Input controls

There are many kinds of input controls. Write a 4-5 page paper in which you: •Explain the function of input controls. •Identify four (4) types of input control and explain the function of each. •Provide an example of a data integrity error that could occur if each of these types of input control were not in place. •Explain the advantages and disadvantages of restricting user interfaces. (User interfaces can often be restricted, limiting the user's ability to navigate to other areas of the system, or out of the system. •Design and build a graphical representation of a Web-based input for making a hotel reservation, using Visio or PowerPoint or an equivalent. •Research and cite at least three (3) reputableacademicsources. Darren Blake Week 6 Assignment CIS210 " An HTML form is a section of a document containing normal content, markup, and special elements called controls. " These controls are commonly referred to as input controls, according to the World Wide Web Consortium. There are many types of input controls that can be used in a web form. They help to provide a framework for the kind of data that will be submitted by users.

Selecting the correct input control for a data field is critical. Text input, select box, radio button, and password are four examples of input controls. As pointed out by Ponce de Leon, most input controls are visual and interactive. There is also something called a hidden input control. They can be used to store system critical data, such as database key data, that the user does not need to interact with. Text type input controls are used to input text. They provide a single-line input field in which any text can be entered. The text type input controls are useful for form data such as names, street addresses, and user names.

This data is viewable on the screen, so it ought to not be used for passwords. Select box input controls are extremely common in web forms. There are two basic types of select input controls: single select and multi-select. This type of input control provides a list of predetermined options that the user can select. They offer strict control of what can be entered into the form. They are used for items that have limited and predefined options. Good examples of this would be things like credit card type, country, state, and language. Usually, this type of input is used when the number of pre-defined options more than two.

If there are only two options, other types of input controls may be more appropriate. If the user is allowed to select multiple options, such as a list ofcareerfield interests, a select input control can easily be set up to permit multiple selections. Radio button input controls are always used within a group. This means that there should be more than one radio button that has the same name. When radio buttons have the same name and different values, only one can be selected at a time. They are used when there are few predefined options.

Predefined option sets of two are usually not put inside of a selection input control. For instance, the options for gender should usually be “ male” or “ female. ” It is more fitting in this case to use two radio buttons. This allows the user to enter their data with one click rather than the two which would be required with a select drop down input control. It is up to the programmer to decide if a select input control or a group of radio buttons is more suitable. In general, if the user can easily view all available options on a single line of the form, the programmer should seriously consider using radio buttons.

Alternatively, if there are enough options that it would p many lines they should be presented inside of a select input control, like selecting a state. Password input controls, on the surface, looks exactly like the text input control. They also, form an allowed content perspective, functions in an identical manner. However, password input controls hide the data that is entered into the control. This means that each keystroke within the control will result in a dot or star instead of the actual data. This is done to prevent other individuals, who may be able to see the user's computer screen, from viewing the password as plain text.

In order to insure the correct amount of keystrokes by the user, the star or dot remains on the screen. However, the text is not displayed for the world to see. Hidden input controls are extremely useful when performing data entry tasks with a database-driven web application. Often, the forms used to edit data are in reference to an entry within the database that has an integer primary key. This key is usually arbitrary integer that increments automatically, provides indexing, and has absolutely no meaning to the user. When the user selects to update the data, it is important that the systems knows what ID is being updated.

However, there is no reason to display this ID to the user. In order to submit the ID of the edited database record along with the modified form fields, the ID can be assigned to a hidden input control. Data integrity with input controls is achieved both by the nature of the controls themselves and basic script validation techniques. As far as scripting is concerned, each data field can be easily verified upon submit before sending the data to the server. For

the types of input controls chosen, selecting inappropriate input controls can result in data integrity issues. A text input control is rather straight-forward.

It is also the easiest field to realize data issues with. Obviously, you would not want to use a select input control for an individual’s name. However, using this type of control opens databases up to SQL injection attacks, entry of HTML entities, and entry of incorrect or bad data. With SQL injection and HTML entities, it is critical that the data entered is cleaned before being processed by the server. For a field like “ First Name”, entry of SQL or HTML should be identified and rejected. In general, you also wouldn’t want to use a password field for something like “ First Name. While it is great to be able to mask data, the user should be able to see if they have entered a typo. Asking the user to verify the entry of every single text field would be unreasonable. The potential for data integrity issues if a select input control is not used when it should be are obvious. If a user is supposed to choose a U. S. state, allowing him to enter text would be deleterious. The user could enter Whoville. They could also enter “ None of your business. ” Restricting entry is important for fields that have limited, predefine options.

Radio buttons are in the same category as select input controls when it comes to data integrity. Selecting to use something like text instead of a group of radio buttons would be undesirable. For instance, if the user was supposed to select gender, he could type enter eunuch. This would not be helpful if that data is critical for the site’s services or interactions. Password fields come with data integrity issues built in, the data within a password input control are masked. Since the user cannot see the entered data, it is

very easy to submit data that contains typos. This is not critical for a log in form.

The user would simply be notified that his log in attempt failed. Conversely, for a registration form, this could result in highly undesirable issues. It is therefore common to place two password input controls on a form like this. The second input control is used to validate the entry in the first input control. The user is able to submit the form only when the values in both fields are identical. If a programmer chose to use a normal text field rather than a password field, the integrity of the entire system could be compromised. This has more to do with systems security than data integrity, but is still an important consideration.

User interfaces are often restricted by logged in status or type of user. For instances, a member of the human resources department would have access to employee information that a member of the software development department should not have, and vice versa. Obviously, a user who has yet to log in should not be able to access any sensitive data from any department. These offer definite advantages to any system. However, there are caveats that come with setting up a system like this. The first is simply the design and setup of these restrictions.

A small bug in the setup or the code can cause an entire department to lose access that they need to do their jobs. Another issues is password management. Designers need to deal with how often passwords must be changed, how strong the password should be, and users forgetting their passwords. Without good forgotten password procedures, employees can be at work-stoppage for a significant amount of time, costing the

organizationmoney. There is also additional overhead when an employee needs to be granted access or removed from access. Finally, an organization can decide to alter the access requirements for an entire section.

This makes it necessary that the system access restrictions can be easily updated. All of this adds a large amount of overhead and requires one or more individuals to takeresponsibilityfor system support. Web Form References Ponce de Leon, D. (n. d). Forms in HTML. Retrieved from http://www. htmlquick. com/tutorials/forms. html W3Schools (n. d.). HTML forms and input. Retrieved from http://www. w3schools. com/html/html_forms. asp World Wide Web Consortium (n. d.). , Forms in HTML documents. Retrieved from http://www. w3. org/TR/html401/interact/forms. html#h-17. 1