Comp 3020 – Human-Computer Interaction I
Assignment #3
Handed out on: November 16th 2015
Due on: December 9th 2015

PART I - Implementation (70%)

This part consists of implementing your application for the "project", i.e. Movie Organizer system based on the "Project Description" document. You need to build the application to sufficiently support the evaluation in Part II of this assignment. As core functionality, your implementation should include management tasks for adding/deleting/editing data, along with **3 major tasks** (ideally you can use the same 3 tasks you listed in Assignment #2, which you developed in the requirements and prototype/design phases). As one of these 3 tasks, you must include an implementation of the **Advanced Search task**, as described below.

Other major tasks that serve the core functionality of what you intend your system to do include (these are simply ideas, and we hope your user elicitations from Assignments #1 and #2 have provided a richer sample):

- Advanced search - search for movies by different fields such as title, director or cast, including combinations of multiple fields. This should look similar to the FilmFinder interface in the figures of the "Project Description" document. It must include a graphical visualization (i.e. scatterplot as pictured) and a filter mechanism to assist the search. However, the particular data fields you choose to display on the axes, and the mechanism you use for filtering data may be changed to suit your original design. (*** This task must be included in your implementation**)
- Recommender system - search for movies based on criteria of other movies you have liked in the past, say you would like to see movies that resemble the genre, maybe same actors. For example, if a user likes Pulp Fiction, allow him/her to find other similar movies.
- Watch list - allow users to create, search and maintain a "like to see" list of movies. The movies in the list could have been suggested by friends, movie critics, etc. or could be a list of movies you have seen in trailers.
- Shopping list organizer - allow users to organize movies they have at home or would like to purchase, say for Christmas, birthday, etc.
- User reviews – a collection of feedback from viewers.
- other tasks you that you can creatively define.

Important Notes: Your 3 major tasks must be in addition to the basic data management features (add/delete/edit). One of your 3 tasks must be the advanced search as described above. Your remaining tasks should not be overly similar i.e. each task must require some unique feature of your implementation. You are not expected to create a robust database with features such as data integrity. You simply need to have a working implementation for maintaining the system data, and be able to add, edit and remove records through the interface.

This part will be evaluated by taking into consideration the following:

- Usability: is there a practical and consistent flow within the application, is there keyboard navigation, do tabs and mnemonics work properly, is the layout properly defined, are the proper widgets being used?
- Utility: does the system do what you indicate it should do?
- Efficiency: can a user quickly access the necessary information?
- Learnability: for someone with very little experience, is the system learnable?

- Implementation depth (these items **should be** implemented for your learning):

  - Data entry: you should include forms for adding, deleting and modifying records

  - **Visual Inheritance (VI)**: include some form of visual inheritance in your project. A good place to do this would be the data management routines.

  - **Thorough input validation:** all data-entry screens should include some level of input validation and your data entry forms should further add thorough input validation. This can be easily achieved using Error and Info Provider widgets (see class notes).

  - **Syntax-free interaction:** include two examples of syntax-free interaction. This could be a combination of dynamic queries and drag-&-drop, or just simply dynamic queries in at least two areas of your application.

  - Help: you should include minimal amount of help. For example tool-tips should be used where a certain syntax is needed, or if you require the user to complete steps in a particular manner.

Hand-In: the source code, the .exe and any other file needed to run the application. **Include an appropriate data file (i.e. .xml or other means as required) to populate the program as it is opened.** The source should be commented where appropriate. Include a text file containing the following:

(1) Project description
A short description of your project with a focus on those tasks you completed. This should include a list of your 3 major tasks and description of each task's purpose.

(2) Interface description
A short description of how the interface is expected to behave. Touch on each major task.

(3) Particular problems you are aware of with your application.

(4) Installation procedure
Include all descriptions for running the program. Submit a one paragraph description of how to put the pieces together, i.e. which directory to insert the data files, image files, etc.

PART II - USER EVALUTION (30%)
This part involves a simple evaluation of your project. You need to produce a brief report (2-3) pages about your evaluation outcomes.

- Users: You need to test and evaluate your application with at least 5 users. All users are preferably the same individuals or have similar profiles as your initial users from A1 and A2;
- Tasks: You will need to devise as many sub-tasks as possible to test the application. Typically your test should include 8-10 sub-tasks. The word task here is not referring to the high-level tasks you have implemented but instead to sub-tasks of these high-level functional pieces in your interface.
- Variables: Choose the appropriate dependent variables (time and number-of-errors) for your tests and insure that the data is well recorded.

- Other evaluation techniques: You can choose to complement your user testing with one other technique such as an observation or a semi-structured interview. Be sure to indicate the techniques used, the results, and an analysis of the results.
- Analysis: Provide an analysis of your data using reporting and presentation techniques discussed in class. The clarity of your presentation will be evaluated.
- Evaluation Criteria:
    - You will be evaluated on the adequacy of your evaluation methodology and reporting of your data.
    - Your report should include a description of each of the sections mentioned above.

In a separate section of your report, you should also describe your experience doing the user testing: (a) what you learnt, (b) what you would do differently the next time around, and (c) how you can integrate user testing into your work/coop/future projects.

Include the complete project files for all coding parts of this assignment. Place all clearly labeled .exe files in a single folder called "execs". Place all parts of this assignment in a single .zip file named "comp3020_group<number>_a3.zip" and upload to the course dropbox on UM-Learn.

Format and guidelines

(a) All text documents should be typed using an appropriate word processor and submitted in **.pdf** format.
(b) Your document should be properly structured.
(c) Where appropriate, always provide a rationale for your answer.