Content

## Amendment History----------------------------------------------------------3

## Purpose of the Project-------------------------------------------------------3

## Background-------------------------------------------------------3

## Aim-----------------------------------------------------------------3

## Objectives-----------------------------------------------------------------3

## Viewpoints----------------------------------------------------------------------3

## Indirect-----------------------------------------------------------------------4

## Interactor--------------------------------------------------------------------4

## Domain-----------------------------------------------------------------------4

## Mandated Constraints--------------------------------------------------------5

## Solution Constraints-------------------------------------------------------5

## Implementation Environment-------------------------------------------5

## Partner Applications-------------------------------------------------------5

## Commercial off the Shelf Packages------------------------------------5

## Anticipated Workplace Environment----------------------------------5

## Timescale---------------------------------------------------------------------5

## Financial Budget------------------------------------------------------------5

## Facts and Assumptions-------------------------------------------------------6

## External Factors-------------------------------------------------------------6

## Assumptions-----------------------------------------------------------------6

## Scope of the Work-------------------------------------------------------------6

## Scope of the Project-------------------------------------------------------6

## Scope of the Product------------------------------------------------------6

## Requirements-------------------------------------------------------------------7

## New Problems-----------------------------------------------------------------11

## Tasks-----------------------------------------------------------------------------12

## Steps to Deliver the System---------------------------------------------12

## Development Process----------------------------------------------------12

## Development Phases-----------------------------------------------------12

## Risk Assessment--------------------------------------------------------------12

## Cost Assessment-------------------------------------------------------------13

## User Documentation--------------------------------------------------------13

## References---------------------------------------------------------------------14

1. Amendment History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Description** |
| 1 | Neil Rafferty | 14/11/2016 | Original Version |

2. Purpose of the Product

**2a. Background:**

Email messaging connects over 2.5 billion people world-wide *(Radicati Group Inc., 2015)* and still continues to be the leading method of business communication today *(Radicati Group Inc., 2016)*. However, after continual use an email address will eventually fall into unwanted hands. Third parties who obtain our email addresses can spam us with marketing emails, they can spam us with dangerous phishing and malware attacks, they can build online profiles about us (sharing and selling our information) and they may even have the ability to hack into our accounts.

This project will attempt to address the issue through the development of a new type of email service. The concept behind the service is that it will manage a large amount of email addresses for each user, one per contact. This way the user will have the ability to delete unwanted contacts permanently and identify where third party spammers retrieved their contact information. In addition to this, the service has the potential to increase user privacy and security online. This is because the user would disclose a private username for authentication rather than a public email address and only the domain (which would potentially be utilised by multiple people) would link a users’ varied addresses.

**2b. Aim:**

To create an innovative, reliable, secure and user-friendly client/server email service that gives users more control over their contact information than conventional email. This through the use of managing multiple email addresses.

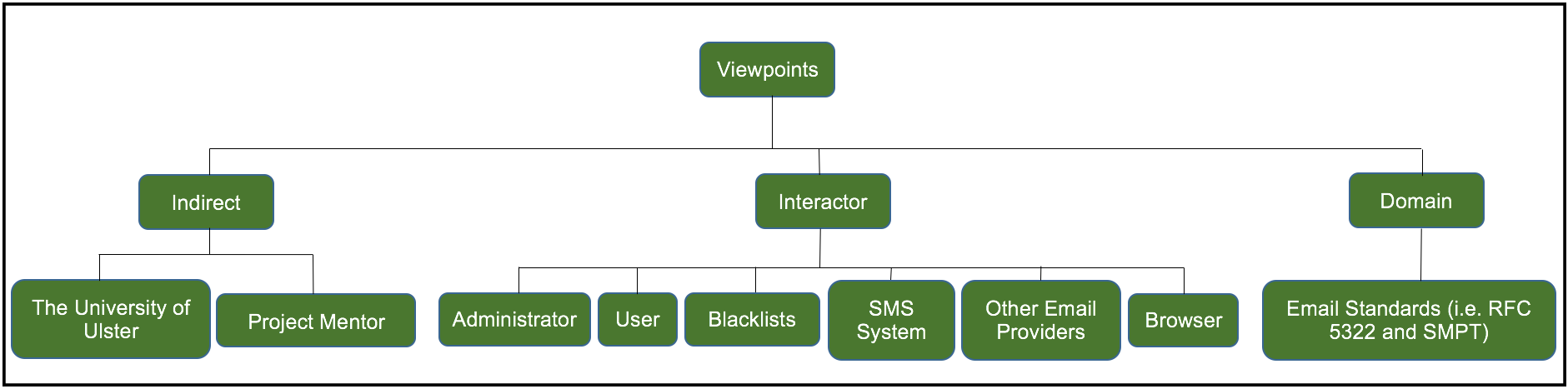
**2c. Objectives:**

* Create a basic client/server email service.
* Alter the service to allow users to manage multiple addresses.
* Add functionality to quickly generate email addresses and link them to a given contact.
* Allow users to report contacts on a public board for a range of offences.
* Add more common email features.

3. Viewpoints

The following diagram (figure 1) outlines the relative stakeholders, systems and standards for this project.

*Figure 1: Viewpoint Hierarchy Diagram.*

****

**3a. Indirect:**

The University of Ulster – As this us a university project, the university may have some influence over the direction and outcome.

Peer Support Group - Will offer support and guidance.

Project Mentor – The project mentor, Dr. Ian McChesney, will continue to offer guidance throughout the duration of the project.

**3b. Interactor:**

1. Administrator – This person will be responsible for managing User and Super User accounts.
2. Super User - This person will be responsible for managing User accounts.
3. User – A person who uses the application for communicative purpose.
4. Blacklists – Online lists of malicious emailers.
5. SMS System – This may be used for account recovery.
6. Other Email Providers – Other email providers that the system will communicate with.
7. Browser – Some browser extension functionality may be added for a better user experience.

**3c. Domain:**

1. Email Standards – Basic email standard need to be adhered too.

4. Mandate Constraints

**4a. Solution Constraints:**

1. The product must be a web application.
2. The application must management multiple email addresses.

**4b. Implementation Environment:**

1. The product will be expected to work on multiple platforms (including desktop and mobile devices)
2. The application will be hosted online.

**4c. Partner Applications:**

1. This application will send and receive correspondents to other email providers (as mentioned in 3b.3).
2. An anti-virus will be needed to scan emails for threats.

**4d. Commercial off the Shelf Packages:**

Open source resources and services should be investigated during sprint planning to reduce the project workload. This should be investigated for a:

Spam Filter.

Email Server.

Email Client.

Appointment System.

**4e. Anticipated Workplace Environment:**

1. The majority of development will be done from home on a mac.
2. Meetings will be held with the project mentor and peer support group on the university grounds.
3. Relevant academic materials will be studied in the library and online.

**4f. Deadlines:**

1. There will be a project presentation on the 12th of December 2017.
2. There will be a project showcase in week 5 of semester 2.
3. The absolute deadline for this project is in week 11 of semester 2 which will be in April 2017.

**4g. Finance:**

There is no budget for this project. However, if a cost does arise it may be paid for by the student or a request for funds can be made to the university.

5. Naming Conventions and Definitions

**5a Point of Contact:**

In relation to this project, a point of contact is a line of communication that is created when an email address is disclosed to another party.

**5b Maximum Email Size:**

This value is initially being set to a 10Mb as that is a competitive size. However, stable this could be improved if justified by testing.

6. Facts and Assumptions

**6a. External Factors:**

*no relevant external factors were found.*

**6b. Assumptions:**

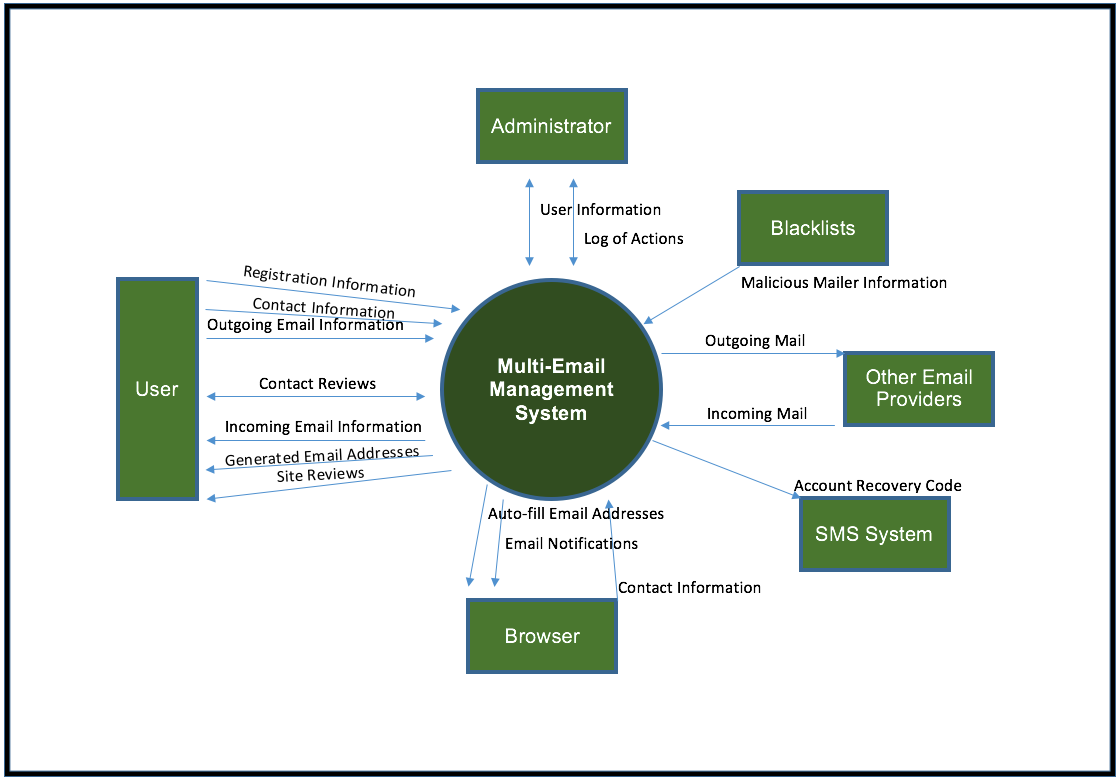
1. Email communication will remain to be a popular form of communication.
2. Users interest in reducing spam and protecting their points of contact.

7. Scope of the System

**7a. Context of the System:**

Figure 2 had been used to illustrate the flow of information between the system and its interactors from **3b.**

*Figure 2: System Context Diagram.*

****

**super user**

**7b. System Components:**

1. Email Server – The email server will send, receive and store user messages.
2. Email Client – The email client will retrieve emails from the server and present them to the user. It will send notifications to the user and provide functionality for the user to send messages also (through use of the email server).
3. Website – The website will host the email client and possibly a calendar too. It will provide register and log in capabilities as well as a download like for the Browser Extension.
4. Browser Extension – May be used to auto fill email addresses, store contact details and warn of reported websites.
5. Calendar – A calendar could be added to help users manage appointments.

8. High-Level Requirements

The following requirements have been gathered by using common knowledge of email services, referring to a previous email requirement documentation (Office of Information Technology, 2007) and through brainstorming sessions. The requirements are separated into four categories. The first are the minimum viable product requirements. These requirements are the highest priority requirements because if they are not delivered they will jeopardise the success of this project. The delight requirements, in this case, are all features that revolve around the concept of managing multiple email addresses per user. The additional requirements are common emailing features that are not necessary to implement a MVP. Because they are neither necessary for creating a MVP nor do they revolve around the main concept of the project they will be designated with a lower priority. And finally the non-functional requirements set an extremely low benchmark to ensure that the system operates reliably during the demonstration. Unfortunately, due to the nature of using multiple email addresses using third party email clients will not be possible.

**MVP Requirements:** **Administrator**

* A user must be able to register on the web site.
* There must be a secure log in capability.
* The application must have a change and recover password capability.
* Users must be able to view received email messages online.
* Users must be able to send emails to any valid email address (this must be from the relative linked email address).
* There must be a contact directory.
* There must be an email generating tool that stores recipient’s details (with the linked address) in the contacts directory.
* Users must be able to delete an email address from their collection.
* A user must be able to create multiple email addresses linked to their account.

**Delight Requirements:**

* Users should be able to flag contacts that have committed range a of offences.
* Smartphone apps could be created to copy a newly generated email address into a users’ clipboard.
* A browser extension could be added to automatically enter a new email address into online forms.
* The browser extension could also notify users when they are supplying details to a reported contact.
* A user should be able to unfollow or hide a contact without deleting the linking email address or messages.
* There could be an option to only accept emails from specific addresses or domains within each user address.

**Additional Requirements:**

* The user should be notified when they receive an email.
* A spam filter could be applied to remove spam from the inbox.
* Emails could be scanned for virus and malware.
* An email search tool could enable users to search through email content.
* Auto respond capability/forward capability.
* The user should be able to send emails to multiple users at once.
* Attachments could be sent using directory selection and/or drag-and-drop capability.
* Attachments could be previewed on the site.
* Two factor authentication could be implemented.
* The web-client should display if the mail has been read or replied too.
* An administrator should be able to ban or delete accounts.
* Administrator actions should be recorded.
* Users should be able to save/delete/send drafts.
* Users should be able to delete emails.
* Contact aliases could be used.
* A calendar/scheduling system could be implemented.

**Non-functional Requirements:**

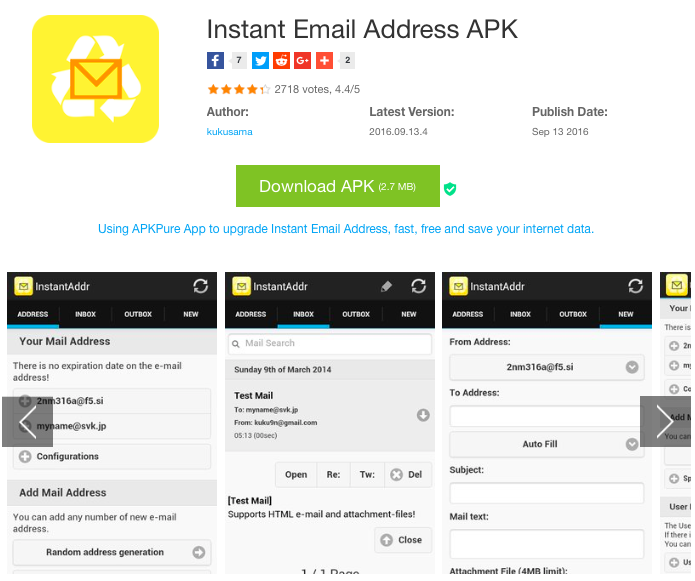
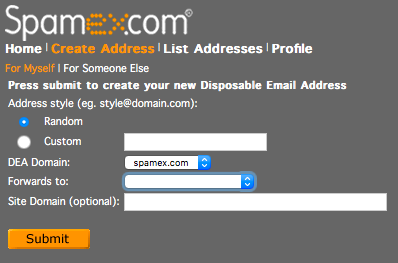
* Sent emails should be received within 5 minutes.
* Each user account should store at least 500MB of email information with a set upper limit.
* The email service should support a minimum of 10 users with 10 email addresses each.
* Data must be backed up.
* Server should be hosted on a cloud environment.
* Timeout disconnects of five minutes could be set for idle clients.

**Competition .**

There are many disposable email services online but few of them are tailored towards continued use. Here is an outline of two competing products that were found.

**Instant Email Address** *(Figure 1)* is an Android app that allows the management of quickly generated emails. Being an android app as opposed to a web-app means that it will only work on the android platform. However, this means that Instant Email Address can make use of push-notifications.

**Spamex** *(Figure 2)* is hosted on a website and therefore can be accessed on multiple platforms. The website will forward emails received by a users’ spamex email addresses to the user’s external email address. It will also forward user replies to the original correspondent. However, this means that the email management tool is separate from other email functionality making it inefficient for users.

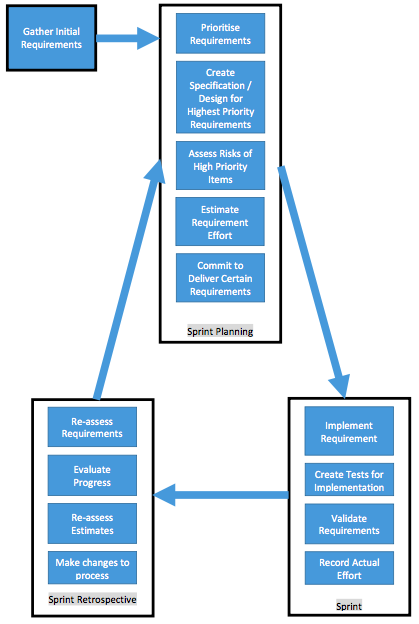
**** 

*Figure 1: Instant Email Address on the Play Store. Figure 2: Spamex User Interface.*

This project is different from the competition as it forces the linking of contacts with a unique email address supplied to only them. Doing this will afford the project the potential to provide some interesting features. As a side note, spam filtering and features like Outlooks’ Clutter and inbox ruling could be considered competition for this project. However, there is no reason why similar features could not be incorporated into a future release of this product, in turn, making it more effective.

**Project Life-Cycle .**

Due to the current lack of technical knowledge with regards to email hosting this project carries a certain level of risk. To reduce this risk and help ensure that a working MVP is produced for the projects demonstration, an incremental software process will be used during development. This will reduce time on spent planning features that may not be implemented within the projects initial time-scale. The spiral methodology was considered and it was therefore decided to prioritise the most risky and essential features. However, a custom life-cycle has been chosen to develop this project *(Figure 3)*. It is similar to scrum, however, it has been created for individual project development.

****

*Figure 3: Project Life-Cycle.*

**Project Plan .**

This project is going to consist of three or more sprints. Each sprint will last four weeks and will consist of no less than 30 hours of work. The layout of which is shown in figures 3 and 4.

**Prioritising Requirements:**

Items of work needed to create the MVP will be completed first. Then surveys will be used to gather user delight and annoyance factors towards a feature being delivered and not being delivered respectively. A calculation will then use this information along with the developer effort estimates to assess the priority of the remaining features. This calculation will place more value on the delight and effort factors than the annoyance factor.

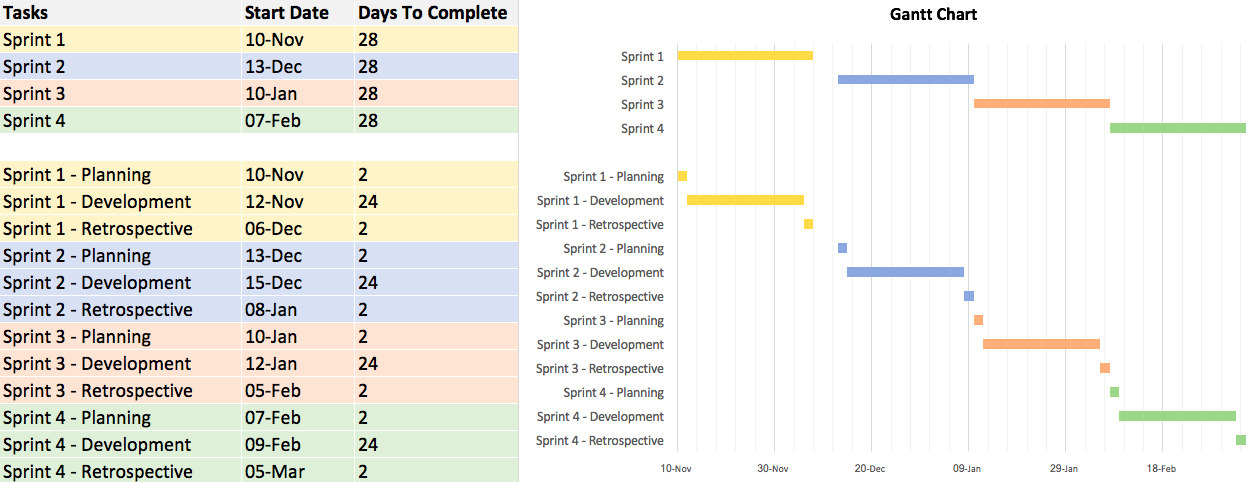
**Estimates:**

Estimates will be made using story points in a Fibonacci sequence in the sprint planning phase. During development estimates may be changed by the developer but a reason will need to be recorded reflecting why. Actual time will be recorded during development and the average amount of time per story point will be recorded at the end of each sprint. This should help the developer decide how much to commit to during the subsequent sprint.

**Work Item:**

A work item will be created for each requirement being brought into the sprint and backlog. Each work item will outline a more detailed description of the requirement, testing criteria and relevant system and user interface designs. The item will be moved through a Kanban board until it is designated as done. The work item may be altered by the developer at any time as long as the original is recorded with a reason.

At the end of the sprint the developer will make a list of things that went well, went poorly and thinks that could be changed. This will be discussed with the project mentor who in turn will help decide if a change will be implemented or not. The developer may add new requirements to the project. However, these new requirements will need to be of high priority to be brought into the sprint.

****

*Figure 4: Gantt Chart*

**References .**

Office of Information Technology. (2007)

*Undergraduate Email System Requirements*

Available: <http://www.oit.ucsb.edu/committees/CNC-EMail/requirements.asp>

Last accessed The 22nd of October 2016

PSDgraphics, (2013)

*male silhouette (Cover page image)*

Available: <http://www.psdgraphics.com/file/male-silhouette.jpg>

Last accessed The 25th of October 2016

Radicati Group, Inc. (2015)

*Email Statistics Report, 2015-2019*

Page: 3

Available: <http://www.radicati.com/wp/wp-content/uploads/2015/02/Email-Statistics-Report-2015-2019-Executive-Summary.pdf>

Last accessed 29th Sep 2016

Radicati Group, Inc, (2016)

*Email Statistics Report, 2016-2020*

Page: 2

Available: <http://www.radicati.com/wp/wp-content/uploads/2016/01/Email_Statistics_Report_2016-2020_Executive_Summary.pdf>

Last accessed 25th Oct 2016