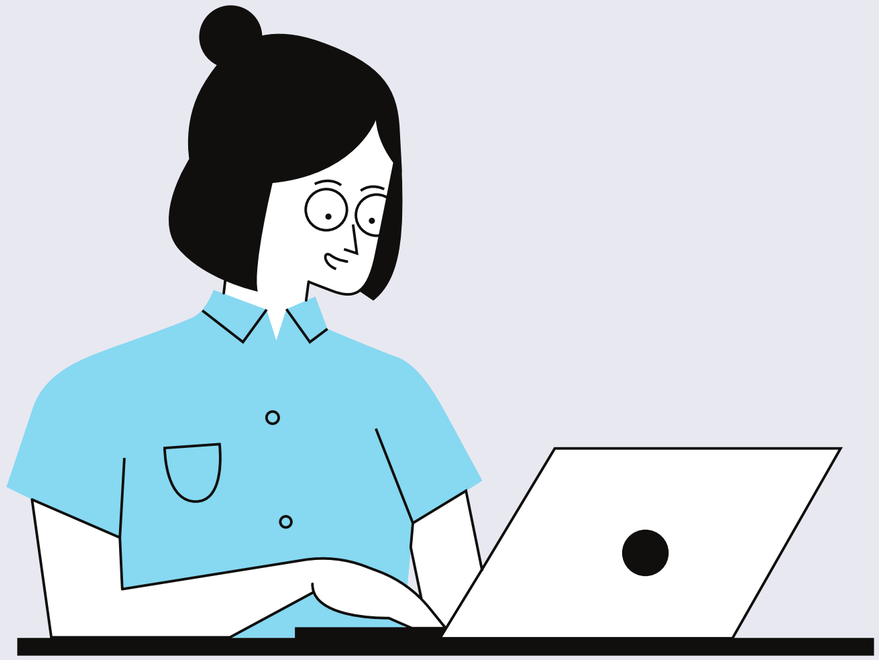
PayClear

"On track and in the know about your money"

APL Individual Assignment #1
Done by: **Murphy Facey**

Key terms

interpreter. compiler. platform-independent. JIT. file manipulation. frameworks. exception handling. data structures. simple syntax. transparency. object-oriented paradigm.



Case Scenario

PayDay is a start-up financial tech company found in the Caribbean.

PayDay is aimed at creating a personalized payment processing system that lets how your money is being spent.

Their payment processing system is affectionately called, PayClear.

Case Scenario:
Features

PayDay has commissioned me to create our system, PayClear.

PayClear should have the basic features of other payment processing systems.

Payclear should also have the following key features.

1. Transparency
2. Personality
3. Security
4. Localization
5. Availability
6. Quick Transactions

Interpreted vs Compiled

My Proposed Language

The programming language chosen to create the system is Ruby, which is an interpreted language.

An interpreted language is a language that is compiled and ran through an interpreter.

A compiled language, on the other hand, does not need an interpreter and runs on the local machine.

Features: Transparency

Transparency refers to both clients' and investors' ability to access financial information about companies. This provides clarity about their charges.

We will build an API to interface with the system of the banks the company partners with.

Application Programming Interface (API), which is a software intermediary that allows two applications to talk to each other.

Features: Transparency

Some of the things to consider when building an API:

1. Comfort and Ease to Develop
2. Supported frameworks to assist in development
3. Communities for the language

Example of Simplistic and Elegant Syntax

```ruby
# Output "I love Ruby"
say = "I love Ruby"
puts say

# Output "I *LOVE* RUBY"
say['love'] = "*love*"
puts say.upcase

# Output "I *love* Ruby"
# five times
5.times { puts say }
```

Features: Personality

It is said that about 2.5 quintillion bytes are collected daily in the year, 2020. Information has become one of the world's greatest currencies.

To manage this precious commodity, Ruby has a set of data structures that will be used to stored and manage the information in an efficient way.

There is also the object-oriented paradigm that Ruby supports.

Features: Security

Our company will be storing sensitive information about our clients through the process of getting the system to adapt to them.

It should be known that interpreted languages are not the best for security

HOWEVER,

There is an exception handling system within Ruby. It also allows for the developer to create our own custom exception handling system.

Example of Exception Handling in Ruby

```ruby
begin
  puts 'This is Before Exception Arise!'

  # using raise to create an exception
  raise 'Exception Created!'
  puts 'After Exception'

# using Rescue method
rescue
  puts 'Finally Saved!'
```

Features: Localization

Localization is the process of adapting or translating software to a specific locale’s language, culture, and legal requirements.

The translations for our supported languages will be stored in files. Ruby allows for file manipulation so translations can be read from the file and then display to our clients.

Example of File Manipulation in Ruby

```
# File Handling Program

# Creating a file
fileobject = File.new("sample.txt", "w+");

# Writing to the file
fileobject.syswrite("File Handling");

 # Closing a file
fileobject.close();
```

Features: Availability

Availability speaks to the fact our system should running all the time so, our client can get access to it at any time.

Ruby, being an interpreted language, creates platform-independent code which assists in making the system more accessible to the different platforms.

There are several hosting options for Ruby as well as its popular framework, Ruby in Rails.

Features:

Quick Transactions

Compiled languages are usually faster than interpreted languages. However, Ruby is one of the fastest interpreted languages which will guarantee good performance.

Ruby also has a Just In Time (JIT) interpreter which lessens the overall execution time.

Features:

Quick Transactions

Just In Time Interpreters is an intermediary between interpreters and compilers. It allows for a portion of the code to be compiled and run through the virtual machine.

Disclaimer

Ruby is not the best programming language for everything. There are limitations to only using the interpreted languages.

So Ruby is proposed as a base for which all of other languages and frameworks can be added.

This is to make our system more complete in its development, and feature set that it will later provide.