M3 Domain Design

Requirements
1. Create team page and indicate your team members.
2. A brainstormed list of Domain classes
3. A list of candidate classes after filtering
4. A set of CRC cards for the candidate classes. Cards should be filled out on both sides (role stereotypes and responsibilities/purpose). These should real index cards, not a word document.
5. A set of scenarios that cover typical uses of system and exercise the CRC Cards. The number of scenarios is left to you, but they should cover the major uses of the system.
6. Meet with assigned TA to discuss your design and get corrections

Domain Classes:
User, MediaLibrary, MediaFile, Client, Search, SearchResult, NetworkInterface, Server, StatKeeper

Example of CRC cards:

<table>
<thead>
<tr>
<th>MediaFile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Know name of MediaFile</td>
</tr>
<tr>
<td>Know rank</td>
</tr>
<tr>
<td>Know size</td>
</tr>
<tr>
<td>Know shared</td>
</tr>
<tr>
<td>Know type</td>
</tr>
<tr>
<td>Know path</td>
</tr>
</tbody>
</table>
Our goal on M3 is to brainstorm a list of Domain classes.
Suggestion: Should be creative.

M4 - Application/GUI Design

**M4 Requirements**

- Create your Software Architecture and Trust Boundaries
- Identify your Application and Utility Objects (those new objects specifically developed for the design. You do not have to say which are which, just id the ones that were added from the domain analysis)
- Add necessary CRC Cards and any new scenarios needed (for newly discovered domain classes if any)
- Submit a UML Class diagram of your refined design (shows all domain and application/utility objects)
- Submit 2 UML Sequence diagrams that shows your full design handling one or more scenarios
- Submit screen mockups that show your preliminary user interface. These screens can either be hand-drawn, or prototyped with the VW Painter and then captured.
- Submit a written contract (1 per person on the team) for an object or responsibility in the design
- Submit a short paragraph on your error handling and exception handling design strategy.
UML Sequence diagram:

Screen mockups:
Contract’s example:

Name: generateSearch (Client)

Responsibilities: Creates a new Search object from search criteria specified in the user interface.

Preconditions: This method requires the user to set their search criteria.
Postconditions: The Search object is created and sent to the server to be processed.

Error and Exception Handling Design Strategy:

A main source of error comes from invalid input from the user. To handle this we plan to check all the data that is input by the user to make sure the data entered is correct, and display a prompt with the proper warning if the user enters something incorrectly. Another error that we must worry about is the user doing something that’s not possible like deleting a file that doesn’t exist. In order to prevent such impossibilities we would disable buttons that the user shouldn’t be able to select and only show the fields that the user can safely edit. For example when a user logs in we check if the user has a library yet and load that and enable the library delete button. If the user has multiple libraries then enable the switch library button also. We would do the same with the file buttons and also the server base buttons. Another source of error that we can foresee is networked communication. For example, if the user enters the address of the server incorrectly, the client will be unable to connect. In this case, we would wait for the connection to timeout, and report to the user that the server is unavailable for some reason. Or, what if one user is downloading a file and the other user disconnect before the download finishes? To handle this we’ve added an upload speed indicator to the client as a way for the client to know someone is downloading from him. Also if possible we would tell him that a person is still downloading a file and does he want to quit automatically after he’s done. If he still wants to quit, we would have to send a message to the downloading client to stop the download, and then close the connection.

M5 Domain Implementation:

The goal of M5 is to implement all the domain class
M5 Requirements

1. Implement Domain Objects
2. Create SUnit tests to unit test your core Domain Objects (you do not need to write tests for trivial accessor/modifier methods or for network communications)

Criteria Breakdown:
  * User Creation/Login and Library Access
  * Library Creation/Deletion
  * File Management (files add/remove to a library)
  * Customizable Extensions
  * Basic Connection to the Server
  * Load/Save of Client Information
  * Search for files by different criteria
  * SUnit tests for Domain Objects

What Worked:
User Creation/Login and Library Access
Library Creation/Deletion
File Management
Customizable Extensions
Basic Connection to the Server
Load/Save of Client Information
SUnit tests for Domain Objects

What Didn't Work:
Search for files by different criteria

M6 Complete Implementation with UI
Criteria:

- User/Library Creation Screens
- File Management Screens
- Server Interaction Screens
- Search/Report Screens
- Download Screens/functions
- File Displays sortable by multiple Criteria
- Constraints enforced (only shared files visible)
- SUnit tests (for any non-gui classes added)
- Good Smalltalk code and style

What Worked

Everything is working.

Screen mockups of our final design:
M7 User Interface Evaluation

**Checklist of Things to Complete**

1. Clearly identify the team whose prototype you evaluated and provide a one-page description of your evaluation plan and the rationale behind the plan, i.e., why you chose to conduct that evaluation on this prototype.

2. Results of execution of usability plan, providing clearly documented printed evidence that you conducted the evaluation plan correctly. For instance, this includes the actual raw data sheets and evaluator notes in addition to the synthesized data.

3. 3-page usability report highlighting the three top concerns arising from the usability evaluation. The report should identify clearly what each usability concern is and the evidence you have to support this problem being a real problem that should be corrected by the design team. One copy of this part of the report goes to Bob, and one copy should be given to the team you evaluated.

4. 3-page rebuttal of the usability report submitted to your project team. Be careful not to get defensive in this report (resist refuting that the findings of the evaluation team are incorrect). Make a clear argument for how you would address each usability concern. Assess whether the object-oriented design you developed for your prototype made it any easier to address these usability concerns.

Our team has chosen to use the heuristic method to do our UI evaluation. We were initially leaning toward this method because of its low overhead (i.e. no “user” is required) and its straightforward process. Upon receiving other team’s M6, we have discovered that everyone is unable to file in the team’s code (and at the time of performing the evaluation, no working code has been provided), which means we will be unable to actually run the program and test it hands-on. This effectively makes doing a heuristic evaluation our only choice. We have been provided GUI mock-ups for all of the application’s screens, so we should be able to do a decent evaluation without the code. Using the heuristics found on the Xerox checklist (based on Nielsen) as our guideline, we hope to identify as many usability problems as possible, so that we may produce a meaningful usability report for the team we are reviewing.

Things we have to improve for our UI:

1. The lack of prompting for deletion of the library and media file
2. The “moving file button” was not described in a proper way
3. The filter list had some problems with.

And yet after M7, our team improved our UI design.
M8 The TWIST

Required Feature

- Login to application. Incorrect attempts must be handled properly.
- Display all the libraries for the logged in user.
- Allow the user to add a new library.
- Display the files associated with a library.
- Allow the user to add a new file.

Grading Criteria

Basic Requirements (30)

- use of style sheet
- query strings
- session vars
- dynamic web page
- html forms
- include file

Mockups:

Welcome to the Insert Team Name Here CS2340 login page.

Please Login

User Name:
Password:

Login
What We Learned Through This Project:

1. Having a regular meeting
2. Read the document in the VisualWork folder
3. Ask TA/professor for help if you face difficulty.
4. Go to class
5. On networking, we recommend using opentalk.
One last we recommend people on sharing code:

Create a tab on T-square

It is very helpful on sharing team material since each member can download the version of code, and if that member updates something, he could upload on T-square.