CSC 642-842 Human Computer Interaction individual project assignment

Prof. D. Petkovic

Fall 2021

20+2 bonus/100 grade points

10/16/21

DUE: TBD (likely late November)

1. Goals of the assignment

General goal of this assignment is for students to learn the design and the implementation of a typical form (a common GUI component) *at professional quality* in terms of design, implementation, dealing with user errors, field validation and robustness to attacks, cross browser and platform compatibility and usability. Students may also practice (optionally) automated GUI testing.

This assignment focuses on design, implementation and testing of WWW *data survey form* following given specifications.

Results of this assignment will be valuable for student portfolio as well.

2. Assignment description

This assignment consists of several parts:

- Design and development of WWW data survey form as per given specifications and following best practices for forms design and development (consult class slides)
- Design and development of WWW results verification form, showing entered data, for verification purposes
- Automated testing using UI test automation (optional, for extra points)

Personae (who is a typical user, their skills and pain points)
User of any gender and age, with medium skills in using WWW. Moderate level of anxiety with use of WWW and generally very busy hence does not have much time. Conscious of not entering wrong data so wants to verify input before submitting it.

Use case - context:
Your company is doing voluntary survey of customers over the WWW. You know the customers are busy and your goal is to make the survey easy to use e.g. to engage as many participants as possible. You offer input verification page before submission to make sure data is entered correctly. Ease of use I critical since survey is on voluntary basis. Correctness of input data is also paramount including enforcement of mandatory inputs. Primary platform is WWW browser but form also must render well on mobile devices (e.g. WWW version must be responsive to different platforms)

2.1. Design, development and automated testing of WWW *data survey form*

Students shall develop two WWW pages as follows: a) **data survey form** for user input, with the specifications below; and b) **results verification form** displaying captured information from the form a).

Specifications:
Input fields for the form a) are provided below and must be followed precisely including the order of the fields. Fields have to be validated (details below). It is your job to design the form layout, error handling, field validation, enforcement of mandatory fields and all details following The design and implementation must follow UI principles and best practices for forms design (check class slides) as well be consistent with above persona and use case. Both forms a) and b) shall be designed primarily for the WWW browser delivery on laptops but also must render well on mobile device (e.g. your code must be responsive)

a) **Data survey form**:

Title:

"CSC 642 848 Fall 2021 Individual Assignment <student first last name here>"
"Data survey form"

Registration for shall contain the following fields, (field names are underlined) * denotes mandatory fields:

- User last name then first name * (validate each for up to 40 characters)
- Chose preferred title * (*options: none; student; professor; staff; retired)* *
- Your Height (feet, inches)
- Phone (validate to 10 digits)

- Address* (validate to each entry up to 40 alpha numeric characters; Zip code: positive 5 digit number)
- Check all services you require (*options: e-mail; phone; Facebook; Tweeter;, surface mail; personal visit*)
- Your monthly budget for services (*options: less than $ 50, between $ 50 and $ 100; above $ 100*)
- Your e-mail (verify basic e-mail format e.g. has @ sign et)*
- "I Agree to terms" check * - (has to ensure this has been checked, but no need to have actual terms page, the link can be dummy link*)*
- Captcha * (*use any open source captcha to prevent robots registering or design your own)*

Place then appropriate action buttons, where the main action button leads to form b) on a separate page

b) **Results verification form,** for verification of input data is to be shown after page a) has been submitted, in a separate WWW page or appended below the page a). This page will contain results for all input fields in a), arranged as per student design, with address being shown as text and as a map with one marker (red drop). This page will also be graded for UX and has to be well designed in terms of GUI and presentation since it shall be viewed by the user for quick verification.

Title:

"Results verification page <student first and last name>"

In the results form display of all the input data from form a) arranged according to UI principles., in same order as in form a.

Form design details (for a) and b)) are up to you. Please pay attention to the flowing (each will be graded):

- Proper UI design patterns and layout
- Following of best practices for WWW form design
- Proper text labels and instructions for each entry field
- Field validations:
  - first and last name: up to 40 characters;
  - phone: 10 positive digits; in format you specify;
  - address : each entry up to 40 alpha numeric characters; Zip code: positive 5 digit number,
  - e-mail: verify basic e-mail format e.g. has @ sign etc.
- Mandatory fields: Must enforced fields marked *

- Error handling: erroneous entry must be handled appropriately (consult best practices and class slides).

- Works on minimum two major browsers, last and one older version of each

- Responsive implementation: must render well on mobile devices (test it by resizing browsers)

- Correctness: data must be captured correctly (verified in page b)

Automated testing (optional – for extra points): Use some tool like Selenium and create automated test for the WWW form for minimum two fields of your form. Execute the test and provide screen shots of tool output showing test results to document you implemented automated testing.

Resources to help: see end of this assignment and also check class slides.

2.2. Implementation:

In terms of user interface design (layout, organization, error handling etc.) both forms a) and b) have to be designed following UI principles and patterns specific to forms (material as covered in the class, see also comments above and guidance from the resources below).

Students are required to develop pages a) and b) using their choice of a modern open source free framework which also provides its own functions for input field validation, responsive design, cross browser compatibility etc. We recommend bootstrap framework https://getbootstrap.com/ . **Note**: the primary platform is WWW browser but the both forms (a) and b)) also must render well on mobile device, hence your WWW implementation must be responsive. This can be verified by resizing the browser. For maps we suggest Google maps. For captcha/"I am not a robot" use any open source service, see resources, or design yours like answering a formula question.

Forms a) and b) shall be hosted on WWW server of student choice (has to be free), and this server may be shared among students members of a student team project. You can also use github to host and deliver this assignment app since it does not involve server backend. Data captured in form a) can be stored in any way, as long as it can be displayed back by the form b) and it does not have to be stored in the DB, you can simply use browser store. Make it simple!

Suggestion: use your github to store all the work related to this assignment to serve for your job search portfolio, e.g. a folder with the following information,:

- Readme
- Documentation (including this assignment and copy of your delivery e-mail as below),
- URL to run the app (make sure it points directly to the forms code and is accessible t all)
- Code for the form
- Code and results logs/images for automated testing if you do it.

Please test the access to the form and related info on some other laptop before submission to ensure it will work out for the instructor.

NOTE on asking for help: Students are allowed to ask advice from their team mates and to consult WWW resources **but they must design and develop the assignment on their own.** Students may share WWW servers set up by others in the team. All additional resources (besides those in references below) and names of the persons who helped must be reported in submission e-mail.

2.3. Automated testing using UI test automation (optional, for extra credit)

For extra credit, students can choose a tool for automated UI testing and write a test and execute it, on at least 2 fields of the form (e.g. two fields for test input in form a) then test for correct output in form b)). One suggestion is to use Selenium tool https://www.seleniumhq.org/ . As a proof of testing student must submit images of log of test results, as provided by chosen testing tool (see submission instructions below) and have automation code in assignment github.

NOTE: This assignment development will not include usability review of instructor prior to submission since this assignment is among other things testing ability of students to apply UI design principles on their own. Students can ask others for feedback though.

3. Assignment submission and delivery

Assignment shall be submitted by the deadline (TBD) via e-mail to Petkovic@sfsu.edu and ajsouza@sfsu.edu as follows (this process must be followed in all its details and text in quotes must be used as below or assignment will be returned). Please make sure you enter your name in subject line as noted below.

e-mail subject line: "CSC 642 842 Fall 2021 Individual Assignment <student first and last name>"

e-mail body text and content (required text is in "quotes", your input is in < >):

"Here is the link to test my individual assignment URL <*URL*>" **(Provided URL should directly execute the form). Make sure the URL is accessible to all (not only Prof. Petkovic))**

"For the development I used <*enter framework you used*>"

"For field validation I used <*explain*>"

"For maps I used <*explain*> "

"For captcha I used <*explain*>"

"For testing I used <*explain*> and tested the following form fields <*explain, minimum 2 fields must be tested*>" **(optional)**

"In the attachment please find screen shots of the results of the automated testing <*you would attach screen shots of your test logs – 1-2 pages*>"

"For this work I was helped by <*list names of students who helped you*>"

"Documentation and Code I developed for this assignment are in github <link to github folder with your assignment**> (make sure the link is accessible to all (not only Prof. Petkovic)**

"Regards"

<*your first and last name*>

1. Grading

Total points for this assignment are 20/100 (and 2 bonus points for automated testing), as follows:

- Form GUI design and ease of use for both pages a) and b) (e.g. following UI best practices and design patterns for forms (layout, alignment, handling of errors etc. – check details in section 2.1) - 9
- Correctness of operation (e.g. does the form work correctly including dealing with mandatory fields, field validation, cross browser compatibility, is the data captured correctly, input error handling) – 9
- Rendering on mobile device – 2
  Total: 20

- Automated Testing – 2 additional bonus points

Specific guidance for above grading rubrics is based on best practices and design patterns covered in the class and in references below. Graders will also spot check the code from github.

If graders submit the code has been coped that will be investigated and 0 may be given for the assignment.

Ways to store and retrieve input data will not be part of the grading, so students have flexibility here.

2. Resources

Here we provide some resources and of course students may also look for their own. Check class slides too

Design patterns for forms
http://mono.company/journal/design-practice/the-10-commandments-of-good-form-design-on-the-web/
https://www.nngroup.com/articles/web-form-design/
https://code.tutsplus.com/tutorials/validation-and-exception-handling-from-the-ui-to-the-backend--net-36697

Bootstrap
https://getbootstrap.com/
https://getbootstrap.com/docs/4.3/components/forms/

Selenium for automated UI testing
https://www.seleniumhq.org/
https://www.softwaretestinghelp.com/selenium-tutorial-1/

Google maps
https://cloud.google.com/maps-platform/

Some choices for captcha (use any you wish)
https://captcha.org/