Rose-Hulman Institute of Technology

University of Wisconsin–Madison | Department of Computer Sciences
Human-Computer Interaction Laboratory

Milestone 1

Trey Cahill Chris Gropp Samad Jawaid Kevin Risden

November 8, 2011

Contents

1 Executive Summary

This document's purpose is to detail the participant scheduling system proposed by the of Wisconsin-Madison. It is the first document describing this project, and contains information on the lab's needs, proposed features, and how the project will affect the lab. The project exists because the lab wishes to unify their schedule information and provide a simple, intuitive interface for prospective participants to sign up for experiments.

2 Introduction

The Human-Computer Interaction Lab at the University of Wisconsin-Madison wants a web-based system to better manage the scheduling of participants for their studies. These studies range from one-on-one experiments to group interactions, and many of them involve the robot used by the lab. Currently, each researcher arranges studies independently via email and is responsible for scheduling rooms, avoiding conflicts, and notifying participants of changes; unifying this information onto one system simplifies all of these tasks. To the client, the most important benefit of a unified system is the ability for participants to easily browse all available experiments, which is not possible over email. However, a variety of other functionality should be integrated into this utility to take advantage of the unity of information; most notable is recognizing room conflicts when scheduling studies, since the lab has only one robot and it cannot be moved.[1]

Project information will be documented as follows: Milestone 1 provides an overview of the project, from client background to key features and requirements. Milestone 2 covers the behaviour of the system, including use cases and data flow diagrams. Milestone 3 details constraints, back-end requirements, and elaborates upon the user interface. Testing and maintenance information can be found in Milestone 4. Milestone 5 will include usability data and interface re-design related to such data.

3 Client Background

The client is the Human-Computer Interaction Lab at the University of Wisconsin-Madison. Their research focus is the on the way humans perceive computers, and how this perception influences their actions. The main goal is to learn about this interaction through making hypotheses, experimenting, analysing the data, and then publishing papers on the results. They draw the participants for their experiments from a wide range of people, usually ranging from 18-65 years of age and from diverse technical backgrounds. As such, any system they use must be designed for all levels of technical competency.

4 Current System

Each researcher has their own method of handling participant scheduling. For most, the current system is to have the participants email the individual researcher and then that researcher records the time slot in some sort of Excel spreadsheet. Other researchers have tried Google Calendar appointment slots; while this is a better system, not everyone uses it and the client believes it is too complex for most participants and some researchers. Addressing the lack of unified data and superfluous effort on the part of the participants is the primary goal of the project.

5 User/Stakeholder Description

5.1 User/Stakeholder Profiles

The stakeholders for our system fall into two distinct categories, participants and the people in the research lab. These categories then fall into a few subsections that will be described in detail. The participants themselves fall into the categories of technical and non-technical users.

The technical users will desire a rich user experience. Furthermore these users will want the system to be more robust and provide a large number of features to make them feel comfortable. Success to them will include a system that they can customize and work into their already technical work flow.

The non-technical users may or may not have had experience with any sort of web interface. This provides an extra challenge for our solution because we must make the interface intuitive enough that people who are not comfortable with computers can use. A success for the non-technical user would be the ability for them to use the software in a way that does not interfere with their normal daily work flow and they can overcome any hesitancy they have with using computers to schedule their participation in an experiment.

The other class of stakeholders involves people in the research lab. The lab consists of undergraduate students, graduate students, and professors, and their use of the proposed system is varied. Furthermore, since the research lab will be controlling and maintaining the system once in place there is a subclass of the people in the research lab who act as both administrators and maintenance technicians.

Since the research lab is involved with Human-Computer Interaction, many of the people in the lab will be experienced with computers. However, from the client we learned that a few outside researchers who may use the system we develop may not have a large degree of technical knowledge. Therefore we must design the system in a way that permits researchers, students, and professors to setup experiments and look at participant lists in an effective manner. Since these people have been using a rudimentary system before, we must make the new system capable of integrating into their work flow without introducing any more work than done previously. The success for the researchers, students, and professors comes from being able to schedule participants in a way that works effectively and simply from their point of view.

The administrators and maintenance people from the research lab we know are all technically proficient users. This means the system must be simple to maintain and possible expand at a later date if needed. The key responsibilities for the administrator include being able to add and remove other users from the system and ensure that the system is functioning correctly. The maintenance people must ensure that the system is stable and in a working condition when required. The success for the administrator can be defined as having an intuitive management interface that provides them the flexibility they need to perform their duties. For the maintenance people, success can be defined as having a system that is stable and requires little work to keep it operational even over a large number of years.

5.2 User Environment

The user environment is basically the same between the classes of users listed above. The working environment consists of mainly just one user at a time on their own machine joining an experiment. This will stay constant but there may be many users at once on separate computers for any number of experiments. The task length should be fairly minimal and involves them finding an experiment they would like to participant in and then signing up. The entire length of the activity should only take a few minutes as to not inconvenience the user. The only environmental constraint that will be imposed on the user is that they must use a web browser of some sort. This means that the user will be required to have some device that has internet access. The system today involves the user going to a website to find the contact email and then sending the contact an email specifying that they would like to participant in an experiment. The users may have some sort of calendar system that they use so we may have to integrate with it but it is not a hard requirement but instead a possible feature.

5.3 Key Needs

5.3.1 What is the need?

The problem of...	listing and browsing all currently running experiments
Affects...	the possible participants of the experiments and the researcher
And results in...	less people signing up for the available experiments
Benefits of a solution...	would permit possible participants to see currently running experiments in one place and have easy access to them all in a standard format

5.3.2 How is it solved now?

There is no current solution to the problem.

5.3.3 What is a possible solution?

The possible solution involves having a single page where all the current experiments being held are listed in a format that is easy to browse.

5.3.4 What is the need?

The problem of...	signing up for an experiment is a complicated nonstandardized process
Affects...	the possible participants of the experiments and the researchers collecting the data
And results in...	wasted time and effort signing up and compiling lists of participants for each experiment
Benefits of a solution...	allow participants and researchers to get used to a standardized format of signing up and collecting experiment participants that would be quicker than manually creating the lists

5.3.5 How is it solved now?

The current solution is to manually email a researcher that you want to do an experiment with and the researcher must then respond and compile a list of people and times which takes a large amount of time

5.3.6 What is a possible solution?

The solution would be to have a sign up sheet that looks the same for each experiment with a few optional fields that would then be placed in a database that can be queried later to easily determine the list of participants for each experiment

5.4 Alternatives and Competition

5.4.1 Google Calendar Appointment Slots

The issues the client mentioned about Google Calendar appointment slots was that there was no way to list all experiments in an easy format and no way to do multiple experiments at once. Furthermore, it would

be a hassle to have to compile all the appointment slots into a list of participants for the experiment. The process involved in using Google Calendar appointment slots was also very manual as for each experiment the researcher would need to go through and outline all available times. The approach also did not work well for participants since they would need to navigate to each calendar separately to look at all available times for an experiment.

6 Product Overview

This section provides a high-level view of the product capabilities, interfaces to other applications, and system configurations.

6.1 Product perspective

The participant scheduling system will be a new product. It will be used to schedule experiments and participants in the Human-Computer Interaction Lab at the University of Wisconsin-Madison. The product is independent and totally self-contained, besides a few external software packages; it is not a component of a larger system.

6.2 Elevator Statement

For the researchers in the Human-Computer Interaction Lab at the University of Wisconsin-Madison who currently schedule experiments and participants with rudimentary tools such as pencil and paper, email, or Google Calendar, the participant scheduling system will be a web application that will streamline the lab's scheduling process. Unlike current solutions, this application will be the same for every researcher, so it will also be easier for participants to be a part of multiple experiments.

6.3 Summary of Capabilities

Here are the major benefits and features the product will provide.

Customer Benefit	Supporting Feature
List of participants for an experiment	Reports
Room availability (avoid conflicts)	Overall lab schedule
Simple sign up	Intuitive user interface
Track all experiments	Experiments manager
Access from anywhere at any time	Web application

6.4 Assumptions and Dependencies

- The participant scheduling system will be a web application.
- The server has the necessary operating system and software.
- There is no integration with any other system.
- There is no import of existing data.

6.5 Rough Estimate of the Cost

There is no monetary cost for this project, because the software development, as part of a college class, is free. Similarly, all software used is open-source. Furthermore, the client will be provided with free servers through the University of Wisconsin-Madison for the finished product. The client will perform maintenance and management on their own.

7 Features

ID	Feature	Status	Priority	Effort	Risk	Stability	Target Release	Reason
1	Browse Experiment	Approved	Critical	Medium	Medium	Medium	1st	Lets experiments be advertised better and to display the experiments
2	Persistent Experiment Storage	Approved	Critical	Medium	Low	High	1st	Store experiment for the data to be web based.
3	Levels of Authentication	Approved	Useful	Medium	High	Medium	3rd	Have levels of administrators, workers and participants in order to control privacy issues and other sensitive data
4	Participant Schedule Experiment	Approved	Critical	Medium	Medium	Medium	1st	Participant can schedule experiment slot
5	Filter Experiments	Approved	Useful	Medium	Low	High	2nd	Filter the experiments when browsing according to Time, Date, Payment, etc.
6	Experiment Participants	Approved	Important	Low	Low	High	2nd	View all of the participants by administrators and workers only of individual experiments
7	Cancel Experiment Appointment	Approved	Useful	Medium	Medium	Medium	3rd	Cancel participant scheduled appointment
8	Add Experiment	Approved	Important	Medium	Medium	Medium	2nd	Add experiment from administrators view
9	Modify Experiment	Approved	Important	Medium	Medium	Medium	2nd	Modify or Edit experiment from administrators view
10	Notify Participant when Creating Appointment	Approved	Useful	Medium	Low	High	4th	Send an email reminding participants of participation in an experiment
11	Notify Participant Appointment Reminder	Approved	Useful	Medium	Low	High	4th	Send an email or text reminding participants for their experiments
12	Notify Participant Appointment Cancellation Reminder	Approved	Useful	Medium	Low	High	4th	Send an email or text reminding/telling participants of cancellation of their experiments
13	Export Experiment Participant List	Approved	Useful	High	Low	High	4th	Reports on experiments scheduled with an option for Individual experiments reports
14	All Experiments Calendar	Approved	Useful	Medium	Low	Low	4th	Have an overall schedule viewer
15	Remove Experiments	Approved	Important	Low	Low	High	4th	Allow for workers or administrators to remove schedules
16	Tracking of Consent Payment Forms	Rejected	Useful	Medium	Low	Medium	N/A	Allow for workers to check off participants when filling out consent/payment forms
17	User Report	Rejected	Useful	Medium	Low	Medium	N/A	Allow participants to have a report on new experiments
18	Accounts	Approved	Critical	High	Low	Medium	N/A	Accounts for participant
19	Prevent Scheduling Conflicts (Participant)	Approved	Useful	High	Low	Medium	N/A	Prevent participants from scheduling 2 experiments at the same time
20	Prevent Scheduling Conflicts (Administrator)	Approved	Useful	High	Low	Medium	N/A	Prevent 2 rooms from being scheduled at the same time
21	Install Scripts	Proposed	Useful	High	Low	Low	TBD	Install scripts for installation
22	Documentation for Maintenance and User	Approved	Useful	High	High	Low	Ongoing	Documentation

The **Browse Experiments** feature and **Persistent Experiment Storage** both had a Priority of Critical since they both must be implemented for even a very basic version of the scheduling System. The effort on both was a medium as with a team of two, there would be a manageable amount of work. **Browse Experiments** has a stability of medium since it is up for change upon the client seeing the UI. **Persistent Experiment Storage** has a stability of high since once implemented has little chance of being changed.

Participant Schedule Experiment has a priority of critical since the participants must be able to sign themselves up for an experiment for the project to be successful. Again, the effort is medium since with two people the work would be manageable. The risk is high on this feature, since the success of the project has a dependence on the feature. The stability would be medium since the steps are unlikely to change, but the UI could easily change.

Levels of Authentication is a useful priority because it would not be necessary for there to be an actual Administrator since the users trust each other, but this would be a nice feature. The effort and stability are medium since the feature may change some, but only smaller parts of the feature, while still being a very manageable task.

Filter Experiments has a priority of useful, since it might only apply to users in certain situations. **Filer Experiments** and **Experiment Participants** have a low risk, since the project does not depend on their success. They both also have a stability of high, since changes are unlikely to happen. The effort on **Filter Experiments** is medium, since there are some areas of the feature, such as what to filter by, that have not been established. The effort on **Experiment Participants** is low since a simple SQL query will do most of the job.

Cancel Experiment Appointment, **Modify Experiment**, and **Add Experiment** get and effort of Medium, since most deal with SQL and some logic on the back end. They also have a risk of medium, since a mistake while implementing these features could create a difficult to find bug elsewhere. The stability is medium, since parts of the database could change slightly.

Notify Participant When Creating Appointment, **Notify Participant Appointment Reminder**, and **Notify Participant Appointment Cancellation Reminder** all have a priority of useful since they would be nice to have, but are not vital to the projects success. They all have an effort of medium since they involve a persons email, but could be reduced to low, depending of the framework used. Their risk is low, since a failure here creates no problems else where in the project, nor does a mistake spread else where in the project. The stability is high on these since they are unlikely to change.

Export Experiment Participant List is a useful feature, that has a high effort due to formatting of the report. The risk is low though, since the feature is not critical in the release of the product. Stability is high due to the feature being very specific.

An **All Experiments Calendar** would be useful for the future participants. The effort is medium because it would be an extension of the **Browse Experiments** feature.

Remove Experiments has a priority of Important, since, although rare, experiments may be cancelled. The effort is low since most of it will be taken care with an SQL query. Also, a stability of high is given because of how specific the feature is described.

Accounts, **Prevent Scheduling Confilicts for the Participant**, and **Prevent Scheduling Conflicts for the Administrator** all have a priority of useful, except for **Accounts** which Critical, since the other two features mentioned rely upon the **Accounts** along with cancellation of the experiment slots. The effort is high for all the features due to the logic needed when implementing the features.

Tracking of Constent Payment Forms and **User Report** have all been rejected, since the client does not need theses features.

Documentation and **Install scripts** are both useful priority. The effort will be high, since there is complexity associated with the Install scripts and Documentation is difficult to keep up to date. The stability would be low since the definition could change.

Items 1, 8, 15, 18, and 21 will be assigned to Kevin Risden. Items 3, 4, 10, 11, and 12 will be assigned to Samad Jawid. Items 6, 9,14 and 20 will be assigned to Chris Gropp. Items 2, 5, 7, 13 and 19 will be assigned to Trey Cahill. The entire team will work on item 22.

8 Solution Constraints

Source	Constraint	Rationale
Systems Mandate	Must be able to be ran on Red Hat Enterprise Linux Server 6.1.	This is the operating system that the client currently uses.
Technology Mandate	Must use PHP or Python as the programming languages	These are the languages supported by the client
Databases Mandate	Must use MySQL or PostgreSQL database management systems	These are the database management systems supported by the client.
Time	The time constraint on the project is the end of Second Term (Rose Hulman time)	At this point the group is reduced to 1 person.
Equipment Budget	No new equipment can be bought for the project	The software will be place on an existing server and we have no budget.
Privacy	Participants must not be able to see who else is participating in a project.	Privacy is key for the experiments.

9 References

[1] University of Wisconsin-Madison. Human-Computer Interaction Laboratory, 2010.

10 Appendix

11 Glossary

PostgreSQL open source object-relational database system. 10

SQL Structured Query Language is a programming language designed for managing data in relational database management systems. 8

Index