Rose-Hulman Institute of Technology

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

Milestone 3

Trey Cahill Chris Gropp Samad Jawaid Kevin Risden

October 17, 2011

Contents

1 Executive Summary **2**

2 Introduction **2**

3 Client Background **2**

4 Current System **2**

5 Product Overview **2**
5.1 Product perspective . . . 3
5.2 Elevator Statement . . . 3
5.3 Summary of Capabilities . . . 3
5.4 Assumptions and Dependencies . . . 3
5.5 Rough Estimate of the Cost . . . 3

6 Requirements **3**
6.1 Usability . . . 3
6.2 Performance . . . 4
6.3 Reliability . . . 4
6.4 Supportability . . . 4
6.5 Hardware and Software Interfaces . . . 5
6.5.1 Software Interfaces . . . 5
6.5.2 Hardware Interfaces . . . 5
6.6 Documentation . . . 5
6.7 Installation . . . 5
6.8 Legal/Licensing . . . 5

7 Solution Constraints **6**

8 User Interfaces **6**
8.1 Browse Experiments Screen . . . 6
8.2 Experiments Screen . . . 7
8.3 New User Sign Up Screen . . . 7
8.4 Log In Screen . . . 8
8.5 Researcher Experiment Screen . . . 9
8.6 Create Experiment Screen . . . 9
8.7 Modify or Delete Experiment Screen . . . 10

9 References **10**

10 Appendix **10**

11 Glossary **10**

Index **10**

1 Executive Summary

This document's purpose is to detail the participant scheduling system proposed by the Human-Computer Interaction Lab of Wisconsin-Madison. It is the third document describing this project, and specifies requirements not captured by use cases, as well as a detailed user interface. 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 Product Overview

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

5.1 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.

5.2 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.

5.3 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

5.4 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.

5.5 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.

6 Requirements

6.1 Usability

- For anyone who has used a web application before, there should be zero training time required before being able use this system.

- For anyone who has never used a web application before, there should be minimal training time (a maximum of ten minutes) required before being able to use this system thanks to the on-screen help in the form of a walk-through.
- No more than half of the time required to actually visit, email, or call a researcher to schedule an appointment will be needed to use this system.
- The new system shall be judged by 99% of the user community to be at least as useful as the existing system and by 90% of the user community to be at least as useful as competing state of the art system.
- On-screen help will be an accessible option for struggling users.
- There will be no drastic interface changes from existing web applications, so there will be no more than a slight learning curve for the user.

6.2 Performance

- The average response time for the next screen to appear after the user has selected an option that will change the screen is thirty milliseconds and the maximum is sixty milliseconds.
- A minimum of one page load per second will be supported.
- Each individual session will support one concurrent user, while the overall system will support up to twenty users using the web application in parallel.

6.3 Reliability

- The system will be up 99.99% of the time (less than one hour of downtime per year), independent of the maintenance crew taking it down up to twice a year for eight to ten hours at a time.
- On average, failures will not occur within three months of one another.
- Due to innovative remote diagnostics, updates, and repairs, 90% of all system failures will be able to be repaired within five minutes and 99.9% of all failures will be able to be repaired within one hour, depending on the maintenance crew.
- All appointments will be accurate to the nearest minute 100% of the time.
- Appointments will be successfully scheduled 99.99% of the time, and in the case a scheduling fails, the user will be able to try again 100% of the time.
- There will be no more than one bug per thousand lines of code.
- There will not be any critical or significant bugs. However, there will be up to one minor bug per thousand lines of code. Minor is defined as not affecting the usability of the system or scheduling data.

6.4 Supportability

While the development team will not be involved with supporting the system once it is handed over to the customer, the system will still have some support. The team will create documentation throughout the development process. Also, the customer will receive all source code when the system is handed over. In order to facilitate supportability for the customer, the system will be written in Python and use PostgreSQL, both with which the current customer is familiar. With the code in a familiar language and the documentation, the customer should be able to fully support the system by themselves.

6.5 Hardware and Software Interfaces

6.5.1 Software Interfaces

The software required for the system will be an operating system of Red Hat Enterprise Linux Server 6.1. Also, the software will be written in Python and use PostgreSQL database management system.

6.5.2 Hardware Interfaces

There are no hardware specifications that we must conform to since we are writing a hardware independent web application.

6.6 Documentation

As part of the usability requirements, the system we provide should be intuitive enough that the end user should have no difficulties navigating the system. However, the client prefers that we provide tool-tip type documentation for the entry forms. This documentation should provide clear, concise direction for the user when entering their information. On the back-end, similar tool-tip documentation will provide the administrator with the guidance to complete their tasks such as modifying experiments or researchers. Additionally, documentation for the installation will be provided in the form of step-by-step instructions as detailed in the installation requirements section. Finally, developer documentation will be in the source code. There will be documentation relating to the use of each class and public method.

6.7 Installation

Since our client is technically proficient, they prefer either step-by-step installation instructions or an install script. Installation instructions are typically more robust than an install script so we will be providing installation instructions with our solution. The installation instructions must be detailed enough that our client can install the web application on the machine provided by them. We have replicated to the best of our abilities the platform on which the web application will be deployed to be able to provide proper installation instructions.

6.8 Legal/Licensing

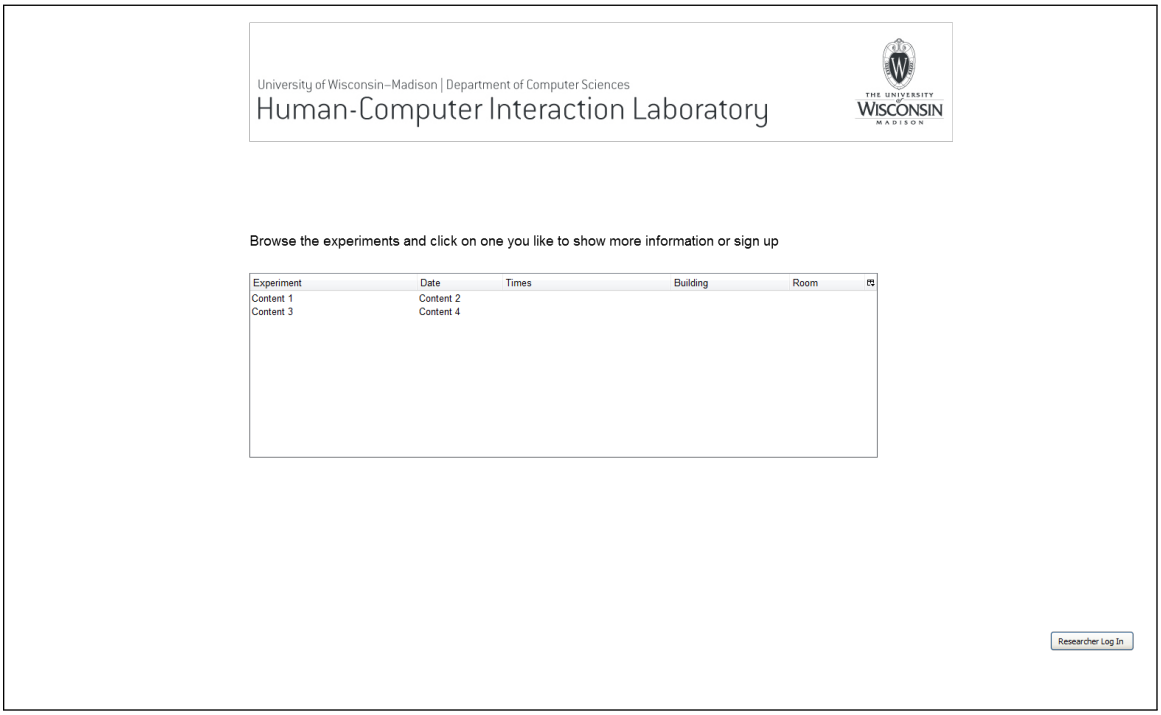
The client has specified that there are no legal or licensing requirements for this project. The code will be released under an open source license that is yet to be determined. The code is hosted publicly on GitHub and is freely available. The users will have to agree to the terms of the client when creating an account but these have not been finalized yet.

7 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.

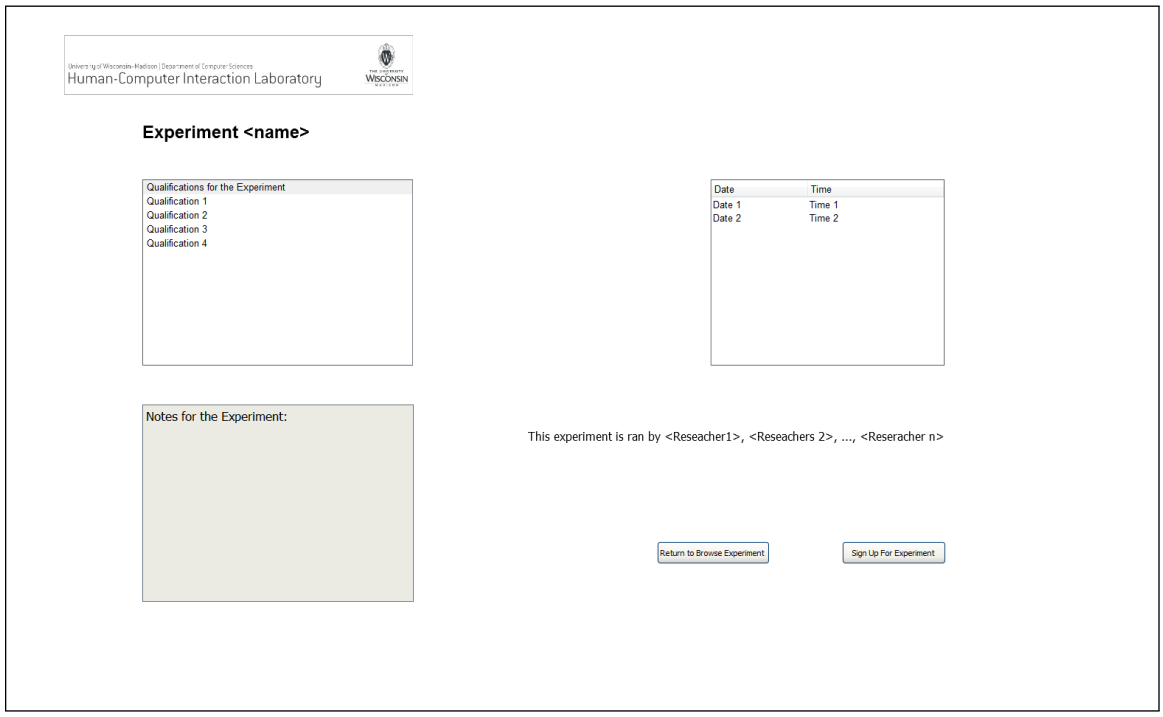
8 User Interfaces

8.1 Browse Experiments Screen



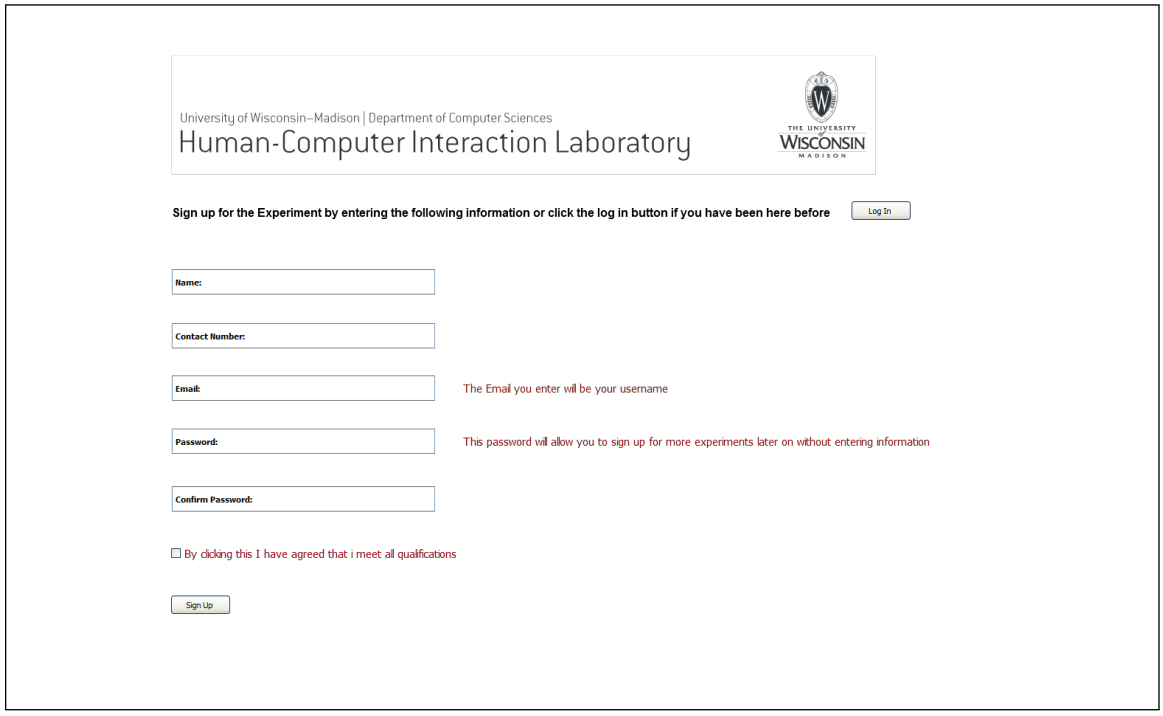
This screen lets a future participant browse all the experiments that have not be executed. If a future participant clicks on an experiment the user will be sent to the Experiments Screen.

8.2 Experiments Screen



This screen shows the specifics of an particular experiment and allows a future participant to sign up for the experiment or return to browse more experiments.

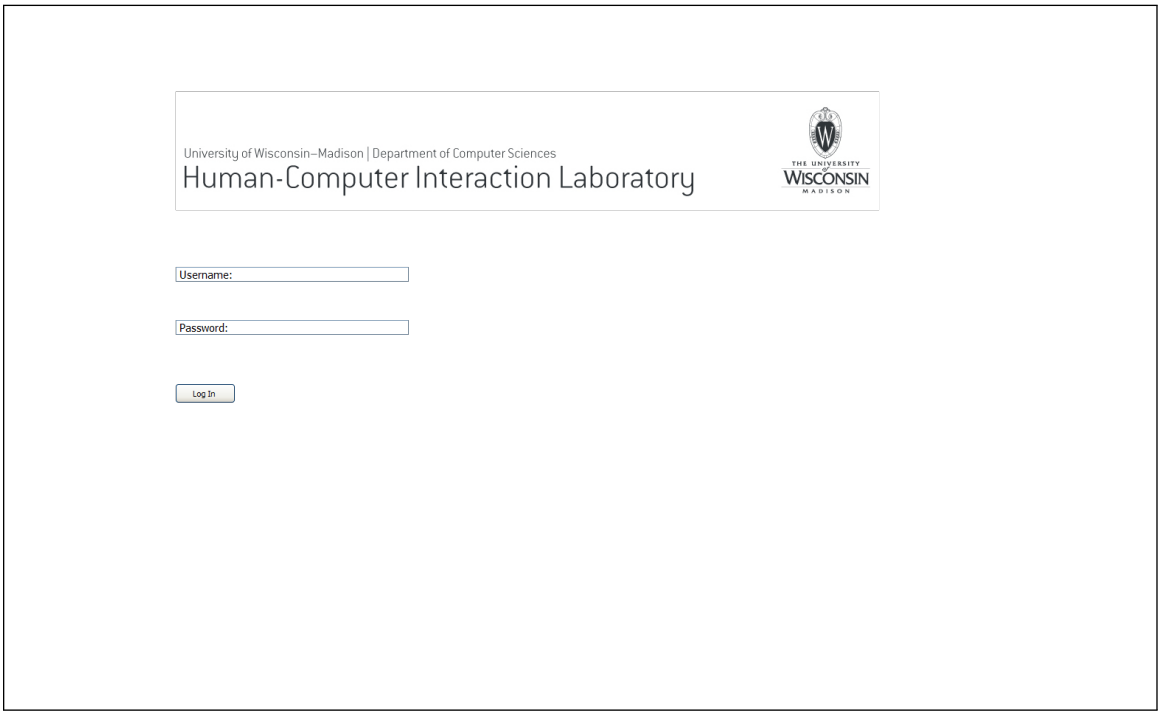
8.3 New User Sign Up Screen



This screen allows a participant who has not participated in an experiment before to sign up and create an

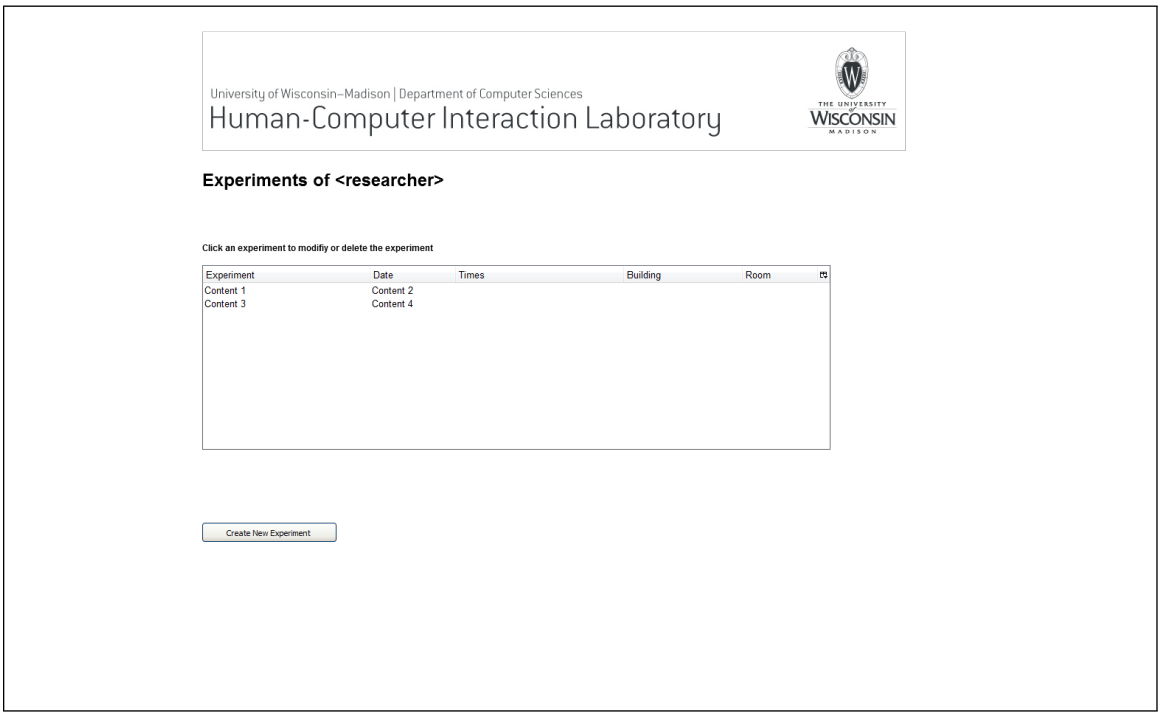
account to sign up for an experiment. If the user has participated before, the user can click the log in to log in to their account.

8.4 Log In Screen



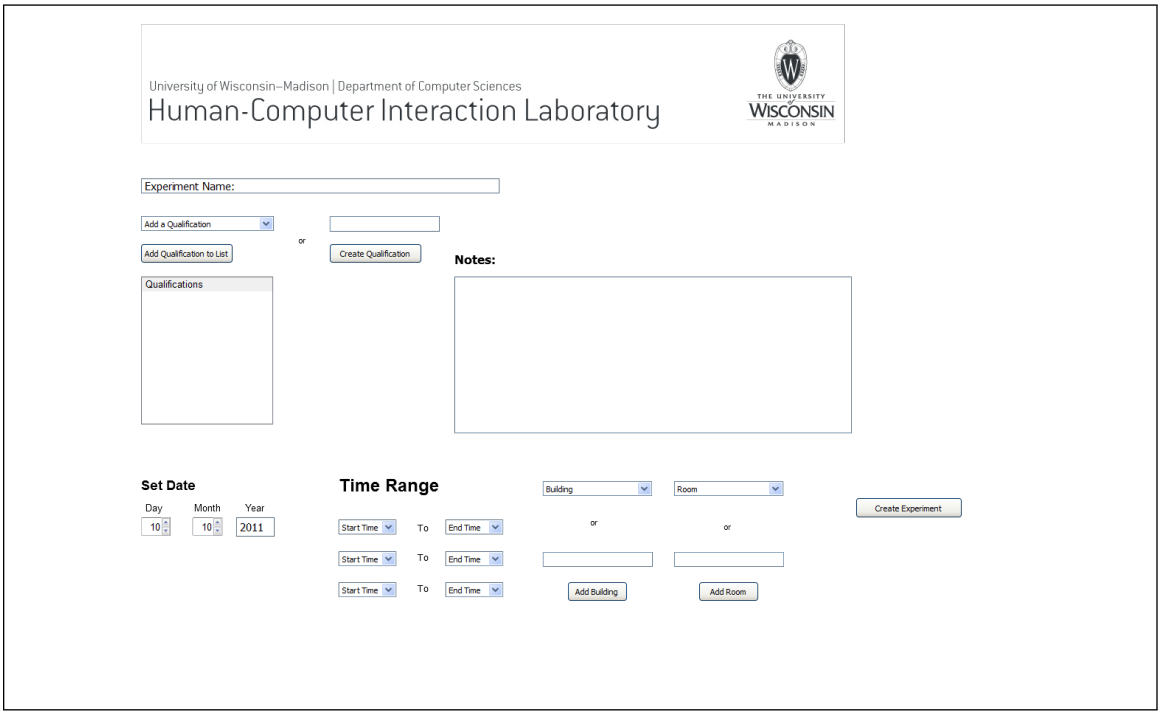
This screen allows a participant or researcher to Log in to their accounts.

8.5 Researcher Experiment Screen



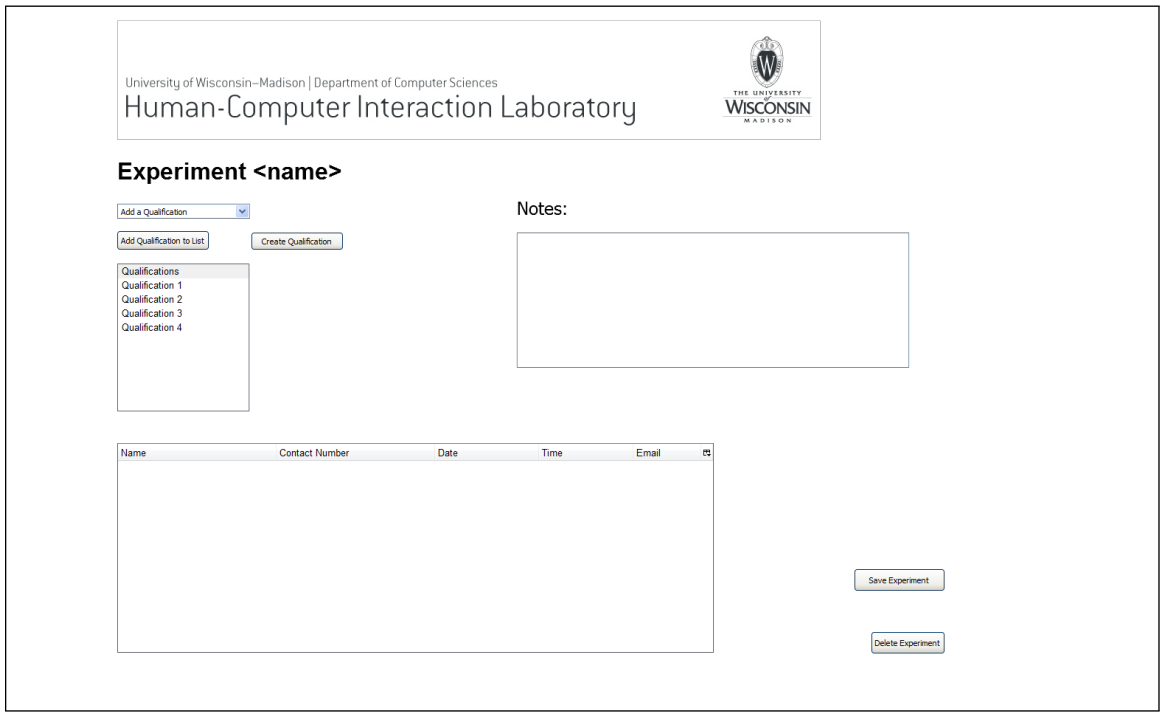
This screen is what a researcher will see when they log in to their accounts.

8.6 Create Experiment Screen



This screen allows a researcher to create a new experiment.

8.7 Modify or Delete Experiment Screen



This screen allows a researcher to modify or delete an experiment.

9 References

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

10 Appendix

11 Glossary

PostgreSQL open source object-relational database system. 4–6

Index

Human-Computer Interaction Lab, 2