CEN 4010 Principles of Software Engineering

Summer 2021

Milestone 3 Project Proposal and High - level description

Activity Connects (WORKING TITLE)

Team 4: Event Masters

Revision	Date
1	6/15/2021
2	6/30/2021
3	7/7/2021

Team:

Nathaniel Martin: Development Manager; nathanielmar2019@fau.edu

Aaron Sharari: Scrum Master (Github Master); ashahriari2019@fau.edu

Terrence Charitable: Product Owner; tcharitable2018@fau.edu

Sebastian Nunez: Developer; sebastiann9956@gmail.com

Juan Merlos: Developer; jmerlosjr2017@fau.edu

Executive Summary:

Our application, Activity Connect (working title), which allows end-users to find and sort through events to go to with friends or acquaintances that are relatively Covid safe for the average end-user. More personalized experience for the end user to provide a recommended list of events that they can go to. Providing live up to date weather for the events as well as a calendar to see upcoming events and the associated weather with them. Email notification once the end user has booked the event and then a notification the day before the said event.

To promote socialization/events through Covid friendly activities and beyond. As well as engage with local areas to create a sense of community.

Competitive Analysis:

Activity Connect (Working title)	EventBrite
Search Bar	Search Bar
Easily maneuverable site	Cluttered(browsing through events is messy)
Able to follow friends	Only able to follow the event itself
Able to see the weather for the event	No weather info shown
Share schedules	Only see your own schedule
In-Depth Filters	In-Depth Filters
Community oriented	More independent
Able to see event location	Able to see event location

_____Our goal in this website is to attempt to make a website that is more meant to be used with friends, in trying to properly schedule activities to do. Which is a challenge on its own to do now-a-days due to friends having work, school, or other personal activities to work on. Activity Connect is meant to close the gap in this challenge by introducing quality of life features that EventBrite does not have. We feel that their browser page is cluttered by having four events per line. By reducing the amount per line we can allow events more room to stand out rather than a small box. The ability to add and communicate with friends on the app itself would be a great feature, as of right now the main way to share an event on EventBrite is to use a 3rd party social

app to send the info. By having an integrated feature we make this much easier, and allow for friends to share schedules. Weather is a vital variable in deciding on what activities are doable, beach day may not be viable on a stormy weekend, but a play would be a great alternative. As mentioned earlier in the friend feature, we will have the ability to see friends schedules and share our own. Easier to track what days all friends are able to hang out together and coordinate. Our site will also show more community made events, such as fundraisers, or charity events on the home page.

Data Definition:

Users:

- UID(int)(8) - id of user

- UN (string)(30) - name of user

- UE(string)(30) - users current email

- UPW(string)(15) - users current password

- UUN (string)(15) - users current username

- UPN (int)(10) - users current phone number

- UBID (string)(8) - user buddies list id

- UCID (int)(8) - chat id to connect user to his/her ongoing chats

User Activities:

- UAID (int)(8) - id of users activities

- UP (string)(50) - user list of preferred activities

- UCA (string)(50) - list of current user activities

- UPA (string)(50) - list of planned user activities

Buddies:

- BID (string)(8) - id of Buddy

- BN (string)(30) - name of Buddy

- BP (string)(10) - user list of preferred activities

- BCA (string)(5) - list of current user activities

- BPA (string)(5) - list of planned user activities

- BCID (int)(5) - chat id to connect user to his/her ongoing chats

Events:

- EID: (int)(5) - id of event

- EN: (string)(30) - name of event

- ED: (string)(5) - activities being done

- ESD: (int)(12) - start date/time

- EED: (int)(12) - end date/time

- EBA: (string)(3) - All the buddies attending

- EL :(string)(30) - Location of the event

- ER: (int)(5) - Rating

Overview, Scenarios, and Use Cases:

In this site we will provide an exceptional way for the average end user to schedule and collaborate events to do with family and friends. Ranging from Outside activities such as the beach or park to Inside activities such as parties and board game nights. All will be provided to increase the socialization of the community around you.

Let's do an example: Joe and Jane, two new users who have never ever seen the website, want to host an event but want it to be Covid safe. How can they accomplish this? With Activity Connect (Working Title) Joe and Jane can schedule an event using our filters to find something that is Covid safe for their event. Wait, they also want it to be affordable so using our filters they can also find an option that is both fun and affordable. Joe and Jane settle on having an event at the beach from here our weather algorithm will be able to dictate which day is good for a beach day.

Through this Joe and Jane are able to create and host an event to spend with their friends. Once the event is made a group chat is created to allow the group to communicate outside of the event till the event is over. Thus providing an adequate socialization that we all crave during these difficult times of the pandemic.

As you can see Activity Connect (Working Title) allows and even encourages the end user to have fun and try new things leading to new life long friends. By showing the user a step by step filter system we are able to narrow down their options to give them exactly what they so depressly need. Overall making an excellent tool to increase socialization during the pandemic and even beyond

Functional Requirements:

1. (MUST HAVE)

    ● Site shall have a filter mode which will change the events shown to the user, filters will include aspects such as Covid Safe events, non-mask wearing activities, pricing (free -> max), distance from user, start date/ end date, etc.

    ● Site shall provide weather information for both the day and location to help show the user good days to choose outside events. Will also have a part in the filters with an "Outdoor"/ "Indoor" filter to make it easier to find events to suit any weather.

    ● Site shall have a social aspect of being able to create friends (buddies) where users can send event info, or chat, and even see the events their buddies have marked down. This allows users to coordinate and plan activities with buddies as well as attend them together.

    ● Site shall have a login feature for users to never lose track of info they want stored, this will be implemented through the use of Google API to also allow the correct synchronization of Calendars

2. (DESIRED)

    ● Site shall show a list of events located near the user location by default only to be changed when either location is changed or when a filter is created, this will try to make the user more aware of community made events. From fundraisers, to charity events, the goal is to try and make users more connected with their own community.

- Site shall sync with google calendar so that once a user has finalized a plan they can easily access both the date and information wherever they go. This will also allow for both seeing friends' schedules and sharing of your own schedule.

3. (IF WE CAN)

- Site shall allow for user made events to be both created and shown (via User Created Filter) so that smaller events, such as school fundraisers or clubs can gain more following

Non-Functional Requirements:

1. Security:

    a. Account information will be secure and only available to the user although it will be used to make recommendations on events they should attend

    b. User is capable of making both public and private events so that they are only seen by the intended audience

2. Accessibility:

    a. User capable of connecting account through google instead of making an account from scratch

    b. Make sure works correctly on all mobile and other personal devices

    c. Works on any operating system

3. Performance

    a. Site loads up as fast as possible
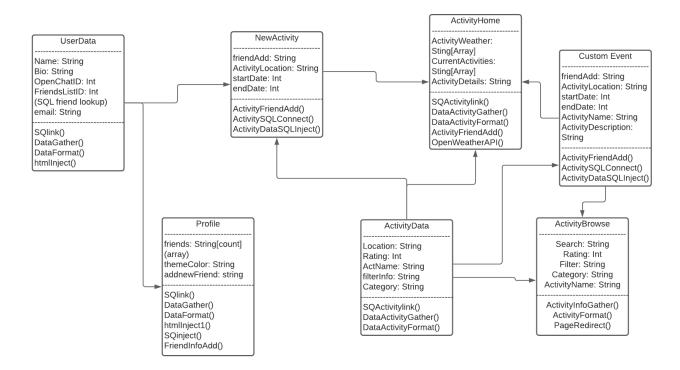
    b. Site doesn't cause crashes due to errors

4. Storage

    a. Stores info using sql database

Risks:

- Currently a majority of the team does work and are taking other classes, so exams or homework may result in an issue. Hopefully this does not occur
  - To solve this we will continue to communicate ahead of time what days are available, and who can perform which tasks. If an event or issue comes up for someone, they are more than able to let the rest of the team know or ask for assistance.
- Programming the storage of information required for events to the site may become a difficult task as SQL is still new to some of the team.
  - To solve this, the ones that have knowledge in SQL will assist in teaching and helping others understand. We will also use youtube, or other helpful sites to help solve the trouble we may be facing.
- May be a risk to place big organization events on the site due to legal reasons
  - To solve this, we will be more focused on community events that are free, school events such as the ones FAU does or any others that shouldn't be an issue to showcase
- Events may contain certain rules or laws that people must follow, trouble may arise if those aren't followed.
  - To solve this, we will link to the section of their website showing any rules needed to be followed

UML DIAGRAM

UserData

Name: String
Bio: String
OpenChatID: Int
FriendsListID: Int
(SQL friend lookup)
email: String

SQlink()
DataGather()
DataFormat()
htmlInject()

NewActivity
----
friendAdd: String
ActivityLocation: String
startDate: Int
endDate: Int

ActivityFriendAdd()
ActivitySQLConnect()
ActivityDataSQLInject()

ActivityHome

ActivityWeather:
Sting[Array]
CurrentActivities:
Sting[Array]
ActivityDetails: String

SQActivitylink()
DataActivityGather()
DataActivityFormat()
ActivityFriendAdd()
OpenWeatherAPI()

Custom Event
----
friendAdd: String
ActivityLocation: String
startDate: Int
endDate: Int
ActivityName: String
ActivityDescription:
String

ActivityFriendAdd()
ActivitySQLConnect()
ActivityDataSQLInject()

Profile
----
friends: String[count]
(array)
themeColor: String
addnewFriend: string

SQlink()
DataGather()
DataFormat()
htmlInject1()
SQinject()
FriendInfoAdd()

ActivityData

Location: String
Rating: Int
ActName: String
filterInfo: String
Category: String

SQActivitylink()
DataActivityGather()
DataActivityFormat()

ActivityBrowse

Search: String
Rating: Int
Filter: String
Category: String
ActivityName: String

ActivityInfoGather()
ActivityFormat()
PageRedirect()

High-level architecture:

- HTML

- CSS

- JAVASCRIPT

- GOOGLE API

- MYSQL

- VISUAL STUDIOS

- GOOGLE CHROME SUPPORTED

Database Org:

Using our Data Definition section as our guide, this will be how the main tables will be created. The table 'Users' would be focused on personal info such as what is their user-name, password, email, current event plans, etc. The foreign key for this table would be UID, so we may access any User related info easier. The table 'Buddies' would be more focused on the social media aspects such as storing the chat logs, and storing what events your friends have saved so it's easily shareable. Like the user table the Buddie table will also contain a foreign key called BID. The table 'Events' will store all information related to the events such as address, rating, times, etc. It will also contain a foreign key for use named EID.

STORAGE:

_____For this site, we have no current plans to implement videos or any audio as it is not needed for our goal. The main media addition to the site would be photos of the events and anything associated with them. To properly store this we shall store all images in our database using php. So that images properly show up only when their event is called.

Search / Filter:

_____For this program, when events are created they will have tags assigned to them such as 'Outdoors' or 'Free'. With these tags assigned to each event it allows for filters of events to be easy for the user. SQL will help us greatly for this. By having the user decide what tags they want for the events, it will narrow down the list to only include those that contain the tags. The user will also have the ability to manually search up the event by name if they desire.

Checklist:

- ☑ ~~Team decided on basic means of communications~~

- ☑ ~~Team found a time slot to meet outside of the class~~

- ☑ ~~Front and back end team leads chosen~~

- ☑ ~~Github master chosen~~

- ☑ ~~Team ready and able to use the chosen back and front-end frameworks~~

- ☑ ~~Skills of each team member defined and known to all~~

- ☑ ~~Team lead ensured that all team members read the final M1 and agree/understand it before submission~~

- ☑ ~~Created a prototype on Balsamiq - GUI MOCKUP~~

- ☑ ~~Revising project document~~