Project Analysis Report
Team B – Improve Campus Morale

20 July 2019

Christopher Morcom
William Javier Purificacion

CS 487 – Software Engineering

Summary:

Stress is a common trait among students, faculty, and generally everyone involved on-campus. This causes a loss in morale over time. The objective of this system is to facilitate morale-boosting by increasing the opportunities for users to learn about and attend stress-relief activities. Stress-relief activities are sorted into three major areas: Social, Physical, and Academic. Activities are then sorted into subsections of these three areas

User Categories:

University members that possess a university email

- Undergraduate students tend to be more likely to engage with other students
- Graduate/PhD students have a higher focus on life outside of campus and are less likely to use per-class resources from previous years
- Faculty sometimes engage in on-campus activities but tend to focus off campus for social events

Requirements:

Functional:

1. *Identify the levels of stress that the user has*
    a. The user could be experiencing some stress that they may not be aware of
    b. The user may be experiencing stress, but unable to measure how stressed they are.
        i. This should help clear things for the user if they are unable to make an evaluation on how stressed they are.
2. *Describe what kind of psychological/mental issues the user could be experiencing*
    a. If the user is actually experiencing stress, then it could be possible to identify what kind of stress the user may be currently experiencing.
    b. The user must be able to identify what kind of psychological/mental issues they COULD be experiencing (if they are stressed)
        i. Ex. Depression
3. *Describe what kind of help could be given to the user depending on their preference*
    a. If the user is willing to get help if they are undergoing stress, then they must get the appropriate help *that they want*.
        i. Ex. A reminder to write a journal every night, or to finish a checklist that they made. Or to have an option to get notifications if they want to participate in any events
4. *Identify critical data that relates to user interactions with the system*

    a. The app will record how the user interacts within the app, and must see which data is relevant (user preference and satisfaction scores) which will increase querying precision for future information provided to the user.
5. *Constantly update user interactions with the system*
    a. User choices will affect their preferences in the app. Erroneous selections from the user should be handled by restoring the user-preference score to its previous state or by re-normalizing the user preference score if previous states are unrecoverable.
6. *App speed and query precision*
    a. The app stores data locally and must be able to read it/sort it within seconds to display to the user. Preserving BCNF sorted lists in app data is important and will make querying quicker. Loading changing preferences should take no more time than a query.
7. *Background tasks*
    a. Background tasks that save app states are essential for handling interrupts like a phone call or text message, or quick app switching. Background tasks are also useful to provide the user with reminders for events they have subscribed to, or to remind the user to check the app.

Non-Functional:

1. *Easy to read and use*
    a. Users should be able to quickly open and integrate the app into their daily life. A clean interface that is easy to understand and use should reduce the chance of mis-selects and confusion in not being able to find what the user wants.
2. *Portability*
    a. The system must be able to interface with social media applications (Facebook, Google, etc.). The system must connect with the user's academic account for verification and the user must be able to access the app outside of network access (using saved app states and sessions).
3. *Privacy Policy*
    a. The app must use a secure login interface that has a recovery option and secure input.

Testing Approaches:

1. *Scenarios and Test Cases*

    We would need significant user testing to evaluate what events are most common, what categories would be most popular, and generalize each user group's preferences for maximum satisfaction. For test cases, we would need to ensure our algorithm that

identifies probable user choices from the set of user preferences is accurate based on an accuracy score calculated per event.

2. *Traceability*

   Each user needs to have a personal, verifiable login to make the app secure and preserve user data across sessions.

3. *Usability*

   The user will be presented with a login page that will verify their affiliation to their home campus via email verification. They will be asked their user preferences on first use and be provided an option to change preferences in a universal “settings” portion of the application.

4. *Edge cases and Exception Handling*

   All user preferences may not be readable or may cause undesirable/inaccurate suggestions so the most likely way to create accurate results would be a long series of simple questions (less than 35) that have preprogramed options (like a likert scale). Scores from the questions would be compared to scores for events and events will be sorted and displayed. All events will be shown. As the user attends more events, the event score will be normalized into the user preferences score to generate better results later on.