Operational Concepts

The goal of this project is to create a whiteboard-like internet application (creatively nicknamed ‘WhiteBoard’). There is no intended audience for this project; the intent is, rather, to add enough flexible functionality that can allow any group of people to easily communicate with a common point of discussion, e.g. a diagram that all participants can alter.

Beyond the interface, though, another goal is to create an application that not only runs but runs quickly and ‘correctly’ (i.e. when someone alters the whiteboard, this changes are reflected quickly and accurately in other users’ whiteboards as well). This is a synchronization and networking problem that, without proper implementation, will invalidate any work done on the interface.

System Requirements

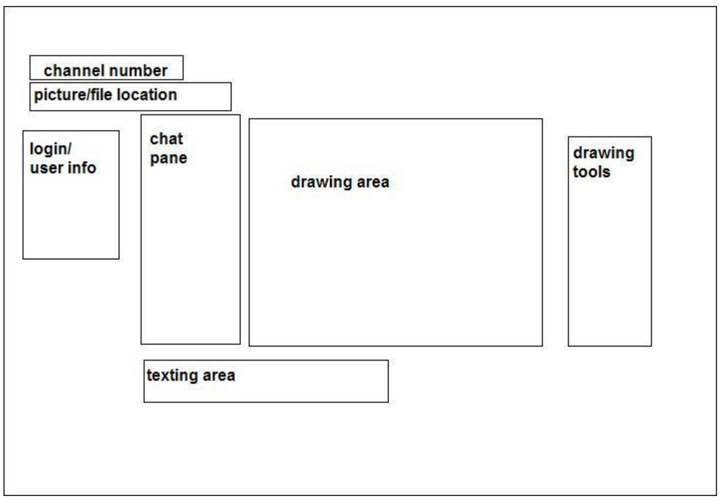
The features that will have to be included in this project are:

1. a common ‘drawing’ area
2. text/chat ability
3. loadable pictures/customizable drawing template. An example usage is loading a picture of a slide that a group of students can draw on and discuss.

A few other features that are not absolutely integral to completing the application, but important to giving it enough functionality so that it will actually be useful, are:

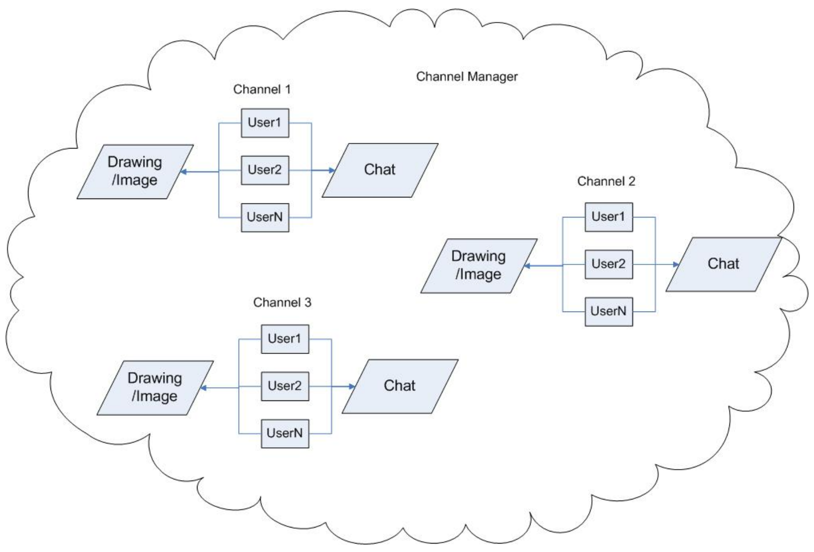
1. properly prioritized changings (e.g. if two people are trying to make a change at the same time, the person that first starts making a change should have his changes reflected in the image)
2. multiple channels/rooms that users can log into (so that different groups of people can use the application at the same time)

The following is a (very) rough outline of what the application will look like:



System and Software Architecture

The two large components of this project are how the users will interact with each other, and how the groups will interact with each other. A rough example of how these will be organized follows:



The architecture can be divided into two modules: a channel and a channel manager. The channel's functionality should be to coordinate activity between individual users and to manage the text pane and the drawing area. The channel manager's functionality should be to coordinate each groups activity.

The majority of this project will be implemented in the Java language. Different tools might be used for networking problems, but haven't been decided on yet.

Lifecycle Plan

The developers for this group should be composed of roughly the following:

- Java programmers. This group should have between two and four programmers.
- Programmers with networking knowledge; at least one, but preferably two. This group should necessarily be a subset of the group of Java programmers, since the majority of the application will be coded in Java.
- Web designer. Since this is intended to be created as a web application, at least one person with a knowledge of useful tools for web development is needed.

Every programmer should partake in the testing since this is a much more efficient way to use programmer time (and given the fact that programmer resources are limited). A rough schedule is projected as follows:

final project planning Approximately .5 week spent on finalizing general design decisions (e.g. what type of networking tools will be used). A general idea about how the channel manager can be implemented should be established now, since this will influence how individual channels should be implemented.

channel module implementation 2 to 2.5 weeks. Write the code that will represent the user, the drawing area and tools, the text pane, and the coordinated communication between these three things. This will necessarily have some overlap with the time it takes to implement the channel manager, since a more concrete understanding of how the channel manager will be structured than was described during initial planning is required to produce a channel module that can actually communicate with the channel manager. However, by the end of this two weeks there should be some sort of user interface that will allow developers to see if the code that makes an individual channel actually works.

channel manager module 1.5 to 2 weeks. Write the code that will represent and coordinate activity between groups of users. At this point there should still be a very rough UI, but enough functionality to it such that the programmers can easily measure how much progress has been made.

heavy testing and UI design 1.5 to 2 weeks. Quality assurance and most testing should be done during this phase. A rough UI should have been designed by now (in order to represent what code is currently functioning in a useful way), so this should be the 'pretty' version of a UI.

final testing and final touches 1 week. Final bug fixes and small changes made to the UI should be done during this time period.

Feasibility Rationale

This project is feasible. There are currently several applications that are similar to WhiteBoard in how it would (theoretically) work. One example is iSketch (a Pictionary-like online game); although WhiteBoard is intended to have a less specific purpose, the basic idea of having a drawing that everyone can see and talk about is similar to what is intended for WhiteBoard.

One of the assumptions being made is that the resources required for this project (e.g. whatever networking tools are needed) are easily and readily accessible to the programmers, along with the assumption that the projected lifecycle/implementation schedule will allow for completion, e.g. the time for testing isn't too short. However, these are risks associated with every project and do not imply that this project is not feasible.