Software Requirements Specifications
for
Multi-user Infinite Canvas Thingy (MICT)

Don Huckle, Ben Kaplan, Mark Wyrzykowski, Rob Wiesler

September 27, 2010

Contents

1 Introduction

1.1 Purpose

This document describes the features and limitations of the Multi-user Infinite Canvas Thingy (MICT). Its purpose is to describe the interface and components of the software. It will be used by the members of the project team and all interested third parties.

1.2 Scope

The Multi-user Infinite Canvas Thingy (MICT) will allow users to log in and edit the canvas. A more detailed description of its scope may be found in our vision and scope document[1].

1.3 Reference

[1] Huckle, Donald, et al. Multi-User Infinite Canvas Thingy Vision and Scope Document.

2 Overall Description

2.1 Product Perspective

MICT is a collaborative image editor. It allows digital artists to work on their images from any computer in the world, and to collaborate with others. MICT uses an ever-expanding canvas so the artists never have to worry about running out of room on the canvas.

2.2 User Classes and Roles

Unconnected: May attempt to connect to the canvas.

Viewer: May only view and move around on the canvas. The viewer may not view locked sections that he does not have permission to view.

Editor: A user who has permission to edit sections of the canvas.

Artist: An Editor who has been elevated to control a section of a canvas.

Operator: May change non-Operator and non-Administrator users to roles other than Operator or Administrator, and may kick, ban or pardon Viewers, Editors, or Artists.

Administrator: The user in charge of the server. He or she has all permissions on the server, and may view and edit all sections even if someone else has claimed control of it and locked it. Additionally, Administrators are

in control of server configuration, and can be assumed to have direct control over the physical server the canvas server program is running on.

2.3 Operating Environment

OE-1: The Multi-user Infinite Canvas Thingy will operate on modern Windows, Macintosh OS X, and Linux systems with Java 1.6 available and installed. The web version of the client will run in Internet Explorer 9, Mozilla Firefox 3.6 and 4.0, Chrome 6 and 7, and Safari 5.

OE-2: The server will run on Ubuntu 10.04 with installed and configured Java 1.6, PostgreSQL 8.4.4 or 9.0, and Apache 2.2.14 or 2.2.16.

OE-3: The system will not limit access to the server. All Internet-connected computers will be able to access it.

2.4 Design and Implementation Constraints

CO-1: The system shall be implemented in a platform-independent manner

CO-2: The system shall use the PostgreSQL Database.

CO-3: All code shall be written in Java or Python.

3 System Features

3.1 Navigate a Canvas

3.1.1 Description and Priority

An arbitrary with Internet access may, through either a simple Java web interface or the standalone client application, access a canvas residing on a server in read-only mode, assuming the server configuration has not been modified from the default. Changes made to the canvas, as described in section 3.2, will be conveyed in real time to the viewport of these users. [High Priority]

3.1.2 Functional Requirements

Canvas.Connect	User connects to canvas and authenticates if possible and/or necessary.
Canvas.View	User requests a segment of canvas and registers to receive updates to said segment. If the user has permission to view the segment, the server will transmit it to the user. The server will send view denied messages for the sections of the canvas the user does not have permission to view
Canvas.Pan	User changes viewed area and issues a corresponding request to the server. If the user does not have permission to view the requested area, the server returns an error message. Otherwise, the server returns the image data for the requested segment, registers the user to receive updates to that segment, and sends location updates to all users following the user.
Canvas.Zoom	User increases or decreases the size of the viewed area and issues a corresponding request to the server. The server checks the user's permissions and if appropriate, sends the user the requested segment and registers the user for updates to that area. Otherwise, in the case that the user does not have permission to view an area, the server returns an error message. Graphics on the user's view interface scale appropriately.
Canvas.Jump	User changes the area he or she is viewing by issuing a request to the server for the appropriate canvas area. The server checks the user's permissions and if appropriate, sends the user the requested segment and registers the user for updates to that area. Otherwise, in the case that the user does not have permission to view an area, the server returns an error message.
Canvas.ListUsers	User requests a list of other users on the server. Based on the user's permission and the settings of the other users on the server, the server returns a list of locations of users both currently on the server and the last locations of users no longer present on the server.
Canvas.JumpToUser	User jumps to the location of another user. All details of Canvas.Jump follow.
Canvas.FollowUser	User A jumps the the location of anther user (User B). If User B is inactive on the server or else invisible to User A, the procedure of Canvas.JumpToUser is followed. Otherwise, the procedure of Canvas.View is followed, with the exception that the server registers User A to receive updates to whatever area User B is accessing at the time, while maintaining User A's access permissions.
Canvas.Mark	Store the coordinates of the current canvas view. Allows the user to easily jump back to that location.

3.2 Edit a Canvas

3.2.1 Description and Priority

A user with Internet access and sufficient permissions upon authentication may use a suite of bitmap tools to edit the content of the canvas. Changes to the canvas made by users will appear in real time to other such users viewing the same areas of canvas. [High Priority]

3.2.2 Functional Requirements

Canvas.Select	Users selects a section of the canvas to perform operations on using the select tool.
Canvas.Select.Copy	using the copy tool will copy the currently selected section. If no section is selected (or if the area selected is 0) the operation will do nothing.
Canvas.Select.Paste	If there is a item stored in the clipboard and the paste tool is used then the item stored in the clipboard will be placed (with is top left corner) at the currently selected spot of the canvas.
Canvas.Select.Rotate	Rotate the selection 90Degrees Clockwise.
Canvas.Select.Scale	Increase or decrease the relative size of the selection.
Canvas.Select.Shear	Shear the selection. If nothing is selected do nothing.
Canvas.Tool.Select	Clicking on a tool in the toolbox sets that tool to be active.
Canvas.Tool.Draw	Use the active tool to modify the canvas. The specific behavior will depend on the selected tool.
Canvas.Undo	Revert the last canvas modification that this user performed. Will only revert modifications done by the user that triggers this action. In addition, it will only revert modifications done in the current session.
Canvas.Redo	Re-apply the last undone canvas modification for the user undone during the current session. Only available if the previous operation was an undo.

3.3 Control a Section of the Canvas

3.3.1 Description and Priority

The controller of a section of a canvas is a user with the ability to restrict access to non-administrator users for that section. Any user with the Artist role can request control of a section of the canvas as long as no other user controls the section. [Low Priority]

3.3.2 Functional Requirements

Canvas.Region.Aquire	Adds the current user as the controller for a section of the canvas. You can only add acquire a section if the current user is an Artist and if no other Artists control the section.
Canvas.Region.Lock	Lock a region that you control to prevent other users from editing it.
Canvas.Region.Unlock	Unlock a locked region that you control.
Canvas.Region.AddPermittedUser	Allow another user to edit an area you control even if it is locked.
Canvas.Region.RemovePermittedUser	Remove the user's permission to edit the locked region.
Canvas.Region.Release	Remove yourself as controller for the section. If the section is locked, the lock is removed.

3.4 Administer the Server

3.4.1 Description and Priority

An administrator or operator with sufficient privileges will be afforded access to configuration and administration tools upon authentication. Changes that affect the canvas or users will be executed while the server is still running. [Medium-High Priority]

3.4.2 Functional Requirements

Canvas.User.Kick	Disconnects a user from the server, and prevents that user from reconnecting for a customizable amount of time.
Canvas.User.Ban	Disconnects a user from the server, and prevents that user from reconnecting ever again, or until unbanned. (uses IP addresses as the basis for user identity)
Canvas.User.Pardon	Unbans a user.
Canvas.User.Permissions	Modifies the permission set for a user.
Canvas.User.Groups	Modifies the user groups the user belongs to.
Server.User.Permissions	Modifies permissions for a user for all canvasses at once.
Server.Groups.Permissions	Modifies the permission set for a group.
Canvas.Start	Allow users to connect to a canvas. If the canvas does not exist, a new one is created.
Canvas.Stop	Kick all users currently editing the canvas, save the current state, and prevent any users from connecting to the canvas.
Canvas.Autostart	Sets whether or not a canvas starts when the server starts.
Server.Start	Starts the server, starting all canvasses set to autostart.
Server.Stop	Gracefully stops all canvasses, and kills the server process.
Canvas.MaxUsers	Modifies the maximum number of users that can be connected at a single time to a single canvas.
Server.MaxUsers	Modifies the maximum number of users that can be connected at a single time, across all canvasses.

4 External Interface Requirements

4.1 User Interface Requirements

UI-1: A help menu will be available throughout the entire program. There will be a help section for each tool as well as instructions on how to use the program as a whole.

UI-2: The system will require both a keyboard and mouse to use all of the features.

4.2 Software Interfaces

SI-1: Standard User Client:

- SI-1.1: The standard user client shall be available as a standalone client as well as embedded in a web page.
- SI-1.2: The client shall display a log-in screen on launch. The user will be prompted to specify a server, a canvas name, a username, and their password. Once those are entered, the user can attempt to connect to the server by pressing the enter key or by pressing the log in button.
  - SI-1.2.1: If anonymous viewing is enabled, then a user can leave the username and password fields blank.
  - SI-1.2.2: If a default canvas is specified, then a user can leave the canvas field blank.
- SI-1.3: Once a client connects to the server, they shall see a section of the canvas. The window will also have the view manipulation options. If they have logged in and have edit permissions on the canvas, they shall see a set of tools on the side of the window.
- SI-1.4: An authenticated user has a section control panel in the window. If the section is controlled by another, the options in the panel will not be available. If they control the section, they will see the interface for the actions specified in section 3.3. If no one controls the section, they will have the option to acquire control of it.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

PR-1: Allow up to 20 users to simultaneously work on a single server without a reduction in performance from the single-user scenario.

PR-2: Have a delay smaller than 4 seconds between one user editing a canvas and the edit appearing on other user's screens. This limit is based on the processing time on the server and clients, not on the latency of the user's Internet connections.

PR-3: The time difference between scrolling within a block and scrolling between blocks should be less than 500ms.

5.2 Safety Requirements

No safety requirements have been identified.

5.3 Security Requirements

SE-1: All authentication will be performed over a secure channel.

SE-2: Users will be required to log in before they can perform edits, assuming default server configuration.

SE-3: The system will only permit designated administrator users to modify other users' accounts.