# BCIT

# Comp 4952 Technical Programming 2

# Technical Programming Option

# Option Head Mirela Gutica

# Fall 2014

Total: \_\_\_\_\_\_\_\_\_/ 100

**Project**

You will practice the concepts discussed in this course by designing and implementing a web application. This task gives you the opportunity to implement the web application of your choice, to research concepts and/or details of implementation that are not covered in lectures or labs. However, you have to follow the requirements indicated by your instructors.

The web application will be implemented in three stages.

In this exercise, you will use an iterative approach to design a website using: prototyping, task-centered and user-centered design methodology.  The website should be implemented based on the technologies discussed in this course (ASP.NET). The design and implementation will follow an agile model and will be divided into three stages: low-fidelity prototype, medium-fidelity prototype, and high-fidelity prototype.

You will work in pairs.

**Stage 1: Low-Fidelity Prototype**

A low-fidelity prototype is a prototype that is sketchy and incomplete, that has some characteristics of the target product but is otherwise simple, usually in order to quickly produce the prototype and test broad concepts.

At this stage you will focus on:

* articulating good task descriptions
* interviewing or otherwise consulting potential users with regards to their work
* using the task descriptions to decide upon system requirements
* brainstorming design alternatives for your system  based upon the above
* sketching out the rough ideas
* evaluating the various prototypes through a task-centered walk-through
* evaluating a prototype informally with a small number of users (usability study)

The outcome of this stage will be:

1. A document named Low-Fidelity Prototype containing:
2. A description of your project:
   1. Description of the interface (what the users see and interact with)
   2. Description your system/application as a list of requirements
3. A justification of a user-centered design (based on the heuristics discussed in lectures)
4. A list of risks and issues
5. Pictures or screen shots of your prototype

1. A document named Task Analysis.

Depending on your users, the task analysis will contain:

* + 1. Workflow analysis
    2. Job analysis
    3. Task sequence

1. A document named Usability for each usability study containing:

A usability study (3 or more users)

1. Usability report

Marking (maximum): 100/100

1. (15) A description of your interface
   1. Describe your system/application as a list of requirements + Design
2. (25) A justification of a user-centered design (based on the heuristics discussed in lectures)
3. (25)Task analysis
   1. Depending on your users, the task analysis will contain:
      1. Workflow analysis
      2. Job analysis
      3. Task sequence
4. (25) A usability study (3 or more users)
   1. Usability report
5. (10) A list of risks and issues

**Task analysis**

**Use tasks when you describing your website.** List at least 5-7 concrete task examples. Keep task descriptions short and to the point. Each task should be accompanied by a paragraph that describes the class of the expected user (eg, a typical customer), the relative importance of the task (e.g. frequently done and important, infrequently done but still important, rare and not important, etc), and whatever other nuances you feel should be included.

**Requirements for concrete task**:

* Says what the user wants to do but does not say how the user would do it
  + Do not make any assumptions about the system interface
  + You can use this stage to compare different interface design alternatives in a fair way
* Are very specific
  + Say exactly what the user wants to do (select, pay, search, etc.)
* Describes a complete job
  + List all aspects of the task, from beginning to end (this forces you when design to consider how interface features will work together)
  + Use this to contrast how information input and output is carried through the dialog, i.e.:
    - where does information come from?
    - where does it go?
    - what has to happen next?
* Says who the users are
  + Use particular people, if possible (shopper, customer, admin, student, etc.)
  + Reflects real interests of real users

**Stage 2: Medium-Fidelity Horizontal/Vertical Prototype**

A medium-fidelity prototype allows for a combination of both high-level and detailed views; rapid, iterative changes; and the ability to conduct meaningful user tests to evaluate complex functionality and to help determine system specifics.

* *Redesign* your interface. To do this, you should *review* your Stage 1 prototype(s) and walkthrough results. You should also apply the design knowledge you have gained in class to your design. You may want to develop a few more low-level prototypes here and do further walkthroughs to check your ideas out. This part is up to you.
* Decide the technology that works the best for your web application
* *Implement* your design as a medium-fidelity horizontal/vertical prototype. Use the appropriate tools of your choice; implement your page(s) and the visual parts of your interface. Implement the most appropriate solution (i.e. choice of technology, user-centered design, separation of concerns, modularity, etc.)
  + Horizontal: all the main components of your interface are included. This gives the illusion of a fully functional prototype.
  + Vertical: select the primary tasks and make sure that a substantial part of the vertical functionality required for those tasks is supported. 'Substantial part' includes screens, error messages, handling of unexpected input, defaults, robustness, etc. You may program in 'stubs' for sub-tasks you are not implementing at this time (e.g., certain actions may return some kind of 'Under development' message).
* *Usability study* (you can use the same users or new users).

The outcome of this stage will be:

* A website that you can demo.
* New usability study (3 or more users)
* Design document
* Testing plan document
* Revised:

1. Description of your interface (with real screen shots)
2. Justification of a user-centered design (use the heuristics discussed in lecture)
3. Task analysis
4. Risks

Marking (maximum): 100/100

1. (60) A website that you can demo
   1. Layout
   2. Functionality (primary tasks should be implemented)
   3. Robustness (recovery from errors, no crashes)
   4. Solution (choice of technology, user-centered design, separation of concerns, modularity, etc.)
2. (10) New usability study (3 or more users)
   1. Usability report
3. (20) Design Document
   1. High-level entities diagram
   2. Low-level design
      1. For each object (or set of objects) describe:
         1. Interface for the object
         2. Pseudo-code
   3. Interaction and state diagrams
   4. Depending on your project other aspect i.e., database schemas, configuration, etc.
      1. Please plan, do not generate the design document form code
4. (5) Testing plan and unit tests
5. (5) Revision (update in yellow)
   1. Description of your interface (revise with real screen shots)
   2. Justification of a user-centered design (if changes have been made)
   3. Task analysis (if changes have been made)
   4. Risks (update)
6. References (if not present -10)
   1. List of resources (code, documentation, existing systems, etc.) that you used for your project

**Stage 3: High-Fidelity Horizontal/Vertical Prototype**

The high-fidelity prototypes will be developed with the full set of graphics, functionality, session management and dynamic data sources. For you, this will be the final product.

The outcome of this stage will be:

* A website that is ready to be deployed on a web server (with instructions to be deployed).
* Final usability study that confirms the strengths of the application and offers suggestions for future improvements (3 or more users).
* Documentation:
  + Design document (revised)
  + Testing plan document (revised)
  + Installation and User manual
  + Test results

Marking (maximum): 100/100

1. (60) A website that is ready to be deployed on a web server (with instructions to be deployed).
   1. Layout
   2. Functionality
   3. Robustness (recovery from errors, no crashes)
   4. Solution (choice of technology, user-centered design, separation of concerns, modularity, etc.)
2. (15) Final usability study (3 or more users)
   1. Usability report
3. (10) Design Document Revised (update in yellow)
   1. High-level entities diagram
   2. Low-level design
      1. For each object (or set of objects) describe:
         1. Interface for the object
         2. Pseudo-code
   3. Interaction and state diagrams
   4. Depending on your project other aspect i.e., database schemas, configuration, etc.
      1. Please plan, do not generate the design document form code
4. (5) Testing plan and unit tests (revised)
5. (5) Testing results and bug reports
6. (5) Installation and user manual
7. References (if not present -10)
   1. List of resources (code, documentation, existing systems, etc.) that you used for your project
8. Self-evaluation of your project (the mark you claim for your project based on the project requirements and your project status). Please mention if you and your partner worked equally and deserve the same mark. (if not present -10)

**Presentations**

* For stage 2, you will give an in-class demo.
* The final presentation will be scheduled in class during the week of Nov. 24, 2014. Each team will present for 10 min.

**Schedule, Grading & Due Dates**

Stage 1 Due Sept. 21, 2014 @ 9:30pm 20%

Stage 2 Due Oct. 19, 2014 @ 9:30pm 30%

Stage 3 Due Nov. 23, 2014 @ 9:30pm 40%

Presentation Week of Nov. 24, 2014 10%