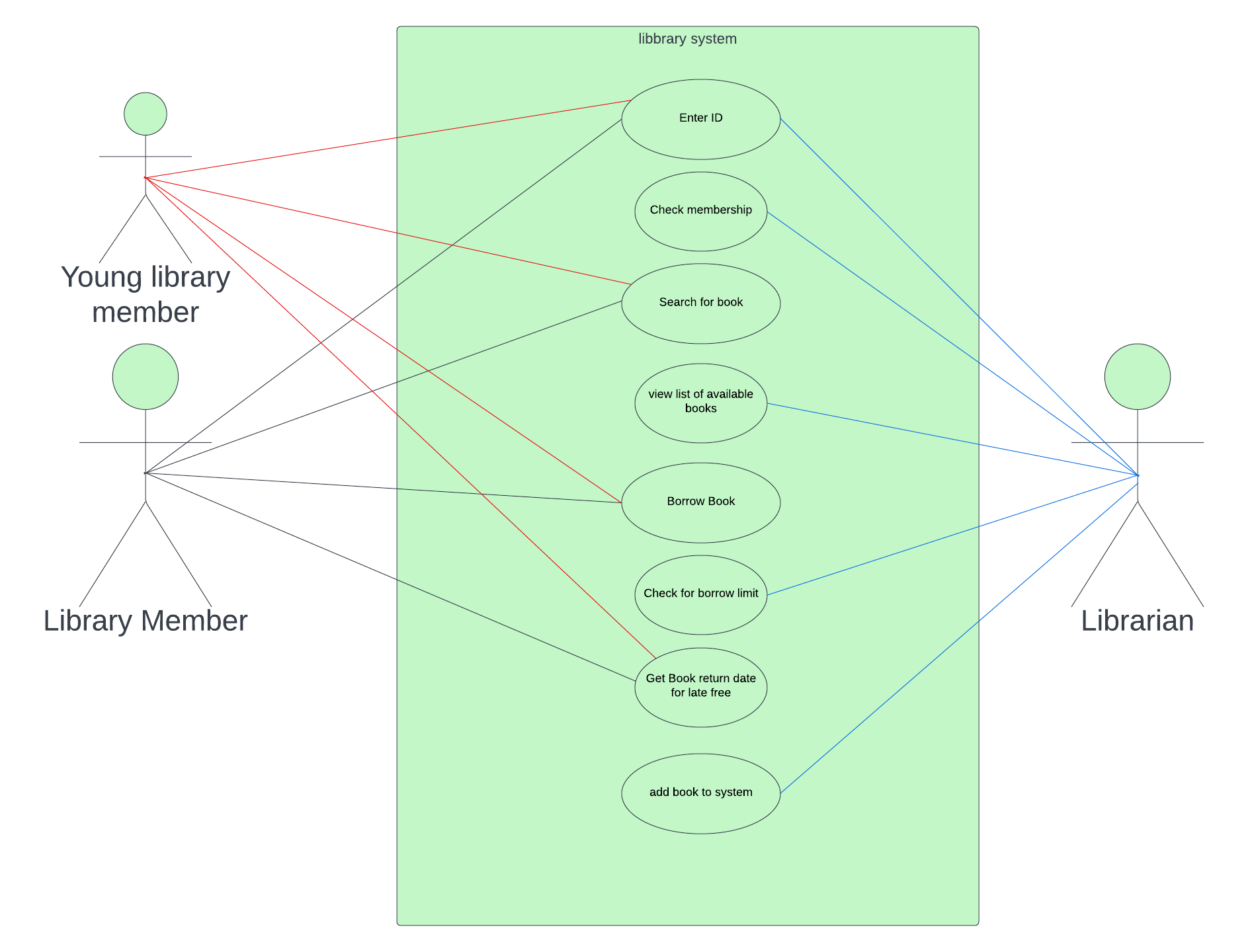
# Introduction

(Intro goes here)

# Task 1

In this task I created a use case diagram that explains the steps that a user would take to use the library system. It also gives scenarios for the librarian or young member and the steps they would take to complete some of the options on their respective library menus.

## Use Case Diagram



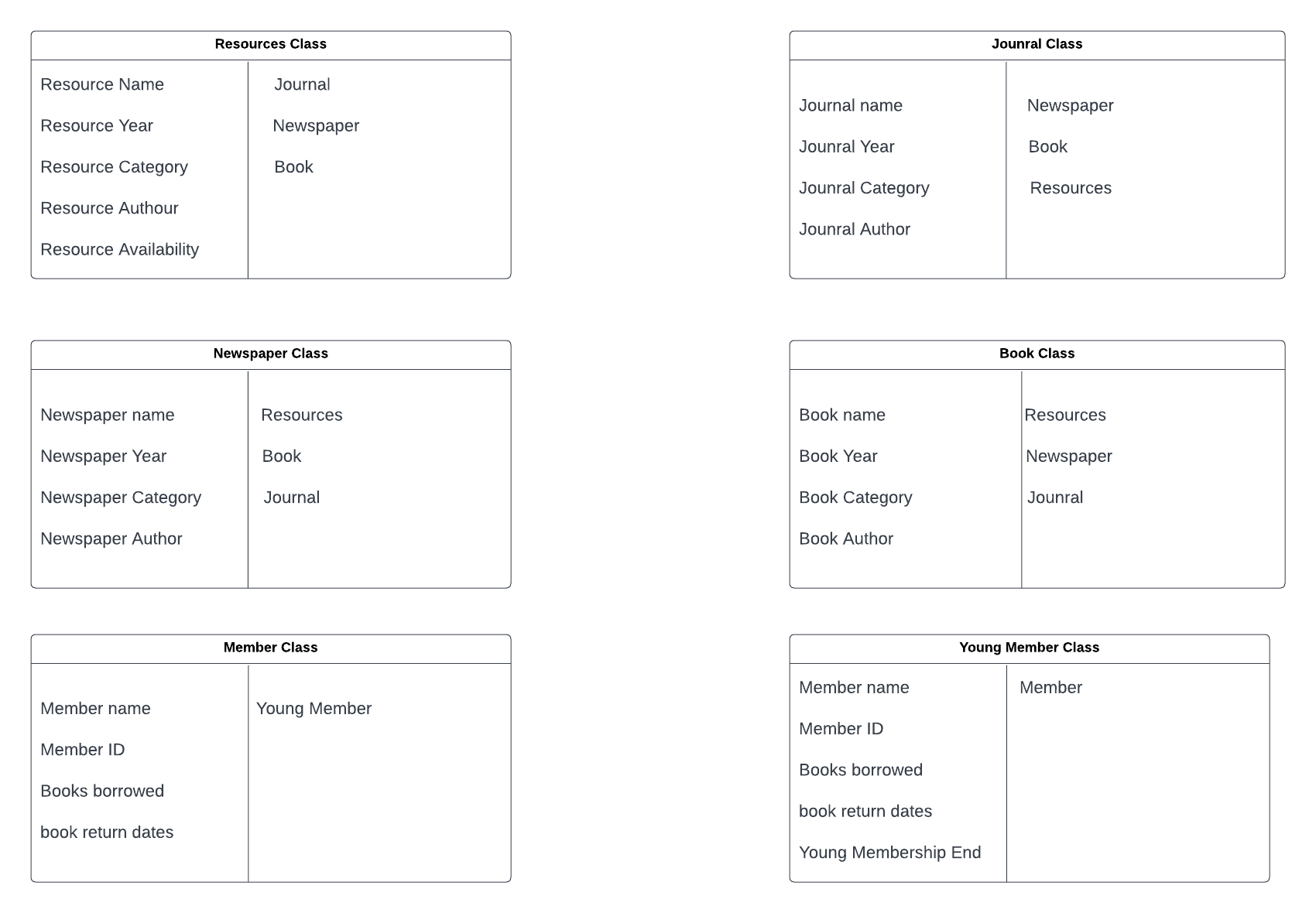
# Task 2

Here I have created CRC cards. These show each individual class that will be used, and which other classes will be linked to them.

I also designed class and object diagrams which act as a blueprint for the entire project, highlighting the parent class and how the child classes inherited features from it.

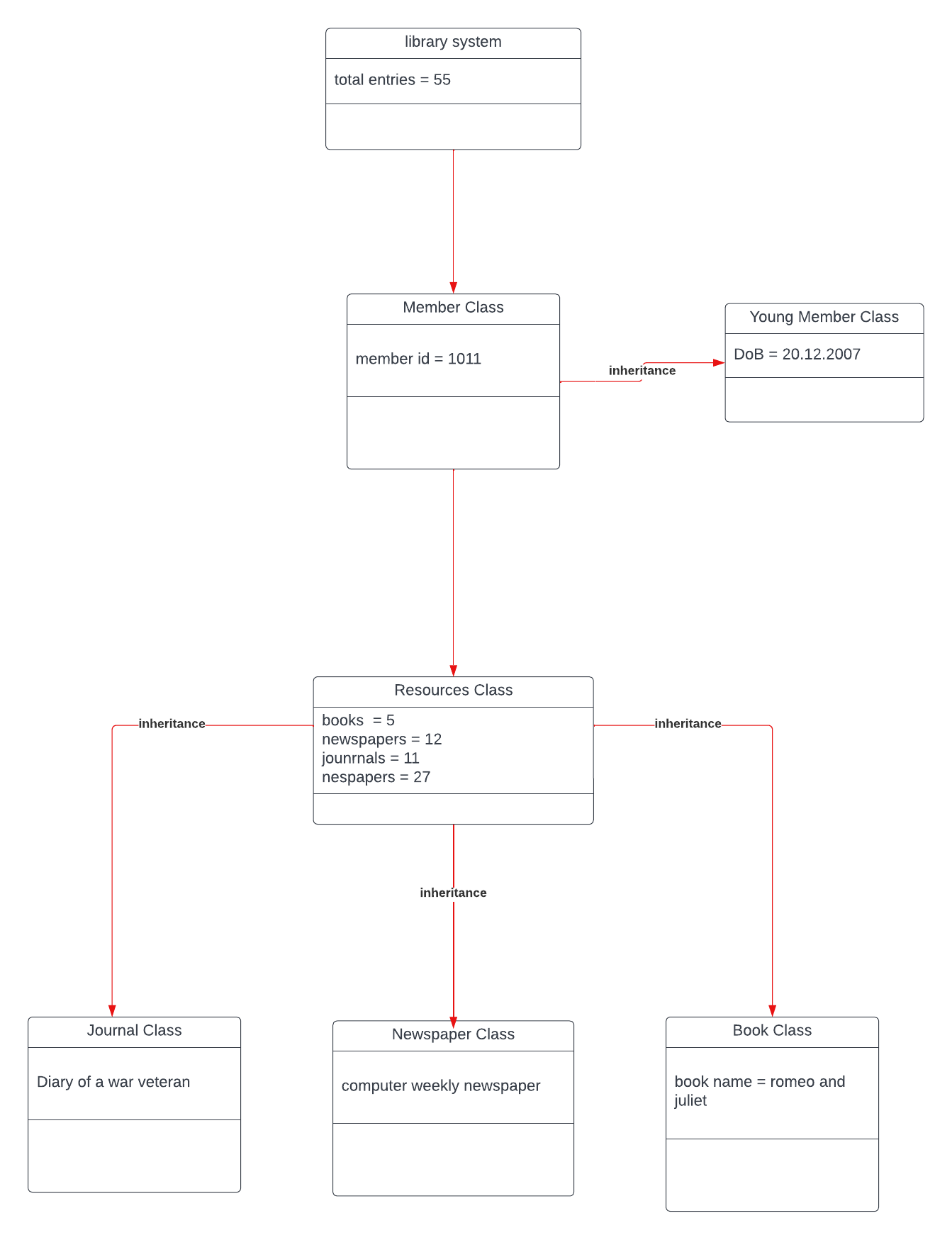
Finally, I took these designs and created pseudocode based on them.

## CRC Cards



## Class Diagram

## Object diagram



## Pseudocode

Text

Description automatically generatedText

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

# Task 3

This is the part where the actual code is implemented. I have provided screenshots of each step.

## User Menu

Text

Description automatically generated

Text

Description automatically generated

Code for initial menu. Code to keep the menu there if user does nothing

Text

Description automatically generated

Code to check user ID

Code to add borrowed book to text file

If book is unavailable

Text

Description automatically generated

Checks borrow book list to see if book can be returned

Checks for return date. If return date is more than 3 weeks, £5 added to fines record

Text

Description automatically generated

Shape

Description automatically generated

Quit option for quit button

Age check, if age is under 18, user account is young person.

## Manager Menu

Text

Description automatically generated

Text

Description automatically generated

Code to add book in format from the book class

Adds book to file of current books

# Task 4

After I was finished programming the system, I designed a plan showing each of the tests I did to make sure the code was working as It was supposed to.

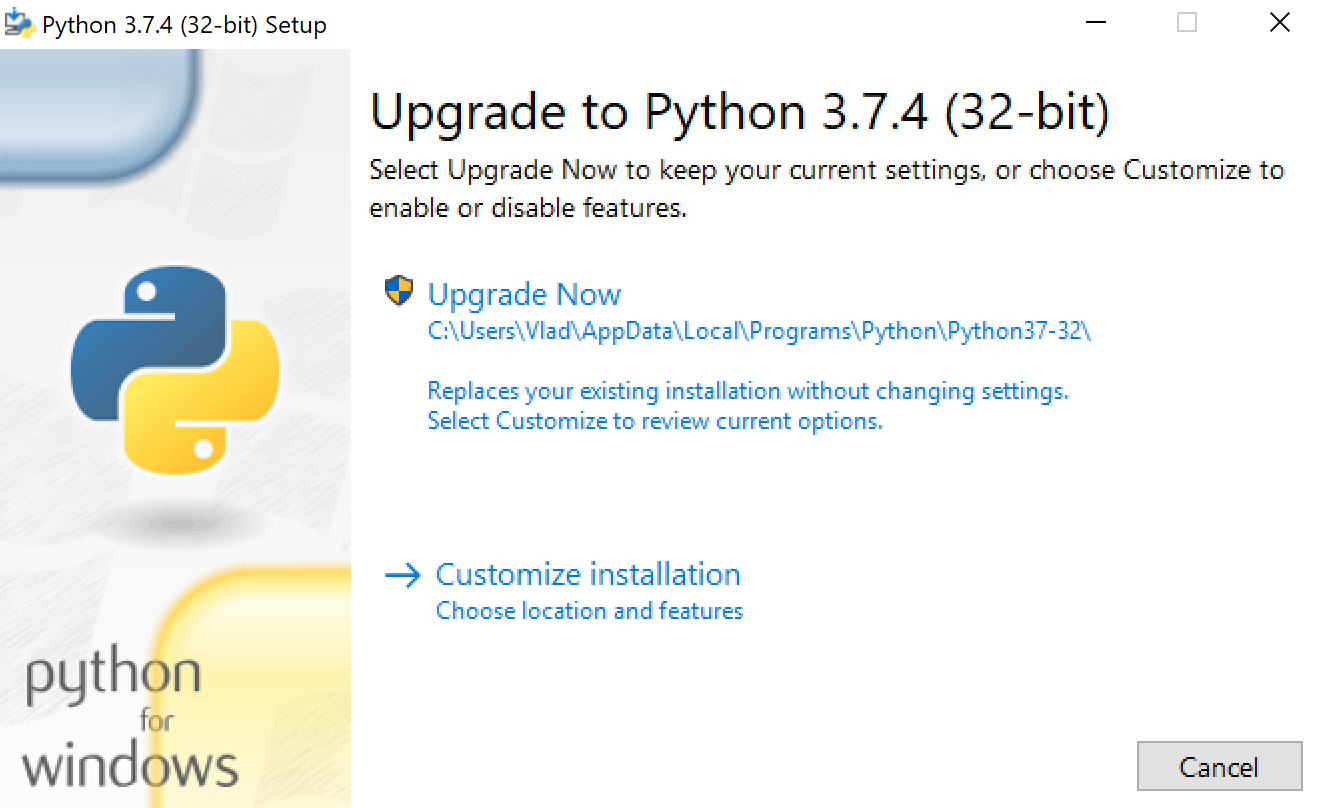
Text

Description automatically generated with medium confidence

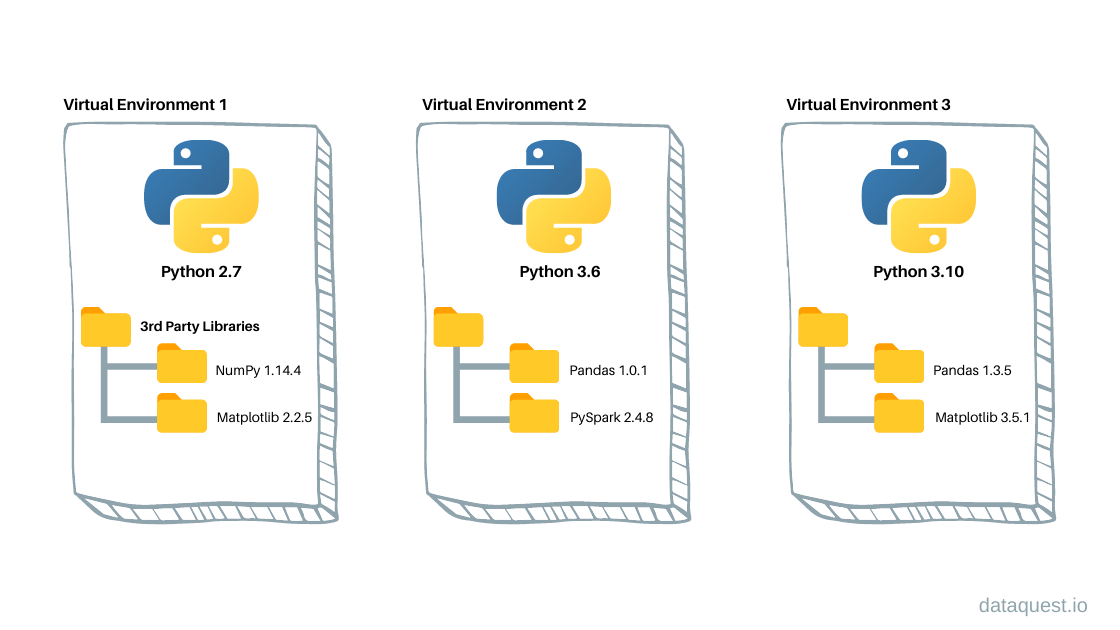
# Task 5

To keep in line with cyber security practices, this section will cover the best ways python developers can keep their code safe and secure.

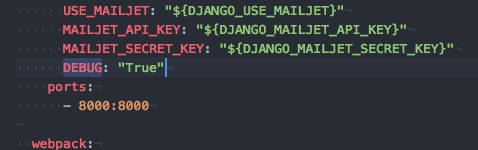
## Method 1

The easiest precaution programmers can take is to make sure their version of python is up to date. Using outdated software can leave you exposed to cyber threats. This is because the older versions may have security vulnerabilities that hackers can exploit.

## Method 2

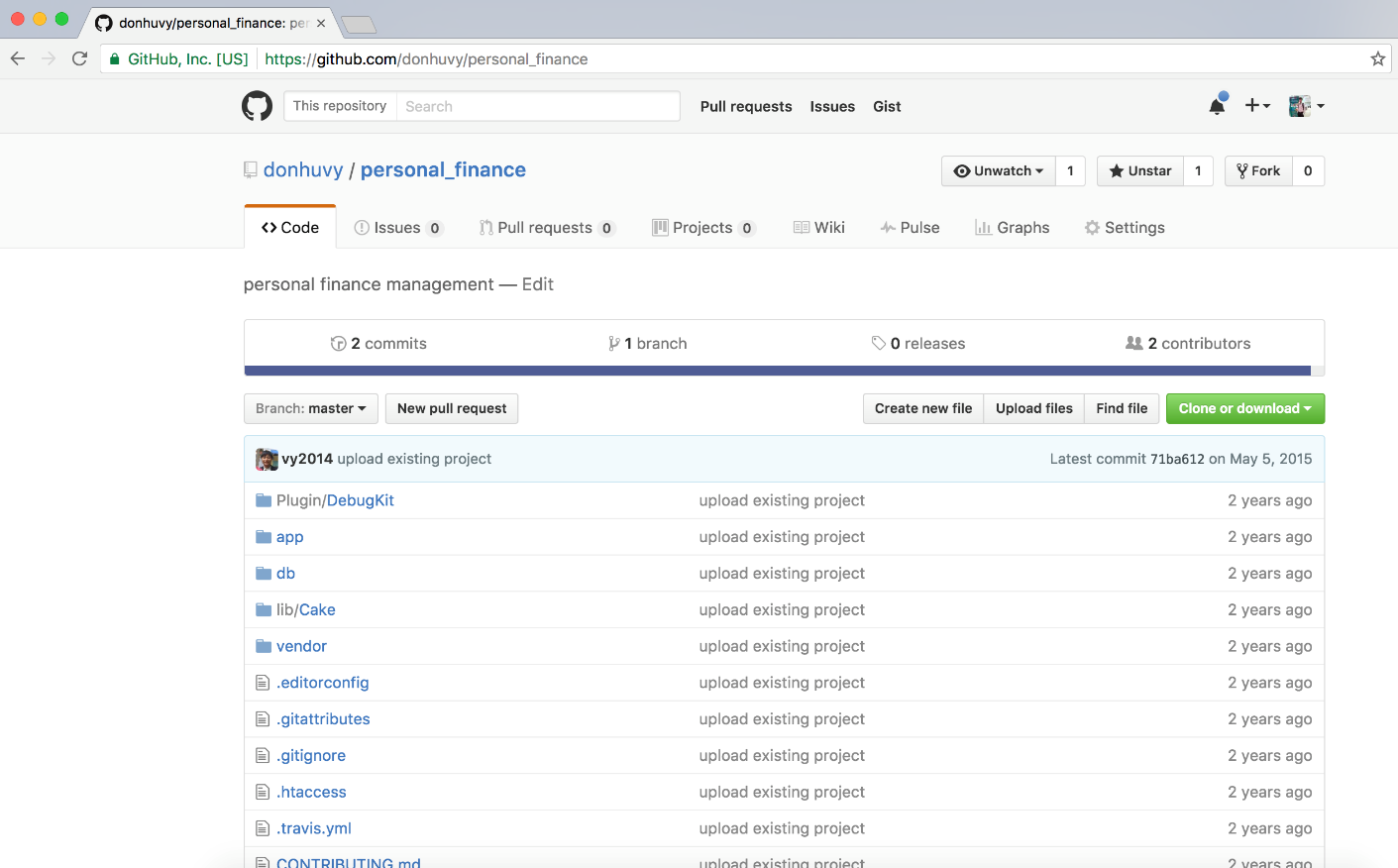
The 2nd thing python programmers can do is to use a Virtual Environment. This allows the developer to install packages to a non-physical storage space on a different installation of python. This is important as a user might install a malicious package, but it won’t affect the user’s system since it was installed to a virtual environment.

## Method 3

When programmers are ready to publish their code, they like to set the debug tool to TRUE. This helps them identify possible errors that may occur. This can make them more vulnerable when the code goes live. Everyone who has access to the program can see where the errors are and find potential exploits.

## Method 4

Another mistake programmers can make when publishing their code to sites like GitHub is leaving password files in the final folder. Once there, any user can simply locate the file in the database and exploit it. Storing the passwords separately will help keep private files protected.



## Method 5

Finally, python programmers need to be careful when installing packages. Its one of the easiest ways for hackers to infect your system with malicious code since its downloaded directly by the user. Its important to only get python and node.js packages from trusted websites that are on the most up to date version.