Help for Project #5

CSE681 – Software Modeling and Analysis

Fall 2013

Jim Fawcett

Sections in Project #5 Document

Here are some suggested sections for your Project #5 document:

1. Code Repository Executive Summary
2. Context - SDCS Collaboration System
    a. Repository
    b. Project Server
    c. Test Harness
    d. Clients – As a member of SDCS
3. Code Repository
    a. Server – focus on Repository functionality
    b. Clients – focus on Repository functionality
4. Core Services and Policies
    a. Services – Described in concrete detail so won't have to be repeated in other sections
    b. Policies – May be brief summary if covered in more detail in appropriate section
    c. Services Samples – Communication, Metadata, Notification, Navigation, Queries, …
    d. Policies Samples – Ownership, Notification, Versioning, …
5. Parts that users interact with directly
    a. Samples – Check-in, Extraction and/or Check-out, Viewing, Editing, …
6. Appendices:
    a. Prototype for navigation using Metadata files
    b. Another prototype of your choice
    c. Note: The prototypes must be discussed in an appropriate section of the document along with results of the prototype leading to some interesting conclusion(s).
7. Make sure you look at the note at the end of the Project #5 statement.

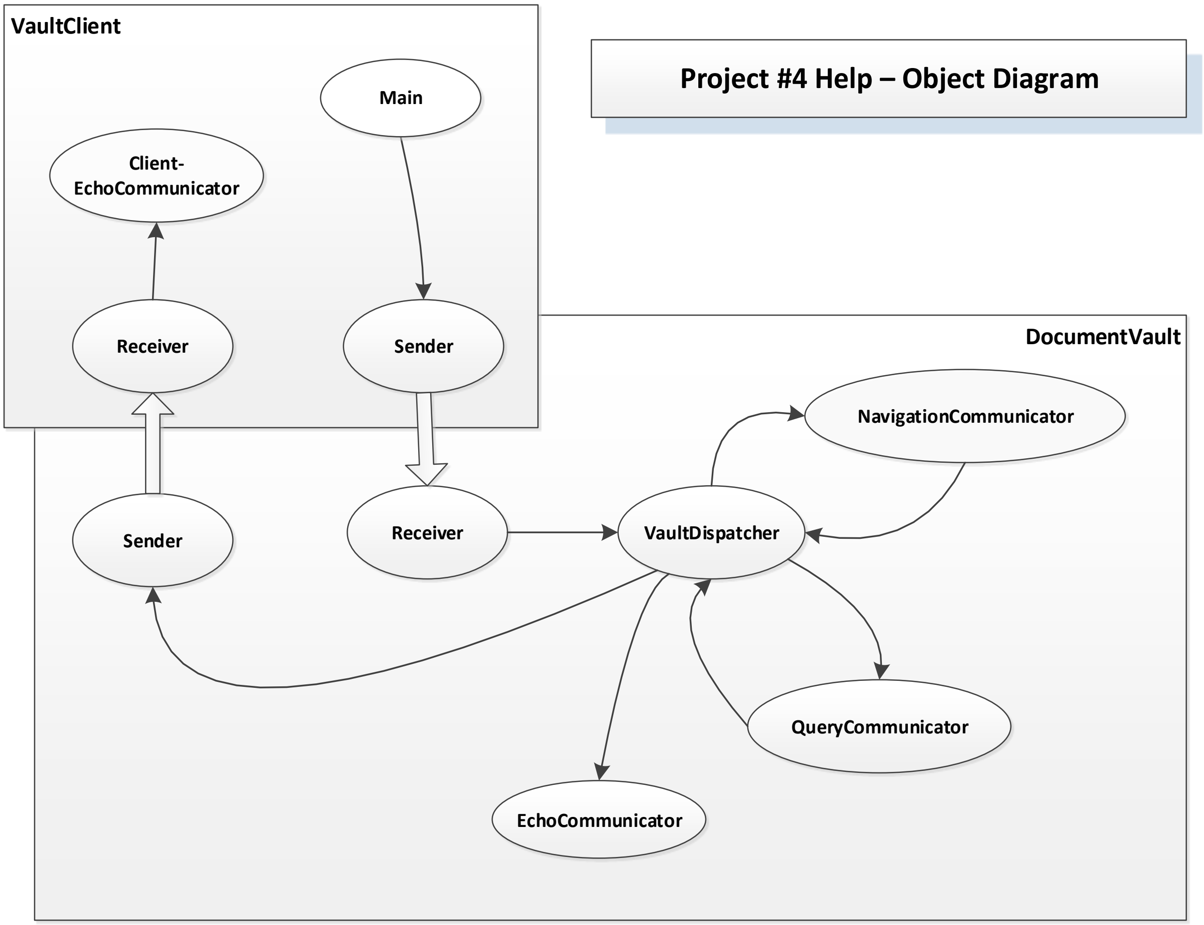
Section Template for Project #5

1. **Section Title**
   Name some part or service
2. **Concept**
   Brief discussion of the core idea, goals, and organizing principles for this part of the system.
3. **Uses and users**
   Who will use this functionality, what tasks will this support, what it the impact on design of supporting this functionality.
4. **Tasks**
   Enumerate the main tasks, discuss the activities for these tasks using an activity diagram with brief discussion of concrete activity details.
5. **Structure**
   Present packages that implement this functionality using a package diagram and discussion responsibilities and interactions of the packages.
6. **Views**
   Show mockups of GUI views, data structure diagrams, database tables, or any other graphics that will help readers to understand the concept and operation of this part of the system. This may be a good place to integrate your prototype data and conclusions.
7. **Interfaces**
   Show message formats, interface contracts, factory signatures for communicating between this part and the rest of the system. If there is a lot of detail then put the detail in an appendix and discuss briefly here with a reference to the appendix.
8. **Critical Issues**
   Enumerate issues that may affect the developer's ability to implement a safe, correct, robust system that is easy to use and conveys the right amount of information about ongoing computations and results. Each issue should be defined with a concise concrete statement of the issue and accompanied with a draft solution if that is feasible.
9. **Note:**
   Not every part deserves this much detail. Some parts may only have a Concept statement, some may have a few of these subsections, but the most important should have all of them.

Diagrams in Project #5 Document

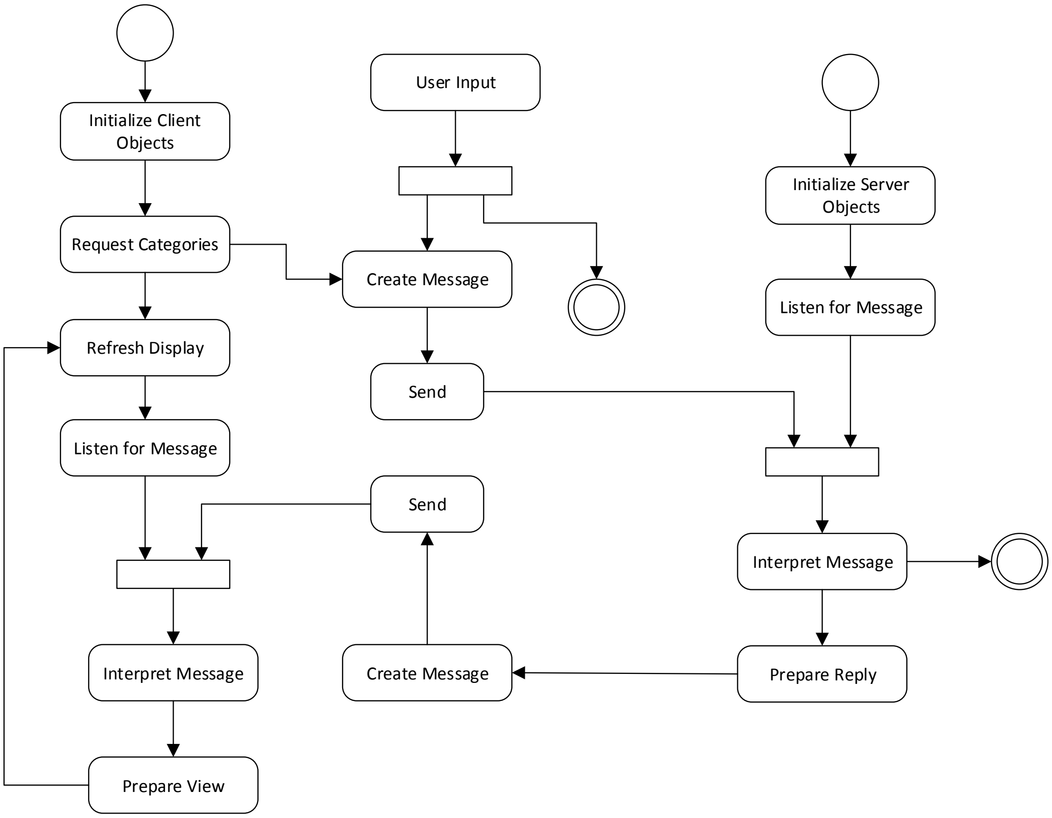
1. Each diagram should be accompanied with text that states:
   a. Why is this diagram here?
   b. What does it show?
   c. What conclusions can we draw from the diagram?

2. Object Diagram Example



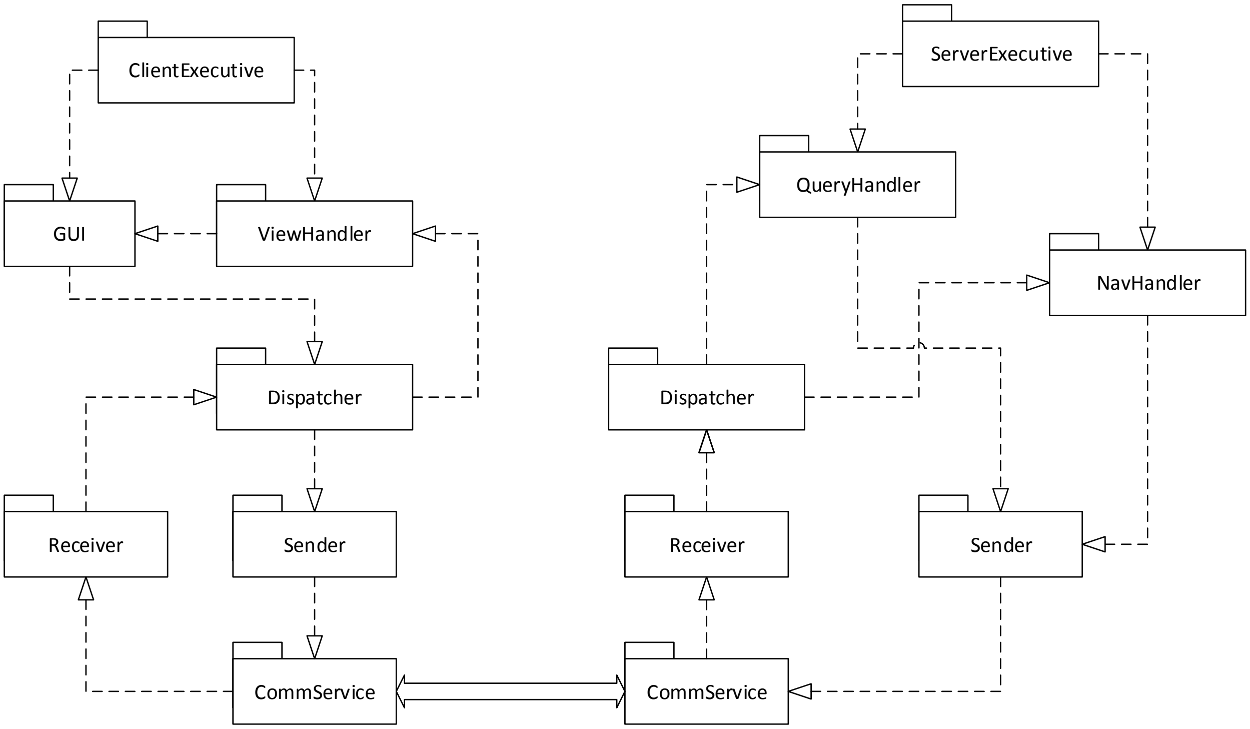
   a. This non-UML diagram helps your reader understand how the various parts of your system communicate with each other if your concept works that way.
   b. You will want to replace the Main object with a GUI object and have the Client-EchoCommunicator connect to the GUI. The name of that communicator should change to something more appropriate to your concept.
   c. Note that notification is entirely supported by this infrastructure. You just need a policy to describe when and why notifications happen.

3. Activity Diagram Sample:



a. Synchronizing bars show waiting, all flows go to activities or synchronizing bars or Exit
b. Every diagram has start and exit
c. Beware of deadlocks
d. For the server “Prepare Reply” can be expanded into the functional parts of the server services, e.g., query, navigation, file handling, etc.
e. For the client “Prepare View” can be expanded into the functional parts of the client.
f. You can show choices with if then symbols or with multiple flows leaving a single activity.
g. If you use the later, you will need a few words to help the reader understand what the multiple flows mean.
h. With this much detail you will probably want to have a separate activity diagram for client and server, but it is a good idea to first show the combined diagram with less detail. You are welcome to use this diagram if it matches your concept.

4. Package Diagram Sample:



a. It would be a good idea to show a combined diagram like this, but then provide separate diagrams for client and server, and, in later sections, for some of the parts where appropriate.

b. For the client and server diagrams you will want to show the additional communicators you are using, e.g., FileHandler, Check-in, etc.

5. You don't have to use Visio if you don't have a copy. However, you can download a free version from MSDNAA which can be installed on two computers. You can also use Gliffy which is a free drawing package available from the Chrome Launcher application provided in the Google Play Store. I've found it easiest to use Google Chrome browser to set this up.
6. You can find the object diagram here: http://www.lcs3.syr.edu/faculty/fawcett/handouts/CSE681/code/Project4HelpF13/Project4HelpF13-demoMsgPassing/.
   The activity and package diagrams are found here: http://www.lcs3.syr.edu/faculty/fawcett/handouts/CSE681/MidTerm/MTF13/.