

# Design Iteration 4

*Open-source web-app for questionnaires and surveys*

---

Group 1000<sub>2</sub>:

Kevin Brotcke

Maria Carrasco

Andrew Furusawa

Joshua Papa

Version: 3.0

Winter 2011

University of California Irvine

# Contents

|  |           |
|--|-----------|
| <b>1. Introduction</b>                               | <b>3</b>  |
| 1.1. Purpose . . . . .                               | 3         |
| 1.2. Project Description . . . . .                   | 3         |
| <b>2. Data Design</b>                                | <b>4</b>  |
| 2.1. Database Description . . . . .                  | 4         |
| 2.2. Temporary Data Structures . . . . .             | 4         |
| <b>3. Design Pattern : MVC</b>                       | <b>5</b>  |
| <b>4. Architectural and Component-level Design</b>   | <b>7</b>  |
| 4.1. System Structure . . . . .                      | 7         |
| 4.1.1. Architecture Diagram . . . . .                | 7         |
| 4.2. Class Diagram . . . . .                         | 9         |
| 4.3. Sequence Diagrams . . . . .                     | 10        |
| 4.3.1. Request Survey . . . . .                      | 10        |
| 4.3.2. Submit Survey . . . . .                       | 11        |
| 4.3.3. Login . . . . .                               | 11        |
| 4.3.4. Upload Experiments . . . . .                  | 12        |
| 4.3.5. Download Experiments . . . . .                | 12        |
| 4.3.6. View Experiments . . . . .                    | 13        |
| 4.4. Activity Diagrams . . . . .                     | 14        |
| 4.4.1. Survey creation . . . . .                     | 14        |
| 4.4.2. Survey Management . . . . .                   | 15        |
| 4.5. Use Case Diagrams . . . . .                     | 16        |
| <b>5. Restrictions, Limitations, and Constraints</b> | <b>18</b> |

## CONTENTS

---

|           |   |           |
|-----------|---|-----------|
| 5.1.      | Description For Experiment Component . . . . .    | 18        |
| 5.2.      | Description For User Component . . . . .          | 18        |
| 5.3.      | Description For Administrator Component . . . . . | 19        |
| 5.4.      | Description for the Output File . . . . .         | 19        |
| <b>6.</b> | <b>User Interface Design</b>                      | <b>20</b> |
| 6.1.      | Description Of The User Interface . . . . .       | 20        |
| 6.1.1.    | Screen Images . . . . .                           | 21        |

# 1

## Introduction

### 1.1 Purpose

---

This document describes the design layout for the Survey Generator program by Team 1000 for their Informatics 117 class during Winter 2011. Major design components such as the data design, architectural and component-level design, and user interface design will be discussed. This is meant as a reference for team members during implementation as well evidence for stakeholders.

### 1.2 Project Description

---

The project uses a database back-end to generate surveys on-the-fly using a web app. The main components consists of a SQLite database, HTML front-end of surveys and administrator panel, and business logic to generate the surveys.

# 2

## Data Design

### 2.1 Database Description

---

The data will be stored in the form of an embedded SQLite database.  
(See figure 4.2)

### 2.2 Temporary Data Structures

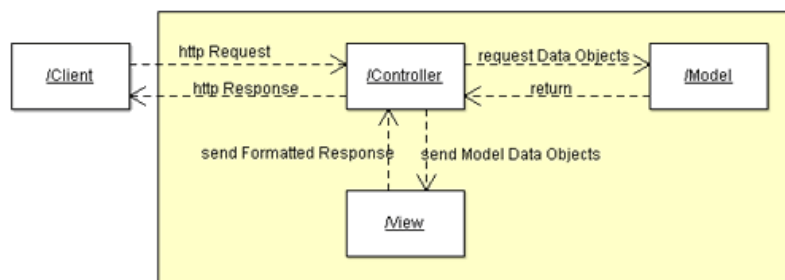
---

This will need to import and export XML files in the form XLSX as well as CSV files. These will only be temporary since they are being converted into and out of the SQLite database.

# 3

## Design Pattern : MVC

For our application we decided to use the *Model View Controller* pattern. This decision was made based on the fact that this pattern is the most used for today's world web applications.



**Caption :** MVC Collaboration Diagram.

The *MVC* pattern divides our application in three main modules:

### Model

This module is responsible for managing the data. It is responsible for storing and retrieving information from the database, in our case *SQLite*.

---

## View

The view is the module that display the data provided by the model in a specific format.

## Controller

The controller is responsible to handle the other two models working together. This module receives a request from a client, invokes the *model* to perform the requested operations and send the resulting data to the *view*. The view is in charged to format the resulting data to be presented to the user. This format is in HTML format.

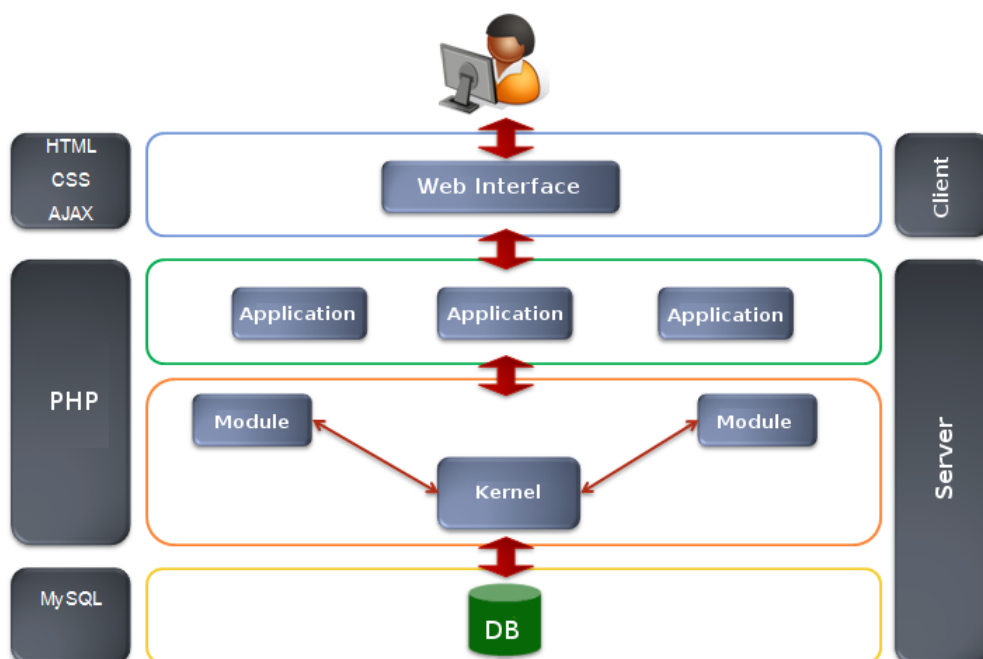
# 4

## Architectural and Component-level Design

### 4.1 System Structure

---

#### 4.1.1 Architecture Diagram



**Caption :** Block Diagram showing major components.



## 4.1. SYSTEM STRUCTURE

---

The figure shows the *Architectural Diagram*. This diagram shows the main components in our application. The main characteristic the application will be divided in different levels of interaction.

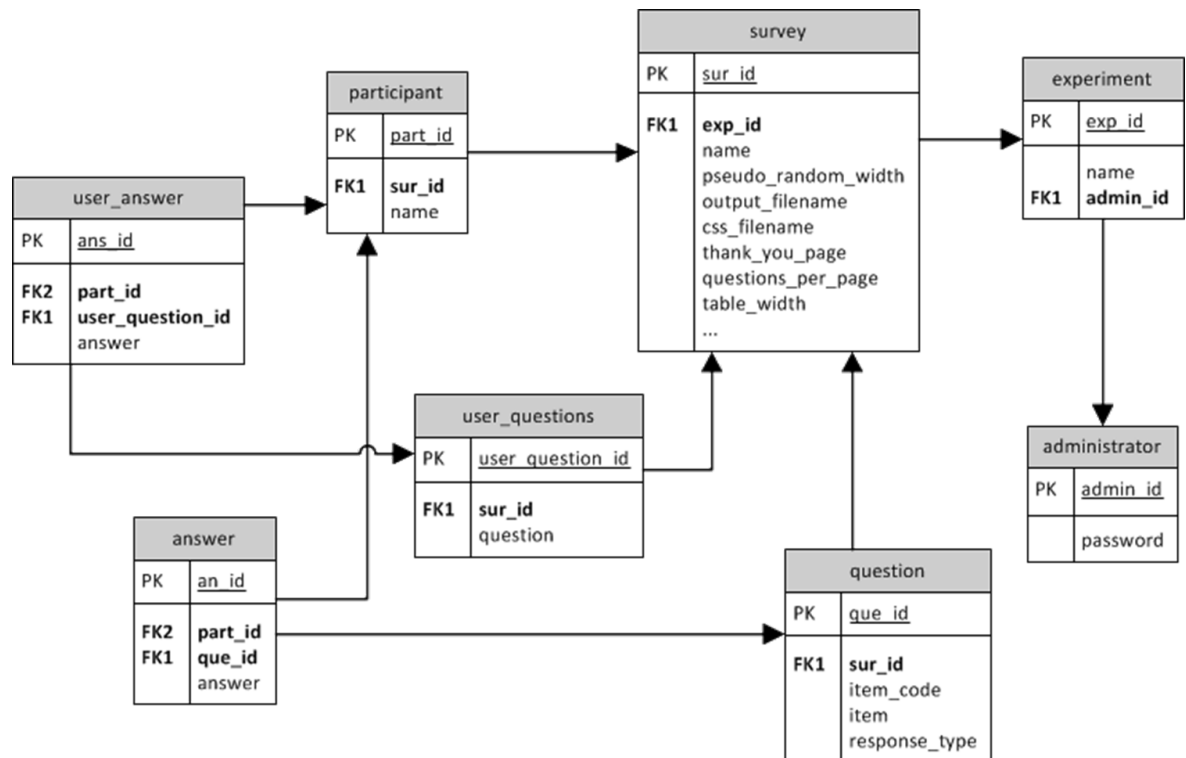
The database will only interact with the *Modules*, that will be responsible for retrieving and storing data from the database.

The *Application* will be responsible for controlling the interaction between the user and the database. Through the *Web Interface* the user will request information that the application will receive invoking the correct module.

The tools used to develop the System are HTML, CSS, PHP and SQLite.

## 4.2. CLASS DIAGRAM

### 4.2 Class Diagram



Caption : Class Diagram.

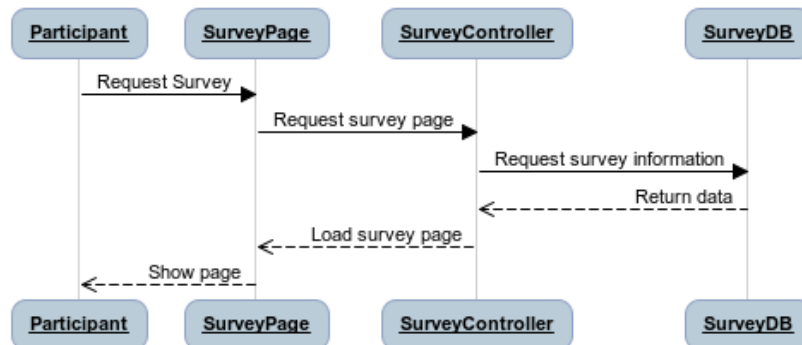
---

## 4.3 Sequence Diagrams

---

These interaction diagrams show the order of operation of the different processes or objects in our system. They interact with each other by sending messages (represented by the horizontal arrows).

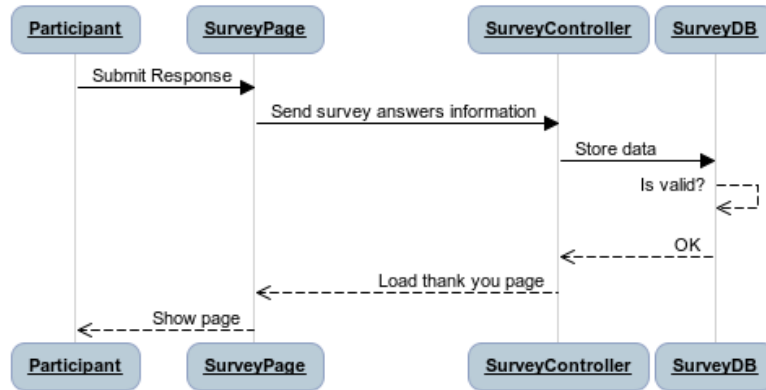
### 4.3.1 Request Survey



**Caption :** Request Survey Sequence Diagram.

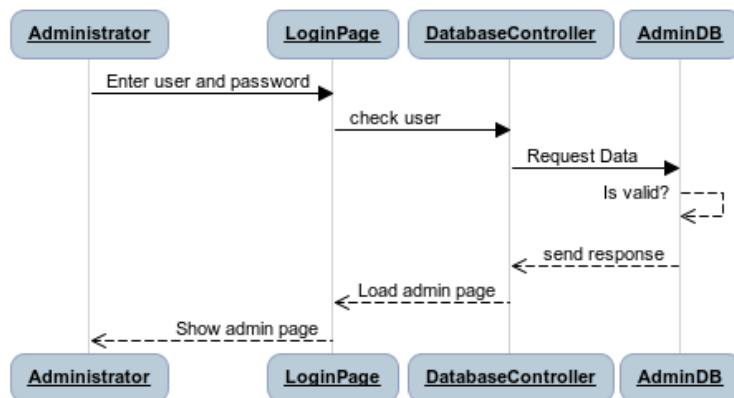
## 4.3. SEQUENCE DIAGRAMS

### 4.3.2 Submit Survey



Caption : Submit Survey Sequence Diagram.

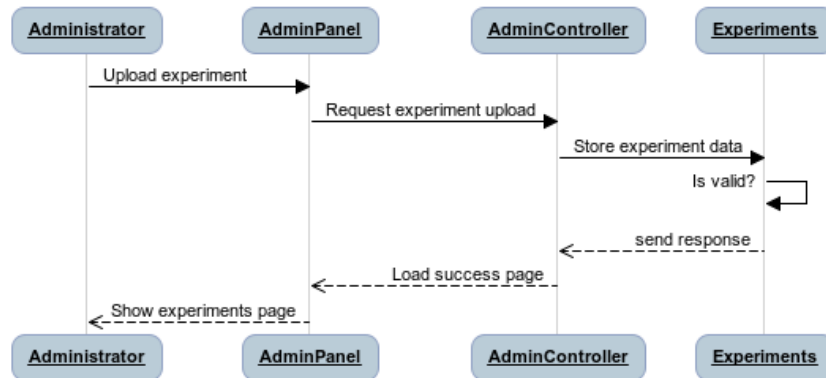
### 4.3.3 Login



Caption : Administrator's Login.

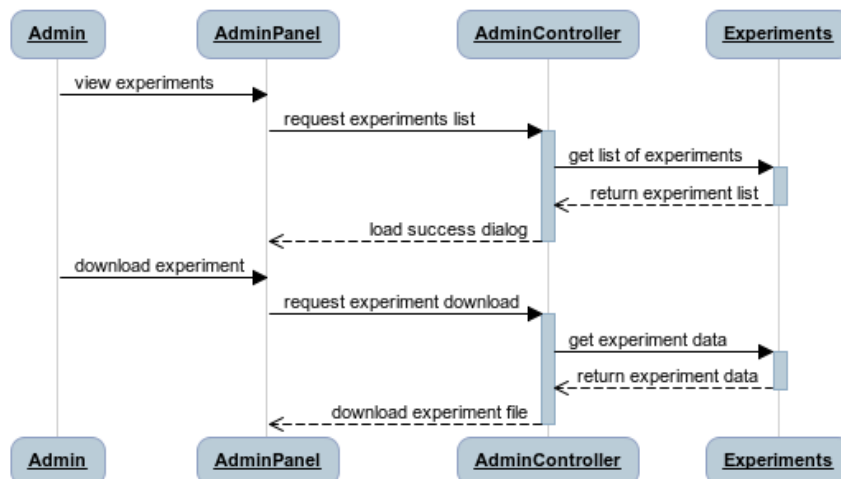
## 4.3. SEQUENCE DIAGRAMS

### 4.3.4 Upload Experiments



Caption : Upload experiment.

### 4.3.5 Download Experiments

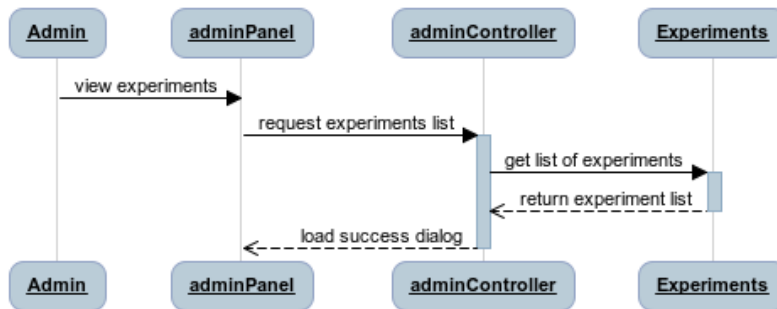


Caption : Download Experiment.

---

## 4.3. SEQUENCE DIAGRAMS

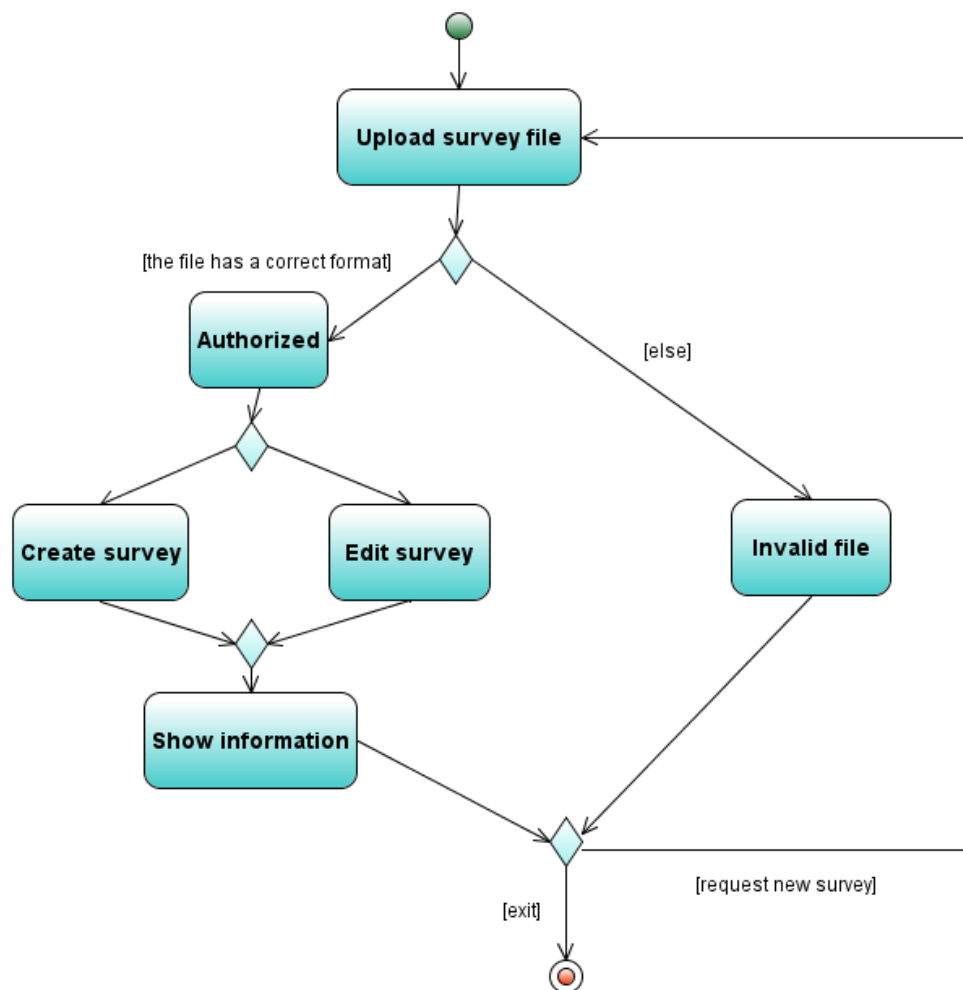
### 4.3.6 View Experiments



Caption : View Experiments.

## 4.4 Activity Diagrams

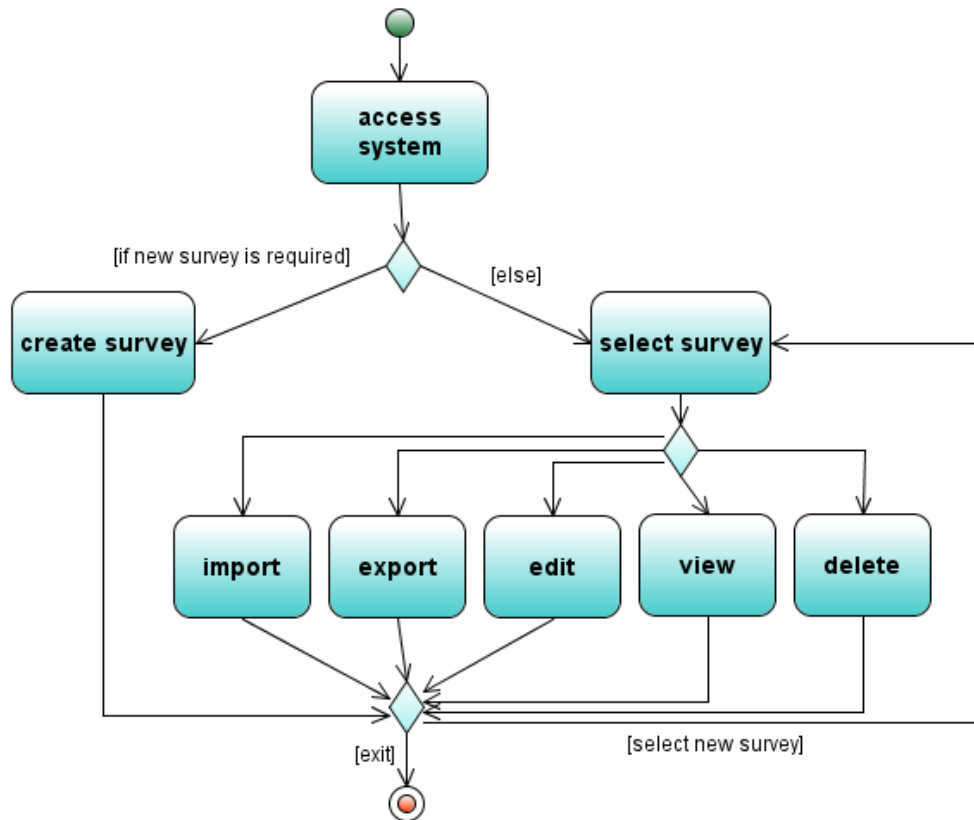
### 4.4.1 Survey creation



**Caption :** Activity Diagram for survey creation.

This is a graphical representation of workflows of the activity of survey creation. It shows the steps needed to be followed in order to upload a new survey (or as we call it *experiment*). The administrator selects a file to upload. The system will check the validity of the file format, if correct the user can either make final changes or create it. If the format is incorrect, the system will show the error and request a new file.

## 4.4.2 Survey Management



**Caption :** Activity Diagram for survey management.

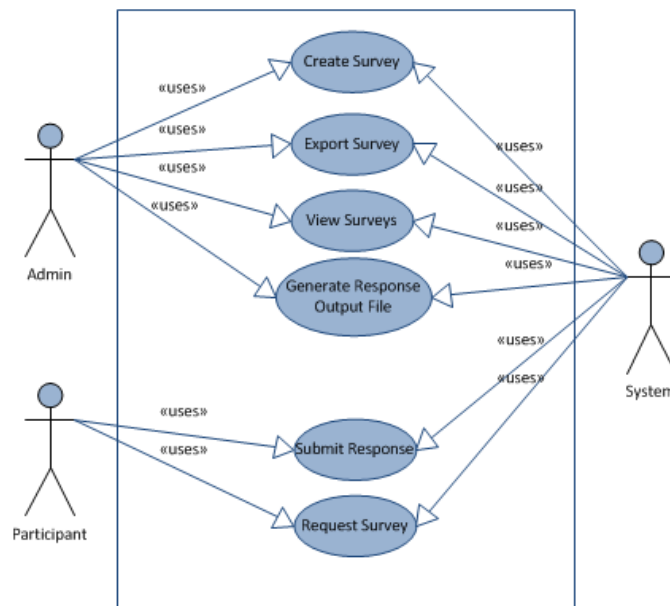
This activity diagram represents operational step-by-step work-flows of the survey management. The main options that the system will allow the administrator to perform related to experiments is their creation or edition. In order to perform basic operations such as *import*, *export* or *delete*, you need to select the survey to which you are going to apply the operations.



---

## 4.5 Use Case Diagrams

---



**Caption :** Main Use Case Diagram.

This use case shows the main basic functions of our system. There are two main actors that will interact with the system, the *administrator* and the *participant*. Each of them will be able to perform some tasks, as specified by the previous figure.

---

## 4.5. USE CASE DIAGRAMS

---



**Caption :** Survey Management.

This use case represents the basic interaction of the administrator with the Surveys. The surveys is the main focus of our application, that is why the Administrator has capabilities for total interaction with them.

# 5

## Restrictions, Limitations, and Constraints

### 5.1 Description For Experiment Component

---

An experiment will consist in a set of surveys. Each survey should be able to contain three types of inputs:

**Text box.** It will consists of 0 or more integers.

**Radio box.** Containing one number. It is inclusive.

**Yes/No radio box.** Allowing a two choice answer.

The system will support infinite surveys and experiments.

### 5.2 Description For User Component

---

The user or participant shall be able to perform the following actions:

- Request Survey by URL  
Correct survey automatically assigned to user based off URL.
- Submit Survey Response

## 5.3 Description For Administrator Component

---

Although our design will allow the possibility of future additions of administrators easily, initially we will only have one. The administrator shall be able to do the following actions:

- Create Survey  
transfer a new `xlsx` file via ftp to an input folder
- Edit Survey  
overrides the existing `xlsx` file
- View Survey  
downloads the `xlsx` file from the server
- Download Results

## 5.4 Description for the Output File

---

The exported results will be in `csv` format. The mush contain the following restrictions:

- Additional column for survey title.
- Do not have to support excel.

# 6

## User Interface Design

### 6.1 Description Of The User Interface

---

The main features that the software will present are:

- *Sets of surveys*, allowing the user to automatically loop through them.
- *Pseudo-Randomization* of the order of the surveys.
- *Result file*. This file will contain these main fields: Subject ID, survey number, code and responses.

#### Appearance Of Surveys

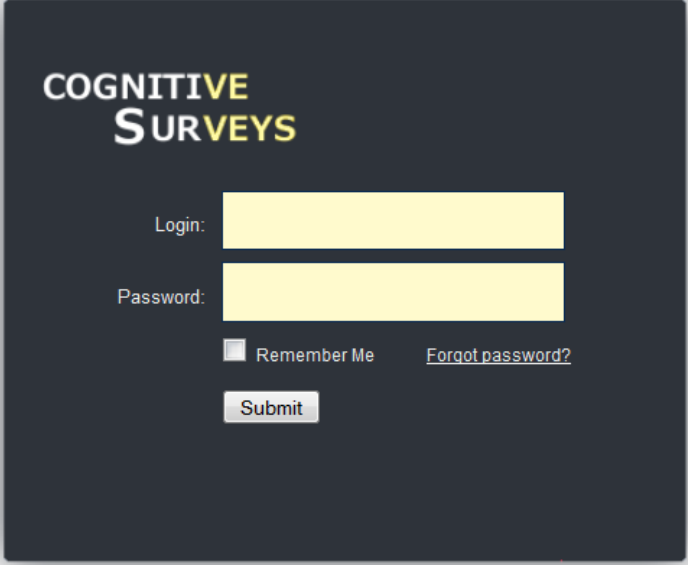
- the appearance of the sentences on the web form (font, size, color, spacing, location, etc) needs to be customizable through CSS/HTML tags
- the web form needs to accept a couple of different response types (radio buttons, text boxes, etc)
- Left/Center/Right justifiable
- autocomplete off
- css template file option
- sentences per page

---

## 6.1. DESCRIPTION OF THE USER INTERFACE

### 6.1.1 Screen Images

#### Login Screen



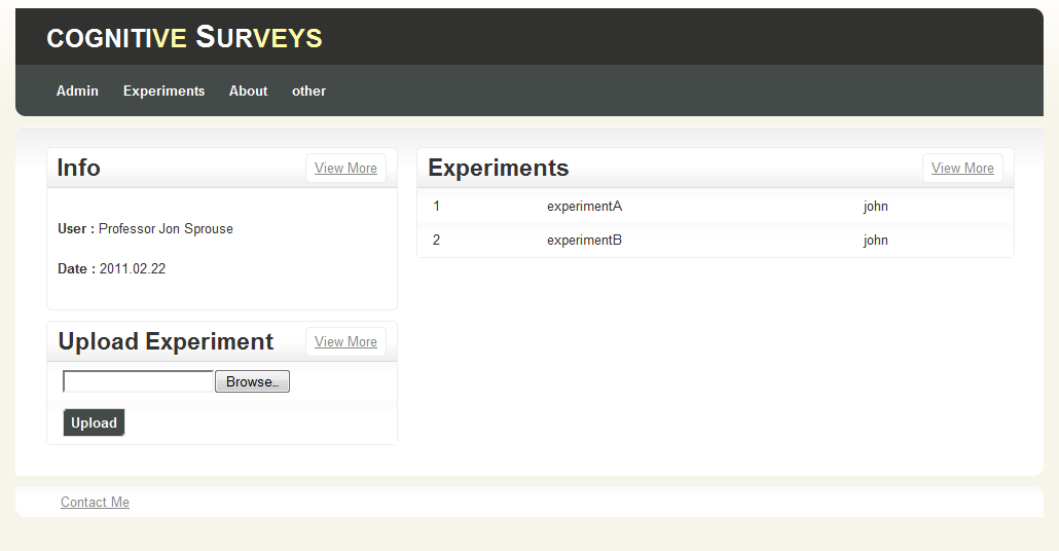
The image shows a login screen for an application titled "COGNITIVE SURVEYS". The title is in a bold, sans-serif font, with "COGNITIVE" in white and "SURVEYS" in yellow. Below the title, there are two yellow input fields. The first field is labeled "Login:" and the second field is labeled "Password:". Below the password field, there is a checkbox labeled "Remember Me" and a link labeled "Forgot password?". At the bottom of the form, there is a yellow button labeled "Submit". The entire form is set against a dark blue background.

**Caption :** Login Screen.

This is the first page that appears to the administrator while the application. It will request an user name and the password. If the input is correct (matches with the records in the database), the main administration panel will show up.

## 6.1. DESCRIPTION OF THE USER INTERFACE

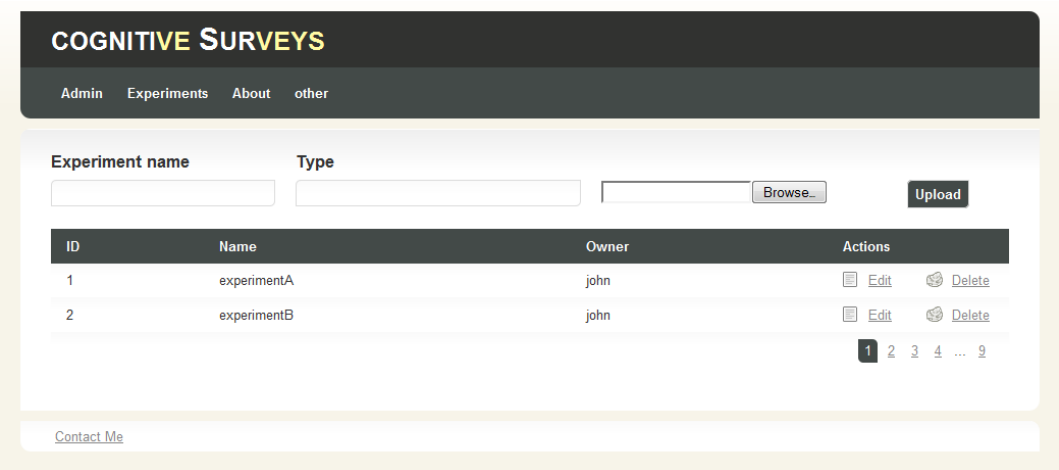
### Admin Screen



**Caption :** Adminsrator Screen.

Main page where the administrator can perform the main operations of survey management.

### List of Experiments Screen



**Caption :** Screen showing list of experiments.

## 6.1. DESCRIPTION OF THE USER INTERFACE

---

### Survey Screen

The screenshot displays the 'COGNITIVE SURVEYS' web application. The header is dark grey with the title 'COGNITIVE SURVEYS' in yellow and white. Below the header is a navigation bar with links: 'Admin', 'Experiments', 'About', and 'other'. The main content area is titled 'Submit Survey'. A prominent red error message bar at the top of the form states 'Something went wrong.' Below this, the form contains three input fields: 'Name' (with a hint 'Must contain alpha characters.'), 'Location' (with a hint 'Must contain alpha characters.'), and 'Major' (a dropdown menu currently showing 'Computer Engineering'). Below these is a large text area for 'Question' (with a hint 'Will be displayed in search engine results.') and another text area for 'Other Question' (with a hint 'Markdown Syntax.'). At the bottom of the form are two buttons: 'Reset' and 'Send'. A footer bar at the very bottom contains a link 'Contact Me'.

**Caption :** Survey Screen.

Sample screen of how a survey might look like. The red high-lighted text shows when the user sent the survey but he did not fill all fields correctly.