**Open WebLabUX Specification**

**Version:** 15 January 2014 **Specifications Maintainer:** Elisabeth Cuddihy ([ecuddihy@uw.edu](mailto:ecuddihy@uw.edu))

1 Page Templates 5

1.1 Logged out page templates 6

1.2 Logged in page templates 8

2 Login Page 11

2.1 Login Problem-handling pages 14

3 All Studies 22

3.1 All Studies 23

3.2 Edit Study 25

3.2.1 Edit Study: General Info 26

3.2.2 Edit Study: Variations 27

3.2.3 Edit Study: Protocol 29

3.2.4 Edit Study: Scheduling 32

3.3 Study Data 33

3.3.1 Study Data: Measured variables 33

3.3.2 Study Data: Manage data sets 33

4 Resources 33

5 Collaborators 33

6 Help 33

7 Client-Side Javascript Data Tracker 33

7.1 What the Javascript Data Tracker Does 33

7.2 User Experience 33

7.3 Configuring the JavaScript Data Tracker 33

7.4 How the Javascript Data Tracker Works 33

8 Definition of a Study 33

8.1 Example Study Designs 35

8.1.1 STUDY DESIGN 1: A “Reading To Learn” hypertext study design 35

8.1.2 STUDY DESIGN 2: A “Reading To Do” hypertext study design 36

8.1.3 STUDY DESIGN 3: An “Info Identification” page study design 37

8.2 What Constitutes a Study from the Researcher’s (User’s) Perspective 38

8.2.1 What Constitutes a Variable from the Researcher’s (User’s) Perspective 39

8.3 Object Model for Study and Study Parts (minus variables) 44

8.4 Database Schema for Study and Study Parts (minus variables) 51

8.5 Object Model for Study Variables (part of Study) 51

8.6 Database Schema for Study Variables (part of Study) 54

8.7 Object Model for Scheduling Study Periods 56

8.8 Database Schema for Study and Study Parts 57

8.9 ORIGINAL DATABASE SCHEMA --- OLD BUT HERE FOR REFERENCE 58

9 User Accounts and Authentication 59

9.1 Overview 59

9.2 Scenarios 59

9.2.1 Creating a WLUX Account 60

9.2.2 Logging into an Existing WLUX Account 60

9.2.3 Delegating/Removing Access for a Study 60

9.2.4 Accessing an Authorised Resource 60

9.3 User Information 60

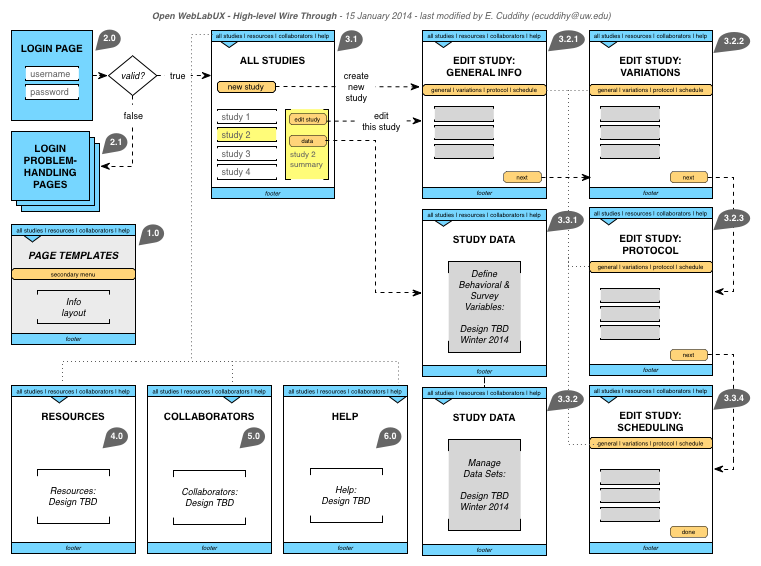
9.4 Authentication and Security 61

9.4.1 Password Storage at Rest 61

9.4.2 Inflight Encryption 61

9.4.3 Logging Concerns 62

9.5 Authorisation and Gating 62



# Page Templates

Page templates define the headers, footers, menus, fonts, and styles for Open WebLabUX.

|  |  |  |
| --- | --- | --- |
| Template | Used For | Filename(s) |
| Logged Out – Home template | 2.0 – Login page | **Photoshop:** Hompage\_V3.psd – *template needs to be separated from content.*  **HTML:** Front\_End\_Zulka\_Jessie – *needs review and updating – template needs to be ported into our templating system* |
| Logged Out – Account interaction | 2.1 – Login problem handling pages & account request pages | **Photoshop:** Hompage\_V3.psd – *template needs to be separated from content, needs review of white space, fonts, text…*  **HTML:** |
| Logged In – Top-level Three Regions | 3.1 – All Studies | **Photoshop:** LayoutA.psd – *needs to be documented – may need to have content separated out.*  **HTML:** … |
| Logged In – Top-level Two Regions | (not yet used) | … |
| Logged In – Sub-level Submenu | 3.2.1 – Edit Study: General Info  3.2.2 – Edit Study: Variations  3.2.3 – Edit Study: Protocol  3.2.4 – Edit Study: Scheduling  3.3.1 – Study Data: Measured Variables  3.3.2 – Study Data: Manage Data Sets | **Photoshop:** *needs to be created*  **HTML:** … |

## Logged out page templates

|  |  |
| --- | --- |
| **Logged Out – Home template – *incorrect* – needs review, needs to remove page specific content**  Macintosh HD:Users:ecuddihy:Dropbox:WLUX-Prototypes:0 Home Page:V3_Screens:HomePage_V3-1.jpg | **State in which template is used:**   * User not logged in * User not inside login sequence * User not inside account request sequence   **TO DO:**   1. Need to determine which if any of these menus will be supported in the initial release.    1. Where do these links go? What is the minimal viable subset of static content. 2. Stripped down page template needs to be created from existing photoshop file    1. Top menu (revised links?)    2. Bottom footer    3. Empty content area    4. Optional overlay with sign-in box 3. Areas of template need to be named (CSS IDs and classes), styles/fonts need to be defined, CSS needs to be updated. 4. Template CSS needs to be put into our templating system (probably: blade/laravel, alternatively bootstrap/laravel) |
| **Logged out – Account Interaction – Incorrect – needs to remove page specific content**  Macintosh HD:Users:ecuddihy:Dropbox:WLUX-Prototypes:0 Home Page:V3_Screens:HomePage_V3-2.jpg | **State in which template is used:**   * User not logged in * User inside login sequence *OR* user inside account request sequence   **TO DO:**   1. THIS IS A VARIANT OF THE TEMPLATE ABOVE 2. Stripped down page template needs to be created from existing photoshop file    1. Top menu and bottom footer are the same as template above.    2. Empty content area    3. Define the error/account interaction box size, colors, and fonts.    4. Define the messages box size, colors, fonts. 3. Areas of template need to be named (CSS IDs and classes), styles/fonts need to be defined, CSS needs to be updated. 4. Template CSS needs to be put into our templating system (probably: blade/laravel, alternatively bootstrap/laravel) |

## Logged in page templates

|  |  |
| --- | --- |
| **Logged In – Top-level Three Regions -- *incorrect* – needs review, needs to remove page specific content**  Macintosh HD:Users:ecuddihy:Dropbox:WLUX-Prototypes:0 Page Layouts:Layout_A.png | **State in which template is used:**   * User logged in * User in a top-level section   **TO DO:**   1. **IMPORTANT! -** We need to consider the UI widgets needed by other pages that use this template! Review all of the EDIT STUDY wireframes for ideas – also see the “Two Region” variant on the next page! 2. Stripped down page template needs to be created from existing photoshop file with simple examples of how the same clean look and feel will be used on the ALL STUDIES and EDIT STUDY pages.    1. Top menu and bottom footer (note – footer needs to be defined)    2. Layout regions (empty of content)    3. Optional/potential widgets (page dependant).    4. All colors, fonts, etc. 3. Areas of template need to be named (CSS IDs and classes), styles/fonts need to be defined, CSS needs to be updated. 4. Template CSS needs to be put into our templating system (probably: blade/laravel, alternatively bootstrap/laravel) |
| **Logged In – Top-level Two Regions – is this layout something we will use in edit study?**  Macintosh HD:Users:ecuddihy:Dropbox:WLUX-Prototypes:0 Page Layouts:Layout_B.png | **NOT YET USED…**  **State in which template is used:**   * User logged in * User in a second-level section   **TO DO:**   1. **IMPORTANT! -** We need to consider the UI widgets needed by other pages that use this template! Review all of the EDIT STUDY wireframes for ideas – also see the “Two Region” variant on the next page! 2. Stripped down page template needs to be created from existing photoshop file with simple examples of how the same clean look and feel will be used on the ALL STUDIES and EDIT STUDY pages.    1. Top menu and bottom footer (note – footer needs to be defined)    2. Layout regions (empty of content)    3. Optional/potential widgets (page dependant).    4. All colors, fonts, etc. 3. Areas of template need to be named (CSS IDs and classes), styles/fonts need to be defined, CSS needs to be updated. 4. Template CSS needs to be put into our templating system (probably: blade/laravel, alternatively bootstrap/laravel) |
| **Logged In – Sub-level Submenu – NEED TO REVIEW GRAPHIC DESIGN OF MENU – needs removal of page specific content – needs documentation**  **Macintosh HD:Users:ecuddihy:Desktop:WLUX WIN 2014:Week 01:WLUX-Specs-Pack1:Study Definition Wireframes :1-EditStudy_General.jpg** | **State in which template is used:**   * User logged in * User within subsection of a top-level section   **TO DO:**   1. **IMPORTANT! -** We need to consider the UI widgets needed by other pages that use this template! Review all of the EDIT STUDY wireframes for ideas – also see the “Two Region” variant on the next page! 2. Stripped down page template needs to be created from existing photoshop file with simple examples of how the same clean look and feel will be used on the ALL STUDIES and EDIT STUDY pages.    1. Top menu and bottom footer (note – footer needs to be defined)    2. Layout regions (empty of content)    3. Optional/potential widgets (page dependant).    4. All colors, fonts, etc. 3. Areas of template need to be named (CSS IDs and classes), styles/fonts need to be defined, CSS needs to be updated. 4. Template CSS needs to be put into our templating system (probably: blade/laravel, alternatively bootstrap/laravel) |

# Login Page

|  |  |  |
| --- | --- | --- |
| ***STORY*: LOGIN**   1. User types in a username and password. 2. User clicks “sign in” button. 3. If system has record of user+password, a user session is authenticated. System loads the “ALL STUDIES” page for that user. 4. If system does not have record of user+password, the system initiates the LOGIN PROBLEM HANDLING story. | | |
| Macintosh HD:Users:ecuddihy:Dropbox:WLUX-Prototypes:0 All Studies:Photoshop Mockups:HomePage-UserFlow.pdf | **Routing URL** | “/” |
| **View name** | loggedout/index.blade.php |
| **Page template** | *(logged out home template)* |
| **Page title** | OpenWebLabUX – Log in |
| **Expects:**   * $username * $password | **Validates:**  *…[details here]…* |
| **Starts on:** logged out home page  **Performs:** on “sign-in”, attempts authentication of username password *(… auth details here…)*  **Successfully ends on:** logged in all studies page  **Unsuccessfully ends on:** login problem handling | |

|  |  |  |
| --- | --- | --- |
| ***STORY*: NON-AUTHORIZED REQUEST OF INTERNAL WEBPAGE**   1. User’s browser sends request for a URL that requires authentication in order to view (i.e., any view that is not in the “loggedout” directory. 2. WLUX automatically loads home page for LOGIN story. | | |
| Macintosh HD:Users:ecuddihy:Dropbox:WLUX-Prototypes:0 All Studies:Photoshop Mockups:HomePage-UserFlow.pdf | **Routing URL** | Any route that requires authentication to view – rerouted as “/” |
| **View name** | loggedout/index.blade.php |
| **Page template** | *(logged out home template)* |
| **Page title** | OpenWebLabUX – Log in |
| **Expects:**  **--** | **Validates:**  **--** |
| **Performs:** when no user is authenticated, re-routes password protected pages to “/”  **Successfully ends on:** login page | |

|  |  |  |
| --- | --- | --- |
| ***STORY*: USER BROWSES STATIC INFO PAGES WHILE LOGGED OUT**   1. User is not logged into the system. 2. User browses links on header or footer, providing static info pages that describe WebLabUX.   *\*\*\* Page content needs to be designed and written. This is a low priority. We should determine the minimal set of pages that should be statically served.* | | |
| Macintosh HD:Users:ecuddihy:Dropbox:WLUX-Prototypes:0 All Studies:Photoshop Mockups:HomePage-UserFlow.pdf | **Routing URL** | “/”  “/about”  “/howitworks”  “/contact”  … (tbd) … |
| **View name** | loggedout/index.blade.php  loggedout/about.blade.php  loggedout/howitworks.blade.php  loggedout/contact.blade.php  …(tbd)… |
| **Page template** | *(logged out home template)* |
| **Page title** | OpenWebLabUX – Log in  OpenWebLabUX – About  OpenWebLabUX – How It Works  OpenWebLabUX - Contact |
| **Expects:**  **--** | **Validates:**  **--** |
| **Performs:** Serves static pages as user browse when not logged in. | |

## Login Problem-handling pages

|  |  |  |
| --- | --- | --- |
| ***STORY*: USER ATTEMPTS TO LOG IN WITH UNRECOGNIZED USERNAME+PASSWORD**   1. System presents login error message to user, displaying message inside the “Sign In” box. 2. User can attempt to sign in or click “forget password” or “sign-up for an account” | | |
| Macintosh HD:Users:ecuddihy:Dropbox:WLUX-Prototypes:0 All Studies:Photoshop Mockups:HomePage-UserFlow.pdf | **Routing URL** | “/” |
| **View name** | loggedout/index.blade.php |
| **Page template** | *(logged out home template)* |
| **Page title** | OpenWebLabUX – Log in |
| **Expects:** | **Validates:** |
| **Performs:**  ERROR MESSAGES NEED TO BE DESIGNED.   * What is the error message? * Where in the box is it displayed? (Font, color…) | |

|  |  |  |
| --- | --- | --- |
| ***STORY*: USER HAS FORGOTTEN THEIR PASSWORD**   1. User clicks “forgot password?” on the home page’s sign in. 2. User given the open to enter username or enter email, and click “next” 3. If an unrecognized email or username entered, system displays error message on the first screen. 4. If a valid email or username entered, system sends password reset information to their email address and second screen is shown | | |
| Macintosh HD:Users:ecuddihy:Dropbox:WLUX-Prototypes:0 Home Page:V3_Screens:HomePage_V3-5.jpg  Macintosh HD:Users:ecuddihy:Dropbox:WLUX-Prototypes:0 Home Page:V3_Screens:HomePage_V3-6.jpg | **Routing URL** |  |
| **View name** |  |
| **Page template** |  |
| **Page title** |  |
| **Expects:** | **Validates:** |
| **Performs:**  NOTE: ERROR MESSAGE FROM #3 NEEDS TO BE DEFINED.  MESSAGES ON SECOND SCREEN NEED TO BE REWRITTEN – THE LOCAL ADMINISTRATOR HANDLES ACCOUNT PROBLEMS, **NOT** THE WEBLABUX TEAM.  MECHANISMS FOR RESETTING PASSWORD NEED TO BE CONFIRMED WITH PEOPLE HANDLING AUTH. | |

|  |  |  |
| --- | --- | --- |
| ***STORY*: USER SIGNS UP FOR AN ACCOUNT**   1. User clicks “sign up for account” on sign in page 2. System prompts user for name, email, password, and password verification and verifies these have valid format (and resolves errors) 3. System forwards information to site administrator (defined during system configuration/installation) 4. System shows user information that their account will soon be verified. | | |
| Macintosh HD:Users:ecuddihy:Dropbox:WLUX-Prototypes:0 Home Page:V3_Screens:HomePage_V3-2.jpg  Macintosh HD:Users:ecuddihy:Dropbox:WLUX-Prototypes:0 Home Page:V3_Screens:HomePage_V3-2a.jpg  Macintosh HD:Users:ecuddihy:Dropbox:WLUX-Prototypes:0 Home Page:V3_Screens:HomePage_V3-3a.jpg  Macintosh HD:Users:ecuddihy:Dropbox:WLUX-Prototypes:0 Home Page:V3_Screens:HomePage_V3-3b.jpg  Macintosh HD:Users:ecuddihy:Dropbox:WLUX-Prototypes:0 Home Page:V3_Screens:HomePage_V3-4.jpg | **Routing URL** |  |
| **View name** |  |
| **Page template** |  |
| **Page title** |  |
| **Expects:** | **Validates:** |
| **Performs:**  SCREENS NEED MINOR REDESIGN TO REFLECT STORY:   * Password needs to be part of signup – add password and confirm password to screen * Too much white space in sign up box – make design nicer. * On verification screen, remove the two check boxes + text for newsletter and ToS, Not needed. * Display information about account being verified by the site administrator and display email address (which is defined whenever WLUX is installed). * Clean up white space on all of these. | |

# All Studies

User activities handled within the All Studies section of WebLabUX:

**All Studies** section page:

* View how the number of many studies are currently:
  + Associated with the user’s account
  + Running at that moment -- STATUS == RUNNING
  + Finished running and have data – STATUS == COMPLETE
  + In draft (not yet ready to run) – STATUS == DRAFT
  + Archived – STATUS == ARCHIVED
* View a list of all studies associated with the user’s account, with high level information displayed for each study (i.e., study status, study name, first X characters of study description, number of participants so far, scheduled dates for the study)
* For a particular study, the user can:
  + View additional details that are displayed in a study side panel. Those details include: scheduling information for each session and number of participants so far.
  + Edit the study (*takes user to the* ***Edit Study*** *sub-section of All Studies*)
  + Download the data for the study (*this needs more definition…*)
  + Create a new study (*takes user to the* ***Edit Study*** *sub-section of All Studies*)

## All Studies

|  |  |  |
| --- | --- | --- |
| ***STORY*: SYSTEM DISPLAYS ALL STUDIES PAGE**   1. System queries all studies associated with user’s account AND the status of each study. System tallies how many studies in each status of RUNNING, COMPLETE, DRAFT, and ARCHIVED. 2. System displays the summarized tallies in the top box 3. System displays a list of studies in the main content box (define: in what order will the studies be displayed?) | | |
| Macintosh HD:Users:ecuddihy:Desktop:WLUX WIN 2014:Week 01:WLUX-Specs-Pack1:Page Layouts:Layout_A.png | **Routing URL** |  |
| **View name** |  |
| **Page template** |  |
| **Page title** |  |
| **Expects:** | **Validates:** |
| NOTE: This screen is **incorrect** – no study would be highlighted and no sidebox on right would be shown. What would this look like instead? How does a user discover how to display the side box? | |

|  |  |  |
| --- | --- | --- |
| ***STORY*: USER SELECTS STUDY TO DISPLAY MORE DETAILS IN SIDEBAR**   1. User selects a study from the list (define: How? What interaction? Hover? click?) 2. System displays additional information about study schedule with all periods run so far 3. (DEFINE: DO WE WANT ADDITIONAL INFORMATION IN THE SIDEBAR?) | | |
| Macintosh HD:Users:ecuddihy:Desktop:WLUX WIN 2014:Week 01:WLUX-Specs-Pack1:Page Layouts:Layout_A.png | **Routing URL** |  |
| **View name** |  |
| **Page template** |  |
| **Page title** |  |
| **Expects:** | **Validates:** |
| Is it clear how to download data or work with data?!?! This should be made more clear on the sidebar. | |

## Edit Study

**Edit Studies** sub-section pages: (*these pages need more definition and wireframes need to be laid out in photoshop template*)

* Edit study - general information (i.e., name, description, researcher contact information).
* Edit study - manipulated variations of the website being tested (i.e., CSS/PHP used to vary the website, descriptions of variables, names of variables, …)
* Edit study – protocol that declares the linear order of study steps (i.e., pre-study components such as information, consent, surveys; one or more study tasks; post-study components such as surveys and thank you/gratuity forms).
  + Editing specifics of pre-study and post-study items
  + Selecting how the tasks will be ordered and whether the website will vary per task
  + Editing specific task information – define specific information about the task.
* Edit schedule – define a study period, when it will run, etc…

### Edit Study: General Info

|  |  |  |
| --- | --- | --- |
| ***STORY*: CREATE NEW STUDY / EDIT EXISTING STUDY – GENERAL INFO AS STARTING POINT**   1. If **create** **new** study, user fills out general information describing study and clicks continue. Upon clicking continue, the system creates the data objects for the new study AND sends the user to the next screen – protocol. (or should this next screen be (manipulated) variables?!) 2. If **edit existing** study, system queries for study information and populates this screen.    1. If the user updates information in the fields AND clicks continue (should this be save?), the system updates the data model/database.    2. If the user updates information in the fields BUT clicks away from this screen, the system should give a warning that study will not be updated. | | |
| Macintosh HD:Users:ecuddihy:Desktop:WLUX WIN 2014:Week 01:WLUX-Specs-Pack1:Study Definition Wireframes :1-EditStudy_General.jpg | **Routing URL** |  |
| **View name** |  |
| **Page template** |  |
| **Page title** |  |
| **Expects:** | **Validates:** |
| Same screen will be used for creating new study and editing existing study. Need to fix text that says “create new.”  **To Do:**   * Standardize screen name - tbd:   + Could have a single name?   + Could have “create new” vs “edit [$study\_name] depending on the situation? * Button name: Continue? Save? ??? * Design the error message when the user clicks away without saving * Compare against API/data model – what fields are these? Do we need others? * Clean up wireframe and layout in the appropriate photoshop template for this page type * What is the correct names & order for the submenu items (general, protocol, variables, sessions) * Rename “sessions” to “study periods” | |

### Edit Study: Variations

|  |  |  |
| --- | --- | --- |
| ***STORY*: EDIT STUDY – EDIT VARIATIONS**   1. On their own website, the user *already* has CSS or PHP files that can be used to define different variations of the website they are testing. Here they define these variations in WLUX by:    1. Naming each variable (e.g., “website color scheme”) and then naming the different variations of that variable (e.g., “orange sunset color scheme”, “midnight blue color scheme”, “spring green color scheme”) and giving the URL of the CSS or PHP file that controls/displays the variation.    2. User may define one or more variables. Each variable may have two or more variations. 2. When user clicks “next” (should be “save” or “continue”), the URLs need to be validated and error messages need to be displayed. | | |
| Macintosh HD:Users:ecuddihy:Desktop:WLUX WIN 2014:Week 01:WLUX-Specs-Pack1:Study Definition Wireframes :2-EditStudy_Variables-b.png  Macintosh HD:Users:ecuddihy:Desktop:WLUX WIN 2014:Week 01:WLUX-Specs-Pack1:Study Definition Wireframes :2-EditStudy_Variables-a.jpg | **Routing URL** |  |
| **View name** |  |
| **Page template** |  |
| **Page title** |  |
| **Expects:** | **Validates:** |
| To Do:   * Button name: Continue? Save? ??? * Design the error message for bad URL * Compare against API/data model – WE NEED A DATA STRUCTURE THAT REPRESENTS VARIABLES. * Clean up wireframe and layout in the appropriate photoshop template for this page type and reconcile issues from prior page (submenu names, create study vs edit study…) | |

### Edit Study: Protocol

|  |  |  |
| --- | --- | --- |
| ***STORY*: EDIT STUDY – EDIT PROTOCOL**   1. User creates/edits the study protocol that declares the linear order of study steps (i.e., pre-study components such as information, consent, surveys; one or more study tasks; post-study components such as surveys and thank you/gratuity forms).    1. Editing specifics of pre-study and post-study items    2. Selecting how the tasks will be ordered and whether the website will vary per task    3. Editing specific task information – define specific information about the task. 2. When user clicks “next” (should be “save” or “continue”), does anything need to be validated and error messages need to be displayed? | | |
| Macintosh HD:Users:ecuddihy:Desktop:WLUX WIN 2014:Week 01:WLUX-Specs-Pack1:Study Definition Wireframes :3-EditStudy_Protocol-a.jpg  Macintosh HD:Users:ecuddihy:Desktop:WLUX WIN 2014:Week 01:WLUX-Specs-Pack1:Study Definition Wireframes :3-EditStudy_Protocol-b.jpg | **Routing URL** |  |
| **View name** |  |
| **Page template** |  |
| **Page title** |  |
| **Expects:** | **Validates:** |
| To Do:   * Button name: Continue? Save? ??? * Do we need any error messages * Compare against API/data model * – NOTE HOW SURVEYS ARE A URL TO AN OFFSITE SURVEY TOOL – HOW ARE THESE BEING HANDLED IN THE INTERFACE? ADD AS NEEDED… * – How is task order and variation type storied in the Data model? * Clean up wireframe and layout in the appropriate photoshop template for this page type and reconcile issues from prior page (submenu names, create study vs edit study…) * Define the click interactions for the accordion style display… | |

|  |  |  |
| --- | --- | --- |
| ***STORY*: EDIT STUDY – EDIT PROTOCOL TASK**   1. User creates/edits details for a single task 2. When user clicks “next” (should be “save” or “continue”), does anything need to be validated and error messages need to be displayed? | | |
| Macintosh HD:Users:ecuddihy:Desktop:WLUX WIN 2014:Week 01:WLUX-Specs-Pack1:Study Definition Wireframes :4-EditStudy_Task.jpg | **Routing URL** |  |
| **View name** |  |
| **Page template** |  |
| **Page title** |  |
| **Expects:** | **Validates:** |
| To Do:   * Button name: Continue? Save? ??? * Do we need any error messages * Compare against API/data model – * Clean up wireframe and layout in the appropriate photoshop template for this page type and reconcile issues from prior page (submenu names, create study vs edit study…) * This is a sub-subsection – reconcile with our display/templates and navigation model! | |

### Edit Study: Scheduling

|  |  |  |
| --- | --- | --- |
| ***STORY*: EDIT STUDY – EDIT SCHEDULING PERIODS**   1. User creates/edits details for scheduling study periods in which data will be collected 2. When user clicks “next” (should be “save” or “continue”), does anything need to be validated and error messages need to be displayed? | | |
| Macintosh HD:Users:ecuddihy:Desktop:WLUX WIN 2014:Week 01:WLUX-Specs-Pack1:Study Definition Wireframes :5-EditStudy_SchedulePeriod.jpg | **Routing URL** |  |
| **View name** |  |
| **Page template** |  |
| **Page title** |  |
| **Expects:** | **Validates:** |
| To Do:   * Button name: Continue? Save? ??? * Do we need any error messages * Compare against API/data model – * Clean up wireframe and layout in the appropriate photoshop template for this page type and reconcile issues from prior page (submenu names, create study vs edit study…) * HOW CAN THIS HANDLE MULTIPLE PERIODS? | |

## Study Data

### Study Data: Measured variables

### Study Data: Manage data sets

# Resources

# Collaborators

# Help

# Client-Side Javascript Data Tracker

## What the Javascript Data Tracker Does

## User Experience

## Configuring the JavaScript Data Tracker

## How the Javascript Data Tracker Works

# Definition of a Study

This section describes:

1. Example study designs that WebLabUX needs to be able to run,
2. An in-progress definition of a study and its component parts,
3. The object models for a study (and its component parts), which will be accessible within WLUX for display and manipulation by the Views, and the required business logic (object’s instance methods) that absolutely needs to be supported for a minimal implementation.
4. The underlying database model for a study (and its component parts), which will be mapped to the object model (#3) by the ORM (defined in the database-ORM mapping).
5. The object models for scheduling study periods, which will be accessible within WLUX for display and manipulation by the Views, and the required business logic (object’s instance methods) that absolutely needs to be supported for a minimal implementation.
6. The underlying database model for a study period and study scheduling, which will be mapped to the object model (#5) by the ORM (defined in the database-ORM mapping).

## Example Study Designs

This section contains three examples of the typical kinds of studies that WebLabUX needs to be able to run in order to be a minimally viable system.

### STUDY DESIGN 1: A “Reading To Learn” hypertext study design

Study seeks to learn which website page design, information design, and/or navigation design helps people learn information on a multi-page website without compromising user experience. This kind of study only has one task. It can have one or more pre-study and pre-task surveys, and one or more post-task and post-study surveys.

1. Introductory information about the study with informed consent
2. Optional pre-study survey (as little as a couple of questions but it could be a few pages long – for instance this could include general demographics questions)
3. Optional pre-task survey (as little as a couple of questions but it could be a few pages long – for instance, this could include questions that measure the amount of knowledge a person has about the website’s TOPIC)
4. Optional pre-task instructions page (gives information and instructions on what the participant is expected to do).
5. Task instructions: *Browse this website for X minutes in order to learn about TOPIC.*
   1. Participant is shown one variation of the website and they are allowed to freely click through the site while reading information.
   2. (The participant decides when they are done browsing.)
6. Post-task survey on usability perceptions and user experience (one or more pages of questions)
   1. RIGHT NOW: The surveys live in a 3rd party tool – we know the surveys by URL, but each survey is “complete.”
   2. EVENTUALLY WE NEED A MECHANISM FOR MANAGING SURVEY INSTRUMENTS: This could include a set of questions that belong to a known survey instrument (such as the SUS – the systems usability survey), this could also include a set of questions from multiple survey instruments; the researcher may already have these questions entered in WebLabUX and they just want a copy of it included in this study.
7. Post-task survey measuring reading comprehension / info understanding (one or more pages of questions)
8. Optional post-study survey (as little as a couple of questions but it could be a few pages long – for instance this could include general demographics questions)
9. Parting information such as: “*Thank you for participating. If you have any questions, email* [*researcher@u.edu*](mailto:researcher@u.edu)*. Please enter your email address below if you wish to be included in a drawing for $25 an amazon gift certificate.”*

### STUDY DESIGN 2: A “Reading To Do” hypertext study design

Study seeks to learn which website page design, information design, and/or navigation design helps people find the information they need to complete one or more tasks. User experience may also be a variable measured in such a study. This kind of study can have one or more tasks. It can have one or more pre-study and post-study surveys. Each task probably has no more than one pre-task and/or post-task surveys.

1. Introductory information about the study with informed consent
2. Optional pre-study survey (as little as a couple of questions but it could be a few pages long – for instance this could include general demographics questions)
3. Performing the Study Tasks:
   1. Researcher must set up – Task order:
      1. Fixed order for all tasks
      2. Random order for all tasks.
   2. Researcher must set up – Variations shown to user
      1. Each participant only sees one variation of the website’s design for all tasks
      2. Each participant sees a different variation per task, and the order in which they see variations may be randomized.
   3. Task 1:
      1. Optional pre-task survey (as little as a couple of questions but it could be a few pages long – for instance, this could include questions that measure the amount of knowledge a person has about the website’s TOPIC)
      2. Optional pre-task instructions page (gives information and instructions on what the participant is expected to do).
      3. Task 1 instructions: *Use this website to find X.*
         1. Participant is shown one variation of the website and they are allowed to freely click through the site while scanning and reading information.
         2. When the participant finds the needed information, they have a place (on the task bar) for entering that information.
      4. Post-task survey on usability perceptions and user experience (one or more pages of questions)
   4. Task 2 (and more… for how ever many): …
4. Post-task survey on usability perceptions and user experience
5. Parting information such as: “*Thank you for participating. If you have any questions, email* [*researcher@u.edu*](mailto:researcher@u.edu)*. Please enter your email address below if you wish to be included in a drawing for $25 an amazon gift certificate.”*

### STUDY DESIGN 3: An “Info Identification” page study design

Study seeks to learn which website page design or information design allows users to rapidly and correctly determine if a page has the information they need. User experience may also be a variable measured in such a study. This kind of study can have one or more tasks. It can have one or more pre-study and post-study surveys. Each task probably has no more than one pre-task and/or post-task surveys.

1. Introductory information about the study with informed consent
2. Optional pre-study survey (as little as a couple of questions but it could be a few pages long – for instance this could include general demographics questions)
3. Performing the Study Tasks:
   1. Researcher must set up – Task order:
      1. Fixed order for all tasks
      2. Random order for all tasks.
   2. Researcher must set up – Variations shown to user
      1. Each participant only sees one variation of the website’s design for all tasks
      2. Each participant sees a different variation per task, and the order in which they see variations may be randomized.
   3. Task 1:
      1. Optional pre-task survey (as little as a couple of questions but it could be a few pages long – for instance, this could include questions that measure the amount of knowledge a person has about the website’s TOPIC)
      2. Optional pre-task instructions page (gives information and instructions on what the participant is expected to do).
      3. Task 1 instruction: *Does this page contain info on X?*
         1. Participant is shown one variation of a single page.
         2. When the participant finds the needed information, they have a place (on the task bar) for entering that information or saying “no, info not found.”
      4. Post-task survey on usability perceptions and user experience (one or more pages of questions)
   4. Task 2 (and more… for how ever many): …
4. Post-task survey on usability perceptions and user experience
5. Parting information such as: “*Thank you for participating. If you have any questions, email* [*researcher@u.edu*](mailto:researcher@u.edu)*. Please enter your email address below if you wish to be included in a drawing for $25 an amazon gift certificate.”*

## What Constitutes a Study from the Researcher’s (User’s) Perspective

**Aspects of a Study**

**GENERAL INFORMATION**

Study Name

Study Description

Researcher or Organization Name

**VARIABLES**

**PROTOCOL**

Pre-Study Protocol (Ex: Consent Form, Demographics Survey, Prior Knowledge Survey…)

Tasks

Task Order (Fixed Order or Random Order)

Website Variation (Same for all Tasks or Randomized for Each Task)

List of Tasks

Post-Study Protocol (Ex: Post-Study Survey, Thank You Form…)

**SCHEDULE** (A List of Sessions)

**Aspects of a Variable**

Variable Name

Number of Variations

(For each variation, there is Variation Name and Variation URL.)

**Aspects of a Task**

Pre-Task Page (Optional)

Instruction Text

Pre-Task Survey (Optional)

Taskbar Text (Summary of Instructions)

Task Type

- Question on Taskbar

User Response Type

- Text Field

- Dropdown Menu

Menu Options

- No Question on Taskbar

Button Text

Post-Task Survey (Optional)

**Aspects of a Study Session**

Participant Pool Description

Participant Pool Size

Start Time

End Time

List of Participant Sessions (One person doing the study.)

### What Constitutes a Variable from the Researcher’s (User’s) Perspective

**IMPORTANT TO DO:** THIS DEFINITION NEEDS TO BE REVIEWED AND REVISED BY ELISABETH BEFORE THE END OF JANUARY. INFORMATION REGARDING SURVEY VARIABLE WILL CHANGE

**IMPORTANT NOTE:** FOR THIS QUARTER, WE WILL DEFINE THE MOST SIMPLE VERSION OF VARIABLES. DURING SPRING QUARTER, WE WILL EXPAND THAT DEFINITION. MOST IMPORTANT RIGHT NOW:

* BEHAVIORAL VARIABLES SUCH AS *TIME ON TASK, TIME ON PAGE*, etc.

**IMPORTANT NOTE REGARDING SURVEY VARIABLES:** WEBLABUX WILL INTEGRATE WITH ONE OR MORE EXTERNAL SURVEY TOOLS SUCH AS **SURVEY GIZMO** AND **LIME SURVEY**. Surveys will be defined in those tools, administered by those tools and results (and questions?) will be imported into WLUX. WLUX will be responsible for POSTPROCESSING those survey results. Survey variables will be defined in order to handle post-processing for flagging disqualifying responses, adding/calculating composite responses, scoring, etc., and storing these post-processed results to survey measured variable names associated with participant IDs.

**Measured Variables - Overview**

Every **WebLabUX Study** will have one or more **Measured Variables**. Measured variables provide descriptions of the kinds of measurements that a study will collect.

Measured variables can be:

* **Webform Measured Variables:** a measured variable associated with one or more Webform survey responses.
  + If the measured variable is associated with only *one* question in the Webform survey, it’s value is the same as the participant’s response to the survey question.
  + If the measured variable is associated with a set of questions in the Webform surveys:
    - These questions must be of Webform type “select”
    - The value of the measured variable is calculated by adding the custom key values for the participant’s responses to those survey questions.
* **Behavioral Measured Variable:** A measured variable that is based on a calculation made by WebLabUX that measures a predefined, supported behavior. The first three jkltypes of behavioral measured variables that WebLabUX will support are:
  + Time on task: time between start and end of each task
  + Total time: total amount of time spent in the set of webpages
  + Total hits: total number of clicks on navigational links (note: this is a measure of all “clicks” which can be more than the number of webpages in the set)
  + Total views: total number of unique pages viewed (note: unique page views cannot exceed the total number of webpages in the set)
    - Definition elements of each measured variable:
      * Update the definitions to work on a per-participant basis
      * output variable type
      * transaction records used
      * SQL Query to process records and produce output variable
      * WebService parameters for query
* **Researcher-defined Measured Variables:** a measured variable defined by the researcher. They are responsible for providing their own instrumentation and passing a value to fill it.

**Webform Measured Variables**

The webform measured variables needs to be built on top of the Webform Drupal module (http://drupal.org/project/webform) such that additional options and functionality is made available to the 7.x version of Webform.

A **Webform Measured Variable** has:

* **Measured Variable ID** - database generated id for the variable
* **Name** - text
* **Description** - text describing the purpose of the variable
* **Type** - must be one of “text”, “nominal”, “ordinal”, or “scale” -- see [http://www.spsslog.com/2006/05/03/nominal-ordinal-and-scale/](http://www.google.com/url?q=http%3A%2F%2Fwww.spsslog.com%2F2006%2F05%2F03%2Fnominal-ordinal-and-scale%2F&sa=D&sntz=1&usg=AFQjCNEbEC__J-x_Sx7aKpfnEYdHT3mE_w) for definitions of these terms
* **Measurement**  - a set of {*nid, cid*} pairs -- (nid= node id; cid=component or question id) -- indicating which survey question(s) will be used for the measurement, where *nid* indicates with the drupal node identifier for the webform, and *cid* indicates with the drupal component identifier denoting the question within the node. Measurement is dependent on variable’s **type:**
  + If type is “text” or “nominal”, the measurement can only be a **single {nid, cid}** and the value of the measurement will be calculated as value of a participant’s response to the question indicated by the {nid, cid} pair.
  + If type is “ordinal” or “scale”, the measurement can be a **set of one or mode {nid, cid} pairs (i.e., surveys questions)** and the value of the of the measurement will be calculated as the sum of the values of a participant’s responses to the set of the question indicated by the {nid, cid} pair.
* (*optionally, a measured variable can also act as a*) **Disqualifying variable** - a boolean rule that compares the measurement to researcher-defined value. If the disqualifying variable is set to true, then the measured variable will also have a:
  + **Boolean operator** - a selection of one of the following: “=” “<=” “>=” “<” “>” “!=”
  + **Comparison Value** - a number or string that is checked against the calculated **measurement** value

**Researcher-defined Measured Variable**

This is a measured variable that is associated with custom instrumentation created by the researcher.

This would have a:

* **Measured Variable ID** - database generated id for the variable
* **Name** - text
* **Description** - text describing the purpose of the variable
* **Variable definition** (string of researcher defined data... *further design to be determined*)
* (*optionally, a measured variable can also act as a*) **Disqualifying variable** (boolean) and it would behave the same as the definition above

**Behavioral Measured Variable - Total Browsing Time**

This behavioral measured variable calculates the total browsing time within a set of pages. These pages can be webpage URLs for pages contained on the study website and/or study “nodes” (e.g., instructions, surveys) in the drupal-based weblabux site. By default, WebLabUX will automatically set up a total browsing time variable for all pages in the study website.

A **Browsing Measured Variable - Total Browsing Time** has:

* **Measured Variable ID** - database generated id for the variable
* **Name** - text
* **Description** - text describing the variable
* **Page set** (webpage urls on the study site or “nodes” on the drupal-based weblabux site)
* (*optionally, a measured variable can also act as a*) **Disqualifying variable** (boolean) as defined above

**Behavioral Measured Variable - Total Page Clicks**

This behavioral measured variable calculates the total number of page hits (navigational clicks) within a set of pages. These pages can be webpage URLs for pages contained on the study website and/or study “nodes” (e.g., instructions, surveys) in the drupal-based weblabux site. By default, WebLabUX will automatically set up a total browsing time variable for all pages in the study website.

A **Browsing Measured Variable - Total Page Clicks** has:

* **Measured Variable ID** - database generated id for the variable
* **Name** - text
* **Description** - text describing the variable
* **Page set** (webpage urls on the study site or “nodes” on the drupal-based weblabux site)
* (*optionally, a measured variable can also act as a*) **Disqualifying variable** (boolean) as defined above

**Behavioral Measured Variable - Unique Pages Visited**

This behavioral measured variable calculates the number of uniquely visited pages within a page set. These pages can be webpage URLs for pages contained on the study website and/or study “nodes” (e.g., instructions, surveys) in the drupal-based weblabux site. By default, WebLabUX will automatically set up a total browsing time variable for all pages in the study website.

A **Browsing Measured Variable - Unique Pages Visited** has:

* **Measured Variable ID** - database generated id for the variable
* **Name** - text
* **Description** - text describing the variable
* **Page set** (webpage urls on the study site or “nodes” on the drupal-based weblabux site)
* (*optionally, a measured variable can also act as a*) **Disqualifying variable** (boolean) as defined above

**Measured Variable Submissions - Overview**

Measured variable submissions occur as the study receives participants and are calculated based on the participant’s survey responses, navigation behaviors, etc. These are calculated values that are derived from other entries in the WebLabUX database.

**Measured Variable Submissions**

The Measured Variables Submissions are values calculated for each study participant as they complete a study:

* **GUID** uniquely identifying the study participant
* **Measured Variable id** - identifies which measured variable this is
* **Value** - a calculated value based on the definition of the measured variable
* **DQValue** - 0 or 1 - an optional calculated value if the measured variable is also acting as a disqualifying variable

-------

Need some examples of modules that extend webform:

\* WebForm Report Module: <http://drupal.org/node/716142>

Things to be added that are findable in the transaction log:

*Task-N* start

*Task-N* end

Time on *Task-N*

Page durations

Page groups

## Object Model for Study and Study Parts (minus variables)

**IMPORTANT TO DO:** CLIENT SIDE OBJECT MODEL NEEDS TO BE DEFINED/UPDATED – WHAT IS INCLUDED BELOW IS AN OLDER VERSION THAT IS CLOSE TO ACCURATE … BUT NOT QUITE DUE TO RECENT DESIGN CHANGES -- ELISABETH WILL START THIS IMMEDIATELY WHILE COORDINATING WITH BOB AND HAVE FOR INITIAL REVIEW NEXT WEEK. AFTER THAT, STUDENTS CAN BEGIN ADDING TO THIS MODEL AS THEY TAKE OWNERSHIP OF DIFFERENT COMPONENTS! ☺

**= STUDY DEFINITION =**

**OBJECT: Study -** A *Study* is used to create and run A/B tests or multivariate tests that measure website users' navigation behavior and their responses to survey questions. A *study* contains all information associated with a study. Each study object contains at least the following elements/attributes (note- other elements/attributes need to be added, based on our current screen designs):

* Name: *String* - the name of the study, which the researcher gives when creating the study and can edit at any time.
* Description: *String* - a textual description of the study that the researcher gives when creating or editing the study.
* Study Protocol: *StudyProtocol* - an list of study protocol steps (attribute defined below) that the researcher creates when defining/editing a study.
* Date created: *DateTime* - system generated timestamp when study is first created.
* Date last modified: *DateTime* - system generated timestamp that is updated each time the researcher updates the study Name, Description, or Study Protocol.
* Status: *StudyStatus* - a system generated attribute that is updated whenever the Study changes status from Draft to Running to Completed to Archived (attribute defined below).
* Session List: *StudySessions* - a list of StudySessions, where each **StudySession** represents a container object that describes all information associated with the data collection of a study. The StudySession object is defined at the end of this section of this document.
* … also need to include Favorite status, etc... (as in Dashboard UI design)

***ATTRIBUTE: StudyProtocol*** - A *StudyProtocol* attribute of the *Study* object is an ordered list of study components. These study components need to be defined by the researcher, added to the Study, and put into a linear order to define the steps that the study participants will go through. Each study component may be associated with one or more study variables (e.g., *MeasuredVariables, ManipulatedVariables*) All of the variables in a study much be associated with an appropriate study component. Currently, only 3 kinds of study components are defined:

1. **Surveys** - a *Survey* is a complex content type thatcontains one or more *SurveyMeasuredVariables* plus the set of questions associated with each *SurveyMeasuredVariable*. A survey may also contain textual instructions and other non-measurement elements (e.g., page breaks).
2. **Forms** - a *Form* is a complex content type that contains information to be displayed to study participants (e.g., a consent form, general study instructions) although it make contain simple input fields such as a CAPTCHA challenge input box.
3. **Tasks** - a *Task* is a complex content type that contains information describing the task that participants will do on the website that is being studied along with all other information for varying the display of the website.

The study component objects are partially defined in this document. The Measured Variables are defined in another document: <https://docs.google.com/document/d/1JCMKyNHsF3vnV0tH8nwPhhq0QGuJmrFPPjfE0GI7LrM/edit?usp=sharing>

***ATTRIBUTE: StudyStatus*** - The *StudyStatus* attribute describes the current overall state of the *Study* object. This needs further definition work.... Current states are:

* Draft - the study is still in draft mode and has not yet been scheduled to run.
* Scheduled - the study has one or more *StudySessions* that are scheduled to run sometime in the future.
* Running - the study has one or more *StudySessions* that are running, potentially collecting data.
* Completed - all *StudySessions* associated with this study have stopped running. (Note: a researcher could schedule/run a new session and this study would then move from Completed status to either Scheduled or Running status.)
* Archived - this state is true when all of the following three conditions are true: the study has 0 scheduled sessions, the study has 0 running sessions, and the researcher has chosen to archive the study.

**OBJECT: StudySessions -** A *StudySession* is a container object that describes all information associated with the data collection of a study. Whenever the researcher wants to collect study data, they must schedule a *StudySession.* The only time that a study collects data is when a *StudySession* is running. Multiple *StudySessions* can be scheduled and/or running simultaneously. Each *StudySession* is used to recruit participants to a study. Each study session is used by the researcher to clump together all participants who have been recruited at a specific time and using a specific recruitment method (such as sending to an internal company email list versus posting an advertisement on a website). The attributes/elements of a StudySession include (note-- this list is incomplete and may have additional attributes/elements that are in our current page designs):

* Name: *String* - the name of the study session, which the researcher gives when initially scheduling the session by selecting "Schedule New Study Session" on the UI. The researcher can edit this name at any time.
* Description: *String* - a textual description of the study session that the researcher gives when initially scheduling the session and the researcher can edit this name at any time.
* StartDate: *DateTime* - time that the study session is scheduled to start or was started. When the study sessions starts, the announcement email will be sent (if it exists) and the study protocol pages will become published/active (rather than hidden/password protected).
* EndDate: *DateTime* - the time that the study session is scheduled to stop or was stopped.
* AnnoucementText: *String* - optional email message text that will be sent to potential study participants when thes session starts.
* RecruitingList : a list of email addresses of potential study participants who will receive the AnnoucemenText at StartDate
* …. Also need to include alerts, etc. (as in Dashboard UI design)

**= TASK DEFINITION =**

**OBJECT: Task -** A *Task* is a reusable study component. Tasks drive what a study participant can see when they visit an external website that is being tested by WebLabUX. That external website must include the WebLabUX JavaScript Data Collector on all pages that are being studied in order for behavioral clickpath data to be collected and in order for website variations to be shown.

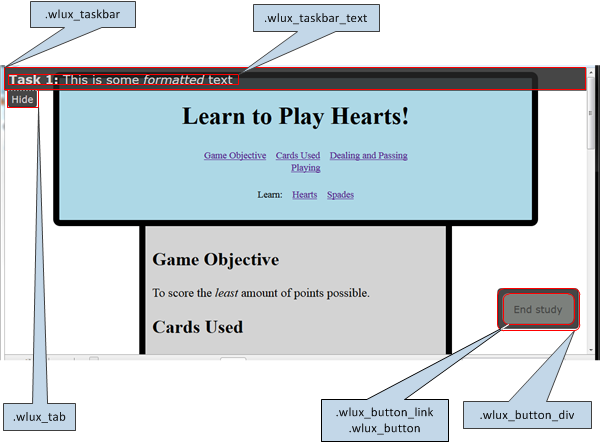
The user interface for tasks needs to allow the researcher to define how the task will be displayed on the external website. All components for a task are listed below in logical groups.

**General Attributes:**

|  |  |
| --- | --- |
| *name* | string - names the task |
| *description* | string - describes the task |
| *manipluatedVariables* | list of one or more manipulated variables that are associated with this task. Object Defined below. |

**Taskbar and Task Button Definition Attributes:**

|  |  |
| --- | --- |
| *taskBarCSS* | CSS for styling the taskbar and taskbutton. (See image below). |
| *buttonText* | The text to display on the task button on the external study website. Default is "End Task". |
| *taskText* | The unformatted text to display in the task bar on the external study website. |
| *taskHTML* | Formatted HTML to display in the task bar on the external study website. If both this field and the *taskText* are defined, only this field will be used. |
| *tabShowText* | Text to display in the show/hide task button when the task bar is hidden. |
| *tabHideText* | Text to display in the show/hide task button when the task bar is visible. |



**OBJECT: ManipulatedVariable -** A *ManipulatedVariable* is an object that **must** be associated with a *Task*. It cannot live on its own. Tasks contain a set of *ManipulatedVariables* (see above in Task - General Attributes)

**Attributes:**

|  |  |
| --- | --- |
| *name* | string - names the manipulated variable |
| *description* | string - describes the manipulated variable |
| *variableList* | a set of ordered pairs in the form [*VariationName, URL*]   * *VariationName* - string naming the variation * *URL -* url pointing to a css file or a php file that the researcher is responsible for putting on the external website that is being tested. this css/php file controls the variation that is shown. |

## Database Schema for Study and Study Parts (minus variables)

## Object Model for Study Variables (part of Study)

**IMPORTANT TO DO:** SOME MODELING EXISTS – NEEDS REVIEW AND PRIORITIZATION BY ELISABETH AND BOB.

**OBJECT: SURVEY MEASURED VARIABLES:**

**Demographic variables:**

These consist of information like the *Ethnicity*, *Gender*, *Age*, *Professional Background information.*

*Phase I: Variable Definition*

|  |  |  |
| --- | --- | --- |
| Variable Details | Source of information |  |
| *Id* | Unique Identification number | Database generated |
| *name* | <String> - Describes the variable ; Predefined | Variable definition |
| *type* | Text |  |
| *description:* | Text describing the purpose of the variable. | Variable definition |
| *measurement type* | Measurement can only be a single {nid, cid} and the value of the measurement will be calculated as value of a participant’s response to the question indicated by the {nid, cid} pair | Variable definition |
| *Disqualifying flag* | Indicates whether variable can be used for participant disqualification. | Variable definition |

*Phase II: Association with study*

Researcher should decide during the **study creation stage** whether they want to include these variables or not.

*Phase II: Association with Survey components (Webforms)*

|  |  |  |
| --- | --- | --- |
| nid |  |  |
| cid |  |  |
| Measured variable id |  |  |
| *Disqualifying flag* |  |  |
| *Disqualifying criteria* | Researcher needs to define the criteria   * Boolean operator - a selection of one of the following: “=” “<=” “>=” “<” “>” “!=” * Comparison Value - a number or string that is checked against the calculated measurement value |  |
| *Value* | calculated value based on the definition of the measured variable |  |

**System Usability Scale (SUS) variables:**

This is measured as a summation of responses to **10 standard questions** associated with System Usability scale.

Researcher should decide during the **study creation stage** whether they want to include this variable or not.

|  |  |  |
| --- | --- | --- |
| Variable Details | Source of information |  |
| *Id* | Unique Identification number | Database Generated |
| *name* | <String> - Describes the variable ; Predefined | Webform Component |
| *description:* | Text describing the purpose of the variable. | Webforms Component |
| *addn description* | Additional information (if any) to be provided to the participants. | Webforms Component |
| *measurement type* | Summation of | Study creation wizard |
| *Disqualifying criteria* | Yes / No  (Researcher needs to define the criteria   * Boolean operator - a selection of one of the following: “=” “<=” “>=” “<” “>” “!=” * Comparison Value - a number or string that is checked against the calculated measurement value ) | Study creation wizard |

**USE variables:**

**Researcher Defined Variables:**

## Database Schema for Study Variables (part of Study)

**IMPORTANT TO DO:** SERVER SIDE DATABASE SCHEMA EXISTS – NEEDS REVIEW AND PRIORITIZATION BY ELISABETH AND BOB.

\*note: red values are required when creating a new row in the database.

Measured Variable Definition Table - each row is a unique measured variable

* study\_id (int: foreign key) – identifies the study that this m.v. is for
* variable\_id (int: primary key) – auto generated by database, uniquely identifies this m.v.
* name (char) – name of this m.v.
* description (text) - text description of the m.v.
* variable\_type (enum) – choose from: “text”, “nominal”, “ordinal”, “scale”
* calculation\_type (enum) – choose from: sum, max, min, etc.
* max\_components\_no (int) – maximum number of components that this m.v. can be associated with (auto value = 0, which means no limit)
* associated\_components (text) – text array of {nid, cid} pairs, identifies which Drupal Webform components are associated with this m.v.
* curr\_components\_no (int) - initial value = 0, count of how many components have been added to this m.v. so far
* disqualifying\_variable (boolean) – whether or not this m.v. uses disqualifying value
* disqualifying\_variable\_operator (char) – initially empty, select an operator: >, <, >=, <=, =, !=, etc.
* disqualifying\_variable\_value (int) – initially empty, the disqualifying variable to compare against
* clone (boolean) – whether or not this m.v. is an original or a clone; a clone is a m.v. that uses values from a previously defined m.v. (can still be edited)
* use\_as\_template (boolean) - whether or not this m.v. should be used as a template for future m.v.; a template is a m.v. that will appear in the “search results” when creating a new m.v.

Measured Variable Data – each row is a m.v. value for a study participant

* participant\_id (int: foreign key) - identifies the participant that this row of values is for
* study\_id (int: foreign key) - identifies the study that the m.v. is in
* variable\_id (int: foreign key) – identifies the m.v. that the value is for
* value (int/char/text) – the computed value for the participant’s measured variables

Measured Variable Notes

* SUS, USE, Demographic, Comprehension M.V – Use “templates”, PHP scripts with preset values, however can still be edited by research
* Getting M.V. value for a study: go through M.V. table and use values with matching variable\_id

Behavioral Variable Definition Table - each row is a unique behavioral variable

* study\_id (int: foreign key) – identifies the study that this b.v. is for
* variable\_id (int: primary key) – auto generated by database, uniquely identifies this b.v.
* name (char) – name of this b.v.
* description (text) - text description of the b.v.
* variable\_definition (char) - Text from selection criteria . Auto generated from selection
* sql\_script (text) – sql script used to run query for this behavior variable

Behavioral Variable Data – each row is a b.v. value for a study participant

* participant\_id (int: foreign key) - identifies the participant that this row of values is for
* study\_id (int: foreign key) - identifies the study that the b.v. is in
* variable\_id (int: foreign key) – identifies the b.v. that the value is for
* value (int/char/text) – the computed value for the participant’s behavioral variables

Behavioral Variable Interface:

* Something like:
* First select box: type of data to for behavioral variable: page clicks, time spent, etc.
* Second select box: group type -- study, task, survey, page, page group(?)
* Third select box: specific group label
  + Study: Nothing more specific, shows all of data type for the entire current survey
  + Task: List of tasks for this study, e.g. #1, #2, #3...
  + Survey: List of surveys in this study, e.g. survey A, B, C...
  + Page: List of specific pages (pages being pages on client website), e.g. About Us (about.html), Home Page (index.html), etc.
  + Page Group(?) Not sure if there will be page groups? These might just be tasks.
* Will have a list of previously created behavioral variables

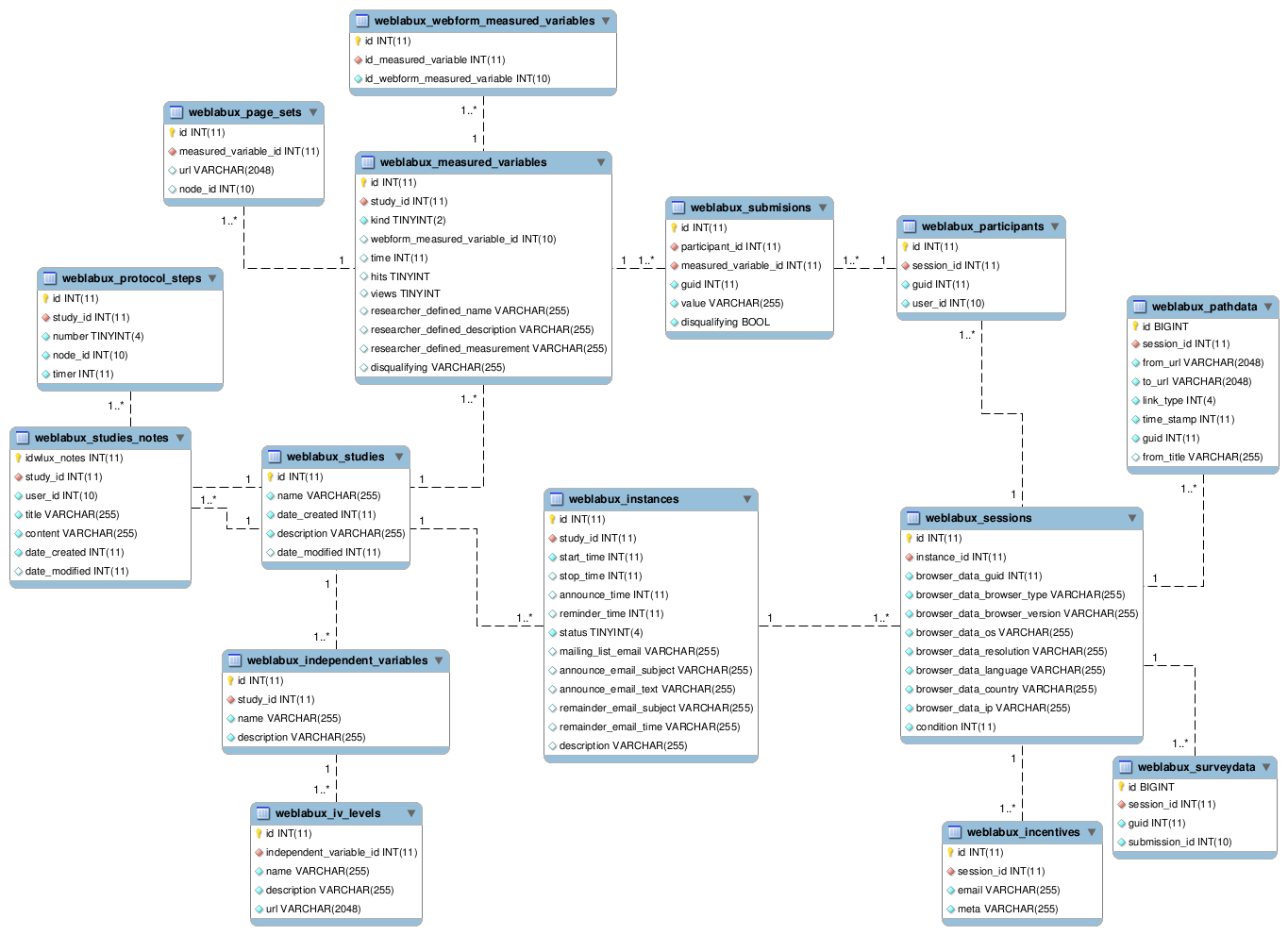
## Object Model for Scheduling Study Periods

UPCOMING TO DO: CLIENT SIDE OBJECT MODEL NEEDS TO BE DEFINED – ELISABETH WILL STAT ON THIS IN LATE-JANUARY AND TURN OVER TO STUDENTS ONCE WE ARE READY TO FLESH THIS OUT FURTHER AND IMPLEMENT.

## Database Schema for Study and Study Parts

UPCOMING TO DO: SERVER SIDE DATABASE SCHEMA NEEDS FLESHING OUT – BOB WILL WORK ON THIS AND COORDINATE WITH ELISABETH TO MAKE SURE THAT SERVER-CLIENT DATA MAPPING IS DEFINED.

## ORIGINAL DATABASE SCHEMA --- OLD BUT HERE FOR REFERENCE



# User Accounts and Authentication

For the minimal viable system, we will have a very simplistic notion of a user (a researcher) who is logged into WLUX in order to create a study. The user account specification will be fleshed out during the Winter 2014 Quarter.

THIS SPECIFICATION STILL NEEDS REVIEW BY BOB AND BY ELISABETH. REVIEW PENDING….

## Overview

In order to increase the accessibility of the WebLabUX platform it is being extended to include a dashboard and control panel for creating, updating, and monitoring the status of studies. Access to these resources is limited to their creators and their delegates.

This specification outlines the functional requirements for the following aspects of the application:

* User Information & Authentication
  + Creating Users:
    - Required Fields, Database Definition
  + Authentication and Security
    - Password Storage at Rest
    - Inflight Encryption
    - Logging Concerns
* ACLs/Permissions
  + Database Definition
  + Default User Roles
  + Granting & Revocation of Permissions
* Authorisation & Gating
  + Request Lifecycle and Authorisation
  + Gating Behaviour

## Scenarios

Examples of system and user behaviour within the purview of this specification.

### Creating a WLUX Account

Tim is excited about the possibilities of the WLUX platform and decides to create an account to test it out. He navigates to $login\_path and clicks on to the $create\_account\_path. He finds a form for “Create User”, fills it out, and submits it. He receives a transactional email with a one-time use link to verify his registration of a WLUX account. Clicking the one-time use link verifies his account, authenticates him, and redirects him to the WLUX dashboard.

### Logging into an Existing WLUX Account

Bobby already has a WLUX account. He navigates to $login\_path and fills out the login form, but mistypes his password. He is redirected to the same page with the incorrect fields highlighted and an error message explaining the issue. He rekeys his information correctly, submits the form, and is authorised and redirected to the WLUX Dashboard

### Delegating/Removing Access for a Study

Tim wants to grant Bobby access to review the results of one of his studies. He navigates to the Study, clicks on the $study\_preference\_link, and then scrolls down to the “Permissions” section. He selects “Add Delegate” and enters the email address that Bobby uses for his WLUX account in the “User” field. He then grants him “View” permissions.

Tim then realises Susan still has access, but is no longer working on the Study and does not require access. He selects her record in the “Permissions” section of the Study Preferences page and deletes it. No email is generated.

### Accessing an Authorised Resource

Bobby receives an email indicating that Tim has given him permissions to access Tim’s Study. Bobby clicks through to access it. He can view results, but not modify any aspect of the Study.

## User Information

The foundation of an authentication system is the user data that authentication and authorisation operate upon. For the purposes of WebLabUX, this constitutes:

* A Unique User Identifier; their Email Address
* Password/One-Factor Authentication (Something they know)

Additionally, for purposes of addressing the user in the 3rd person in the User Interface, we will also record:

* “Friendly Token”: Given Name

It is important to consider the [various misconceptions](http://www.kalzumeus.com/2010/06/17/falsehoods-programmers-believe-about-names/) applications have about names, especially those unfamiliar with the joy of i18n and localisation. As it is not particularly important to the WLUX application that we know the user’s personal information, we can use a much simpler “Friendly Token” and ask the user how they would like to be referred to in the application’s UI.

#### 8Database Mapping

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Column Type | Column Index | Column Constraint |
| user\_id | ID/PK | Yes | PK Default |
| user\_email | VARCHAR(254) | Yes | UNIQUE |
| user\_password | VARCHAR(255) | No | NO |
| user\_name | TEXT | No | NO |

The above table indicates how this aforementioned information may be stored in the database.

## Authentication and Security

For first-party authorisation, we need only host a simple log-on form and basic authentication experience. The most important security consdierations are the storage of passwords at rest and inflight encryption of user authentication data.

### Password Storage at Rest

All passwords in the database should be hashed and salted, with the hash regenerated every time the user logs in. Do not rely on your own hash implementation, but defer to an existing encryption solution, e.g. PHP’s [crypt](http://php.net/manual/en/function.crypt.php).

For an example implementation, this article on [secure password storage with PHP and MySQL provides guidance.](http://alias.io/2010/01/store-passwords-safely-with-php-and-mysql/)

### Inflight Encryption

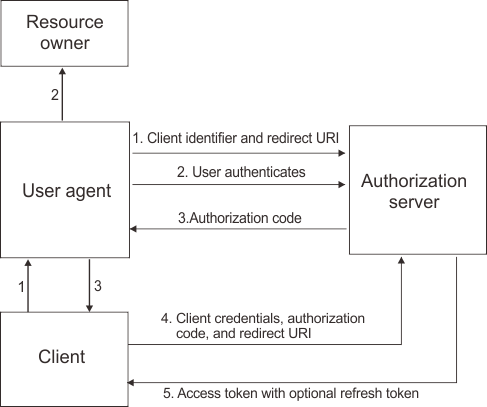
To avoid exposing user credentials to MITM and other proxy attacks, all requests to resources that necessitate or relate to authentication, including the authentication endpoints, should require an HTTPS connection. Non-HTTPs connections should be rejected.

### Logging Concerns

The form data of an authentication request **should never be logged.** Stacktraces, warnings, etc. must all be sanitised to ensure that user data and PII is not inadvertently recorded or exposed to those without direct database access.

#### API Client Authentication

In order to allow third-parties to interact with the WLUX API, e.g. to pull study results into an existing system for analysis, create alternative Dashboard implementations, or otherwise grant access without revealing their own credentials, it is suggested that authentication of API clients should follow the [OAuth 2.0 Specification](http://oauth.net/2/). This behaviour will be discussed in another specification.



## Authorisation and Gating

Several resources in the WLUX platform require authentication for should be limited to authorised users only (i.e. those who created or have granted access data to that data). Within the current scope these are primarily:

* Study Configuration Resources
  + Study CRUD Operations that change study properties, such as:
    - Adding or Removing a Task, Gratuity, or Study Variable
* Authorisation & Gating
  + Request Lifecycle and Authorisation
  + Gating Behaviour