**5 The Final Solution**

In this chapter, a final design of a mass communication system named Myriad will be introduced. To begin with, improvements on the requirements will be compared to those defined for the prototype from the previous chapter. Later on, the actual product features and its architecture will be discussed.

# 5.1 The Improved Requirements

Seeing the prototype in action made us realize the need to revise the requirements and bring new ones. Some of those requirements are shaped according to the feedback we got during our research at the Stanford HCI group, as well as from the organizations who conduct mass email communication on weekly basis in reaching their community. The following section will discuss those new features for the final product.

## 5.1.1 Assistant Support

As described in section 4.1.1, the standard we want to achieve within a mass email communication is the most adequately personalized emails for every individual recipient, with a minimum effort on the researcher’s side. The initial idea, as well as the prototype, included several features for this purpose, as discussed in chapter 4. However, considering the gold standard we would like to achieve as shown in the figure 4.2, the prototype manages to still leave an amount of effort on the researcher’s side in accomplishing a successful mass email campaign.

In order to reduce the effort as much as possible during a mass email communication, the addition of an assistants’ involvement was considered. Therefore, a primary researcher will be able to share tasks with permitted assistants. These might be tasks such as extracting information from the incoming answers, proofreading the primary researcher’s

* 1. *The Improved Requirements*

replies before sending them, or even writing replies to those answers. Therefore, the primary researcher will only need to interact with the flow of a mass email campaign when the situation calls for it. However, the system would still need to provide necessary features that will be seen in the next sections in support for the work flow in a mass email campaign, hence, assistants will only need to interact with this work flow in letting the email campaign carry on by providing answers with using the provided email templates and extract information as KVPs.

## Dynamic Variables and KVPs

We introduced the use of dynamic variables at the initial prototype, however, it was only limited to the salutation of the email. As described in section 3.3.3, email marketing applications support this feature. And because it makes the personalization of emails easier, the final product will also include this feature. As a result, application users can create KVPs and use the said keys in the content of an email message to be replaced dynamically by its value, according to the recipient. Therefore, the extracted information from emails will not just help us gather information in an organized way, but also in personalizing the emails.

As a result, instead of storing the KVPs in the system according to their respective responses, the system should store them according to the recipients itself. This is where the KVP-idea differs from the prototype. With this, we now have profiles of contacts for a campaign having all KVPs of a recipient visible during the whole state of the conversation. However, the system should offer an option to hide individual KVPs to avoid cluttering the view, and make the actively used KVP list the default view.

Importing KVPs can be done in several ways. One option is that the system should be able to synchronize with an online spreadsheet, e.g. Google Spreadsheet, in obtaining the KVPs at the beginning of a campaign. This is a convenient way for researchers, since they are already familiar with spreadsheet environments. Other options should include a campaign-wide view and a contact specific view in the system. Also, it should have an editable keys and values in a campaign.

## Importing and Exporting Contacts and Their Information

In the prototype, the application user has to enter all the basic information of the recipients such as first name, last name, and email address into the system manually. However, we realized that as they perform this, they use a spreadsheet and copy the contact information from there. This was also done for their regular email client used for email campaigns. The applications that were reviewed in section 3.3 all had an option to import data from spreadsheets to ease the process. Therefore, the system should also offer an option similar to this in importing contact information.

However, importing should not only be limited to the basic information of the recipients. Since we already mentioned about importing KVPs from a spreadsheet in the previous section, the system should also be able to detect and import contact information and KVPs related to a specific contact, granting that they are available in the provided spreadsheet.

The system should provide a bi-directional synchronization, and not just to import data support from the spreadsheet. Therefore, the system should provide an option to export contact information and their created KVPs from the system to the spreadsheet as well. This gives a reporting functionality to the application users, where they can see all of their contacts, and the extracted information from a specific campaign, all in one view.

## Interoperability with Other Email Clients

Even though we provided a new system for the users in initiating their mass email communication, there might be other cases wherein a mass email communication was initiated using a regular email client, therefore the system is not aware such campaign, since it was not created with it. The application should be able to provide an option in importing email messages created with a different email client into the system by recognizing specific messages, as annotated by the user.

Allowing the system to import email conversations from other email clients reduces the need for dependency on the application, and while researchers continue on using their own email

clients, assigned assistants can take care of those emails, as imported by the researchers. We saw the same import feature in CRM applications present in section 3.3.1 as well, where a user can easily forward an email to the CRM application’s provided unique email addresses, and the system then takes care of assigning the imported emails to the corresponding recipients in the system.

## Automated Decision-Making and Notifications

Even though the involvement of assistants helps make the primary researcher’s life easier, the system still needs to provide an automated approach in answering emails whose statuses are clear in terms of its mass email communication flow. Therefore, a rule based decision-making mechanism should be used, at which a user sets the value of the keys from KVPs, and it triggers the action of sending emails to qualified respondents whose KVPs satisfies the provided condition.

Since the only purpose of the application is managing mass email communication and each campaign results to a great amount of messages in the inbox, the system should be able to provide notifications regarding what should be done next for each recipient. Labels should be added to email conversations to indicate whose turn is the next in the communication. For an instance, by proving a label saying "You need to reply" to the application user, the state of the communication awaits for an ample action from a researcher, or an assistant. In the same way, unread conversations and conversations waiting for an answer from the recipients should also be annotated in the appropriately. The system should also provide email notifications to the assistants’ email addresses, to notify them whenever there is an action waiting to be taken care of.

# 5.2 Final System

In this section, we will see how the revised and improved requirements from the previous section reflected on the final product, named "Myriad".

## Log-In and Campaigns Overview

Similar to the prototype featured in section 4.2.2, Myriad also requires a Gmail account to work with. The reason behind this is not just the popularity of Gmail, but also given the fact that Stanford University uses Google Apps by default, university wide. Therefore, each member of the university has a Google account ready to use with Myriad. This also provides flexibility, because Myriad does not have a registration form or a sign in screen. All the requested Google’s permissions from a Myriad’s user and their descriptions can be found in appendix C.

After users sign in to the system, the first screen they will see is the campaign overview screen. All created campaigns, including the ones assigned by other user’s as an assistant are shown as in figure 5.1. It has a simple and clean UI, compared to regular email clients, emphasizing its focus on mass email communication.

## Synchronization with Other Systems

Myriad has the ability to get the recipients’ information and their KVPs from Google Spreadsheet1. This is a convenient way in importing recipients’ information into the system, since many people already keep their recipients’ and related information in a spreadsheet environment as discussed in section 5.1.3. Therefore, Myriad offers a bi-directional sync-

1https://docs.google.com/spreadsheet/

ing from and to a spreadsheet, as defined at beginning of the creation of a campaign. Besides this, Myriad also has an option to enter a recipients first name, last name, and email address into the system directly.

The corresponding columns in a Google Spreadsheet start with "first name", "last name", and "email address" as shown in figure 5.2. The rest of the columns will be filled out as a KVP, and imported into the system as well.

As mentioned in section 5.1.4, importing existing email conversations as a campaign message into the system is an important feature. Researchers might have initiated conversations using their regular email client, and if later on they decide to make them more manageable, they can import them to Myriad. Myriad takes advantage of Gmail’s labeling feature, which is equivalent to the IMAP protocol’s folders for the same purpose (Google Inc., 2013b). Myriad creates a Gmail label in the user’s account, and the only thing a user needs to do is the enable the label syncing feature at the campaign creation screen. Next, there will be a label similar to the campaign’s name, and grouped under the root label, "myriad" in the Gmail’s inbox (see the figure 5.3). Hence, researchers will able to import email messages not considered a part of a campaign for some reason, into Myriad.

## Creating an Email Campaign

The campaign creation screen (see figure 5.4) has input fields for a campaign name and Google Spreadsheet’s Uniform Resource Locator (URL) to synchronize with. Other two checkboxes for the spreadsheet are used to manage the frequency of synchronization and disabling the cascading warning messages in case of an erroneous data in the spreadsheet such as an empty email address field for a contact. The option for importing emails from Gmail’s inbox is also in this screen, labeled as “synching section”.

After the campaign is created, all the contacts and their KVPs will be imported as long as the user provided a Google Spreadsheet URL like in the figure 5.5.

## Composing an Email Message

Users can send emails to all of the contacts they entered or imported into the system, or select a subset of them by using the filtering function as shown in figure 5.6. Provided filtering options are according to the values of KVPs, the conversation statuses such as unread, replied, or unreplied, or the message’s template names.

By pressing the corresponding button, users can now readily compose emails for their filtered recipients at the compose screen. The compose email pane (see figure 5.7) contains a section that lists previously sent emails available for reuse. This is the same template concept that was introduced for the prototype in Chapter 4, Section 4.2.2. It also shows the visualization of the state of the communication by using a tree structure, as well as the number of messages sent by using the corresponding template. A more long term campaign’s visualization tree can be found in appendix D. The system also suggests an email template while composing a reply by considering the nodes at the same tree structure.

The compose pane allows users to add dynamic variables in the content of messages to personalize them according to the recipients. These variables are not just limited to the first name and last name, as they can also be of any keys from the assigned KVPs in the campaign.

Each campaign has a link to other campaigns, as long as the recipient is involved in both campaigns, as featured in the next section. A researcher can access conversations and KVPs from other campaigns by switching to the campaign via the provided link. Providing an option to see the earlier campaigns a recipient got involved in gives broader knowledge about a recipient that may help researchers in personalizing the content of the emails more easily and properly. For example, if we have extracted information regarding which sports did the recipients got involved with in an earlier campaign. We can use the same information in making a friendlier and to frankly start a new campaign by mentioning details about the latest events of those sports areas in the country. Such technique also supports social exchange and the diffusion of responsibility theories as discussed in section 2.3. There is also an option to hide the KVPs in a campaign, if they are not related at all or have become obsolete during the flow of a campaign to avoid cluttering the application’s view.

In figure 5.7, at the pane of "after-sending actions", users are able to set the values of the keys right after an email is sent. Therefore, a user does not need to browse to another screen in order to change or set KVPs whose values depends on the email that was recently sent.

At the beginning of this section it was mentioned that a user can filter the recipients list, and then send an email to a subset, according to the filter’s criteria. Myriad saves those filtered criteria under the "Rules" menu as shown in figure 5.8 after the user sent the email. Therefore, the next time the system receives new emails with the same conditions as before, user will see them under the rules menu, and the earlier sent email messages can be sent to those new matching recipients automatically, if user opt to enable the automated response feature or a user can simply press the send button manually in the same screen. For example, in figure 5.8, there are 3 recipients whose "attending" key were set to "yes", and since the system already recognized that we have already sent an email to those attending, new matches for the same condition applies. In this case, the user can simply press the send button, or automate the process by enabling the provided option.

## Extracting Information from Email Messages

Myriad’s reading pane (see figure 5.9) offers a threaded view wherein all of the messages between a specific recipient and a researcher are visually grouped together. The advantage of a threaded view is that it allows a researcher to get a quick overview of the whole state of a conversation, therefore a researcher can write a customized message more easily by focusing on the specific personality of the individual being responded to, considering earlier conversations with him or her at a single glance.

Each of the researcher’s messages is annotated with the name of the message template that was used with, making the latest state of the communication easily recognizable without having to look for its content. In the figure 5.9, emails sent by the user have green labels with indicators of the name of the template used, such as "Accepted" and "Invitation".

While reading a recipient’s answer, the extracted information can be recorded as KVPs at the right-hand side of the reading pane (see figure 5.9). Earlier recorded keys’ values can also be updated at the same pane. Having KVPs along with the conversation thread gives the researcher necessary information about the person being replied to. Under the KVPs pane, the researcher can also see if the recipient is involved with another

campaign, and a link to the involved campaigns are provided with the number of emails exchanged, next to it. This is a helpful feature in reminding researchers about the existence of previous interactions with the recipient, and if necessary, the researcher can switch to other campaigns to get an overview of extracted information as KVPs.

## Enabling Assistants

A researcher can add other researchers as assistants into a campaign by adding their Google account associated email addresses into the Myriad system, as shown in figure 5.10. The task of the assistant can range from extracting information from emails, writing answers to the recipients, to proofreading a researcher’s emails before sending them, and many other tasks that can be generated according to the type of help a researcher needs.

After an assistant is assigned to a campaign, he or she will get a notification email including the link to the campaign. Again, there will be a notification email for each email sent by a respondent to the assistants’ email address to let them know that they need to act on the situation.

Myriad provides status labels for each received or sent email, giving a hint on the next awaiting action similar to what is shown in figure 5.11. These status labels give hints about the next action to be taken according to the state of the conversation, such as if the user needs to read or to send a reply to a message, or the conversation awaits for an answer from the recipient’s side to continue to the communication. There are also status labels related with Myriad’s internal state regarding an email message, such as if Myriad was able to send the messages successfully, or if there was a failure encountered while sending them. The same view also provides a column showing the last message sent by the user. With this, the researchers and the assistants will easily realize what should be done next, and see the status of the communication for each recipient.

# Architecture

Similar to the prototype, Myriad was also developed using a framework stack of RoR, jQuery, Bootstrap, and OAuth2 protocol was used for the authentication with Gmail. The initial project structure was generated by a gem named Rails Apps Composer3, consisting a collection of Rails application templates to start a project. Thanks to the other gems utilized for the project development and completing it on time in a robust way. The following section will discuss the implementation details of Myriad, how it makes mass email communication easier.

## Keeping Track of Recipients

As mentioned in section 4.2, the prototype wasn’t able to keep important information regarding the recipients, only the messages sent. Therefore, extracted information such as KVPs are only limited to email messages related to the recipients. The downside of this approach is its inability in obtaining all the KVPs of the recipients all at one glance. Instead, we need to go through earlier emails of the recipients each time we need to create the same key, since the system was not aware of an existing KVP created for a different email in the same campaign.

2OAuth is leveraged by using OmniAuth library (https://github.com/intridea/omniauth), and its Google authentication strategy is implementation by OmniAuth Google OAuth2. Details can be found at https://github.com/zquestz/omniauth-google-oauth2

3https://github.com/RailsApps/rails\_apps\_composer

In Myriad’s case, we overcame this issue by keeping all the recipients’ information in a separate data model named “contact” as shown in figure 5.12. With this, we were able to keep track of the recipients among different campaigns through different conversations involving them, which is another set of a data model in the Myriad system, as each campaign involves many conversations with the recipient involved with them.

Myriad keeps all of the basic profile information of the recipients, including their first names, last names, and email addresses. These information can be imported into Myriad either by spreadsheet synchronization, by adding them into "To" filed when a user composes an email, or by importing from Gmail’s inbox via the label synchronization. The recipients’ email addresses are unique identifiers for Myriad in order to compare already existing recipient information with the synchronized ones.

## The KVPs and the Templates

Having a separate contact data model in keeping track of the recipients also helped us relate extracted information, specifically KVPs of the recipients, instead of their message in a campaign as shown in figure 5.12. Each campaign has many keys, and as well as their corresponding values, according to the recipients. One advantage of this is that as we browse a specific conversation with a recipient, we’ll be able to see all KVPs of the specific recipient under one view, which is the KVP pane next to the email threads, as shown in figure 5.7. The other benefit is its ability to synchronize with spreadsheets. When a new key is added to a recipient or a value of a key is updated via Myriad, it will be reflected on the spreadsheet as well. Therefore, the spreadsheet will always contain the latest changes done by the user in Myriad.

Similar to the prototype, templates contain emails messages with dynamic variables in it, and the actual emails whose dynamic variables are filled with values are kept in the Email data model as shown in figure 5.12, after they are fetched from Gmail.

To get an overview of the conversation and let the users select templates, Myriad offers a tree structure similar to the prototype. However, the execution of the tree structure is different from the prototype. In section 4.2.3, it was mentioned that the hierarchy between the nodes of the tree was a nested set model. We decided to use an adjacency list model instead of a nested set model to increase the readability of the templates’ relationship with the data model level from the developers’ perspective. Since the relation of the templates is not as completed compared to other relations, the performance drawbacks on this decision was negligible.

## A Campaign Message Identification

Due to the prototype’s dependency on the existing EmailValet’s email fetching process, we manually identify emails belonging to a specific campaign, and then inserting them into the same inbox in EmailValet. With Myriad, we removed this dependency, and considered showing emails belonging to a campaign.

The identification of the emails depends on the three different conditions:

* + - 1. The emails fetched from Gmail and responses to one of to the campaign message.
      2. The emails composed in Myriad as a part of the campaign, whether they are the emails to initiate a campaign or replies to recipients’ emails.
      3. The emails imported into Myriad from Gmail’s inbox via label synchronization (as shown in section \_\_for label synchronization)

Myriad initially retrieves all the email’s Unique Identifier (UID)s4 from Gmail’s inbox for the last 14 days, instead of all the emails in the inbox to reduce the time needed for the fetching process. For those UIDs not stored in the Email data model of Myriad, Myriad retrieves the actual email along with the metadata, such as Gmail thread ID5 and Gmail labels.

For the prototype, we have set a specific message-ID to the emails composed in Myriad. With this, we are able to identify whether an email from Gmail was composed to initiate a campaign or is a reply to one of the recipients of a campaign by a Myriad user during the aforementioned fetching process. Message-ID field is again leveraged for identifying a recipients’ message belonging to a campaign. Myriad retrieves the Message-ID written in the In-Reply-To, and checks if it matches one of the campaign messages in Myriad. Another field that Myriad checks is the "References" field6 to identify messages as a part of a campaign, but was forwarded to another person by a recipient in order to reply the message. Since the forwarded endpoint’s email client sets the Message-ID of the forwarded message into the In-Reply-To, a situation Myriad isn’t familiar with, Myriad checks the "References" field to identify a Message-

4UID (Unique Identifier) is 32-bit integer value to uniquely identify emails in a mailbox in IMAP. Each email added into mailbox will have a higher value than the ones added before, however they are not necessarily contiguous (Crispin, 2003). UID is used as it was in the prototype by the implementation of the EmailValet’s part. It is used to identify if a message is already fetched from Gmail, if it is not, it is

fetched since it is a new message.

5Gmail thread ID is a 64-bit integer that is part of the provided Gmail’s IMAP extension. It helps to associate

groups of messages in the same manner as in the Gmail web interface (Google Inc., 2013a).

6Reference field in another identification field along with Message-ID defined by RFC 5322. It contains one or more Message-IDs used when creating a reply to a message to identify the original message or the other messages when a reply to a message that was itself a reply (Resnick, 2008).

ID belonging to a campaign in this case. This scenario is illustrated in figure 5.13.

If an email is not identified as a part of a campaign message by comparing the Message-IDs with the ones in Myriad’s Email data model, the next step is to evaluate those emails in case they are labelled as part of a Myriad campaign in Gmail’s inbox. In this case, if an email has a label in the pattern of "myriad/campaign-name", and there is already a created campaign in Myriad with the name "Campaign Name", that email will be stored in the data model. Finally, an identified email will be assigned to a contact by looking at its parent email’s Message-ID if it was a reply or if it fails, the contact information will be recorded from the email’s "From" header field. The rest of the emails will be ignored if no matches were found in the mentioned email header fields, specifically Message-ID, In-Reply-To, and References, or Gmail’s Label.

# Experiences

To date, Myriad has 67 users since its beta launch7 last January 30, 2013. Among those 67 users, 30 of them are actively using it, since last month. In this section, statistics will be given about the created campaigns and users’ profile, and continued with some user’s

7Beta version of Myriad was launced at <http://gv.stanford.edu/>and still accessebile from this address.

testimonials.

## Statistics

The beta launch of Myriad has been advertised at the Stanford HCI group and to some of Stanford’s organizations who performs mass email communication regularly with their communities. Some of those campaigns opted to use the following features of Myriad, such as:

* **Invitation to an Event:** User created an initial invitation template and another template for those who rejected the invitation to motivate them for other upcoming events. Contact list is imported via spreadsheet. User also created KVPs to easily identify recipients needing an individual follow up. Salutations of the emails are personalized by their first name through KVPs.
* **Survey via an External Link:** Users send an initial template including a link to an external survey. A detailed list of contact information was imported from a Google spreadsheet. Later on, users continued with a reminder email in a separate campaign.
* **Importing via Gmail Labels** One of the user imported campaign emails from his Gmail account to Myriad via a label synchronization, and continued the campaign in Myriad.
* **Interview Results** One user reached nearly 500 recipients through Myriad in releasing the result of an interview process. User created a rejection template for those who were rejected, and for the ones who were accepted, he created several templates depending on the invitation date, including the specific schedule related to that date. The user also automated the email sending process by the created rules on the filtered search results.
* **Getting User Details via Form** During one of the live demonstrations of Myriad at the Stanford HCI group, we asked to the attendees to browse to a link and fill up a Google Form8 to obtain their contact information in a Google Spreadsheet, where the gathered data is recorded in Google Form by default. Therefore, we were able to easily import contact information into Myriad, and initiate a campaign. There were also 4 assistants extracting information during the live demonstration while the recipients

8https://docs.google.com/forms

were sending answers.

After reviewing some cases using Myriad in real time, the following paragraphs will provide usage statistics regarding Myriad, starting from its beta launch last January 20, 2013 until the beginning of August 2013.

**Campaigns and Their Number of Recipients** Figure 5.14 shows campaigns9 created by Myriad, and their respective number of recipients. The average number of recipients was 105.32, and the total number of recipients was 9585. While the maximum number of recipients for one campaign was 752, the minimum amount was only one recipient.

**Myriad Users and Their Number of Campaigns** Figure 5.15 shows Myriad’s total number of users10, and the number of campaigns that they created with Myriad. There are a total 96 campaigns, after the ones created for test purposes were ignored. The average was

3.92 campaigns per user. While the maximum amount of campaigns created by one user was 13, the minimum amount was one campaign.

9Campaigns’ names are removed due to privacy reasons.

10Users’ names are removed due to privacy reasons.

**The Number of Campaigns per Month** Figure 5.16 shows the ratio of created campaigns per month. 35 campaigns out of 96 were created in April 2013, as the highest amount compared to other months. It might be because it is the middle of spring quarter at Stanford, where lecturers and organizations need to reach their recipients more often. There were an average of 15.5 campaigns create per month if the launch month January is ignored.

## User Testimonials

In this section, testimonials from the very active users of Myriad will be evaluated. Users were asked to evaluate Myriad, keeping the following four questions in mind:

* + - 1. What are the scenarios you use Myriad for?
      2. What were you using before Myriad?
      3. How satisfied are you with Myriad?
      4. How regularly do you use Myriad?

Testimonials written in listing 5.1 and 5.2 belongs to lecturers from the Stanford HCI group. The last testimonial belongs to a Stanford organization who performs mass email communication very often to reach their community.

*5.5 Conclusion*

# 5.5 Conclusion

In this section, the final solution, Myriad, for mass email communication was investigated. We first investigated the improved requirements, comparing them with the initial requirements, and the ones already applied to the initial prototype. Enabling assistant support, using KVPs as dynamic variables in the email, importing recipients list from a Google Spreadsheet, the ability to import email from Gmail’s inbox by annotating email using Myriad’s campaign labels, and automated decision-making through previously applied search filters were some of the main improvements done in Myriad.

Previous sections continued showing how those features were applied for Myriad’s UI, followed by introducing its architecture on how it was possible to enable the said features in support for mass email communication.

Finally, user and campaign statistics were given to show how actively Myriad was used during its beta stage, and the section was followed by testimonials from the active users of Myriad.

In the next section, we will see the possible extensions on the requirements of Myriad as a future work, and the conclusion of this study.