# 

**My signature indicates approval of this Project Plan.**

**Prepared by:**

Andre Davis

Project Manager

**Approved by:**

Andres Giraldo

UX/HCI

**Approved by:**

Paul Hendrick

Technical Writer

**Approved by:**

David Machado

Software Engineer

**Approved by:**

Rhea Prohaska

Integration Engineer

**Approved by:**

Malachi Woodlee

Test Engineer

**Approved by:**

Executive Sponsor



Phish Phinders Project Plans



Version <1.05>

**Table of Contents**

1. **Purpose…………………………………………………………………………………………………………………………………..4**
   1. Document Overview………………………………………………………………………………………………………4
   2. Project Description…………………………………………………………………………………………………………4
2. **Scope Management Plan…………………………………………………………………………………………………………5**
   1. Requirements Definition………………………………………………………………………………………………..5
      1. Project Requirements……………………………………………………………………………………………5
      2. System Requirements……………………………………………………………………………………………8
      3. Nonfunctional Requirements…………………………………………………………………………………8
   2. Scope Verification………………………………………………………………………………………………………….8
   3. Scope Control…………………………………………………………………………………………………………………8
3. **Communication Management Plan…………………………………………………………………………………………9**
   1. Introduction……………………………………………………………………………………………………………………9
   2. Emails…………………………………………………………………………………………………………………………..10
   3. GitHub.com………………………………………………………………………………………………………………….10
   4. Group Collaboration – Group – 5………………………………………………………………………………….10
   5. Discord…………………………………………………………………………………………………………………………10
4. **Staffing Management Plan…………………………………………………………………………………………………….11**
5. **Schedule Management Plan………………………………………………………………………………………………….12**
   1. Work Breakdown Structure………………………………………………………………………………………….12
   2. Schedule………………………………………………………………………………………………………………………13
   3. Tools to Manage Schedule……………………………………………………………………………………………14
   4. Schedule Responsibility………………………………………………………………………………………………..14
6. **Risk Management Plan………………………………………………………………………………………………………….14**
   1. Schedule Delay…………………………………………………………………………………………………………….14
   2. Team Member Resignation…………………………………………………………………………………………..14
   3. Team Member Negligence……………………………………………………………………………………………15
7. **Cost Management Plan…………………………………………………………………………………………………………15**
8. **Quality Management Plan…………………………………………………………………………………………………….15**
   1. Quality Management Method………………………………………………………………………………………15
   2. Quality Standards…………………………………………………………………………………………………………15
   3. Quality Management Tools………………………………………………………………………………………….15
   4. Quality Control Roles and Responsibilities……………………………………………………………………15
   5. Quality Control………………………………………………………………………………………………...............16
   6. Quality Assurance………………………………………………………………………………………………………..16
9. **Change Management Plan…………………………………………………………………………………………………….16**
   1. Problem Reporting and Closure……………………………………………………………………………………16
   2. Action Item and Issue Resolution…………………………………………………………………………………16
10. **References……………………………………………………………………………………………………………………………17**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |
| 10/27/19 | 1.02 | Grammar edits | Rhea Prohaska |
| 10/28/19 | 1.03 | Corrections and adding information about meetings | Andre Davis |
| 10/30/19 | 1.04 | Added new sections and change some of existing information | Andre Davis |
| 11/02/2019 | 1.05 | Added all sign names and added Nonfunctional Requirements | Andre Davis |

**1 Purpose**

**1.1 Document Overview**

This Project Plan outlines the process of developing Phish Phinders’ program. This document will cover the planning, execution, monitoring, control, and the closing of the project on completion. This project plan will provide details about Phish Phinders’ approach to managing the project while ensuring that the project performance is optimized. The project plan will be divided into the following sections:

* **Scope Management Plan**
* **Communication Management Plan**
* **Staffing Management Plan**
* **Schedule Management Plan**
* **Risk Management Plan**
* **Cost Management Plan**
* **Quality Management Plan**
* **Change Management Plan**
* **References**
  1. **Project Description**

The main goal of our program is to create an email parser that 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, creating an API for communication with the mail server. Second, a handler would take the emails from the API, parse it, and then output a CSV file. Lastly, an API would send the CSV file to the Random Forest Classifier (RFC).

The end-user for our application would be developed by another team working parallel with Phish Phinders. This is because Phish Phinders’s project is envisioned to be a part of a larger project that will detect malicious emails. This scenario simulates a real-world possibility for our future, as developers, where we would work alongside other teams creating an application that is a subset of the larger application under development.

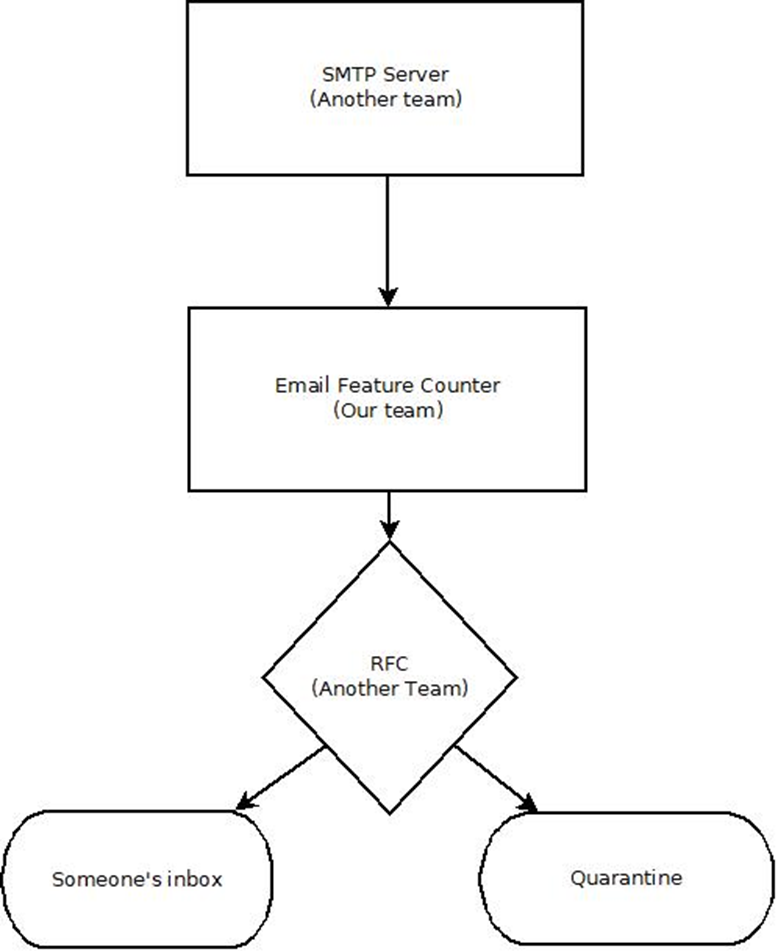
Our email parser is based upon the research of Aviad Cohen, Nir Nissim, and Yuval Elovici. It will scan for features using the novel set of features published by Cohen, Nissim, and Elovici in the article *Novel set of general descriptive features for enhanced detection of malicious emails using machine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

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

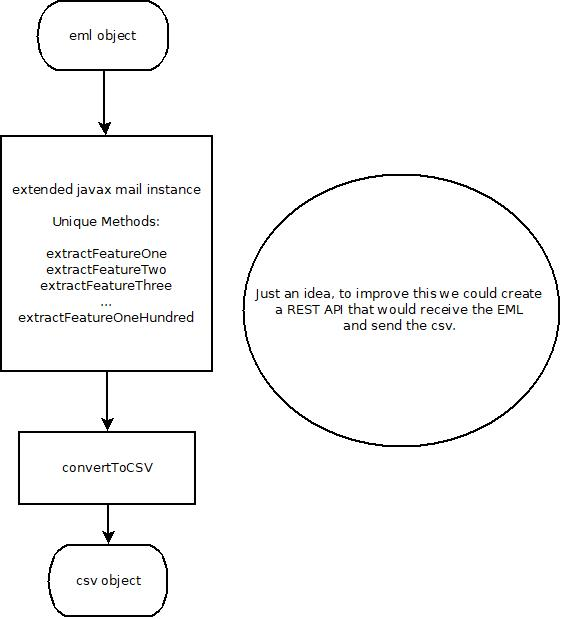
The requirements are broken into two parts; the Project Requirements, and the System Requirements. The project requirements will mainly state the goal of our project. The system requirements will go into hardware and software needs for the project.

* + 1. **Project Requirements**

The main requirement of the project is to parse emails noting the presence of, or lack thereof, features. This data is formatted into a layout and converted into a CSV file. The following figure shows the flow of an email through the program. Once the email is received, it is processed and classified using the RFC. This classification determines whether the email is delivered or quarantined.



The below figure shows the flow of the email parser:



**2.1.2 System Requirements**

The recommended hardware system requirements are as follows:

* 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. **Non-Functional Requirements**

By limiting the size of \*.eml files and anticipating peak periods, the program should perform without delays in system performance. The recommended security requirements are as follows:

* The latest versions of all software packages and libraries to avoid any vulnerabilities of know Common Vulnerabilities and Exposures or CVEs for short.
* The latest patch version of Linus, Mac, or Windows.
  1. **Scope Verification**

The deliverables will be formally accepted through electronic communications as defined 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 sign-offs on all deliverables.

* 1. **Scope Control**

The team will monitor the scope of the project through routine reviews to ensure the program is advancing towards the overall goal, as stated in the project plan. All changes to the project require approval by each team member, 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 for the team, providing timely intercommunication, and ensuring the program’s quality. The below table will detail on what communication actions the team will need to follow.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Communication | Method | Frequency | Goal | Owner | Audience |
| End of Week Report | Discord Meetings | Weekly | Review project status and discuss next week goals and deliverables | Project Manager | Project Team |
| Team Updates | Email | Daily | Discuss what each team member has done and discuss any problems they have encounter | Project Manager | Project Team |
| Product Review | Discord Meetings | At milestones | Present project deliverables to get feedback and discuss the next steps | Project Manager | Project Team |
| Group Discussions | UMGC Discussion | Weekly | Discuss what team will have to do this week and getting feedback from the project sponsor. | Project Manager | Project Team and Project Sponsor |
| Uploading files | GitHub.com | Weekly | Uploading all of the documentation so team members can download them. | Project Manager | Project Team |

* 1. **Emails**

The team will use emails as the primary source of communication. Team members will use this to post daily updates on their assignments in addition to posting questions if team members need any help. The Project Manager will oversee checking everyone’s emails and helping wherever it’s needed.

* 1. **GitHub.com**

The project will be using GitHub.com for software control management. Those features will include:

* Source code management
* Version control
  1. **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 documentation will be posted here for the team to review and provide changes.

**3.5 Discord Meetings**

Discord will be used to facilitate weekly group meetings. Discord provides audio and video capabilities, enhancing collaboration between group members. The meetings will be on Sundays at 6 pm eastern. The scheduling of additional meetings will have to be approved by everyone prior to that day.

1. **Staffing Management Plan**

The staff on this team has been assigned 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 the 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. Work breakdown structure

To help the team focused on what the task needs to be completed for the project, a work breakdown was created.A screenshot of a computer

Description automatically generated

* 1. Schedule

This is the schedule that will show when each milestone is expected to be done and how many days each part of the program will take. The initial submission date is when the milestone will need to be done so other members can view and discuss if any changes need to be made.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Milestone** | **Estimation in Days** | **Initial Submission Dates** | **Final Submission Date** | **Roles assigned to task** |
| Project Plan  GitHub account setup | 5 | October 30, 2019 | November 3, 2019 | Project Manager |
| User Guide  Test Plan  Peer Review 1 | 7 | November 7, 2019 | November 10, 2019 | Technical Writer  UX/HCL  Test Engineer |
| Project Design | 7 | November 14, 2019 | November 17, 2019 | Software Engineer  Integration Engineer |
| Phase 1 Source  Peer Review 2 | 7 | November 21, 2019 | November 24, 2019 | Project Manager  Software Engineer  Integration Engineer  Test Engineer |
| Phase 2 Source | 7 | November 29, 2019 | December 1, 2019 | Project Manager  Software Engineer  Integration Engineer  Test Engineer |
| Phase 3 Source | 7 | December 5, 2019 | December 8, 2019 | Project Manager  Software Engineer  Integration Engineer  Test Engineer |
| Final Report  Peer Review 3 | 7 | December 12, 2019 | December 15, 2019 | Project Manager  UX/HCL  Technical Writer  Software Engineer  Integration Engineer  Test Engineer |

* 1. Tools to Manage Schedule

The tools the team utilizes regularly include scheduled meetings and communications 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 behind schedule. Example remediation includes additional members helping with the task.

* 1. Schedule Responsibility

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

1. **Risk Management Plan**
   1. Schedule Delay

In the event of a schedule delay, the project team will reevaluate the goals and determine if a change to the schedule is needed or a re-design to help get the program back on track. The project manager will keep track of the schedule and will make the necessary changes to resolve any problems that would cause a schedule delay.

* 1. Team Member Resignation

If a team member resigns or leaves, the project team will re-structure the workload of the person who left to ensure the project will stay on track. The project manager will oversee assigning the new tasks to the team members.

* 1. Team Member Negligence

If a team member is negligent, the team will have a direct discussion with that person regarding the issues to see if the situation can change. If the issue does not improve, the team will notify the project sponsor, the professor, about the situation. The project team will oversee each other to make sure everyone is doing their fair share.

1. **Cost Management Plan**

The cost of this project will come from the labor of each member on this project team. Each person has their role which comes with its salary. The team members will spend approximately 14 hours a week on this project which is our labor cost.

1. **Quality Management Plan**
   1. **Quality Management Method**

The Quality Management Method will include a set of requirements that the project team must follow during the development process.

* 1. **Quality Standards**

The project will adhere to the following coding standard of Google Java coding Standard. The standards can be found by following this link: <https://google.github.io/styleguide/javaguide.html>

* 1. **Quality Management Tools**

The project team will use the static code analysis capabilities of the Netbeans IDE to inspect the source code of the project. The capabilities provided by Neatbeans include Java Hints and FindBugs. Java Hints will align the team’s source code with pre-defined coding standard rules. FindBugs will scour the team’s source code for potential inconsistencies and errors. Further, the project team will use the Junit testing framework to test the program. The team will design the code using UML upon which tests will be generated for classes and class methods. Tests will confirm the desired outputs are being returned as well as attempt to identify any side-effects in the source code.

* 1. **Quality Management Roles and Responsibilities**

The Project Manager will assume the role of QA lead for this project. The Project Manager, working with the Test Engineer, will provide verification of the software process and products being developed and tested for this project. The Project Manager will be involved in reviewing and verifying processes and procedures. In addition, the Project Manager will be witnessing and monitoring the execution of test plans. The Project Manager, with the help of the Test Engineer, will ensure any problem encountered is properly reported, documented, and resolved. Team members are responsible for funneling concerns to the Project Manager.

* 1. **Quality Control**

The quality control will occur through the review of all documentation and deliverables for adherence to Google Java Coding Standards. This will be the Project Manager’s job to validate all deliverables meet such standards.

* 1. **Quality Assurance**

Assurance of product quality is the result of spot-checks of code from all members of the team lead by the Product Manager, with the assistance of the Test Engineer. This will also cover the review of documentation. The Test Engineer is responsible for identifying potential issues, but responsibility for assuming risk or imposing a change in the event of a disagreement resides with the Project Manager.

1. **Change Management Plan**
   1. **Problem Reporting and Closure**

When a problem is found, team members will be notified by email as part of the daily update email. The issue will contain at least: a description and a proposed solution. Additionally, the email needs to contain the recommended information that includes the information required to recreate the issue. The closure of the issue will occur with the approval of the Test Engineer, Developers, and Project Manager.

* 1. **Action Item and Issue Resolution**

The team will be notified through email of an issue. The team will then pose a solution in the same email. The team will approve the suggested solution. A project member will implement the solution and test it. The Test Engineer will verify the solution.

1. **References**

Cohen, A., Nissim, N., & Elovici, Y. (2018). Novel set of general descriptive features for enhanced detection of malicious emails using machine learning methods. *Expert Systems with Applications*, *110*, 143–169. doi: 10.1016/j.eswa.2018.05.031