

CSC 642 HCI class SFSU  
Forms assignment  
Fall 2016  
Patrick Aung  
Patbak15@mail.sfsu.edu  
12/15/2016

WEBSITE URL: <http://sfsuswe.com/~patbak15/patbak15.github.io/>

## Introduction:

This report was to practice creating readable and user friendly forms. This assignment was to test our ability in creating a simple form that has to take user's input and validate each of the fields. In this report I will cover the steps I took into developing this form and what tools and processes I used to make sure that each of the forms fields were properly validated. Since the class went through many 'exploratory' options creating forms was an assignment that could be used for our portfolio for software engineering skills. By creating a form, we focus on making the form readable and following user centered design practices allowing the user to easily fill out the required information. Many users fill out forms on an everyday basis (e.g. registration forms) so creating a form from scratch shows that we are focusing on making an interface that people are able to understand and not get frustrated over inputting information.

## UI Design Principles and Patterns used

The assignment's guidelines showed that two main categories of information were being collected in the form which were personal information and information about an incident. The use of grouping of similar information helped by having one section filled with the user's information and another section holding the incident report. After grouping these two categories together I wanted to ensure the user is able to input each item in a reasonable method. While most of the inputs are textboxes I thought that the use of radio buttons for the gender and select picker for the 3+ items was a faster way for a user to input the information.

After reading the 10 commandments of good form design I was able to apply the same ideas to creating each input that is required in the incident form. To make it easier for the user I made each entry aligned to the same spot. Users of the form can easily identify what each input was asking for by looking at the left of the input. Each input box is shaped so it would about the same size as the required input and followed 32px height to be easily clickable. This especially helps with the entries for address area. I also used asterisks to help identify the mandatory items that are needed for the form.

### List of Principles Used

1. 10 Commandments of Good Form Design
  - i) Clear visible labels
  - ii) 14+px font for readability
  - iii) Input Fields have at least 32px height
  - iv) Size Input fields to expected input
  - v) No customization to checkboxes/radio buttons
  - vi) General Errors with highlighted section and reason
  - vii) Asterisk to show mandatory fields
  - viii) Hide non needed items
  - ix) Minimize user input with autofill options
  - ix) Clear showing of what type of input is needed
2. Grouping of similar items (e.g. Personal Information, Relevant topics)
3. Alignment of labels on form

## Screen Shots of Final Version

<http://sfsuswe.com/~patbak15/patbak15.github.io/>

SFSU CS Department - CSC 642 Fall 2016 Project: Form UI  
Patrick Aung, patbak15@mail.sfsu.edu

Please record the problem by filling the form below

### Personal Information

\*required

First Name*	<input type="text" value="First Name"/>
Last Name*	<input type="text" value="Last Name"/>
Gender	<input type="radio"/> Female <input type="radio"/> Male <input type="radio"/> Other
Education	<input type="text" value="Highest Education"/>
Phone*	<input type="text" value="(999)999-9999"/> Format: (999)999-9999
Email*	<input type="text" value="name@domain.com"/>

### Incident Report

\*required

Date\*

### Location of Incident

Address*	<input type="text" value="123 Street Ave"/>
City/Town*	<input type="text" value="city"/>
State*	<input type="text" value="Select a state"/>
Zip*	<input type="text" value="99999"/>
Catagory*	<input type="text" value="Spillage"/> Broken Pipe Electric Hazard Broken Pathway Other

Description\*

Limit is 500 characters

Upload Image\*  No file chosen

☐ I have read and agree to the [Terms and Conditions](#)

☐ I'm not a robot



## Verification Page

SFSU CS Department - CSC 642 Fall 2016 Project: Form UI  
Patrick Aung, patbak15@mail.sfsu.edu

### VERIFICATION PAGE

#### Personal Infomation

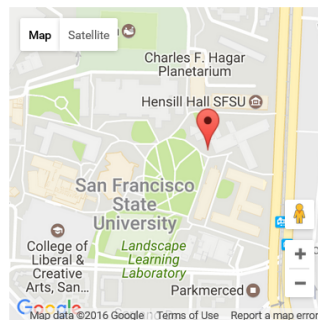
First Name*	Patrick
Last Name*	Aung
Gender	male
Education*	M.S.
Phone*	(555)555-5555
Email*	name@domain.com

#### Incident Report

Date*	03/15/2009
-------	------------

#### Location of Incident

##### Google Maps



Address*	1600 Holloway Dr
City/Town*	San Francisco
State*	CA
Zip*	94132
Catagory*	Broken Pipe
Description*	Broken
Image*	



## Software Tools

### Lamp Stack (AWS Server)

- Linux: Open source operating system that was used on the host server (Amazon Web Services)
- Apache: Component of the LAMP stack web server
- PHP: Programming language that was used to communicate the information that was inputted into the form and displayed on the result.php page.

### API's and Frameworks

- jQuery: Javascript open source library that was used to handle the image processing of the form.
- Bootstrap: Front-end framework that handled the format of the initial form to combine all the web frameworks (e.g. JavaScript, HTML, etc.)
- Google Maps API: API to handle displaying the resulting map to display the location of the incident. The API was to take a resulting address and to display a map of the location on the result.php page.

### QA Automation

- Selenium Mozilla Add-on (<https://addons.mozilla.org/en-US/firefox/addon/selenium-ide/>)

## Implementation

The techniques I used were to mainly stay within the basic web design languages (e.g. HTML, CSS, Bootstrap). Using these design languages to design a “simple” form allowed me to not have to worry about browser compliance. I frequently checked each API’s documentation online and would look at the browser capability. To help with rendering and sizing I used the “<meta name=“viewport” content=“width=device-width, initial-scale=1”>” line to make sure that the form would keep its size regardless of the device screen.

```
Status: Done. Document processed in 6.82 seconds.

Checking link http://www.sfsuswe.com/~patbak15/patbak15.github.io/validator.js
HEAD http://www.sfsuswe.com/~patbak15/patbak15.github.io/validator.js  fetched in 0.02 seconds

Checking link https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js
HEAD https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js  fetched in 1.14 seconds

Checking link http://code.jquery.com/jquery-1.5.js
HEAD http://code.jquery.com/jquery-1.5.js  fetched in 1.02 seconds

Checking link https://www.google.com/recaptcha/api.js
HEAD https://www.google.com/recaptcha/api.js  fetched in 1.21 seconds

Processed in 6.82 seconds.
```

## Results

### Links

Valid links!

*1: W3 validator output*

## Implementation Resources

FileReader API (shows browser compatibility): Used for Images

<https://developer.mozilla.org/en-US/docs/Web/API/FileReader>

W3 Validator (HTML validation)

<https://validator.w3.org>

## QA Test Plan

Test #	Title	Description	Input	Desired Output	Result
1	Phone Number Input	Open webpage: <a href="http://sfsuswe.com/~patbak15/patbak15.github.io">http://sfsuswe.com/~patbak15/patbak15.github.io</a>  Enter Phone Number in Field	(555)555-5555	“No warning signs”	PASS
2	Indicate Required Fields	Open webpage: <a href="http://sfsuswe.com/~patbak15/patbak15.github.io">http://sfsuswe.com/~patbak15/patbak15.github.io</a>  Hit Submit button (No values test)	“Submit”	“Warning Required Fields” (site should have labels turn red)	PASS
3	Open Terms & Conditions Page	Open webpage: <a href="http://sfsuswe.com/~patbak15/patbak15.github.io">http://sfsuswe.com/~patbak15/patbak15.github.io</a>  Open Terms and Conditions Page	“click page link”	“Page Redirects to black page Titled “Terms and Conditions”	PASS
4	Incorrect Phone Input	Open webpage: <a href="http://sfsuswe.com/~patbak15/patbak15.github.io">http://sfsuswe.com/~patbak15/patbak15.github.io</a>  Input into phone number	5555555555	“Please match the requested format: Phone Number (Format: (999)999-9999).” Also Label Turns red	PASS
5	Invalid Address	Open webpage: <a href="http://sfsuswe.com/~patbak15/patbak15.github.io">http://sfsuswe.com/~patbak15/patbak15.github.io</a> Input all valid fields Then Input nonexistent address	1600 abc drive, San Francisco California 94132	“Alert” Geocode Failed ZERO RESULTS	PASS



## Usability Test Plan

Please circle the answer that best describes your feelings to the following statements.

**Question 1:** It was easy to navigate through the form and fill each of the entries.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
----------------	-------	---------	----------	-------------------

Comments:

**Question 2:** I was able to easily upload and image and it displayed correctly on the result page.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
----------------	-------	---------	----------	-------------------

Comments:

**Question 3:** The form indicated the exact format required for each field.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
----------------	-------	---------	----------	-------------------

Comments:

# Automated UI QA testing (Selenium)

TestRun, all values (untitled suite) - Selenium IDE 2.9.1

File Edit Actions Options Help

Base URL <http://sfsuswe.com/>

Fast Slow

Test Case

TestRun, al...

Command	Target	Value
open	/~patbak15/patbak15.github.io/	
type	name=fName	Patrick
type	name=lName	Aung
click	xpath=//input[@id='gender']][2]	
select	id=education	label=M.S.
type	id=phone	(555)555-5555
type	id=email	name@domain.com
type	id=date	03/15/2009
type	id=address	1600 Holloway Dr
type	id=city	San Francisco
select	id=state	label=California
type	id=zip	94132
addSelection	id=catagory	label=Broken Pipe
type	id=description	Broken
type	id=image-input	C:\Users\patbak\Desktop\Images\kitchen-1224845_640.jpg
click	id=agree	
selectFrame	undefined	
click	css=div.recaptcha-checkbox-checkmark	

Command type

Target name=fName

Value Patrick

Runs: 1

Failures: 0

## 2. Test Run with Simple Values (No Failures)

Table Source		
Command	Target	Value
open	/~patbak15/patbak15.github.io/	
type	name=fName	John
type	name=lName	Doe
click	xpath=//input[@id='gender']][3]	
select	id=education	label=Ph.D
type	id=phone	(555)555-5555
type	id=email	hello@world.com
type	id=date	02/04/2016
type	id=address	1600 Holloway Dr
type	id=city	San Francisco
select	id=state	label=California
type	id=zip	94132
addSelection	id=catagory	label=Other
type	id=description	Hi My name is John Doe!
type	id=image-input	C:\Users\patbak\Desktop\Images\kitchen-1224845_640.jpg
click	id=agree	
selectFrame	undefined	
click	css=div.recaptcha-checkbox-checkmark	
Command	click	
Target	id=agree	Select Find
Value		

### 3: Selenium Test with different values

SFSU CS Department - CSC 642 Fall 2016 Project: Form UI  
Patrick Aung, patbak15@mail.sfsu.edu

Please record the problem by filling the form below

Personal Information

\*required

First Name*	<input type="text" value="John"/>
Last Name*	<input type="text" value="Doe"/>
Gender	<input type="radio"/> female <input type="radio"/> male <input checked="" type="radio"/> other
Education	<input type="text" value="Ph.D"/> <input type="button" value="v"/>
Phone*	<input type="text" value="(555)555-5555"/> <small>Format: (999)999-9999</small>
Email*	<input type="text" value="hello@world.com"/>

Incident Report

\*required

Date\*

Location of Incident

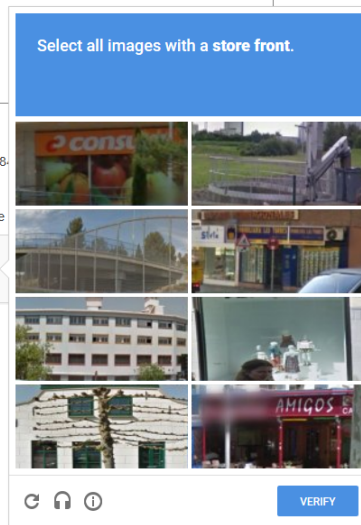
Address*	<input type="text" value="1600 Holloway Dr"/>
City/Town*	<input type="text" value="San Francisco"/>
State*	<input type="text" value="California"/> <input type="button" value="v"/>
Zip*	<input type="text" value="94132"/>
Catagory*	<div>Spillage Broken Pipe Electric Hazard Broken Pathway <b>Other</b></div>

Description\*

Upload Image\*  kitchen-12248

☒ I have read and agree to the

☐



4: Selenium Case 2 results, NOTE: CAPTCHA is activated, needed to be manually completed

## Resources:

### 10 Commandments of UI design

- [https://mono.company/wp-content/cache/page\\_enhanced/mono.company//journal/design-practice/the-10-commandments-of-good-form-design-on-the-web//\\_index.html\\_gzip](https://mono.company/wp-content/cache/page_enhanced/mono.company//journal/design-practice/the-10-commandments-of-good-form-design-on-the-web//_index.html_gzip)

### Selenium

- <http://docs.seleniumhq.org/>

### HTML/PHP Resources

- <http://www.w3schools.com/html/>
- <http://www.w3schools.com/php/>

### Google Maps API

- <https://developers.google.com/maps/>
- <https://developers.google.com/maps/documentation/geocoding/intro>

### Bootstrap Validator

- <https://1000hz.github.io/bootstrap-validator/>