



ROSE-HULMAN INSTITUTE OF TECHNOLOGY

University of Wisconsin–Madison | Department of Computer Sciences

Human-Computer Interaction Laboratory



## MILESTONE 2

Trey Cahill   Chris Gropp   Samad Jawaid   Kevin Ridsen

November 8, 2011

# Contents

<b>1</b>	<b>Executive Summary</b>	<b>2</b>
<b>2</b>	<b>Introduction</b>	<b>2</b>
<b>3</b>	<b>Client Background</b>	<b>2</b>
<b>4</b>	<b>Current System</b>	<b>2</b>
<b>5</b>	<b>Product Overview</b>	<b>2</b>
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
<b>6</b>	<b>Features</b>	<b>4</b>
<b>7</b>	<b>Use Cases</b>	<b>6</b>
7.1	General Behaviour . . . . .	6
7.2	Authentication Use Cases . . . . .	6
7.3	Appointments . . . . .	9
7.4	Experiment Management . . . . .	10
7.5	Reports . . . . .	12
<b>8</b>	<b>Use Cases Features Mapping</b>	<b>15</b>
<b>9</b>	<b>Data Flow Diagrams</b>	<b>16</b>
9.1	Current System Context Diagram . . . . .	16
9.2	Current System Level 0 Data Flow Diagram . . . . .	16
9.3	Proposed System Context Diagram . . . . .	16
9.4	Proposed System Level 0 Data Flow Diagram . . . . .	17
9.5	Proposed System Level 1 Data Flow Diagram for Sign up . . . . .	17
<b>10</b>	<b>Storyboard</b>	<b>18</b>
<b>11</b>	<b>References</b>	<b>22</b>
<b>12</b>	<b>Appendix</b>	<b>22</b>
<b>13</b>	<b>Glossary</b>	<b>22</b>
	<b>Index</b>	<b>22</b>

# 1 Executive Summary

This document's purpose is to detail the participant scheduling system proposed by the of Wisconsin-Madison. It is the second document describing this project, and details project design through use cases, data flow diagrams, and storyboards. 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 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. **Filter 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 an 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 Conflicts 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 Constant Payment Forms** and **User Report** have all been rejected, since the client does not need these 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 Ridsen. 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.

## 7 Use Cases

### 7.1 General Behaviour

Every page on the website possesses a “login/logout/create account” button. If the user is logged in, follow use case “Logout”. Otherwise, follow use case “Login”. In either case, unless noted otherwise, upon completion of that use case, the system will return to the page the button was clicked from. If that page had user-entered fields, they will be in the same state they were when the user clicked the button. A common exception is when a researcher or admin account logs out of a researcher or admin page, in which case they will be returned to the homepage; unauthenticated users cannot view or use researcher-only features.

Every page also possesses a button to return to the system homepage. This will exit any use case they are currently following and discard any temporarily stored information related to that use case, such as entered fields or selected options.

Whenever a table is displayed, there is some concern related to its size; however, due to the low expected number of experiments and participants per experiment, any handling of large tables will be outsourced to the browser (scroll bars being most common). Because the whole table will be loaded at once, browser search functionality is also sufficient to handle most searching needs; tables that can be otherwise sorted, filtered, or searched will be specifically noted.

### 7.2 Authentication Use Cases

#### 1. Name: Login

- (a) Brief Description: User logs in.
- (b) Actors: User
- (c) Basic Flow:
  - i. User clicks the “login/logout/create account” button from any page.
  - ii. User prompted for Email and Password via text boxes.
  - iii. The system sends their login information to the database. [A2] [A3] [A4]
  - iv. System displays a message confirming successful login. The user is now logged in.
  - v. After 10 seconds or when the user clicks a link to do so immediately, the user is navigated out of the login use case as specified in General Behavior.
- (d) Alternate Flows:
  - A1 User navigates elsewhere on the website, through their browser or the “home” button. Unless the page they attempt to visit requires authentication, this simply drops them out of the use case.
  - A2 User entered an email that the database did not recognize. Run use case “Account Creation”.
  - A3 User entered an email recognized by the database but not the password associated with it. Return to email/password entry, displaying the message “Incorrect password, please retry.”

A4 System fails to connect to database. Display the message “Database unavailable; we are sorry for the inconvenience. Please try again later.” Then return the user to the page they entered the use case from.

(e) Pre-conditions:

- i. System is functional.
- ii. User is not logged in.
- iii. User has already created an account.

(f) Post-conditions:

- i. User is logged in, or cancelled login process.

(g) Special Requirements:

- i. N/A

(h) Feature Mapping:

- i. Levels of Authentication
- ii. Accounts

## 2. Name: Logout

(a) Brief Description: User logs out.

(b) Actors: User

(c) Basic Flow:

- i. User clicks the “login/logout/create account” button from any page.
- ii. System displays a message confirming successful logout. The user is now logged out.
- iii. After 10 seconds or when the user clicks a link to do so immediately, the user is navigated out of the logout use case as specified in General Behaviour.

(d) Alternate Flows:

A1 User entered this use case from a page not available while logged out (appointment confirmation or researcher interfaces). The system will return them to the homepage unless otherwise specified.

(e) Pre-conditions:

- i. System is functional.
- ii. User is logged in.

(f) Post-conditions:

- i. User is logged out.

(g) Special Requirements:

- i. N/A

(h) Feature Mapping:

- i. Levels of Authentication
- ii. Accounts

## 3. Name: Create Account

(a) Brief Description: User creates an account.



(b) Actors: User

(c) Basic Flow:

- i. User clicks the “login/logout/create account” button from any page.
- ii. The system navigates the user to the login page.
- iii. User clicks the “create account” button from the login page.
- iv. The system navigates the user to the account creation page.
- v. User prompted for Email, Name, Phone, Password, and Confirm Password via text boxes.
- vi. User clicks “Submit” button. [A2] [A3] [A4]
- vii. The system sends the entered information to the database. [A5] [A6]
- viii. The system sends an email to the entered email address. [A7]
- ix. System displays a message confirming successful account creation. The user is now logged in.
- x. After 10 seconds or when the user clicks a link to do so immediately, the user is navigated out of the create account use case as specified in General Behaviour.

(d) Alternate Flows:

- A1 User navigates elsewhere on the website, through their browser or the “home” button. Unless the page they attempt to visit requires authentication, this simply drops them out of the use case.
- A2 User clicked “Submit” before filling in all fields on the account creation page. System does not leave the page, and displays the message “All fields must be completed to continue.”
- A3 User entered a user name, email, or password that does not meet requirements. See “special requirements”. System does not leave the page, and displays the message “Please check guidelines for account creation, one or more fields were not acceptable.”
- A4 User entered different text in the Password and Confirm Password fields on the account creation page. System does not leave the page, and displays the message “Password confirmation failed; please re-type it.”
- A5 System fails to connect to database. Display the message “Database unavailable; we are sorry for the inconvenience. Please try again later.” Then return the user to the page they entered the use case from.
- A6 User entered an email already present in the database on the account creation page. System does not leave the page, and displays the message “Email already registered.”
- A7 System fails to send an email to the entered address. System does not leave the page, and displays the message “Invalid email, please re-type.”

(e) Pre-conditions:

- i. System is functional.
- ii. User is not logged in.

(f) Post-conditions:

- i. User is logged in with their new account, or cancelled account creation process.

(g) Special Requirements:

- i. Emails must be of the form <name>@<domain>. They are checked for validity when the system attempts to send to them.

- ii. Names cannot include special characters other than . , '
    - iii. Passwords must be at least six characters, and must have at least two of the following; letters, numbers, special characters.
- (h) Feature Mapping:
  - i. Levels of Authentication
  - ii. Accounts

## 7.3 Appointments

### 1. Name: Select Experiment

- (a) Brief Description: Participant views and selects experiment to join
- (b) Actors: Participant (henceforth “user”)
- (c) Basic Flow:
  - i. User can sort or filter experiment table by date, time, and location.
  - ii. User clicks an experiment. The system navigates them to that experiment’s page.
  - iii. Experiment page: Each experiment has a webpage with its name, description, and a list of timeslots, as well as a button to join the experiment.
  - iv. User reads experiment description and required qualifications.
  - v. User can sort or filter timeslot list.
  - vi. User clicks “join experiment” or a timeslot button. This takes them to use case “sign up for experiment”.
- (d) Alternate Flows:
  - A1 User decides to view a different experiment by navigating with their browser or clicking a button on any page. They are returned to the homepage.
- (e) Pre-conditions:
  - i. System is functional.
  - ii. There is at least one experiment currently offered.
- (f) Post-conditions:
  - i. User has clicked “join experiment” or a timeslot button for some experiment.
- (g) Feature mapping:
  - i. Browse Experiments
  - ii. Filter Experiments

### 2. Name: Sign up for Experiment

- (a) Brief Description: Participant enters data and confirms appointment.
- (b) Actors: Participant (henceforth “user”)
- (c) Basic Flow:
  - i. If the user is not currently logged in, run use case “login”. They must be logged in to continue.

- ii. Confirmation Page: This page is available only while logged in. It displays experiment name, required qualifications, and a check box for the user to verify they meet those qualifications. There is a list of timeslots. There is a “Confirm Appointment” button. [A2]
  - iii. The user selects a timeslot (or lets the system do it for them if they did so in “Select Experiment”) and checks the check box. [A2]
  - iv. The user clicks the “Confirm Appointment” button. [A1] [A2]
  - v. If there are no problems with the entered data, the system returns the user to the homepage and displays a message informing them of their successful registration. The system will also send an email containing the experiment and timeslot information to the email account used to register.
- (d) Alternate Flows:
  - A1 User attempts to “Confirm Appointment” before selecting a timeslot or checking the check box.
    - A. The system will return them to the confirmation page and inform them of what still needs to be done.
  - A2 User logs out while on the confirmation page. The system will return them to the experiment page.
- (e) Pre-conditions:
  - i. System is functional.
  - ii. User has selected an experiment via the “Select Experiment” use case.
  - iii. The selected experiment has at least one viable timeslot.
- (f) Post-conditions:
  - i. Database has added appointment to user and experiment data.
  - ii. (Side effect) user is authenticated.
- (g) Special Requirements:
  - i. N/A
- (h) Feature Mapping:
  - i. Participant Schedule Experiment
  - ii. Notify Participant when Creating Appointment
  - iii. Prevent Scheduling Conflicts (Participant)

## 7.4 Experiment Management

### 1. Name: Add Experiment

- (a) Brief Description: Experiments can be created by Administrators and Researchers
- (b) Actors: Administrators and Researchers
- (c) Basic Flow: (user can cancel at any time and follow A1)
  - i. User must click on Add New Experiment link from the Administration “home” page
  - ii. System will display a screen with text boxes to enter experiment name, description, and qualifications, multiple date/time choosers for the schedule times, and a drop down list to specify the length of the schedule slots

- iii. User must enter the experiment information for name, description, qualifications, and schedule slots
    - iv. User can then begin setting up the schedule times by choosing date, begin, and end time for each slot they want to run the experiment
    - v. User then must save the experiment by clicking the Save button
    - vi. System will then save the experiment to persistent storage and provide the user with confirmation that the experiment was created successfully and redirect user to all experiment view [A2]
  - (d) Alternate Flows:
    - A1 User cancels out of creating an experiment
    - A2 Saving an experiment fails
  - (e) Pre-conditions:
    - i. User is an Administrator and/or a Researcher and has authenticated
  - (f) Post-conditions:
    - i. System will have recorded the experiment or the system will notify the user why the creation of the experiment failed
  - (g) Special Requirements:
    - i. End times for each slot must be after begin times.
  - (h) Feature mapping:
    - i. Add Experiment
    - ii. Prevent Scheduling Conflicts (Administrator)
- 2. Name: Modify Experiment**
- (a) Brief Description: Experiments can be modified by Administrators and Researchers to change all assets of the experiment
  - (b) Actors: Administrators and Researchers
  - (c) Basic Flow: (user can cancel at any time and follow A1)
    - i. System will display experiment fields (name, description, qualifications, schedule time, schedule slots, and participant list)
    - ii. User will click on desired field to modify [A3]
    - iii. System will allow field that user chooses to be editable in line
    - iv. User will then change field as desired and click away from the field or save when finished
    - v. System will update the database with the modified experiment information [A2] [A3]
  - (d) Alternate Flows:
    - A1 User cancels out of creating an experiment. System will return user to the page where user came from
    - A2 Saving an experiment fails
    - A3 User deletes an experiment. System will remove experiment from database after user confirmation and display a message to the user indicating this was successful
  - (e) Pre-conditions:

- i. User is an Administrator and/or a Researcher and has authenticated
  - ii. User chose experiment through one of the experiment views
- (f) Post-conditions:
  - i. System will have recorded the modifications to the experiment or the system will notify the user why the modification of the experiment failed
- (g) Special Requirements:
  - i. End times for each slot must be after begin times.
- (h) Feature mapping:
  - i. Modify Experiment
  - ii. Remove Experiments
  - iii. Prevent Scheduling Conflicts (Administrator)

## 7.5 Reports

### 1. Name: List Experiment Participants

- (a) Brief description: Researcher logs in and views a list of all participants for a selected experiment.
- (b) Actors: Researcher
- (c) Basic flow:
  - i. (1) Researcher logs in, using the Login use case with a Researcher account
  - ii. (2) System displays table of researcher's experiments [A1]
  - iii. (3) Researcher selects experiment from table
  - iv. (4) System displays list of all participants for selected experiment
- (d) Alternate flows:
  - A1 Researcher does not own any experiments
    - A. (2) displays an empty table
    - B. He cannot proceed past (2) until he creates an experiment or is added to another researcher's
  - i. Selected experiment has no participants
    - A. (4) displays an empty table
    - B. Nothing is displayed in (4) until a participant signs up for the selected experiment
- (e) Pre-conditions
  - i. System is running
  - ii. System is in ready state
  - iii. Researcher has account with correct permissions/groups
- (f) Post-conditions
  - i. Researcher knows who is signed up to participate in his selected experiment or there are no experiments/participants
- (g) Special Requirements:

- i. N/A
- (h) Feature mapping:
  - i. Experiment Participants
- 2. Name: Cancel Experiment Appointment**
  - (a) Brief description: Participant logs in and cancels an appointment.
  - (b) Actors: Participant (User)
  - (c) Basic flow
    - i. (1) Participant logs in
    - ii. (2) System displays table of participant's appointments [A1]
    - iii. (3) Participant selects appointment from table
    - iv. (4) System displays details for selected appointment
    - v. (5) Participant selects cancel
    - vi. (6) System displays confirmation prompt
    - vii. (7) Participant selects confirm: appointment is marked cancelled and system returns to (2) with an affirmation message
    - viii. (8) Participant selects keep appointment: system returns to (4)
  - (d) Alternate flows
    - A1 Participant has no appointments
      - A. (2) displays an empty table
      - B. He cannot proceed past (2) until he signs up for an experiment
  - (e) Pre-conditions
    - i. System is running
    - ii. System is in ready state
    - iii. Participant has account
  - (f) Post-conditions
    - i. Participant cancelled selected appointment or participant cancelled operation
    - ii. Researcher(s) owning said appointment's experiment are notified via email
  - (g) Special Requirements:
    - i. N/A
  - (h) Feature mapping:
    - i. Cancel Experiment Appointment
    - ii. Notify Participant Appointment Cancellation Reminder
- 3. Name: Report Experiment Participant Lists**
  - (a) Brief Description: When the user is a researcher, the user will be able to export a CSV file, filed with the Experiment name and participant and times.
  - (b) Actors: Researcher
  - (c) Basic Flow:

- i. The researcher will check what experiments to export to the CSV file from the list of experiments in the researcher side view
- ii. The researcher will click “Export to CSV [A1]
- iii. The system will generate a CSV file from the selected experiment displaying the name of the experiment and the names of participants with their times [A2] [A3]
- iv. The system will then start the download of the file to the researcher’s computer [A4] [A5]
- v. When the system has completed 3 and 4, the system will display a message box Export Complete!
- vi. The researcher will click “OK or the exit button on the message box
- vii. The system will return to the researcher side view.

(d) Alternative Flow:

- A1 The researcher did not select any experiment. An error window will appear.
- A2 The system encounters an error when pulling data from the database. An error window will appear
- A3 The system encounters any error when creating the CSV file. An error window will appear
- A4 The system cannot download the file to the researcher’s computer. An error window will appear
- A5 The researcher will deny the download of the CSV. A message box will appear
- A6 The user exits the browser

(e) Preconditions:

- i. The researcher must be logged in as a researcher
- ii. The system is in the researcher side view
- iii. The researcher must already have experiments scheduled

(f) Postconditions:

- i. The system is back in the researcher side view

(g) Feature mapping:

- i. Export Experiment Participant List

#### 4. Name: Calendar/List of All Experiments

- (a) Basic Description: The list will show all ongoing experiments and will allow for a user to click and view more information on the experiment

(b) Basic Flow:

- i. The system displays all experiments that have not yet occurred [A1]
- ii. The user can scroll down the list
- iii. The user selects an experiment, as per use case Select Experiment

(c) Alternative Flow:

- A1 There are no experiments to display. In this case, there is nothing to show the user, and no experiment can be selected.

(d) Preconditions:

- i. The user is on the web page
- (e) Postconditions:
  - i. The system is showing an experiment or the browser is on a new page
- (f) Feature mapping:
  - i. All calendar Experiments
  - ii. Browse Experiments
  - iii. Persistent Experiment Storage

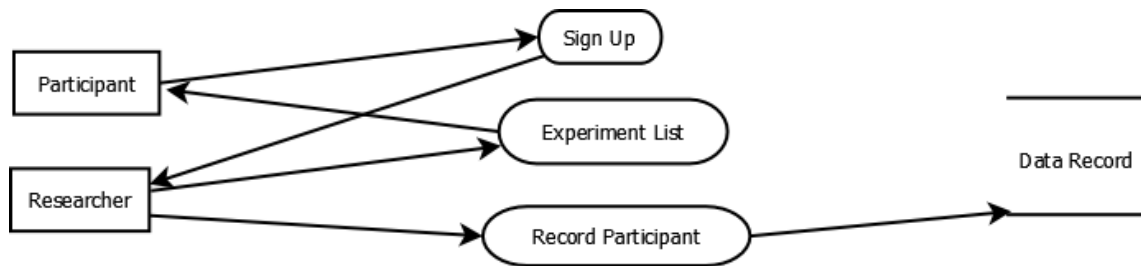
## 8 Use Cases Features Mapping

Use Case	Feature	Feature ID
Log In	Levels of Authentication	3
Log In	Accounts	18
Log Out	Levels of Authentication	3
Log Out	Accounts	18
Create Account	Levels of Authentication	3
Create Account	Accounts	18
Select Experiment	Browse Experiments	1
Select Experiment	Filter Experiments	5
Sign Up for Experiment	Participant Schedules Experiment	4
Add Experiment	Add Experiment	8
Modify Experiment	Modify Experiment	9
Modify Experiment	Remove Experiment	15
List Experiment Participants	Experiment Participants	6
Cancel Experiment Appointment	Cancel Experiment Appointment	7
Export Experiment Participant List	Export Experiment Participant List	13
Calendar/List of All Experiments	All Experiments Calendar	14
<i>None</i>	Persistent Experiment Storage	2
<i>None</i>	Notify Participant when Creating Appointment	10
<i>None</i>	Notify Participant Appointment Reminder	11
<i>None</i>	Notify Participant Appointment Cancellation Reminder	12
<i>None</i>	Prevent Scheduling Conflicts (Participant)	19
<i>None</i>	Prevent Scheduling Conflicts (Administrator)	20
<i>None</i>	Documentation for Maintenance and User	22



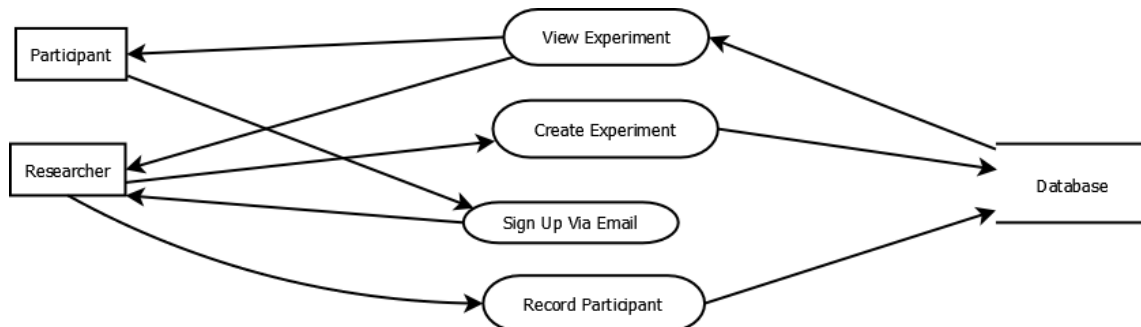
## 9 Data Flow Diagrams

### 9.1 Current System Context Diagram



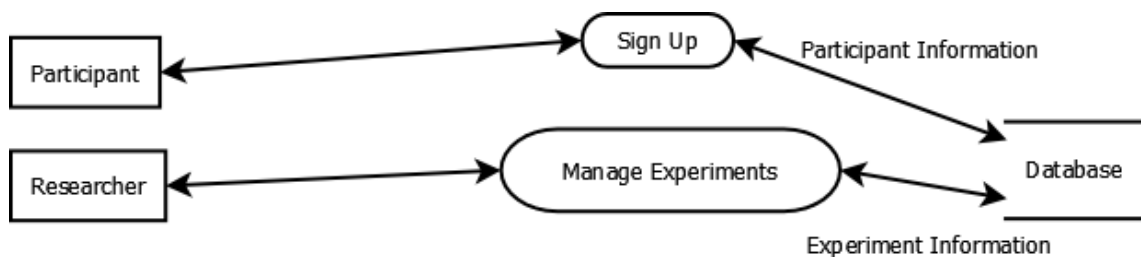
The current system requires the researcher to post an experiment (Experiment List), with the Participant then seeing the experiment from the experiment list and signing up. The researcher then gets the participant's information and records it in their current data record system.

### 9.2 Current System Level 0 Data Flow Diagram



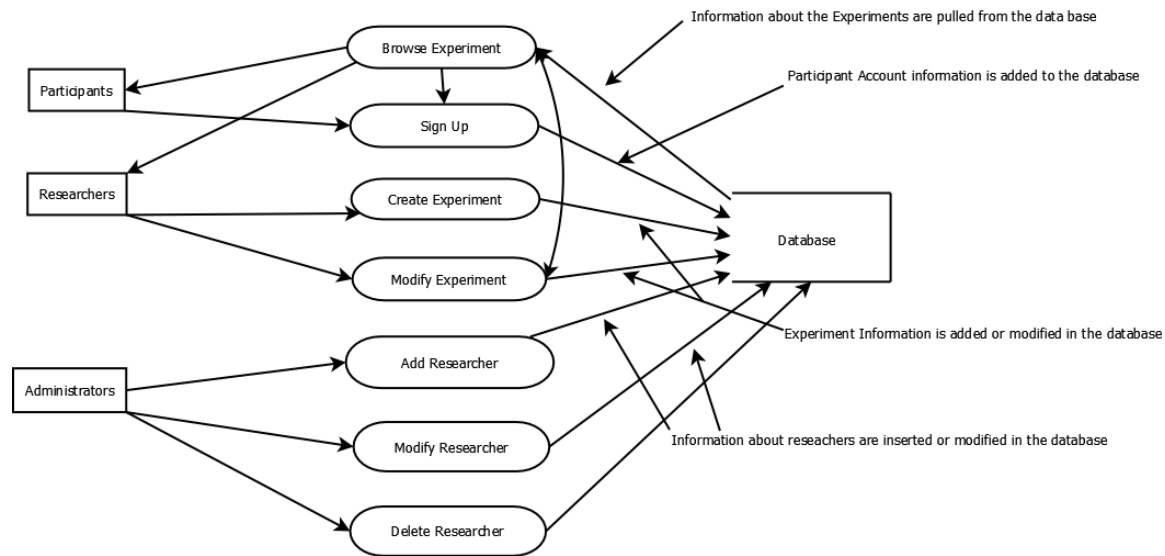
In the level zero data flow diagram, the flow remains the same, but the participant is signing up via email, while the researcher must explicitly create the experiment to be put on the database (for this definition of database the team is including a peg board of information or a campus email or similar types of information distribution).

### 9.3 Proposed System Context Diagram



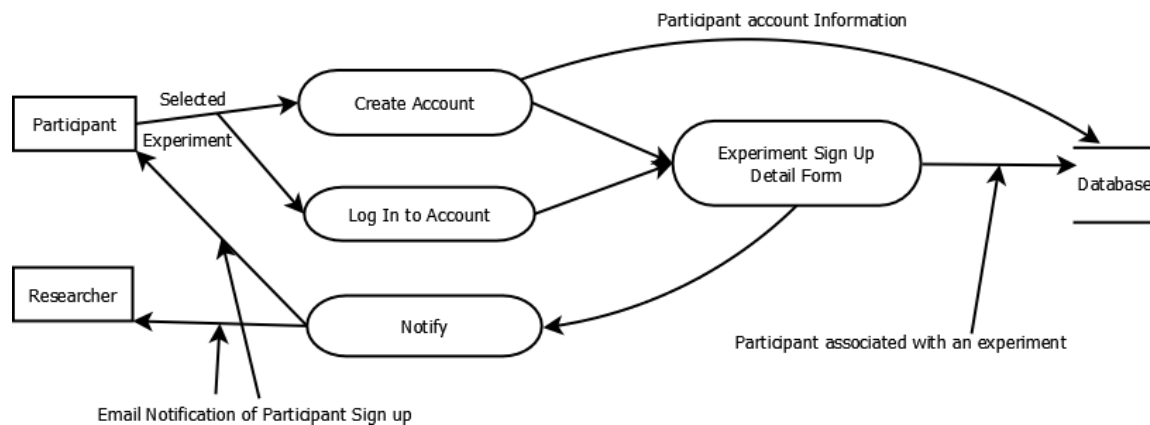
In the proposed system, the researcher will manage experiments via a computer database. This same database will let the user sign up and keep their information for the experiment.

## 9.4 Proposed System Level 0 Data Flow Diagram



Now, in the level 0 diagram, creating an experiment and modifying an experiment are explicitly different (modifying an experiment includes deletion of an experiment), but both have data being stored on the computer database. The participant now browses the experiment and then signs up, instead of the two being combined. The data entered by the participant will be saved on the database. Another actor, the administrator, will have the ability to add a researcher or modify a researcher, which includes deleting a researcher. All of the data about the researchers will be placed on the computer database.

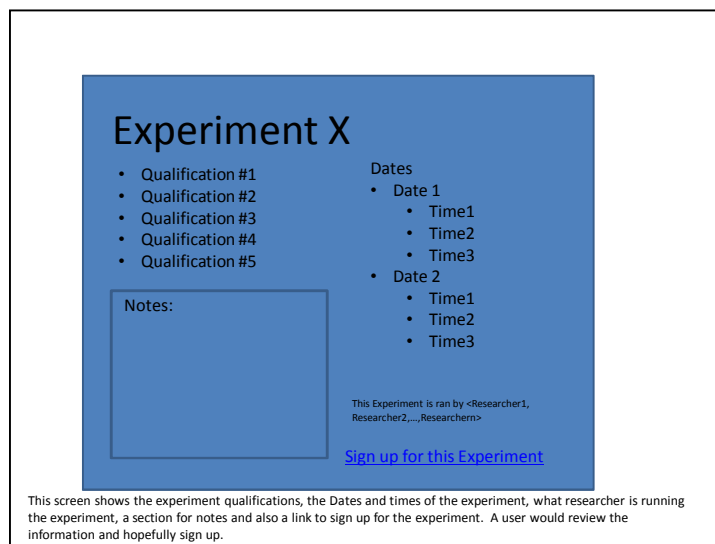
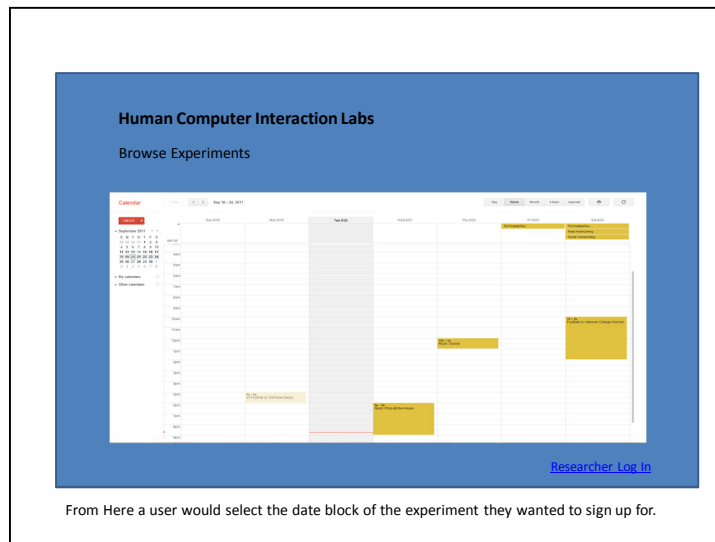
## 9.5 Proposed System Level 1 Data Flow Diagram for Sign up



When signing up the participant has selected a specific experiment to begin with. Then, the participant creates an account, with the information being stored in the computer database, or logs into their account. Next the participant will select a time and confirm that they conform to the qualifications of the experiment. This information will be stored in the computer database. After finishing sign up, the researcher and participant will be notified via email of the new participant.

## 10 Storyboard

10/4/2011



### Sign up Information for Experiment X

Name:

Cell Phone:

Email:

☐ By clicking this I have agreed that I meet all qualifications

[Submit](#)

This screen allows a participant to sign up for the experiment and agree to meet all the qualifications.

### Researcher Log In

Username:

Password:

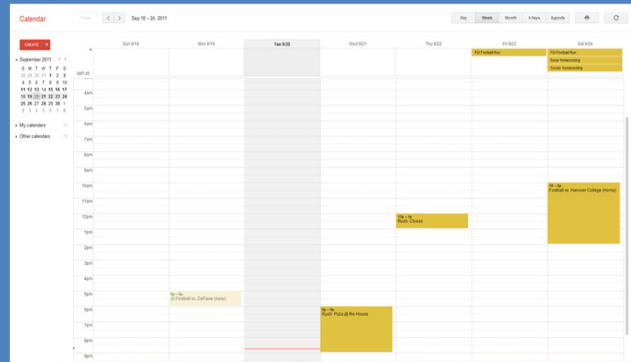
[Log In](#)

A researcher would log in to their account from here.

## Experiments of <Username>

### Filters

- Date
- Room
- Etc
- Etc
- ...



[Create New Experiment](#)

A researcher's possible screen after signing in.

## Experiment X

[Save](#)

### Qualifications:

- Age
- Language
- Etc
- etc

### Description/Notes

### Participant List

Name	Cell Phone	Date/Time	Email
Bill Bob	(555)555555	5/5 – 14:00	<a href="mailto:bob@ma.com">bob@ma.com</a>
Sparks McGee	(403)6185546	5/6 – 13:30	<a href="mailto:sparks@ma.com">sparks@ma.com</a>
Shane Jermey	(209)4723882	5/5 – 13:30	<a href="mailto:jermy@ma.com">jermy@ma.com</a>

From this screen a researcher could add description or notes and also qualifications. Also, the researcher could view the participants.

## Experiments of <Username>

Experiment	Date	Room
<a href="#">Robot Stuff</a>	5/5	F 217
UI Stuff	5/6	F 322
10001010110	5/10	O 231

[Create New Experiment](#)

This screen is an alternative to the calendar view for the researchers.

## Human Computer Interaction Labs

Browse Experiments

Experiment Name	Date – Times	Notes
Robot Stuff	5/5 – 13:00	Etf
Less Robot Stuff	5/5 – 14:00	Etc
More Robot Stuff	5/6 – 12:00	Etc
UI	5/6 – 9:00	Etc
Dan Guiney	5/8 – 20:00	Etc
Computer Runs	5/9 – 15:00	etc

[Researcher Log In](#)

This screen is an alternative to the calendar view for the participants.

## 11 References

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

## 12 Appendix

## 13 Glossary

**SQL** Structured Query Language is a programming language designed for managing data in relational database management systems. 5

## Index

Human-Computer Interaction Lab, 2