CSC 642-842 Human Computer Interaction individual project assignment

Prof. D. Petkovic

Summer 2020

20/100 grade points

06-04-20

DUE: TBD (tentatively late July 2020)

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 will also practice (optionally) automated GUI testing.

This assignment focuses on design, implementation and testing of WWW *data survey form* from provided high level specs.

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

2. Assignment description

This assignment consists of several parts:

- Design and development of WWW data survey form
- 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. 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 enmforcement of mandatory inputs.

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

Students shall develop two WWW pages a) one page **data survey form** 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. It is your job to design the form layout, error handling, field validation, enforcement of mandatory fields and all details following UI principles and best practices and consistent with above persona and use case.

a) **Data survey form**:

Title:

"**CSC 642 Summer 2020 Individual Assignment <student first last name>"**
"Data survey form"

Registration for shall contain the following fields, * denotes mandatory fields:

- User first and last name *
- Address*
- Birth date *
- Education level *(options: high school, college, graduate studies, Ph.D)*
- Height (feet, inches)
- Phone *
- e-mail (and another field to confirm e-mail) *
- "I Agree to terms" check * - (has to ensure this ahs been checked, but no need to have actual terms page*)*
- Captcha (*use any open source captcha to prevent robots registering or design your own)*

Appropriate action buttons, leading to form b)

b) **Results verification form, for verification of input data** is to be shown after page a) has been submitted, in a separate 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 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 name>"

<follow with the display of all input data form a) arranged according to UI principles>

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: 7 positive digits;
  - 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).
- Responsive implementation: must render well on mobile devices (test it by resizing browsers)
- Works on specified browsers
- Correctness: data is captured correctly (verified in page b)

2.2. Non-functional specifications (constrains on design and development) must be followed:

1. Forms shall run on two major browsers of student choice (one has to be Mozilla)
2. Forms shall render well on iPhone and Android mobile devices (via their browsers) e.g. be *responsive* to change of client display size. Can test this by resizing browsers
3. Students shall follow best practices and design patterns for WWW forms to ensure ease of use (layout, error handling, field validation, enforcement of mandatory fields etc.)
4. No entry shall be longer than 40 characters
5. Page shall be accessible by instructors (test that it can be accessed by any user)

2.3. Implementation:

In terms of user interface design (layout, organization etc.) both forms a) and b) have to be designed following UI principles and patterns specific to forms (material will be covered in 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/ . For maps we suggest Google maps. For captcha/"I am not a robot" use any open source service, see resources.

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

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

Note: the primary platform is WWW browser but the both forms (a) and b)) must render well on mobile device, hence your implementation must be responsive. This can be verified by resizing the browser.

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.

2.4. 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 will submit images of log of test results, as provided by chosen testing tool (see submission instructions below).

NOTE: This assignment will not include usability review of instructor since it tests 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 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):

e-mail subject line: "CSC 642 Summer 2020 Individual Assignment <student first and last name>"

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

"Here is my individual assignment <*URL*>"

"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*>"

"Here I list resources I used in order to design, develop and test this assignment <*provide enumerated list of all resources you used (including those listed in references below), if you list URLs provide the last date it was accessed*>

"Regards"

<your first and last name>

6. 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) - 10
- 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 - 1
- Automated Testing – 2 bonus points

Specific guidance for above grading rubrics is based on best practices and design patterns covered in the class and in references below.

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

7. Resources

Here we provide some basic 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/