PorchFest Community Music Festival Organizer Feasibility Study

Table of Contents

Team

Aditi Jain, aj323@cornell.edu
Aditya Agashe, ava9@cornell.edu
Alan Wu, ajw238@cornell.edu
Candy Xiao, clx2@cornell.edu
Carlos Mendez, cam468@cornell.edu
Heather Cai, hmc57@cornell.edu
Kevin Hui, kh523@cornell.edu
Rohit Biswas, rb625@cornell.edu

Client

Robbert Van Renesse, Computer Science Research Professor, 433 Gates Hall, rvr@cs.cornell.edu

Task

A "PorchFest" is a community event in which musicians play for free for their neighbors. The Team will be creating a website to handle the organization of hosting, playing, and attending any PorchFest. The web site will have essentially three classes of users: organizers, musicians, and attendees. An organizer can "create" a new PorchFest, specifying the time slots and area in which the PorchFest will be held and ultimately schedule the bands. Musicians register information about their bands and the locations of their porches. Attendees want to find out which bands are playing when and where.

The task can be split up into 4 parts:

1. Front End
2. Back End
3. E-mail functionality
4. Scheduling Algorithm

Benefits

The current state of organizing and running a PorchFest is slow and complicated. Last year, 185 bands played on their porches in Ithaca's Fall Creek neighborhood during a 6 hour event, and was attended by thousands of people. This event was scheduled by hand upon receiving input from Google documents. This is is a very tedious process and makes it difficult for the

organizers. Creating an automated system will make it easier to hold future PorchFests and hopefully will encourage more communities to organize and host more of these events.

Preliminary Requirements Analysis

The website will need to meet the following requirements:

1. Web Interface
    a. Create Account or Login
        i. Select to host or participate in an existing PorchFest
    b. Browse available PorchFests
2. Database
    a. Store PorchFests with associated bands
    b. Store bands with associated users
    c. Store users with associated email addresses
3. E-mail
    a. Initially, be able to export all emails related to a PorchFest into a CSV file, which can then be fed into the recipient field of an email in order to send out mass emails to everyone.
    b. Eventually set up a customizable mailing service that can be used to contact various groups within a PorchFest.
4. Scheduling
    a. Assign performers specific time slots and porches during a PorchFest automatically given various constraints

Technical Requirements Feasibility

1. *Server* - Our client, Professor Van Renesse, will be running the website after it is finished. He prefers to host it somewhere cheap, e.g. GoDaddy vs Bluehost.
2. *Database* - Our aim is to create a maintainable and functional system that is easy to understand. MySQL is a popular open source database in which we can easily store and model relevant data.
3. *Front End* - There are many front end Javascript frameworks that will be helpful in implementing a good design and interface. AngularJS is one of many frameworks that are great for dynamic views in web applications.
4. *Mapping and Scheduling* - Part of a PorchFest is knowing where the bands are playing. We plan on integrating Google Maps into the website to provide this interface. The constraint optimization problem of scheduling will be interesting to tackle. Even though it is NP hard, our constants are not too large.

Scope

For the purposes of this project we are concerned with building a website to support scheduling and managing instances of PorchFests across various communities. The website will make it easier for the organizers as well as the musicians to create and be a part of these events, and will provide a user friendly interface for the attendees as well. We will be creating and maintaining a customized database schema and will support email functionality, a scheduling algorithm, and front end interface with different view for the different types of users: attendees, organizers, and musicians. The end product will be hosted on a web server and will be viewable on all standard browsers. Ideally, this project will also be mobile compatible for people to check the schedule on the go at the event. We discussed with the client iterative enhancement of requirements so that a working project will be produced by the end of the semester, but may not have all the extra features initially put forth. We have outlined this further in the Milestones section. The creation of this project will hopefully help garner more interest in hosting PorchFests across the country.

Suggested Deliverables

Management Deliverables

1. *Requirements Analysis* - A document that encompasses the formal requirements of the project, both technical and design, in order to ensure that the system will closely match the vision the Client has in mind. Through iterative refinement, the Client will have the ability to modify and clarify design and functional requirements as we proceed.
2. *Design Document* - This document will outline the entire design of the system, including the software architecture and frameworks. The designs will be iteratively made with the client's input. The backend team will handle documentation of the technical portions of the architecture and the front end team will outline the front end frameworks.
3. *Source Code* - Our Team will deliver the final implementation of the website to the client during the last milestone. We will also upload the database and code to the hosting provider that we choose. Our website will have been tested thoroughly so that it is ready for real-world use.
4. *User Interface Design and Testing* - There will be a document describing the final UI design that has been verified, tested, and approved by the client.
5. *Test plan, test examples, and results* - A document with a comprehensive overview of the performance of the scheduling algorithm, including run time complexity analysis and various test sample results.

Technical Deliverables

1. *Database* - The system that will hold and organize all of the necessary data, which includes user information, such as name, email, user type, etc., bands, and PorchFests. We plan to use MySQL to filter out the data and come up with a maintainable schema so that the project can be easily scaled up in the future.
2. *Interface* - This is the actual product that users will interact with. We want to have a functional website where people can register for an account and either create PorchFest instances or participate in one. Additionally, we want organizers to be able to schedule bands in their PorchFests, for which we plan on having Google Maps integration and a drag and drop UI.

Software Development Process

The project will undertake the iterative development model because we want to break down the software development process into smaller chunks. The Client wants to be part of the development process and provide feedback throughout the design, development, and testing process so that he can maintain the application after we hand it off. This method of software engineering will provide us the following benefits:

1. We can start off by creating the high-level design of the application before we actually begin to build the website and define user interface for the entire application. Afterwards, we can iteratively design and build a skeleton of the website, with feedback from the Client at each iteration. Similarly, we will update our backend system by building out the database schema first, then create a basic working implementation of the backend, and finally make the backend more robust and efficient.
2. When we present our sketches and wireframes of the website to the Client for his feedback, we are essentially asking him to imagine how his website will look like and work. So, through iterative development we can get reliable feedback.
3. Since we are building and improving the website at each milestone, we can track bugs and problems in the website early on. This will help us avoid wasting time tracking and fixing bugs at the very end.

Plan Outline (Principal Activities and Milestones)

Milestone 1 (March 15, 2017) - Presentation to client.

- Working, fully functional UI tested and approved by client
- High level backend database schema will be set up

Milestone 2 (April 12, 2017) - Presentation to client.

- Frontend and backend will be connected

- The initial scheduling algorithm will be implemented
- The first iteration of email functionality (CSV file) will be completed

Milestone 3 (May 10, 2017) - Presentation to client.

- Refinement of the initial scheduling algorithm to allow for more fine-tuning by hand
- Build email system (to send emails to different users)
    - Time permitting, add functionality for macros and email queueing
- Security Features (special entities filtering for html and sql injection, restricted endpoints, password salting and hashing)
- Preparing to transfer to client (transfer database and code to server hosting provider)

Milestone 4 (May 18, 2017) - Project Deadline.

- Project source code, documentation, and final project report should be delivered to the Client.

Visibility Plan

External - The Team will conduct weekly meetings with the Client every Thursday from 3:00pm to 4:00pm in Gates Hall unless otherwise arranged. Any other communication will be conducted through email. Notes will be taken at each meeting and posted in Slack to remind The Team about the Client's requirements and inputs.

Internal - The Team will conduct weekly meetings every Sunday from 5:00pm to 7:00pm in eHub. We are relying on three main collaboration tools: Slack for communication, GitHub for source code storage, and Google Docs for document sharing and editing. Within Slack, we are creating different groups based on the different parts of the project (i.e. one for Front End, one for Back End, etc.). Notes will be taken at each meeting, including concerns to be addressed to the Client, and posted in Slack to remind the Team about goals and tasks to be completed.

Business Considerations

When determining the feasibility of the PorchFest project, there are a few factors that must be taken into consideration. As a standalone website that does not rely on other programs, there are no trade secrets or other proprietary information that needs to be shared in the implementation of the system. The only sensitive information that the website stores on registered users will be login information, which should be no more than a provided email and password. As such, the appropriate security measures will be taken to ensure that the pages will not be attacked by malicious users.

The Team owns the copyright to the software created through the course of this project. Upon completion of the semester, the Team agrees to transfer copyright and complete ownership of the project to the Client. However, the Team reserves the right to be able to demo and discuss the project with anyone and be acknowledged as contributors to the system.

A situation may arise where a part of the project may be eligible for patenting. In this case, the Team owns the rights to any patent associated with the project. However, we anticipate that the likelihood of this happening is minimal to none.

Risk Analysis

Time Risks

One of the limitations of this project is that it must be completed by the end of the semester. It is not possible to get an extension, so it is essential that the project is completed by the deadline. This introduces the risk that we may not be able to implement all the requirements that the Client wants given the time constraint.

Resource Risks

Another one of the limitations of this project is that we do not have an unlimited budget. It may not be possible to get the best resources needed for completing parts of the system. Because of the budget constraint, we are considering using open-source software since it is available for free. However, open-source software is constantly changing and makes us rely heavily on its updates and integrity. Hardware problems may also arise and cause our system to crash and data to be lost. Also, since the project is web-based, we will have to account for different ways the user interface will be displayed across different internet browsers.

Functionality Risks

The biggest risk that we face in terms of functionality is not satisfying the requirements that the Client expects. However, this risk is the one we have the most flexibility to deal with, as we can compromise on functionality to ensure that time risks and resource risks are reduced. In this case, we have to be careful to ensure that we are still creating the website that the Client wants.

Risk Management/Backup Plan

In order to minimize these risks, we are ensuring that we follow good software engineering practices. We are using an iterative process to make sure that the Client is satisfied with the final product we deliver and so that we can make improvements at every step. The milestones that we discussed above will ensure that we complete the basic functionality required by the Client by the end of the semester. Weekly meetings will help us ensure we understand and satisfy the

Client's requirements. These check-ins will give the Client opportunities to provide feedback to us and ensure client satisfaction.

Conclusion

From the results of the feasibility study, the Team finds that the PorchFest Community Music Festival Organizer project is feasible in terms of technical difficulty, skill of team members, and time. Given the time constraint of one semester, the Team believes the scope of the project as defined above is manageable and that the Client's requirements can be satisfactorily fulfilled upon system completion. The Team members also possess the adequate skills to implement the system and are familiar with software that may be used in this project. The conclusion of the feasibility report is to go ahead with this software development project.