**Final Project Report**

**Team B – Improve Campus Morale**

9 August 2019

Christopher Morcom

William Javier Purificacion

CS 487 – Software Engineering

**[1] Prototype Analysis and Design**

**Summary:**

Our project’s initial goal the “customer” requirement) was to create a software solution that mitigates increasing morale on-campus. We initially had a number of ideas, but the one we identified as the *most likely* solution that “automates” the task of increasing morale was to create a groups/activities scheduling system that frequently involves users. From here, we identified a number of functional and non-functional requirements. A more holistic description with examples is noted in our project design report and can be seen in our prototype/presentation. The requirements listed below are those identified for our current iteration of the project:

***(Extended) Functional Requirements:***

1. The system must frequently interact with users by using “pings” that request the user to be involved with the application
2. Users for the application will be given different privileges:
   1. Individuals can subscribe to groups and events and create personal events.
   2. Clubs/Groups can create events or remove users from their group. They can also subscribe to existing events/groups in cases of a *partnership*
   3. Admins have the final say in content moderated by the system where the system determines the need for a human user.
3. Users may message each other and will have profiles.
4. The system must provide a means for new users to be created.
5. Data must be validated on a local device before being sent to the server.
6. Users must be able to use a querying and filtering mechanism to make finding user-relevant groups/events quickly and easily.

***Non-Functional Requirements:***

1. *Security:* the application uses defined privilege levels for each user and will have an interface for users to be validated/log into the application
2. *Ease of Use:* The application must perform operations quickly and user tests should determine that the app is easy to use and is personalizable. User interfaces must also be “aesthetically relevant” to the application.
3. *Availability:* The application must work on both Android and iOS devices.
4. *Maintenance:* The app must be updatable and may be funded via personalized ads or campus funding/crowdsourcing.

**Measuring Success:**

We found out that our application’s “success” can be done through numerous methods. Since our application kind of work like social media, there’s going to be a lot of information that will be coming around. Not only that, but in terms of security (especially the misuse of the application), implying that there will be moderators to monitor the application also implies the difficulty of what multiple social media platforms already face with monitoring their respective platforms. Our goal is to improve campus morale, such that our application must have an effect on the users. Using it itself does not mean that the application already works. And if so, even if the application works, does it mean that our goal is met? With this final analysis, we must be able to narrow down the ambiguity of measuring our application’s “success”. Some ideas for testing “success” uses data that answers the questions below:

* How much do they use it?
* What is the frequency and time spent by individuals on the app?
* How are users using the app?
* What are user subscription trends?
* What feedback are users willing to give (comments/ratings)?
  + Is the application a good tool for their student life?
  + Are they be dependent on it?

Using these data, we can create metrics that determine our app’s success. One limitation is that the data needs validation, which is a separate issue. We explain how the use of the data modeling methods to determine success. First, App use frequency for individuals is a quantitative measure of how relevant or useful the app is as a percentage of the entire target user set. Next, a subjective measure of how the users are using the app is important in determining if the app is being used as intended. User subscription trends determine if users are engaging in morale-boosting activities, or if they are shifting to a higher frequency of morale boosting activities/groups. A “morale score” that is assigned to groups/events to create a numerical score that can be used to determine user trends in a numerical sense. Finally, user feedback in terms of reviews and ratings along with the number of users with the app installed and have used the app frequently (determined via trend analysis) will give insight into maintenance updates and functionality to improve, add, or change in the app, which would keep users dependent on the app.

**Prototype Design:**

Our prototype follows our User App State Transition Diagram *(below)*, but does not show the app performing from the perspective of a Group User or an Administrator. The prototype is largely aesthetic-based in a way that simulates functionality. In the source files provided, we use many XML layouts to simulate an android app’s look and feel. Since the final product will be categorized as a social/networking application, we try to use abstract techniques to focus heavily on the social aesthetic and functions that the app would display.



**Individual User App State Transition Diagram**

**[2] Prototype Testing**

**Testing strategies:**

*Notification Pings*

Whenever there is new information that is added within the system , the system sends a “ping” to all the users with those tags. If the user is subscribed to those tags, they will receive that “ping”. If the user interacts with the “ping”, a “ response ping” is sent back to the system which is relevant for determining the application’s usage. This “response ping” will help confirm if the application is working correctly and if it being used. And if it does, application usage will be recorded.

*User/Admin Privileges*

The user and the administrator must be able to use what their respective privileges can do. If the user/administrator doesn’t have a specific privilege, they shouldn’t be able to use it.

*User Creation/Account Validation:*

The new user will create a new account. If the new user is tied to its respective university, then the user will successfully create that account. The university will provide a database of all the user’s accounts to help verify new users.

**Test cases:**

*Notification Pings*

A new event is added in the system. The system reads the event’s tags. After this, the system sends a “ping” to all users with those tags. Users with those tags receives the “ping”. The user that makes a response to the “ping” automatically sends a “response ping” back to the system.

*User/Admin Privileges*

The user will demonstrate functionalities within their privileges. The administrator will also demonstrate functionalities within their privileges.

*User Creation/Account Validation*

The new user creates a new account. The account must use an email from their respective university. If the new user tries to create an account which isn’t tied to the respective university, then the user shouldn’t be able to create that account.

**Results:**

*Notification Pings*

The system acknowledges the “response ping”. In cases where there is no “response ping”, the system also takes note of that. “Response pings” along with the amount of “pings” the user receives is analyzed, and the user will be labeled active or inactive depending on the results.

*User/Admin Privileges*

If the user’s functionalities work, then they must see things like notifications popping up, or if they created an event, they must be able to see the event from the events page of the application. The same thing goes with the administrator, especially with key functionalities that they will most likely use such as editing/removing events.

*User Creation/Account Validation*

The success rate of this functionality must be 100%. No other users must be able to create a new account for the application of the respective university. The account verification must work correctly, or else, users will be at greater risk of application misuse.

**Application Success and Value:**

We think that our application will be very useful to students because it is very simple to use and its key functionalities are straightforward. To be honest, our application’s first steps will be heavily reliant on its partner organization. For example, if IIT tells all the students that this application exists, then if IIT made the incentive to let everyone know, then it means that this application is useful and trustworthy. Aside from that, if the just needs a tool for them to be engaged, all they have to do is create an account, follow the user-interface, do the notification-filters, and that’s it! This application does not stray away from its sole purpose. It aims for the student to have an easier time rather than navigating through their messy IIT inboxes. It is low cost, and the security system isn’t entirely new. The only issue is moderation of the application. Any of its misuse should be expected to occur, and similar approaches of other social media platforms must be copied in order to ease the process. However, many of these approaches are still not very effective, and different approaches may be taken into account.