**Event Management Software Requirements**

**Functionality:**

1. The software will provide a calendar to aid organizers in managing events

1.1. Organizers are defined as

1.1.1. Privileged

1.1.2. Non-privileged

1. Privileged users will be able to add

2.1. events to the calendar

2.2. related tasks to an event

1. The software will track what tasks and events have been completed
2. Organizers will be able to manage participants and their contact information
3. Organizers will be able to create and manage committees
4. Software will let users receive / send emails from/to specific users / groups
5. The software will send automatic email reminders for important dates / events
6. Organizers will be able to manage a budget through the software
7. The software will generate reports on

9.1. budget

9.2. calendar of events and tasks

9.3. event details, i.e. name, location, duration, required tools/equipment, handler/manager/chair

9.4 task progress

1. The software will create forms

10.1. registration

10.1.1 name, address, phone, email, CC, charges, affiliation

10.2. budget

10.2.1. expense

10.2.1.1. Location

10.2.1.1.1. Hourly usage

10.2.1.1.2. Daily usage

10.2.1.1.3. All event usage

10.2.1.2. Food

10.2.1.3. Audio/video

10.2.2. Income

10.2.2.1. registration

10.2.2.2. fundraising

10.3. schedule

10.4 Pre-event

10.4.1. Can have multiple sub-events

10.4.2. Name. type, duration, time, day

10.5 During

10.5.1. Many sub-events

10.6 After-event

10.6.1. Can have multiple sub-events

1. Non-organizer participants will be able to view event / task details now belongs to 1.1.2

**Data:**

1) The software will store data for the

a) users

i) participants

1) everything in 10.1.1, plus list of sub-events selected, type of food asked for, type of badge(s) required

ii) organizers

b) events

c) emails

d) budgets

**Usability:**

1. The first version of software shall be utilized by an event organizer (priviledged user) through a local server . Later versions may be web-based and allow access to participants.
2. The system shall be implemented with a user-friendly GUI.
3. The GUI layout shall be as organized and as simple as possible; it should not be difficult for the user to navigate through this software on their own.
4. The GUI shall display

4.1. Forms

4.2.1. Registration

4.2.1.1. Text areas to enter participants’ name, …..

4.2.1.2. Lists/menus for pre and post sub-events

4.2.1.3. Payment info

4.2.1.3.1. CC info

4.2.1.3.2. Paypal info

4.2.2. Budget entry

4.2.3. Event scheduling

4.2. Reports

4.2.1. Current budget standing

4.2.2. Current task completions status

4.3. Schedule

4.3.1. The whole

4.3.1.1. With details

4.3.1.2. Without details

4.3.2. Only pre-event

4.3.3. Only post-event

5. The software shall allow the user to preview and print different forms and reports to obtain a hard copy of the system’s current information.

6. Any error messages shall be managed through comprehensible text message pop-ups instructing the user on what to do about them.

7. Entered information shall be editable in the system through a button or an editable text field; the system shall allow correction in previous mistakes.

8. It should not be easy to mishandle the program; the system should not allow the user to enter wrong information.

**Reliability:**

1. If you are an events organizer, using this program, you would have to rely on

1.1. The program not to crash

1.2. Your input data would be preserved.

1.3. Having a dynamic calendar

1.4. Updating as soon as changes have been made.

There is nothing more frustrating than making a lot of changes and having to do them again because you forgot to save.

2. System must

2.1. Update data constantly and do so

2.1.1. Consistently

3. Will be archived through tables of data a proven stable and reliable method by

3.1. Updating

3.2. Accessing

4. the larger the program becomes the greater chance for

4.1. Issues relating to reliability and thus

4.1.1. All precautions will be taken to ensure that data is

4.1.1.1. Sent

4.1.1.2. Received

4.2. Messy programming structure and thus

4.1.2. Ensure that program is

4.1.2.1. Documented and commented

4.1.2.2. Properly indented

4.1.2.3. Deprived of unused variables

4.1.2.4. Not hard coded if not needed

**Performance:**

1. Regardless of size of data coming and going, what should be instantaneous is

1.1. Execution speed

1.2. Response time

1.3. Throughput

2. A large amount of data will be flowing through the system:

2.1. Log in: Users will be allowed to

2.1.1. Log in to their accounts, thus sending

2.1.1.1. user name information.

2.1.1.2. password information.

2.2. Use data to

2.2.1. confirm user log in information.

2.3. Participant: To create a participant

2.3.1. Data will be sent through the system

2.3.1.1. With the participant information, whether this includes

2.3.1.1.1. Name

2.3.1.1.2. Phone number

2.3.1.1.3. Email

2.3.1.1.4. What committee the participant belongs.

2.4. Calendar: Users will be able to create events with various different schedules. In this manner, data will be sent pertaining to each event.

2.5. Email system: Emails will be

2.5.1. Viewed\*

2.5.2. Sent

2.5.3. Received\*

2.5.5. Using data

2.6. Report: To produce a report, data will have to be sent through the system to put together a report.

2.7. Budget: To create a budget, data will have to be collected from the user.

3. Data will be sent and received constantly to be able to navigate through the system.

4. A participant cannot

4.1. Be created without sending information pertaining to the user.

4.2. Log in to the system, without information being sent to confirm that the participant exists.

5. A calendar cannot be accessed without receiving information of

5.1. Existing events

5.2. Dates available.

6. The email system cannot be accessed without sending and receiving data of existing emails and email addresses.

7. To create a budget, data will have to be received, and in the same manner, will be sent to display it. The same occurs for the report.

8. The GUI of the entire system will aid in constantly sending, but also receiving data.

**Supportability:**

1. Maintainability:

1.1. No network monitoring due to client sided software.

1.2. Exceptions for recovery without need of technical staff.

1.3. Documentation when needed for organization.

1.4. Upgrading the software will not cause errors of incompatibility.

1.5. Encryption for security over time.

2. Compatibility:

2.1. The software will run of the Java platform.

2.2. The software will be cross-platform, meaning it will run on multiple operating systems.

3. Ease of installation:

3.1. The program will be run from a jar file.

3.2. No installation will be necessary.

3.3. In application update feature(No need to go to website for update).

4. Testability:

4.1. Event logging for failure information.

5. Configurability:

5.1. Enable / disable certain features

5.1.1. Event logging

5.1.2. Profile view status

5.1.3. Settings