



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 Ridsen

October 20, 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	Coding Standards	3
7	Change Control	4
7.1	Change Requests	4
7.2	Managing change	4
7.3	Changes to Project Artifacts	4
8	Test Cases	5
8.1	Experiments	5
8.1.1	List Experiments	5
8.1.2	List Experiment Participants	5
8.1.3	Cancel Experiment Appointment	6
8.2	Authentication	7
8.2.1	Login	7
8.2.2	Logout	7
8.2.3	Login	8
8.3	Appointments	9
8.3.1	Select Experiment	9
8.3.2	Sign up for Experiment	9
8.4	Reports	10
8.4.1	Export Participants	10
9	References	11
10	Appendix	11
	Index	11

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. All information that would be required for Redmine will be needed.

7.2 Managing change

The team will manage change via Redmine. Each change will be accepted or denied after meeting while meeting with the team. From this point the team will determine what is effected 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 and the change is within the allowed scope of change, the team will automatically accept the change and discuss specifics of the change such as effort required, who and where the change will be implemented, etc. 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; other wise the change will be considered dead.

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. By merging a branch, instead of working on the master branch, the team is protected from two people working on the same document and then a team member losing the changes they made.

8 Test Cases

8.1 Experiments

8.1.1 List Experiments

Test Case	Scenario	Description	Precondition	Expected Results
1	Calendar/List of All Experiments	The users has selected the website and the website is loading	The website has responded and the browser has loaded the screen	The website loads the first 10 experiments
2	Calendar/List of All Experiments	Basic Flow: The user scrolls down the list	The website has loaded the first 10 experiments	The user can click for the next 10 experiments
3	Calendar/List of All Experiments	Basic Flow: The user selects an experiment	The system has loaded 10 experiments	The system is showing an experiment
4	Calendar/List of All Experiments	Alternate Flow: The user attempts to log into the researcher page and fails	The user does not enter the correct information or the user is not a researcher	The system returns to the same screen with fields cleared
5	Calendar/List of All Experiments	Alternate Flow: The user attempts to log into the researcher page and has success	The user is a researcher and knows their log in information	The system goes to the researchers main page

8.1.2 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

8.1.1.3 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 deleted and system returns with an affirmation message
2	1	Basic flow: Participant selects cancel	I	V	I	System returns
3	2	Alternate flow: Participant has no appointments	N/A	N/A	V	System displays an empty table of appointments

8.2 Authentication

8.2.1 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.

8.2.2 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.

8.2.3 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.

8.3 Appointments

8.3.1 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.

8.3.2 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.

8.4 Reports

8.4.1 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 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.

10 Appendix

Index

Human-Computer Interaction Lab, 2