Requirements Document

December 6th 2016

Five Pixels

Sponsor:
Joy Knudsen

Faculty Mentor:
Dr. Mohamed Elwakil

Team Members:
Brandon Garling
Xiangzhi Cao
Matthew Nielsen
Mohammad Alsobhi
Clarissa Calderon

Accepted as baseline requirements for the project:

For the Client:

Signature: Joy P. Knudsen Date: December 10, 2016

For the Team:

Signature: Brandon Garling Date: December 10, 2016

Rev 1.2

Introduction

In 2014 the number of graduate and undergraduate international students studying in the United States was over 880,000 [1]! This number has grown by 72 percent since 2000 [1]. These international students often have a more difficult time making friends than students native to the United States, this affects international students in their studies and academic performance. This may be in part because international students sometimes have a difficult time adjusting to American culture. Succeeding in a foreign country can be difficult, especially when you do not know anyone native to the country on a “friend” level basis. In addition, Americans often overlook the culture of foreign countries.

In 2004 Joan and Stan Alf started a non-profit organization called the Flagstaff International Friendship Program, or FlagFriends as it is commonly referred as. This program aims to help alleviate this transition international students may face when traveling to the United States by pairing them up with local Flagstaff residents, commonly referred to as *hosts*. These students and hosts then meet up at least once a month and share in activities such as: family dinners, picnics, holiday celebrations, sports activities, musical events, and other hobbies. In addition to providing a helpful insight into American traditions for international students, hosts also gain insight into the student’s own past experiences and culture. These relationships between host and student sometimes outlast the program and many hosts and students remain friends after the students graduate. FlagFriends is not a home-stay program, they simply help establish these relationships between host and student in the hopes that both parties will learn from one another.

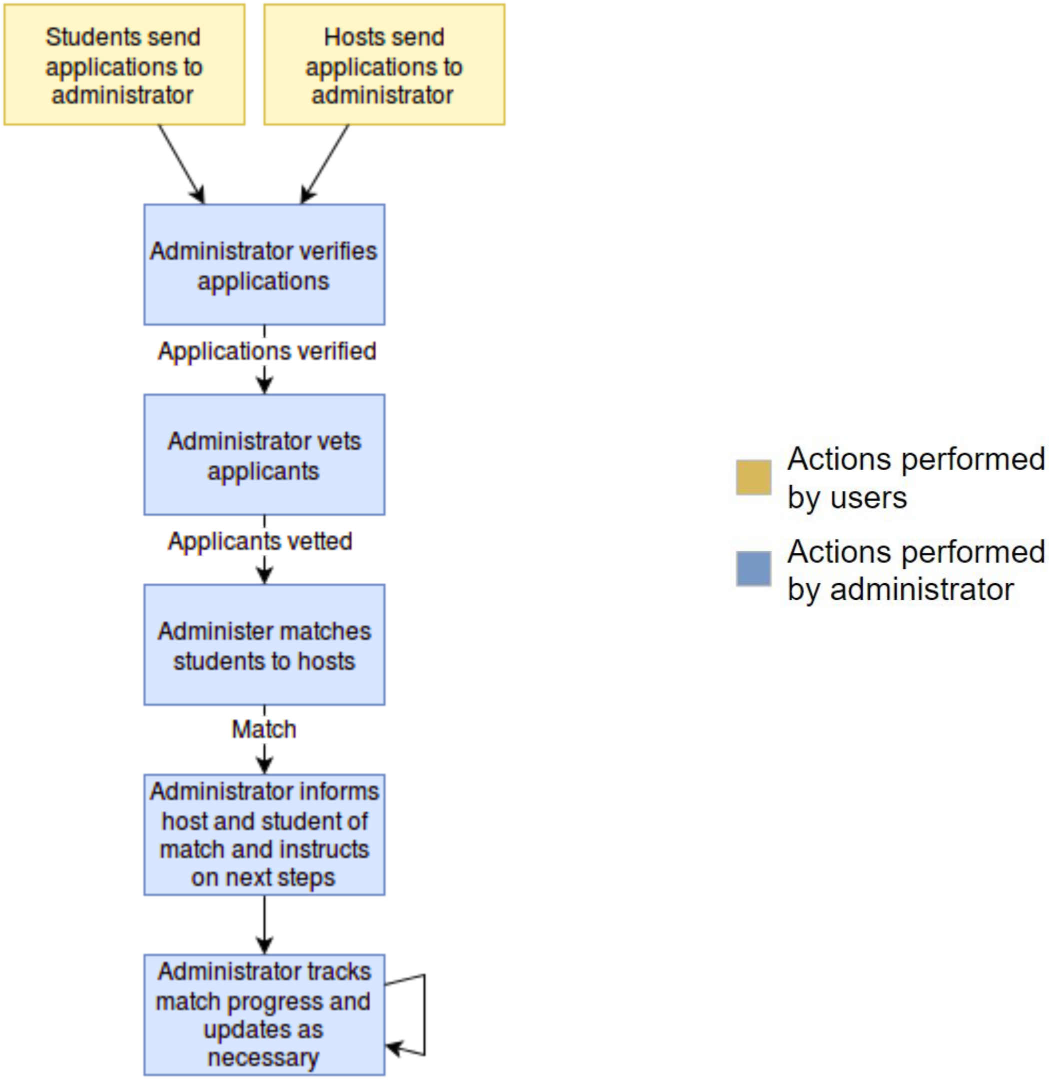
Joy Knudsen, also referred as our sponsor and our client for our NAU senior capstone experience, took over the program in late 2014 and has been working tirelessly to keep the program going and expand its potential. Currently the program serves around 80 international students and 80 hosts.

The purpose of this document is to clearly explain our sponsor's problem, Joy Knudsen, identify gathered requirements, explain our proposed solution, review any potential risks, and identify any possible future work on the project.

Problem Statement

Joy continues to use manual paper-based processes, many of these processes are out of date and difficult for one person to manage. The current workflow for FlagFriends is shown in Figure 1 below. Users which are students and hosts, and administrator which is an account of an administrator, are shown in the workflow with their actions.

Figure 1: FlagFriends current workflow diagram


Students send applications to administrator
Hosts send applications to administrator
Administrator verifies applications
Applications verified
Administrator vets applicants
Applicants vetted
Administer matches students to hosts
Match
Administrator informs host and student of match and instructs on next steps
Administrator tracks match progress and updates as necessary
Actions performed by users
Actions performed by administrator

Joy first distributes student and host applications to prospective applicants. Recruiting students usually happens at the beginning of each school semester, while recruiting hosts happens year-round. She then receives the completed applications, in paper form, and adds them all to their respective piles. As applications are received they are checked for completion and readability, and hosts and students are vetted where applicable. When Joy receives enough applicants she then sorts through all the applications and pairs hosts with students *manually* by comparing preferences, hobbies, and other information conveyed in the applications. After pairing is complete students and hosts are notified and contact information is exchanged for each pair. Joy then occasionally checks in on pairs to ensure that no problems arose from the pairing process. This whole process can take many weeks of time to fully complete. By introducing a digitized and automated process we can greatly expand Joy's ability to spread the word about her program without having to worry about getting too many applicants to handle.

Joy is excited and passionate about the FlagFriends program; however, she faces many problems that prevent her from investing her time expanding the project, and instead she must focus on maintaining the project. Looking again at the workflow presented in Figure 1, one can see an entirely manual process, filled with pitfalls and possible human error. Joy could possibly overlook a choice or lose a form completely. In addition, manually pairing students and hosts could lead to unreliable paring.

Solving these problems are important to the project so that better pairs can be created and so that no forms are lost. Time will also be addressed, if these problems are solved then Joy will not have to manually create the pairs and she could assume better roles in this project such as marketing and spreading the project to other parts of the country and world.

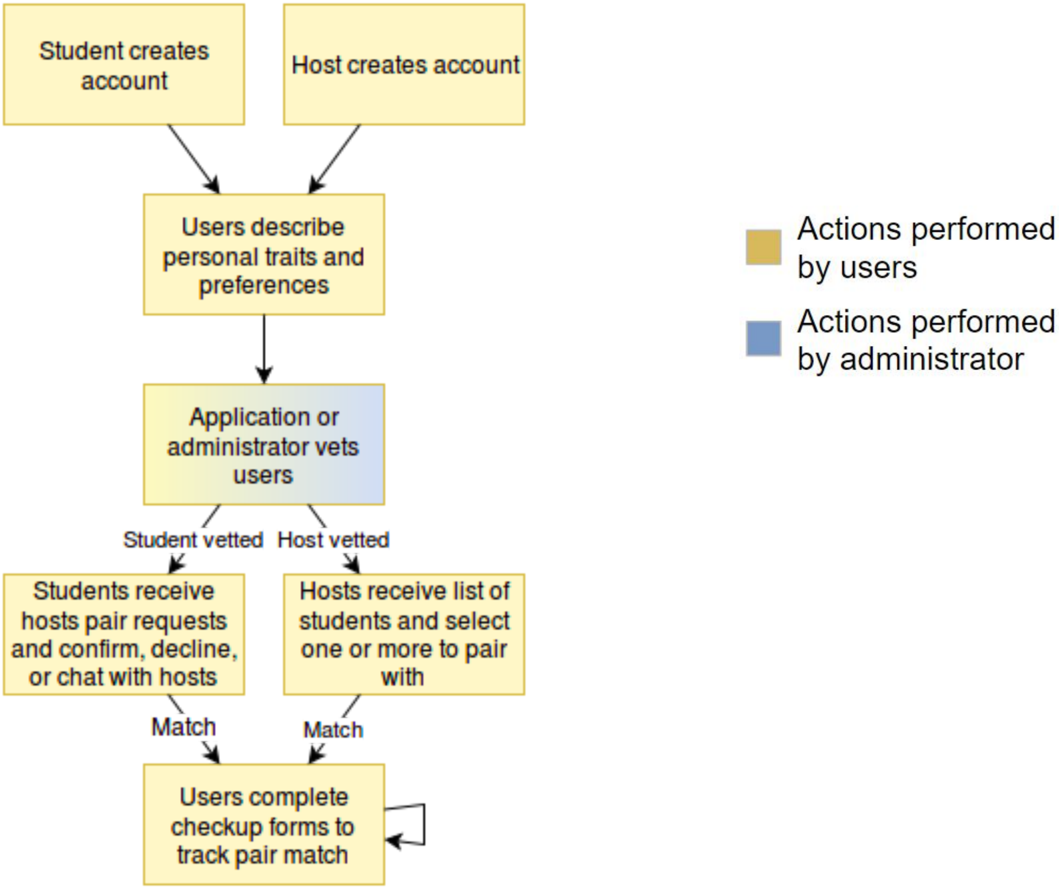
Solution Vision

Our proposed solution can be described as a general group friendship pairing web application we call Groupwise. This application will provide the client the functionality she needs to bring her organization into the digital age as well as provide a foundation for other organizations that want to run similar friendship programs. This application will be free, open source, relatively easy for new organizations to install and setup on their own infrastructure.

- Students and hosts will be able to create accounts, fill out an online application, and be in the system in a few minutes.
- Hosts will be able to view a list of all unpaired students in the application and send a friendship request to any students they may be interested in hosting.
- Students and hosts will be able to chat on the application in real time, before and after pairing.
- An administrator will be able to schedule check-up emails to be sent out to pairs to gather feedback from students and hosts and help to resolve any possible issues that may arise.
- Hosts propose a match to a student, if the student accepts, they are committed to that host and vice versa.

Our solution will first and foremost take much of the pressure off of Joy. Note the increase in the yellow boxes in the figure below.

Figure 2: Proposed solution workflow diagram


Student creates account
Host creates account
Users describe personal traits and preferences
Application or administrator vets users
Student vetted
Host vetted
Students receive hosts pair requests and confirm, decline, or chat with hosts
Hosts receive list of students and select one or more to pair with
Match
Match
Users complete checkup forms to track pair match
Actions performed by users
Actions performed by administrator

By allowing students and hosts to create accounts and fill out an online application this relieves Joy of the responsibility of handling paper applications, as well as the possibility of losing an application during the gathering process. Allowing hosts to pair with students alleviates Joy's time consuming pairing process and allows students and hosts to interact directly before being paired to help ensure a successful pairing.

This process is straightforward and simple, the biggest computational need is to keep persistent data so the application knows the state of the different students and hosts that are in the system. The system will also need to provide some sort of scheduler that allows tasks to be ran on a timed schedule.

Figure 3: Host and student pair process sequence diagram using our proposed online matching application

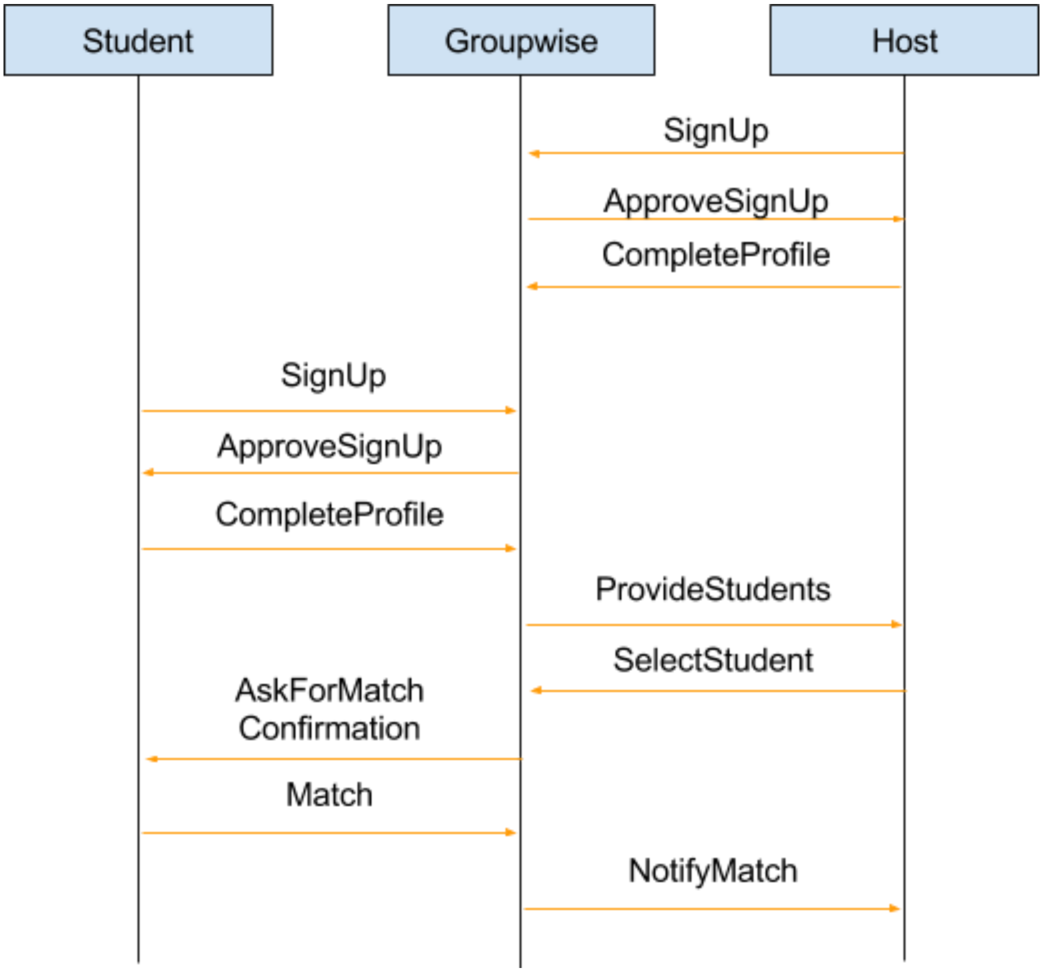


Figure 3 above shows the process of how students and hosts interact with the website for FlagFriends. They first sign up and are approved, then they complete profiles and students apply to hosts. Hosts then select students and approve them, the selected students verify and a pair is created. During the ApproveSignUp transition shown above users will be verified using their email and name that they provided. No other form of identity will be required.

This system will change how our sponsor works dramatically. Our sponsor will not have to spend time creating pairs and will be able to focus on other tasks such as expanding the program.

By working on this project Five Pixels will make matches happen easier. Joy will be working on marketing and if this project is a success, it could be expanded and used by other organizations in many parts of the country or world.

Project Requirements

Domain Level Requirements

D1. The system should support users
- 1.1. Users should have a secure way to access the site
- 1.2. Users should be able to access the site through a specific domain name
- 1.3. Users should be able to remain logged in for 2 weeks at a time, renewed each time they visit the site
- 1.4. Students and hosts should be able to report/recommend other students and hosts to an administrator for review
- 1.5. There should be a subgroup of users called hosts
  - 1.5.1. Hosts should be able to register for a host account on the site
  - 1.5.2. Hosts should be verified during the registration process
  - 1.5.3. Hosts should be able to log in to their account on the site
  - 1.5.4. Hosts should fill out a personal details form
  - 1.5.5. Hosts should be able to modify their personal attributes form after their account is initially created
  - 1.5.6. Hosts should complete an ideal attributes form for their ideal student match
  - 1.5.7. Hosts should receive a notification if a new student matching their list of attributes is added to the system
  - 1.5.8. Hosts should be able to chat with students before students accept matches, and after they are matched using a built-in messaging system
- 1.6. There should be a subgroup of users called students
  - 1.6.1. Students should be able to register for a student account on the site
  - 1.6.2. Students should be verified during the registration process
  - 1.6.3. Students should be able to log in to their account on the site
  - 1.6.4. Students should fill out a personal attributes form to better help hosts match with them
  - 1.6.5. Students should be able to modify their personal attributes form after their account is initially created
  - 1.6.6. Students should complete an ideal attributes form for their ideal host match
  - 1.6.7. Students should be able to chat with hosts that request to match with them, and after they are matched using a built-in messaging system
- 1.7. There should be a subgroup of users called administrators
  - 1.7.1. Administrators should have the ability to customize the introductory process, adding embedded YouTube videos, textual content, or photos
  - 1.7.2. Administrators should be able to review any students or hosts reported for misuse of the system and take appropriate action

1.7.3. Administrators should be able to enable and disable other user accounts

1.8. Users should be able to upload profile pictures

1.9. Users should be able to reset their account passwords or recover their account in case they forget their password

D2. Notifications

2.1. Users should receive notifications via email about chat messages or other events that happen while the user is not logged into the site

2.2. Administrators should be able to send email messages to a custom defined list of students and hosts

2.3. Users should be able to unsubscribe/resubscribe to the email notification feature

2.4. Administrators should be able to get notifications of events such as new user sign up, and new host match proposal.

D3. Matching

3.1. Students and hosts should be able to mutually pair with each other to produce a matched pair

3.2. A student should be able to be matched with only a single host

3.3. A host should be able to be matched with multiple students

3.4. Hosts should be able to see students who exist in the system

3.4.1. Hosts should be able to sort and filter this list

3.5. The system should create a “best matches” list for each host that is based on matching personal attributes as well as ideal attributes

3.6. Students should be able to deny the match request of a host

3.7. Matches should not expire

3.8. Hosts should receive a check-in form every month to fill out

3.9. Students should receive a check-in form every month to fill out

3.10. Students should be able to share that they matched with someone on Facebook

3.11. Hosts should be able to share that they matched with someone on Facebook

3.12. Hosts should be able to retract a match invitation

D4. Reporting

4.1. Administrators should be able to generate and view reports on students and hosts

4.2. Reports should include a report showing new students and hosts

4.3. Reports should include a report showing students who have not yet matched with any hosts

4.4. Reports should include a report showing hosts who have not yet matched with any student

4.5. Reports should be printable from the reporting page

D5. Customizability

5.1. Administrators should be able to change customizability options

5.2. Website name should be customizable

5.3. Landing page content should be customizable

5.4. FAQ page content should be customizable

D6. Setup

6.1. The application should perform an initial setup procedure when a new instance is deployed

6.1.1. This should allow for specifying environment specific settings

6.1.2. This should allow the user to run through all customizability options

6.2. The application should be able to securely allow the creation a new administrator user during the setup process

Functional Requirements

Requirement	Domain Requirement(s)	Explanation
R1.1 Users should access the site using HTTPS with a valid and configured SSL cert.	D1.1	Using SSL provides our users with security and peace of mind.
R1.2 The site should be accessible via the domain flagfriends.org.	D1.2	Allowing our users to use an easy to remember domain name ensures that users can find and return to the service easily and quickly.
R1.3 The site should use cookies and sessions to keep information persistent between visits at extended lengths of time.	D1.3	Using cookies and sessions allows us to easily associate information with a request from each user. This allows us to persist any information we might need between queries to the server.
R1.4 Users should be able to report other users for misuse of the system.	D1.4, D1.7.2, D1.7.3	The administrator will be able to review these reports and ban users from using the system where appropriate. Misuse reports will be stored in the database.
R1.5 The site should allow students and hosts to register for accounts. Administrators should be created on setup and by another administrator account.	D1.5.1, D1.6.1	Account registration requires at least: email, password, name, and phone number. Account data will be stored in the database, passwords will be encrypted prior to being placed in the database.

R1.6 Users need to be approved prior to becoming active in the system.	D1.5.2, D1.6.2	Users with .edu email addresses should be pre-approved, other users should be put into a queue for an administrator to approve before the account becomes approved. This approval status will be stored in the database, associated with the user.
R1.7 Users should be able to log in using a unique email and password combination.	D1.5.3, D1.6.3	After registering and being approved users should be able to log in to the system and begin using it. If a login fails, then users will not be able to access the system.
R1.8 Students and hosts should complete and maintain a personal attributes form	D1.5.4, D1.5.5, D1.6.4, D1.6.5	This form should be completed during the registration process. It can be revised at a later time by the user.
R1.9 Students and hosts should complete and maintain an ideal match form.	D1.5.6, D1.6.6	The ideal match form contains attributes configured by the administrator that
R1.10 Students and hosts should be able to communicate using an in-app communication system.	D1.5.8, D1.6.7	Users will receive notifications when a chat message is received. This chat functionality will be facilitated by the use of WebSockets.
R1.11 Hosts should receive a notification when a user matching their ideal attributes registers in the system.	D1.5.7	This notification will allow users to quickly respond to a possible new match.
R1.12 Users should be able to upload profile pictures.	D1.8	Profile pictures should be constrained in file size and dimensions. There should be a limit of 10 profile pictures for any one user. Profile pictures should be stored in the database.
R1.13 Administrators should be able to add custom content to the sign-up process.	D1.7.1	Custom content includes a simple HTML page editor that allows: YouTube videos, photos, and textual content. Custom content should be stored in the database.

R1.14 Users should be able to reset their account passwords or recover their account in case they forget their password.	D1.9	Allowing users to reset and recover their passwords ensures that users are always able to access their account. This can be implemented using a simple email verification process.
R2.1 Users should receive notifications via email or in-app where applicable.	D2.1	All notifications should be stored in the database.
R2.2 Administrators should be able to send email notifications to individual and groups of users.	D2.2	A page containing this functionality should be available to the administrator.
R2.3 Users should be able to unsubscribe/resubscribe to the email notification feature.	D2.3	Allowing users to unsubscribe and resubscribe to email notifications ensures that Flagfriends is less likely to be reported as spam, and gives the users space in case they don't want to be bothered with the system anymore.
R2.4 Administrators should receive notifications based on events that happen in the system that they are subscribed to.	D2.4	By allowing administrators configure which events they want to be notified about we can help ensure that they are not overloaded with information.
R3.1 Host to student matching should be supported.	D3.1, D3.2, D3.3, D3.4, D3.4.1, D3.6, D3.7, D3.10, D3.11	Students should be able to deny the match request of a host. Matches should not expire. Students and hosts should be able to accept a match request with a host to produce a matched pair. A single student should be able to be matched with only a single host. A single host should be able to be matched with multiple students. While searching for a student hosts should be presented with a searchable, sortable, list of students and quick overview of student profiles. Matches will be stored in the database with user information. Students and hosts should be able to share to Facebook that they found a friend on FlagFriends. This share should direct

		those who click it to the FlagFriends homepage.
R3.2 Hosts and students should be required to fill out check-in forms every month that summarizes activities both parties participated in that month.	D3.8, D3.9	The check-in form should include questions regarding amount of time met, any complications, and other related information. Completed check-in form data should be stored in the database.
R3.3 Hosts should be able to retract a matching invitation	D3.12	Allowing hosts to retract a matching invitation assures that a match proposal was not accidently proposed. This gives the hosts a chance to make a sort of final decision.
R4.1 Administrators should be able to view reports based on users and user data in the system.	D4.1, D4.2, D4.3, D4.4	Report data should be pulled from the database via the server. This data should be filterable and searchable and include features such as the ability to view a report showing students who have not yet matched with any hosts, or hosts that have not yet matched with any students.
R5.1 Customizability should be available to the administrator	D5.1, D5.2, D5.3, D5.4	Custom content includes a simple HTML page editor that allows: YouTube videos, photos, and textual content. The website name should be customizable. Landing page content should be customizable. FAQ page content should be customizable.
R6.1 A setup process should allow new deployments without touching source code.	D6.1, D6.1.1, D6.1.2, D6.2	The setup process should generate a unique key that can be used to setup the system. The site will prompt for this unique secure key to be entered after which the user will be guided through the setup process such as database connection details, email server information, and customizability options.

Performance Requirements

To help figure out a good load time baseline we decided to test the speed of multiple social networking sites such as Facebook and Google Plus, the load times of these services were averaged to around 8 seconds on a 4 Mbps connection, (using connection limiting tools on Google Chrome).

1. Initial page should load in no longer than 8 seconds with a 4 Mbps internet download speed.
2. Switching between pages should take no longer than 2 seconds with a 4 Mbps internet download speed.
3. Users should be able to sign up for an account in 3 clicks from the homepage
4. Users should be able to sign in to their account in 1 click from the homepage
5. Users should get notifications no longer than 30 minutes of them happening if they are not currently online.
6. Users should receive in-app notifications while online within 30 seconds of them happening.

Environmental Requirements

1. Users should accept to a terms and conditions before using the system.
2. There should be a landing page which describes the FlagFriends organization explaining how the program works.
3. There should be a FAQ page that displays common answers and questions about FlagFriends and the application.
4. There should be a contact page to contact a FlagFriends administrator through the website.
5. There should be a link to a handbook further explaining the FlagFriends program that is accessible by everyone.
6. The resulting system should be open source and free for anyone to use.

Potential Risks

This project presents some potential risks to our customer. There is a risk when storing information about users in a database and ensuring that no user information is stolen. FlagFriends must ensure that it takes no responsibility for any actions that any members may perform. The application being full of bugs and being unusable is another risk as well as the application not being used by anyone.

User data is stolen from big websites all the time. This is usually done through exploits in the security of the application. We can take a number of actions to help mitigate this risk and ensure that no user data is stolen from the FlagFriends application. Encrypting network communication using SSL ensures that no “man in the middle” attack can be performed on the application. Using CSRF tokens allows us to ensure that application actions cannot be invoked on behalf of the user by visiting a malicious website. In addition, security patches should be applied to ensure that system to ensure that the system itself is up to date and secure, this will be the responsibility of the client at conclusion of the project. We will also encrypt user passwords in the database to ensure if user data is stolen, passwords will not be compromised.

FlagFriends needs to ensure that they take no responsibility for any activity that happens on the website, instead they are providing a service for people to use, but have no control over how people use it. To ensure that FlagFriends stays out of legal issues, we will include a disclaimer explaining that FlagFriends is providing a service for users to use but claim no responsibility on the use of the platform itself. This will help to protect FlagFriends from legal issues.

Ensuring that users actually use the resulting application could also be a potential issue. If users have a hard time using the application this could cause the project to fail. In order to mitigate this risk, we are planning on doing user testing after we have a solid foundation for the application. This way we can ensure that user’s needs are met before the conclusion of the project.

Table 1: Risk Breakdown

Risk	Mitigation Actions
User data stolen	Encrypt Passwords Encrypt network communication Ensure that security patches are applied Anti CSRF token usage
Legal issues	Include a disclaimer
Minimal to no application use	Ensure that advertising is done Perform user testing to ensure that it fits the user’s needs

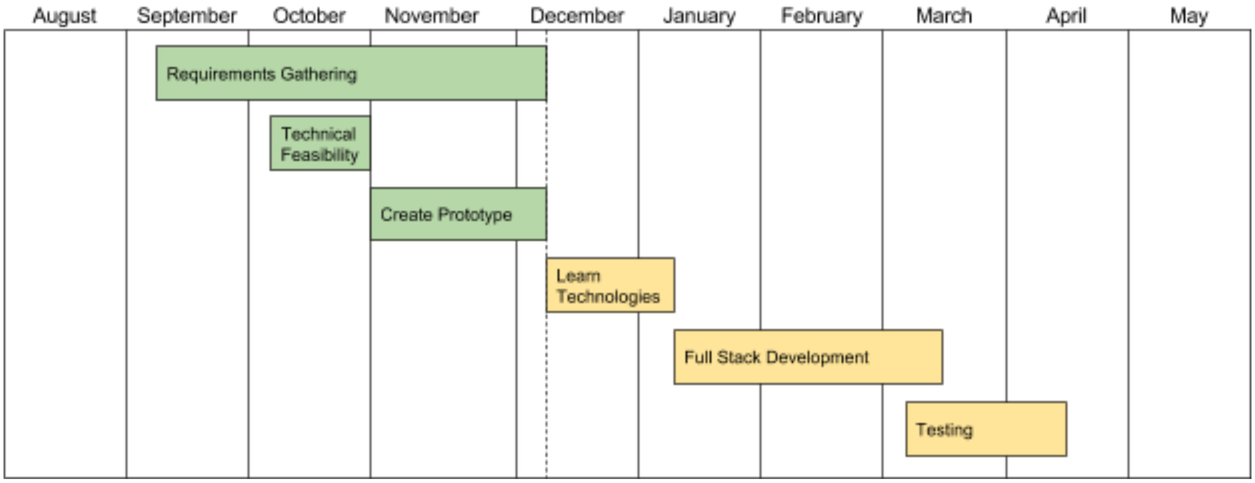
Project Plan

Planning our project roadmap is a big part of ensuring that a project is completed on time. We are planning on using Agile/Scrum methodology, breaking our project into sprints. These sprints together make several milestones that help to track the progress of the project.

Upcoming milestones:

- User account functionality
- Notifications functionality
- Matching functionality
- Reporting functionality
- Customizability functionality
- User testing
- Acceptance testing

Figure 4: Capstone gantt chart



Our project plan is to divide our tasks, as can be seen above in Figure 4. We plan to learn more about the technologies that we will be using in December and finish user support by the end of January. In parallel we plan to be working on notifications, and having that done by early February. Full matching support will be implemented by mid-February, quickly followed by reporting support. Customizability will be worked on in parallel with reporting support, we foresee this as being one of the longer milestones in the project. Finally, we plan to begin user testing in nearly March, while working on fixing bugs and any other final changes to the project. Acceptance testing will be performed at the end of March into April.

Future Work

During the requirements gathering process we identified a number of future enhancements that might be able to be done to this system. These include expanding the application to support additional customizability allowing any number of groups and any number of matches. Limits could be in place in regards to how many matches can occur between groups, as well as the workflow for each group in regards to the matching process. This added functionality would allow the application to work for situations that have two or more groups being matched with each other in some way.

Conclusion

Using an inefficient and outdated manual workflow has led Joy putting all her effort into keeping the FlagFriends project running, and does not leave much time for expanding the program. Using this proposed solution, Joy can cut days out of her workflow and better focus on the parts of running FlagFriends that she enjoys more, such as expanding the program through social activities and other marketing. Our proposed solution will easily support Joy's current user base and allow expandability for whatever the future may hold.

In addition, by providing a simple form of customizability we can allow this software to benefit as many organizations as possible. Organizations similar to FlagFriends will easily be able to pick up the application and have a working platform up and running in no time.

Five Pixels is excited to be able to work on this project and help an amazing group of people. We believe that this project is doable in the time allotted to us. We have the knowledge, skills, and team to get the project complete. We look forward to completing this project in its entirety in the near future.

Glossary

- Personal attributes form
    - A list of attributes that help to identify a user's hobbies, habits, and personal background details
- Ideal attributes form
    - A list of attributes that help to identify a user's preferred hobbies, habits, and personal background details
- Introductory questions
    - A list of questions that may differ between groups that must be filled out when signing up through the application

References

1. Haynie, Devon. *Number of International College Students Continues to Climb*, http://www.usnews.com/education/best-colleges/articles/2014/11/17/number-of-international-college-students-continues-to-climb