Team #12 – UniVent:

1. Altug Gemalmaz
2. Amjad Zahraa
3. Andrew Peterson
4. Anirudh Pal
5. Sultan Al-Shamsi

Prof. Jeffery Turkstra

CS 30700

8th August 2017

# UniVent Product Backlog

## Problem Statement:

**S**tudents, especially new students, struggle to stay up-to-date on events, activities, and other various occasions happening on campus. Fragmented technologies such as BoilerLink, Facebook, Reddit, UniTime 4.2, etc. do not provide a comprehensive solution to increasing student involvement. However, our solution combines these ideas into a real-time, user-friendly application that handles posting, sharing, and matching events to users.

## Background Information:

### Brief Explanation:

College and University students rely on communities, clubs, and organizations to gain valuable interpersonal skills and expand their professional network. Currently, it is the student's responsibility to search through hundreds of clubs, countless news updates, and various social media platforms to plan how they will be involved during their time as a student. For new students, this is a daunting task. We aim to bring those callouts and events to the students in real-time.

### Similar Systems:

There are various systems currently in existence that attempt to solve the problem we are addressing. The Purdue iOS Development Club has done a great job creating an all-in-one mobile application containing useful information for Purdue students. BoilerLink is Purdue's official club information website, containing an updated feed of club callouts and events happening soon. Paper flyers are constantly posted around campus with information about upcoming club and university events.

### Limitations of Similar Systems:

- *Purdue App by iOS Development Club:*
  - Too much information is presented to the user.
  - There is no geo-spatial aspect in their app.
  - The screen is cluttered with features.
- *BoilerLink:*
  - The user must search for information instead of being present with clubs and events near them.
  - There is no mobile-friendly version of BoilerLink.
- *Flyers:*
  - Somewhat outdated method of posting events.
  - Wastes paper, space and money.

  - Details about the event cannot be changed once posted.

### Solutions for Limitations:

- Minimalism allows us to keep it simple for the user and avoid unnecessary feature sets.
- User location will be taken into consideration while presenting information.
- Instead of making the user search, we will recommend things that fit their interest and location.
- The app will be on a mobile platform which is more accessible.
- No need to waste paper when you can get all the information you need right at your fingertips.

## Requirements:

### Functional:

1. As a user, I would like to sign-up with Facebook to simplify account creation.
2. As a user, I would like to login with Facebook.
3. As a user, I would like to enable location services.
4. As a user, I would like to enable push notifications.
5. As a user, I would like to see a tutorial as a first-time user. (If time allows)
6. As a user, I would like to see my location on a map.
7. As a user, I would like to select field of interest i.e. types of clubs.
8. As a user, I would like to enter my class schedule.
9. As a user, I would like to select my search radius to only see relevant events.

10. As a user, I would like to see events near me on a map.
11. As a user, I would like to see events near me on a list.
12. As a user, I would like to be able to select a pin on the map to view details.
13. As a user, I would like to get directions to an event. (If time allows)
14. As a user, I would like to flag an event as inappropriate.
15. As a user, I would like to flag an event as false.
16. As a user, I would like to see the time of an event.
17. As a user, I would like to see the place of an event.
18. As a user, I would like to see a description of an event.
19. As a user, I would like to get push notification about events near me.
20. As a user, I would like to get push notification based on my schedule.
21. As a user, I would like to get push notification based on my interest.
22. As a user, I would like to be able to select if I am attending an event.
23. As a user, I would like to be able to select if I am not attending an event.
24. As a user, I would like to rate an event.
25. As an event organizer, I would like to be able to create an event.
26. As an event organizer, I would like to specify a title for an event.
27. As an event organizer, I would like to specify the time for an event.
28. As an event organizer, I would like to specify the location, description, interest tags etc.
29. As an event organizer, I would like to specify if there should be an attendance count.

30. As an event organizer, I would like to be able to see my most recently posted events.
31. As an event organizer, I would like to be able to edit any of my recently posted events.

**Non-Functional:**

*Geolocation:*

- *Getting Access:* The location of the user is vital for the application. By having user locations; UniVent will be able to sort closest events per user preferences. We will get access to this information through Apple Maps API.
- *Smart Polling:* To avoid draining battery unnecessarily we will use various factors to reduce battery consumption. For example, while walking, we will keep computing new events near the user, we will also not keep computing new data sets in the absence of a GPS connection. Even factors like standing still, background activity and battery levels can be taken into consideration.

*Scalability:*

- *Response Time:* Bluehost is a professional hosting service that can handle large volumes of requests ensuring that we can scale if we need to in the future. It provides guaranteed response times making it possible for us to predict the latency of our system.

- *Testing:* Small scale testing will take place within the team to prove that our systems do work on larger scales. We might create bots to test our backend infrastructure. Tests would include different criteria such as handling big number of users, high number of events to test the database.
- *Facebook Verification:* Using Facebook verification ensures that users don't have to create an account to use this app and at the same time Facebook handles all the user information on their servers.
- *University Verification:* We might implement a university student verification process that will help eliminate malicious bots from infiltrating the system, and by eliminating the threat of malicious bots we secure our servers from overloading.

*Usability:*

- *Map/List Design:* Our UI will follow a split design scheme. One screen will show the map with relevant information on in. And the other will have textual information. This allows the user to go back and forth between two distinct forms of data representation.
- *Visual Data (Map View):* Sticking with the split design all visual and small textual information will be presented in this section. For example; Pins, Map, Event Name.
- *Textual Data (List View):* All other textual information will be in this view. For example; Date, Time, Details.

- *Minimalistic:* We will avoid adding multiple sections and try to make all features cohesive instead of being distinct. This will help increase usability.

*Data Management:*

- *MySQL:* The First database of choice would be MySQL. One of the things that will go into this database will be all the events with fields such as ID, Time, Location, Flag, Attendance etc.
- *RESTful API:* These APIs can help streamline our database management. They also provide certain security features.
- *Local Data:* We will retain a certain subset of the data on the phone to avoid querying the database for every piece information that the user needs. This will reduce the number of data transfers. This will also allow the app to operate in absence of a network connection.
- *POST Command:* Query the database using post will avoid malicious access to the database.

*Resource Management:*

- *Polling Smart:* Resources can be managed smartly by reducing app processes during low battery situations or increase app processes while the student is moving (i.e. actually using the phone).
- *Small Footprint:* We can also reduce memory usage by building a compact and efficient code.

- *Primitive Types:* We try to primarily use primitive types since they have low overheads in terms of memory.
- *Background/Live:* We will also reduce the number of resources used by the app when it is in the background relative to the amount resources used while it is in the foreground.
- *Fewer Data Transfers:* Processing small portions of the data on the phone rather than querying the database multiple times could also reduce our network usage.
- *Memory Leaks:* X-code would be used to identify memory leaks when the application is tested under different scenarios.