Rose-Hulman Institute of Technology

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

Milestone 4

Trey Cahill Chris Gropp Samad Jawaid Kevin Risden

November 2, 2011

Contents

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 fourth document describing this project, and enumerates test cases for the use cases presented in Milestone 2, as well as covering coding guidelines and change control. 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.[2]

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 Coding Standards

Since the project will be coded in Python and use the Django framework, our team will be following the style guides for Python and Django. The style guide for Python is PEP 8 [3] and for Django it is based on PEP 8 with a few modifications [1]. Outlined below are a few of the main key points from both style guides.

Key Points:

- Use 4 spaces for indentation
- Never mix tabs and spaces
- Use underscores, not camelCase, for variable, function and method names
- Use InitialCaps for class names
- Imports should usually be on separate lines
- Avoid extraneous white space

7 Change Control

7.1 Change Requests

The team will receive requests by either email or verbal request from the client. Information required from the client would be a description of the change and the version in which the change should be implemented. Should a change be made internally, the change request will be received during a meeting time. A ticket will then be created in Redmine detailing each change requested.

7.2 Managing change

The team will manage change via Redmine. Each change will be accepted or denied after meeting with the team based upon a team vote, with majority rule. From this point the team will determine what is affected and add the needed information to Redmine. Also any information that would change in the documentation will also be altered. If the client requests a change, the team will discuss specifics of the change such as effort required and who/where the change will be implemented to determine if the change will be accepted or denied. If a team member requests a change then the request will be reviewed by the team during the next team meeting. If the request is accepted, the change will follow the flow described above; otherwise, the change will be considered dead. The priority of the accepted change, will be brought through a vote or as asked by the client.

7.3 Changes to Project Artifacts

The team currently uses GitHub for both documentation and source code. Since documentation and source code are changed frequently, team members create a new branch to do their work in when making changes. After finishing their work, a team member would merge their branch to the master branch.

8 Test Cases

8.1 Experiments

8.1.1 List Experiment Participants

Conditions:

A Researcher does not own any experiments

B Selected experiment has no participants

Test Case	Scenario	Description	Cond A	Cond B	Expected Result
1	1	Basic flow	I	I	System displays list of all participants for selected experiment
2	2	Alternate flow: Researcher does not own any experiments	V	N/A	System displays an empty table of experiments
3	3	Alternate flow: Selected experiment has no participants	N/A	V	System displays an empty table of participants

8.1.2 Cancel Experiment Appointment

Conditions:

A Participant selects confirm

B Participant selects cancel

C Participant has no appointments

Test Case	Scenario	Description	Cond A	Cond B	Cond C	Expected Result
1	1	Basic flow: Participant selects confirm	V	I	I	Appointment is marked cancelled and system returns with an affirmation message
2	1	Basic flow: Participant selects cancel	I	V	I	System returns user to page they came from
3	2	Alternate flow: Participant has no appointments	N/A	N/A	V	System displays an empty table of appointments

8.2 Experiment Management

8.2.1 Add Experiment

Conditions:

A Experiment information (includes name, description, qualifications, date/time schedule, and slot length) - Must check each combination

Test Case	Scenario	Description	Cond A	Expected Result
1	1	Basic flow: Administrator enters experiment information	V	Experiment is created and user is notified that creation of experiment was successful
2	2	Basic flow: Administrator enters experiment information	I	Experiment is not created and user is notified of invalid field entry and the corresponding field
3	3	Alternate flow: Administrator tries to save experiment	N/A	System notifies user that the save failed and returns user to Add Experiment page with pre-filled values
4	4	Alternate flow: Administrator clicks cancel	N/A	System returns user to page they came from

8.2.2 Modify Experiment

Conditions:

A Experiment information (includes name, description, qualifications, date/time schedule, and slot length) - Must check each combination

Test Case	Scenario	Description	Cond A	Expected Result
1	1	Basic flow: Administrator enters experiment information	V	Experiment information is updated and user is notified that modification of the experiment was successful
2	2	Basic flow: Administrator enters experiment information	I	Experiment information is not updated and user is notified of invalid field entry and the corresponding field
3	3	Alternate flow: Administrator tries to save experiment	N/A	System notifies user that the save failed and returns user to Add Experiment page with pre-filled values
4	4	Alternate flow: Administrator clicks cancel	N/A	System returns user to page they came from
5	5	Alternate flow: Administrator deletes the experiment	N/A	System notified the user that the experiment has been deleted and returns user to experiment management page

8.3 Authentication

8.3.1 Login

Conditions:

A Email entered and in database

B Password entered and matches email in database

Test Case	Scenario	Description	Cond A	Cond B	Expected Result
1	Any page	Basic flow: User clicks "login/create account" link from any page.	N/A	N/A	System navigates user to login page.
2	Login page	Alternate flow: User clicks "submit" button.	I	N/A	System navigates user to create account page (see test cases for Account Creation).
2	Login page	Alternate flow: User clicks "submit" button.	V	I	System displays message informing user their password is incorrect.
3	Login page	Basic flow: User clicks "submit" button.	V	V	System displays message confirming successful login.
4	Success message	Basic flow: User clicks "return immediately" link.	N/A	N/A	System takes down success message and navigates back to initial page.
5	Success message	Alternate flow: 10 seconds pass after message displayed.	N/A	N/A	System takes down success message and navigates back to initial page.

8.3.2 Logout

Test Case	Scenario	Description	Expected Result
1	Any page	Basic flow: User clicks "logout" link from any page.	System displays message confirming successful logout.
2	Logout message	Basic flow: User clicks "return immediately" link.	System takes down logout message.
3	Logout message	Alternate flow: 10 seconds pass after clicking "logout".	System takes down logout message.

8.3.3 Login

Conditions:

A Email entered and is not in database

B Password entered and follows guidelines

C Confirm password entered and matches password

D User name entered and follows guidelines

Test Case	Scenario	Description	Cond A	Cond B	Cond C	Cond D	Expected Result
1	Any page	Basic flow: User clicks "login/create account" link from any page.	N/A	N/A	N/A	N/A	System navigates user to login page.
2	Login page	Basic flow: User clicks "create account" button.	N/A	N/A	N/A	N/A	System navigates user to create account page.
3	Login page	Alternate flow: User clicks "submit" button.	V	N/A	N/A	N/A	System navigates user to create account page.
4	Create account page	Alternate flow: User clicks "submit" button.	I	N/A	N/A	N/A	System displays message informing user that email is already in use.
5	Create account page	Alternate flow: User clicks "submit" button.	V	I	N/A	N/A	System displays message informing user that password does not meet guidelines.
6	Create account page	Alternate flow: User clicks "submit" button.	V	V	I	N/A	System displays message informing user that password and confirmation do not match.
7	Create account page	Alternate flow: User clicks "submit" button.	V	V	V	I	System displays message informing user that name does not meet guidelines.
8	Create account page	Basic flow: User clicks "submit" button.	V	V	V	V	System displays message confirming successful account creation.
9	Success message	Basic flow: User clicks "return immediately" link.	N/A	N/A	N/A	N/A	System takes down success message and navigates back to initial page.
10	Success message	Alternate flow: 10 seconds pass after message displayed.	N/A	N/A	N/A	N/A	System takes down success message and navigates back to initial page.

8.4 Appointments

8.4.1 Select Experiment

Test Case	Scenario	Description	Expected Result
1	Home page	Basic flow: User clicks an experiment from the table.	System navigates user to that experiment's page.
2	Experiment page	Basic flow: User clicks "join experiment".	System navigates to appointment page (see Sign Up for Experiment).
3	Experiment page	Alternate flow: User selects a timeslot.	System navigates to appointment page (see Sign Up for Experiment) with timeslot information.

8.4.2 Sign up for Experiment

Conditions:

A User logged in

B User has clicked check box and selected valid timeslot

Test Case	Scenario	Description	Cond A	Cond B	Expected Result
1	Appointment page	Alternate flow: User is on appointment page.	I	N/A	System navigates user to login page.
2	Appointment page	Basic flow: User is on appointment page.	V	N/A	Selected timeslot from experiment page already selected.
3	Appointment page	Alternate flow: User clicks "confirm appointment" button.	V	I	System informs user they must complete the form.
4	Appointment page	Basic flow: User clicks "confirm appointment" button.	V	V	System displays message confirming successful appointment.
5	Success message	Basic flow: User clicks "return immediately" link.	N/A	N/A	System takes down success message and navigates back to home page.
6	Success message	Alternate flow: 10 seconds pass after message displayed.	N/A	N/A	System takes down success message and navigates back to home page.

8.5 Reports

8.5.1 Export Participants

Test Case	Scenario	Description	Precondition	Expected Results
1	Export Experiment Participant List	Basic Flow: The researcher clicks Export to CSV	There are no experiments checked	Error box asking the user to select an experiment
2	Export Experiment Participant List	Basic Flow: The researcher clicks Export to CSV	There are experiments checked	The system generates a CSV file
3	Export Experiment Participant List	Basic Flow: The system starts the download of the file	The CSV creation succeeded	The file downloads and a message box is displayed “Export Complete”
4	Export Experiment Participant List	Alternate Flow: An Error occurs when pulling from the database	The researcher has selected experiments and Clicked Export CSV	The system displays an error message
5	Export Experiment Participant List	Alternate Flow: The system cannot download the file to the researchers computer	The CSV creation succeeded and the download has started	The system displays an error message
6	Export Experiment Participant List	Alternate Flow: The researcher denies the download of the CSV	The CSV creation succeeded and the download has started	The system displays a message box

9 Use Cases

9.1 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.

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.

9.2 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

9.3 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.

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 entering all required information/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)

9.4 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)

9.5 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
      ii. System displays table of researcher's experiments [A1]
      iii. (2) Researcher selects experiment from table
      iv. 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

10 References

[1] Django. Django Coding Style, 2011.

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

[3] Python. Style Guide for Python Code, 2011.

11 Appendix

12 Glossary

Django a high-level Python Web framework that encourages rapid development and clean, pragmatic design. 3

GitHub a web-based hosting service for software development projects that use the Git revision control system. 4, 20

Redmine a free and open source, web-based project management and bug-tracking tool. 4, 20

Index