

## Comp 3020 – Human-Computer Interaction I

### Assignment #3

Handed out on: November 10<sup>th</sup> 2014

Due on: December 3<sup>rd</sup> 2014

#### PART I - Implementation (70%)

This part consists of implementing the remainder of the application for the “project”, i.e. Movie Organizer system. You need to build the application to the point that it can be tested by end-users. The implementation should include only the **major tasks** you developed in the requirements phase and in the prototype/design stage. In particular it could contain the implementation of the **search task**, the **task for placing reviews**, and one for **bookmarking which movies to watch next**. You may wish to align your tasks with the features listed below.

The program should include basic data management tasks, such as adding/deleting/editing content. I would not consider these basic routines as your primary tasks. You are not expected to create a robust database with features such as data integrity. You simply need to have a persistent manner to maintain content, data files and be able to add and remove records through the interface.

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

- Usability: is it easy to 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.

- Other features: in addition to your own set of tasks you should at least ensure that the following features are available in your application:

- all the search features as provided on the FilmFinder interface shown on the website should be replicated. You could use this UI as the primary method through which a user can search for a movie.

- 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.
- 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 like to watch
- allow users to organize movies they have at home or would like to purchase, say for Christmas, a friend's birthday, etc.
- other tasks you that you can creatively define.

**Hand-In:** the source code, the .exe and any other file needed to run the application. The source should be commented where appropriate. The entire submission should be submitted through D2L with a text file containing the following:

(1) Project description

A short description of your project with a focus on those tasks you completed.

(2) Interface description

A short description of how the interface is expected to behave.

(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 must preferably be or have the same profile as your initial set 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.