University of Maryland Global College

CMSC 495

Project PLan

Andre Davis, Andres Giraldo, Paul Hendrick, David Machado, Rhea Prohaska, Malachi Woodlee

**Table of Contents**

1. **Purpose**…………………………………………………………………………………………..2
   1. Document Overview…………………………………………………………..2
   2. Project Description…………………………………………………………….2
2. **Scope Management Plan**……………………………………………………………….2
   1. Requirements Definition……………………………………………………2
      1. Project Requirements……………………………………………….2
      2. System Requirements……………………………………………….3
   2. Scope Verification………………………………………………………………4
   3. Scope Control……………………………………………………………………4
3. **Communication Management Plan………………………………………………4** 
   1. Introduction………………………………………………………………………4
   2. Emails…………………………………………………………………………………4
   3. GitHub.com…………………………………………………………………………4
   4. Group Collaboration – Group – 5………………………………………4
4. **Staffing Management Plan……………………………………………………………5**
5. **Schedule Management Plan…………………………………………………………6**
   1. Schedule……………………………………………………………………………6
   2. Tools to Manage Schedule………………………………………………….6
   3. Schedule Responsibility………………………………………………………7

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| **10/24/19** | **1.00** | **Initial start of project planning** | **Andre Davis** |
| 10/26/19 | 1.01 | Grammar edits | Malachi Woodlee |

**1 Purpose**

**1.1 Document Overview**

This Project Plan outlines the ~~developing~~ process of developing group 5’s program ~~design~~. This document will cover ~~all~~ the planning, execution, monitoring, control, and the closing of the project on completion. This project plan will ~~go into~~ provide details ~~of~~ for group 5’s approach to managing the project while ensuring that the project performance ~~will be~~ is optimized. The project plan will is divided into ~~include~~ the following ~~plans~~ sections:

* **Scope Management Plan**
* **Communication Management Plan**
* **Staffing Management Plan**
* **Schedule Management Plan**
  1. **Project Description**

The main goal of our program is to create an email parser that ~~would go through several emails and separate all the “junk” mail out of them~~ takes an email file, \*.eml, as input, extracts the necessary data and formats the data into the required layout for use in a Random Forest Classifier. The program can be broken up into three different parts. First, having an API for the communication with the mail sever. Second, having a handler that would take the emails from the API, parse it, and then output a CSV file. Lastly, the third is having an API for sending the CSV file to the Random Forest Classifier.

The end user for our application is another application that would be developed by another team, who would be working in parallel with group 5. This is because group 5’s project is envisioned as being part of a larger project, the detection of malicious emails. This scenario simulates a real world possibility for our future as developers where we would work along side other teams creating an application that is a subset of the larger application under development.

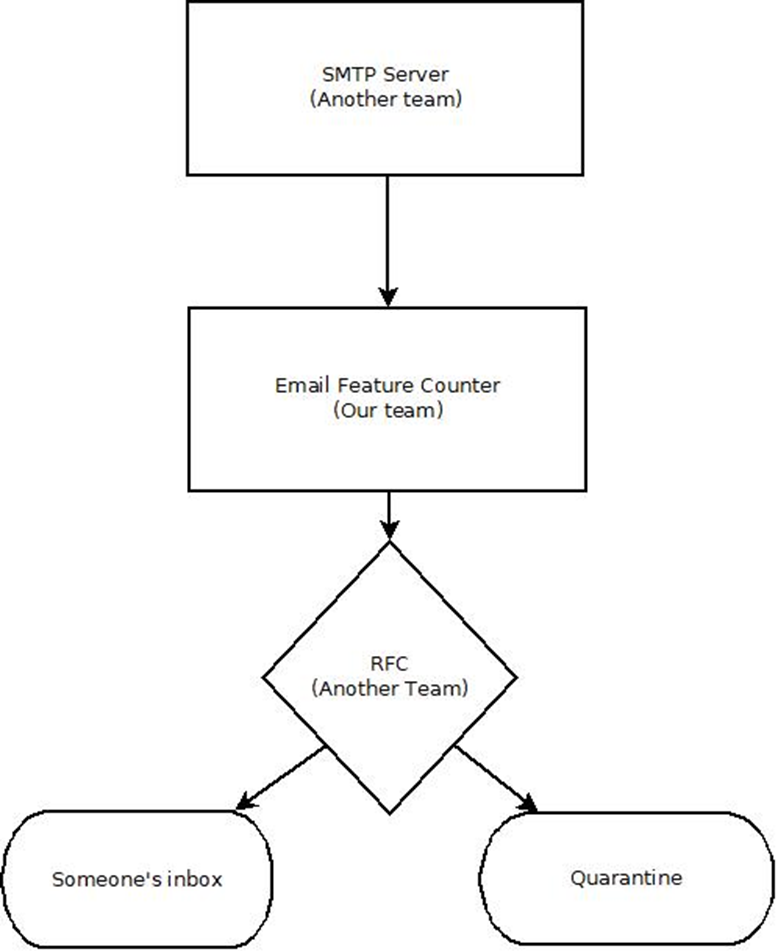
Our email parser is based upon the research of Aviad Cohen, Nir Nissim, and Yuval Elovici. It will scan for features using the novel set of features published by Cohen, Nissim, and Elovici in the article *Novel set of general descriptive features for enhanced detection of malicious emails using machine learning methods.*

1. **Scope Management Plan**
   1. **Requirements Definition**

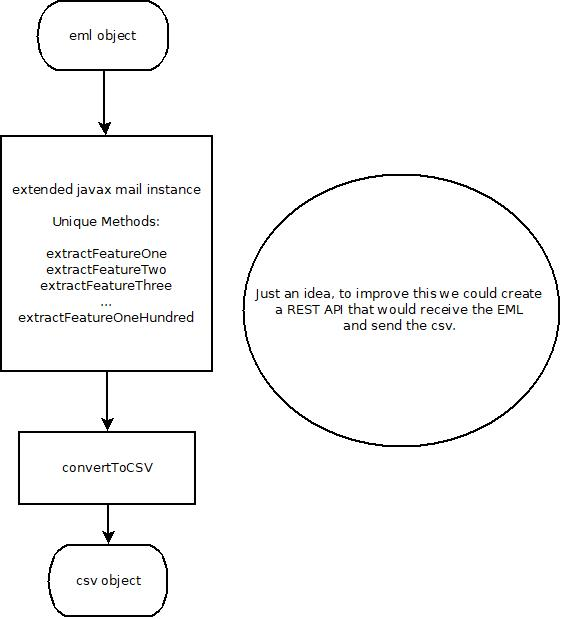
The requirements are broken into two different parts for the program. ~~One~~ First are the Project Requirements and ~~the other being~~ Second are the System requirements.

* + 1. **Project Requirements**

The main requirement of the project is to be ~~able to tell junk mail from other email~~ to parse emails noting the presence, or lack thereof, of features. This data is then formatted into a layout that can be converted into a csv. This chart will show the flow ~~of how the process~~ starting with receiving the email, next the email going through our program, and then being classified by the RFC before being delivered to someone’s inbox or being quarantined.



The next figure shows the flow of the email parser.



* + 1. **System Requirements**

The recommended hardware system requirements will go as follow

* JDK/JRE 1.7 or higher
* Microsoft Windows 7 Professional or higher
  + Processor: Intel Core i5 or equivalent
* Ubuntu 15.04 or higher
  + Processor: Intel Core i5 or equivalent
* OS X 10.10 Intel or higher
  + Processor: Dual-Core Intel or equivalent
  1. **Scope Verification**

The deliverables will be formally accepted through the electronic communications that are define in the Communication Management Plan. The process of a full team review on all deliverables will ensure their quality will meet the requirements of the overall project. The team review will also include testing the application as well as appropriate sing offs on all deliverables.

* 1. **Scope Control**

The team as a while will monitor the scope of the project through regular reviews to make sure the program is advancing to the same goal as stated in the project plan. Any and all changes to the project scope must be approved by all team members and the project plan must be revised to include the changes.

1. **Communication Management Plan**
   1. **Introduction**

The Communication Management Plan describes the planned methods of communication within the team itself. This will help ensure timely intercommunication which will help with ensuring the program quality.

* 1. **Emails**

The team will use emails as the primary use for communication. This is where the team will ask any questions for anything about the project. This is also where team will discuss any changes to the project and vote on matters as well.

* 1. **GitHub.com**

The project will be using GitHub.com for software control management. The features that we will be using is

* Source code management
* Version control
  1. **UMGC Group Collaboration – Group – 5**

This will be used in addition to the emails as a secondary form of communication within the team. In addition, any rough drafts of any documentation will be posted here for the whole team to view over and provide any changes.

**3.5 Discord**

Discord will be used to facilitate weekly group meetings. Discord provides audio and video capabilities enhancing collaboration between group members.

1. **Staffing Management Plan**

The staff on this team has been assign these roles and tasks as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Roles** | **SDLC Lead** | **Responsibilities** | **Team Member** |
| Project Manager | Functional Design | * Project Plan * Organize all project documentation * Schedule meetings | Andre Davis |
| UX/HCI | Requirements | * Assisting Project Design to make program more user-friendly | \*Andres Giraldo |
| Technical Writer | Requirements | * User’s Guide | Paul Hendrick |
| Software Engineer | Development | * Project Design | David Engineer |
| Integration Engineer | Development | * Project Design | Rhea Prohaska |
| Test Engineer | Test | * Test Plan | Malachi Woodlee |

1. **Schedule Management Plan** 
   1. Schedule

This is the schedule that the program will need to follow:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Duration in Days** | **Start Date** | **End Date** | **Roles assigned to task** |
| Project Plan and GitHub account setup | 7 | 10/27/2019 | 11/03/2019 | Project Lead |
| User Guide and Test Plan | 7 | 11/03/2019 | 11/10/2019 | Technical Writer  UX/HCL  Test Engineer |
| Project Design | 7 | 11/10/2019 | 11/17/2019 | Software Engineer  Integration Engineer |
| Phase 1 Source | 7 | 11/17/2019 | 11/24/2019 | Project Lead  Software Engineer  Integration Engineer  Test Engineer |
| Phase 2 Source | 7 | 11/27/2019 | 12/01/2019 | Project Lead  Software Engineer  Integration Engineer  Test Engineer |
| Phase 3 Source | 7 | 12/01/2019 | 12/08/2019 | Project Lead  Software Engineer  Integration Engineer  Test Engineer |
| Final Report | 7 | 12/08/2019 | 12/15/2019 | Project Lead  UX/HCL  Technical Writer  Software Engineer  Integration Engineer  Test Engineer |

* 1. Tools to Manage Schedule

The tools the team utilize regularly scheduled meetings as well as communications that were defined by the Communications Management Plan to measure progress. During the meetings the team will assess the progress of the current task, determine if it is on schedule and provide remediation for any tasks that are behind schedule. Example remediation would be additional members helping with the task.

* 1. Schedule Responsibility

The project lead will be accountable for maintaining and reporting the schedule. The project lead will also be responsible organizing any help, if needed, to ensure the project remains on schedule