**Study Guide for Midterm Exam**

**Comp 4952 2015**

Lecture notes: Introduction to HCI Part 1 and 2; Modules 1 and 2.

Labs: Lab assignments and issues discussed in lab.

Quizzes.

Readings: as specified in assignments and power-point presentations.

**Learning Outcomes / Questions**

1. What are the characteristics of a well-designed interface?

Compresible, Controllable, Predictable. Users feel comptetent, satisfied, responsible for their actions

1. What is usability?

Ease of use and learnability of a human-made object (intuitive-ness)

1. What methodology is employed with respect to usability?

* Prototyping methodology for task-centered design
  + Application must fulfill everything the user must accomplish.
* User-centered methodology for interface design
  + User should accomplish tasks in an intuitive way.
* Low/Med/High-fidelity prototype

1. Describe the methods and outcomes of low-fidelity prototype.

* Needs Assessment
  + The discrepancy between the current condition and wanted condition must be measured
  + How: Surveys, interviews, meetings
* User Task Analysis
  + Figure out how to fulfill the needs as discovered in the Needs Assessment phase
* Usability Studies
  + Studies conducted on real users to determine effectiveness of interface designs
  + Techniques:
    - Think aloud
    - Videotape
    - Discuss Afterwards
    - Surveys
    - Focus groups

1. Name four rules of interface design and describe them (Shneiderman).

* Strive for consistency
  + Colors should be the same for similar controls
  + Controls should be in similar positions
  + Similar things should look similar
* Offer informative feedback
  + Toast
  + Breadcrumbs
* Allow reversal of actions
  + Undo
  + Go back a step
* Enable shortcuts for frequent users
  + Site map

1. Describe four usability heuristics proposed by Jakob Nielsen and give a short description.

* Visibility of system status
  + The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
* Match between system and the real world
  + The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.
  + Follow real-world conventions, making information appear in a natural and logical order.
* User control and freedom
  + Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
* Consistency and standards
  + Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
* Error prevention
  + Even better than good error messages is a careful design which prevents a problem from occurring in the first place.
  + Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
* Recognition rather than recall
  + Minimize the user's memory load by making objects, actions, and options visible.
  + The user should not have to remember information from one part of the dialogue to another.
  + Instructions for use of the system should be visible or easily retrievable whenever appropriate.
* Flexibility and efficiency of use
  + Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users.
  + Allow users to tailor frequent actions.

1. Describe the usability heuristics proposed by Jakob Nielsen that you used in your low-fidelity prototype design and justify them.
2. Indicate practical strategies for usability and describe them.

* Do not wait till you finish your product to test:
  + Test early with real users
  + User interfaces can be tested with fake web pages (no code behind) or just paper
  + Design based on tests
* Determine the goals of your application and write the user tasks
  + Based on these goals, write tasks for the user to perform
  + Be detailed in your tasks
* Determine the user profile
  + Invite users with different profiles to test your application

1. Consider Fitts’s law: MT = a + b \* log2 (2A/W + c). Explain it.

* Refers to issues of modeling human movement
* Predicts time required to move to a target area.
* Function of distance to target and size of target
* The equation yields the ID (Index of Difficulty)
  + ID increases by one unit for each doubling of distance and halving of width.

1. Describe how Fitts’s law can be used for interface design.

* Big targets can be acquired faster: make important controls bigger
* Closer targets can be acquired faster: contextual menus appear next to cursor
* Corners are infinitely targetable because of the boundary created by the edges of the screen: put common controls in corners

1. What are consequences and heuristics of Fitts’s law?

* Same thing as 10

1. How did you apply Fitts’s law in your design?
2. Explain usability and aesthetics.

* Users thought beautiful ATMs worked better than normal ones

1. How did you perform the user task analysis in your low-fidelity prototype design?
2. What is user modeling?

* A field of HCI and AI
* Build design based on how user would do it IRL

1. Consider the Bayesian theorem: P (H|X) = (P (X|H)\* P (H)) / P(X). Explain it.

* An equation that predicts a future outcome based on a previously predicted outcome and measurements of data.

1. Consider the Bayesian theorem: P (H|X) = (P (X|H)\* P (H)) / P(X). Solve a problem with a Bayesian theorem and draw the tree graph.
2. What is the difference between a stochastic and a deterministic system?

* Stochastic is a system in which the outcome cannot be predicted because it is pertaining to chance. Results may vary each time even if the inputs are the same.
* A deterministic system is a system in which the output will always yield the same results given the same inputs.

1. To what extent statistical models can be used for user modeling?

* Adaptive interface design
  + Interface that automatically adjust to accommodate the user’s previous experiences by keeping a user profile and logging and analyzing their activities.
  + Apply the Bayesian theorem
* Adaptable design
  + User can customize the interface to suit his/her needs.

1. Give four differences between ASP and ASP.NET
   1. ASP.NET decouples the code from the interface; HTML and CS are not inline.
   2. ASP.NET can be compiled; ASP is always interpreted
   3. ASP.NET uses the VIEWSTATE variable which is a hidden encoded HTML input field that stores session data.
   4. ASP is more difficult to reuse because it is intermingled with the frontend code.
   5. ASP is procedural; ASP.NET is Object-Oriented.
2. What were the challenges of ASP that were solved with ASP.NET?

* Difficult reuse: solved by making ASP.NET OO to decouple front end and back-end
* Poor code readability (intermingled): OO
* Limitations of session state: VIEWSTATE variable
* Browser support: move code to backend

1. What are Web Forms and Web Services?

Web Forms

* Like Windows forms
* Prgrammable web pages
* Server side controls

Web Services

* API over the internet via http and xml
* Server side controls available to the user without providing a gui interface

1. Explain Postbacks.

* A post to the same page when the user submits a form.
* The state of all server-side controls is maintained in a postback; controls are automatically populated
* Works with all browsers
* Different from refresh, which is a GET request with no parameters back to the same page; all controls will be cleared.

1. What is happening to server-side control objects during postbacks?

* The state of the controls are saved and repopulated when the page reloads.

1. How do technologies ASP and ASP.NET maintain the state of all server-side controls during postbacks?

* The data in the controls are saved into \_VIEWSTATE: a hidden, base-64 encoded input field in the html document.
* The data is compressed and sent to the server for processing, and put back into the \_VIEWSTATE upon postback.
* NOT encrypted

1. In what event Page.IsPostBack is used? Why?

* It is used to check if the page is being loaded for the first time: if Page.IsPostBack returns false, that means we are not in a postback, and we can set or initialize things depending on the context of the page, ie populating a dropdown with options, or providing a tutorial for the first time.

1. Are refresh and postback the same thing? Why or why not?

* A postback is different from refresh, which is a GET request with no parameters back to the same page; all controls will be cleared.
* A postback will send a POST request and include the \_VIEWSTATE variable in the request.

1. Describe server-side controls.

* Elements that reside on the server
* Objects so you can invoke methods on them
* Maps to the controls in the frontend

1. Explain the page life cycle.

* Refer to Module 1 slide 20
* Know that there are several events, but don’t memorize all of them
* Know Initialization, pageload, loadviewstate, saveviewstate
* Begins with http request post or get / it will be created like a package on server and then sent to browser. On the browser, there is init stage, load viewstate (description of the control tree, content of each control), load postback data, load controls and events / after user sees site and invokes another event, …

1. Explain the Page\_Init() event.

* Raises each control’s init event; called only for the first page load

1. Explain LoadViewState() event.

* Only happens on postback
* Any data received will be loaded into the controls

1. Explain Page\_Load () event.

* Called each time the page is requested
* Initialize/reset variables here

1. How are variables stored in the ViewState?

* ViewState is like an object, and has a method like .add()
* Variables are ENCODED, not ENCRYPTED.

1. What are the advantages and disadvantages of storing variables in ViewState?

Advantages:

* Keeps track of variables

Disadvantages:

* Viewstate can get too large
* Performance
* Security

Things to put into viewstate:

* Form elements and other things important to the interface, ie calculator control – small, don’t have to store on server; remember what the previous answer was and add it onto your current answer.

1. What security issues are related to the ViewState?

* It is stored in base-64 ENCODED field; NOT ENCRYPTED. Any user can take this information and decode it.

1. In what format is the ViewState stored? Is the ViewState encrypted by default?

* Same as above.

1. If no variables are stored in the ViewState is the ViewState empty? Explain.

* The viewstate is not empty if you do not store any information. The control state is saved regardless.

1. What is stored in the ViewState?

* Control state – Viewstate for the control that cannot be disabled
* Variables
* State information, ie previous result of your form or your name

1. What is the ControlState?
2. If you are a developer, is it possible to disable the ViewState? What about the ControlState? Why?
3. If you are a user, is it possible to disable the ViewState? What about the ControlState? Why?
4. What is happening when a user requests a page from a remote server?
5. Is it any client-side code executing in your assignment application? If so, how is the client-side code executed in ASP.NET?
6. What is the role of the line:

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="\_Default" %>

1. What is the role of a default web form?
2. What special folders are in ASP.NET and what is their role?
3. What is the difference between web server controls and HTML controls?
4. Explain the property IsPostBack. What is the role of property IsPostBack and how should be used?
5. Describe the validation controls. What control validators are available in ASP.NET and how should be used?
6. What is the role of the property Page. isValid and how should be used?
7. What is happening when the user hits the refresh button?
8. What is MVC and how is it implemented in ASP.NET MVC?
9. Explain in your own words and give an example of model, view and controller in ASP.NET MVC.
10. Explain the role of the Entity Framework. Draw a diagram to explain your answer.
11. What does EF allow developers to do?
12. What are the main (basic) functions implemented in EF?
13. Describe the three aspects of EF: conceptual model, storage model and mapping.
14. What is the cost of EF? (for more details: read <https://msdn.microsoft.com/en-us/library/cc853327(v=vs.110).aspx> )
15. What is the most expensive operation with the EF?
16. What EF design models are implemented in ASP.NET?
17. Discuss the Entity Lifecycle. Refer to the ObjectContext and EntityState.
18. What are DataAnnotations and what do they describe? Give an example.
19. What was the role of the Context class when you created a data model?
20. What was the role of the Initializer class? What did it inherited from?
21. What design model did you used in the ASP.NET MVC lab activity?
22. What are migrations?