James Clark

Hussam Hashem

Dr. Yugyung Lee

CS 551

February 22, 2013

Project Proposal

# Introduction

Group 5 is a University of Missouri – Kansas City (UMKC) student software engineering team comprised of James Clark and Hussam Hashem, instructed by Dr. Yugyung Lee for CS 551: Advanced Software Engineering.

## Purpose

This project proposal introduces Strikethrough: A Whiteboard Communication Project scheduled for final presentation on May 7, 2013.

## Intended Audience and Reading Suggestions

Sections 1, 2, and 3 are recommended for elementary and secondary educators, educational theorists, and computer science students and instructors. The technical aspects outlined in section 4 and 5 are intended for students and instructors from any branch of information technology, particularly computer science.

## Reading Conventions

This document follows a general numeric outline format. The selected diagrams are not intended to fill a complete collection but rather to highlight the major components of all the system features. Diagrams were selected in a manner so as not to serve repetitive information.

## Project Scope

We need a system that demonstrates our proficiency developing software in the agile process and a service-oriented architecture. We have decided to create an HTML5-centric whiteboard and communication application for desktop and mobile users. The system will be used to capture mouse or touch events and store them in a database. Users will access the service after creating a free user account. Users will be able to create and join groups. Groups will empower sharing and collaboration between whiteboard assets. The system will be ideally suited for a “show your work” type of mathematics assignment and collection process.

## References

See the bibliography at the end of this proposal for a full list of references.

# Project Goal and Objectives

## Overall Goal

We aim to provide an electronic medium for educators to collaborate with students in an open online environment that could be useful either in a classroom or across the world.

## Motivation

Two developing trends in education are the reverse classroom and blended learning. In a reverse classroom environment the teacher sits in the back of the room and uses technology (ie computers, educational software, and projection equipment) to instruct lessons. Blended learning is centered on the idea that cognitive overload occurs when instructional presentations are too long. To overcome the challenge of cognitive overload, teachers can break a lesson into a variety of manageable tasks. For example, a lesson on math can include a pre-quiz, one video, a short demonstration, and a set of practice problems. Addressing these two trends, this project is motivated by a desire to create an innovative educational application that empowers blended learning and is useful in a reverse classroom environment. At the same time, this project provides an opportunity to learn about modern trends in software design.

## Specific Objectives (Problem Statement)

The primary objective of this project is to capture canvas drawings into image format and store them in a database or file system. Then, provide an interface where users across a variety of web browsers (especially mobile) can retrieve their data, make changes, resubmit, and generally organize progress. Teachers and students can share problem sets and work together either in a computer lab or across the internet. Teachers monitor and manage their groups while individuals can share outside of a group. Anyone can be a teacher; they simply need students to join a group.

## Significance

A collaborative whiteboard management system provides an opportunity for teachers and students to track their graphical data entries and communicate over the web or side by side in a classroom. A whiteboard experience escapes normal typeface and provides a platform well suited for manual scientific and mathematical calculations. Computers do not readily interpret handwriting, so a whiteboard experience is ideal for human to human interaction. Without the free-writing experience of a whiteboard, students and teachers alike may be challenged by entering mathematical formulas in standard font typefaces.

## System Features

An HTML5 canvas will serve as a virtual whiteboard. Students and teachers can create user accounts. Teachers can create groups and enroll students. Students can enroll in groups, share notes with other students, or simply maintain a personal collection of notes. Canvas can generate PNG graphics.

# User Classes and Characteristics

There will be three user classes: user, owner, and admin. User and owner can be used interchangeably. They are essentially the same with the exception that owner is an attribute applied to a group wherein select group functionality becomes available to the owner, such as invite or remove a member. Admin should inherit user and should include access to additional reporting features and tasks of the application. Admin tasks might include lock an account, unlock an account, view an activity report.

# Operating Environment

This system will run on Windows Server 2008 R2, IIS 7.5, and C#.NET Framework 4.0. Client-side access will be expected from any modern web browser. Clients must use a web browser that supports the HTML5 canvas element, and Javascript must be enabled.

# Project Background and Related Work

## Work Done by Others

This system is going to utilize the Kinetic.js javascript library for animating the HTML5 canvas, and will demonstrate concepts learned by following the tutorials hosted by http://www.html5canvastutorials.com/.

## Similarities and Differences

Market-ready products feature interactive HTML5 canvas approaches, but their toolbars are more in depth, featuring additional education tools like a calculator or set of colors for markup. Better performance may be achieved by writing platform-specific software (apps). The AWW application does not provide a persistent data storage mechanism.

# Related Work

An online learning environment is exemplified in Learn to Be, an online tutoring environment. A good example of collaborative canvas can be found at AWW: A Web Whiteboard. Khan Academy inspired the idea that anyone can be a teacher as long as they have students who can learn from them. There are numerous related and more advanced projects on the market. While it is beyond the scope and timeframe of our project, related applications usually feature real-time communication executed in online chat, video conferencing, voice over IP, and dynamic graphics like a whiteboard that updates for more than one person simultaneously.

# Challenges

Challenges include mixing .NET with JavaScript to handle canvas upload and download as PNG format. A decision needs to be made regarding how to manage storage, retrieval, and isolation of file system data. We need to develop logic for sharing and management between teachers and students. We are inexperienced with adapting a web application to display appropriately for a variety of mobile devices, especially Android.

# Proposed System

## Requirement Specification

### Functional Requirements

### Interface

Drawing should be limited to inside the canvas border. Canvas size should adjust dynamically to platform and available screen resolution. Users should be able to create more than one canvas and link them in a row as a continuous document. Any canvas assigned to a group should not be copied in the database; rather, it should be referenced and then applied as a background layer.

### Business

Data should be stored in the database server in such a manner that a logical log exists which is useful for troubleshooting bugs or errors reported by users. It should be possible to construct SQL queries that yield useful information about visitor activity.

### Non-functional Requirements

### Operational

The system should be able to work on updated versions of Android, iOS, FireFox, and Chrome.

### Performance

With minimum recommended client settings of 1 gb system ram, 1.8 ghz dual core processor, and 5 mb broadband download speed, user interactions with the system should not exceed 3 seconds. User interaction with the HTML5 canvas element should not exceed ½ second. The system should only be unavailable during critical updates, not to exceed 1 hour.

### Security

Users should be able to maintain a section of private, unshared data that is independent of any group membership. Group owners should be able to maintain a collection of data shared one way from any member to the owner. Members in a group should not be able to restrict the group owner from accessing group data.

### Technical Requirements

The system will log minimal activity including error messages, member login activity, relevant times. Passwords will be stored one-way encrypted.

### Business Requirements

Provide a means for educators to distribute a problem set. Students should be able to see assigned problem sets and respond. The instructor should be able to easily access student responses in a logical and organized manner and provide feedback. Completed transactions between instructor and student should be retained in an easy-to-access history summary.

### Business Process

Process Decomposition Diagram



### Workflow Analysis

### Architectural Requirements

This project requires implementing a service-oriented architecture. As a result, a minimum number of web services must be utilized by the project.

### Framework Specification

### Assumptions and Principles

We believe the internet hosts the greatest opportunity for providing a modern education in an efficient, feature-rich manner. We

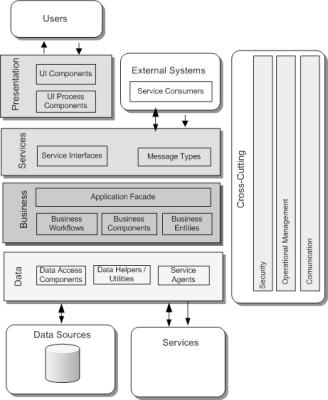
### Methodology and Algorithms

We will apply an agile software approach to implementing this system. See diagrams for basic read-write algorithms with the canvas element and user membership.

### Design Pattern

Creating web methods will allow us to reuse remote code from client to server. When web methods are not feasible or appropriate, the system will be designed in external class files so that NUnit testing will be more efficient.

### System Architecture Diagram



source: http://blogs.msdn.com/b/jmeier/archive/2008/11/24/application-architecture-diagrams-added-to-codeplex.aspx

### System Specification

### Existing Services

### freegeoip.net

An IP address logging mechanism will extrapolate geo location data. This is useful for reporting use statistics, but it also yields an opportunity to demonstrate service oriented architecture in a useful manner. This particular service is a RESTful service that returns JSON string format.

### Google Reverse Geocoding API

Google’s reverse geocoding API can extrapolate physical addresses from latitude and longitude data. This may be useful in setting up user profile data, but it has ethical implications of requesting user permission before accessing information users may regard as private. One unnecessary benefit as a result of including this service in our project might be to learn about the accuracy of the API as well as some security implications across device platforms.

### New Services to Build

### Kinetic Service

This service is intended to support the read, write, assignment, sharing, and notification methods related to the canvas element graphics. Eventually we would like to serve entire images based on the JSON data entries generated by Kinetic’s javascript library.

### Use Case Scenario



### Use Case Specification Template

|  |  |
| --- | --- |
| Use Case Name | Create a group in the dashboard system |
| Scope | Canvas drawing system |
| Primary Actor | User |
| Preconditions | User successfully authenticated with the system |
| Guarantee (Postconditions) | Success message will display.  Group will be added to user’s group list. |
| Main Success Scenario | Users selects “Create a Group”  User enters group name  User submits request  Request is approved  Group is added to user’s group list |
| Extensions | If group name is already taken, notify user |
| Special Requirements |  |
| Frequency |  |
| Miscellaneous |  |

### Class Diagram



### Sequence Diagram



### Service Specification

The Kinetic Service will provide methods for storing, viewing, and recreating canvas content.

### Operational Description

Three key operations will exist: read, write, and image. The read methods should provide lists of user canvases and their series. Write should apply one canvas series to storage. Image should provide a means for downloading a canvas. The image method could be used in a printing system.

### Input/ouput, Pre/post Conditions for Services

Canvas input and output data will be transported in JSON string format. The Kinetic JS library has methods for doing this. The canvas will be parsed to string before it goes into the database, then parsed to Kinetic objects when it reaches the client-side.

### Actors/Objects Involved

Users decide when to save, load, assign, or share their canvas.

### Constraints/Exceptions

Baseline error handling should result in saving a canvas to a user’s personal collection. If problems occur at this baseline, an admin should be notified through an error logging mechanism.

### Service Flow/Alternative Flow

Service calls should not occur in loop-style functions. Logic should be executed in the query structure and application processing. The flow of a service should be two way from client to database, passing through a web server.

### Priorities (Degree of Importance, Difficulty, Etc…)

The two most important areas because of difficulty are the kinetic animation and business logic (ie managing canvas views and access i/o). Finding the right public services for our project has proven difficult, and as we look at new services our project requirements seem to be changed or adapted to accommodate incorporating a new web services.

### Authorization Service

A lack of documentation exists at this time as it is a late concept entry. The basic premise is email authorization. Custom keys sent to users during password recovery can be sent through an authorization service. Upon receiving a call to the service passing the correct key in an acceptable time frame, user account email ownership can be verified.

### Plan by Services

### Tentative Schedules for Project

There are four major project increments, and we have identified four major project components: UI design, kinetic animation, service implementation, access rights and membership, business logic.

### Schedule for Four Different Increments

### Increment 1

UI design and kinetic animation

### Increment 2

Service implementation

### Increment 3

Access rights and membership

### Increment 4

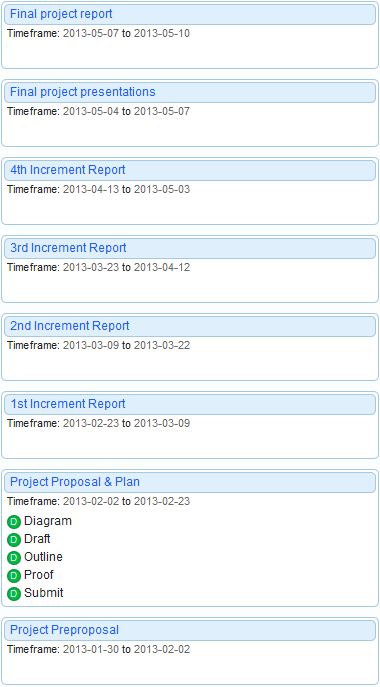
Business logic

### Design by Service

### Build by Service

### Project Timelines, Members, Task Responsibility

At this time, both contributors are scheduled for all project tasks and responsibilities.



### Internet Access

### Project Website

<http://vhost1738.site1.compute.ihost.com/>

### Agilefant

<http://vhost1377.site1.compute.ihost.com:8080/agilefant/login.jsp>

### Github

<https://github.com/JamesWClark/Strikethrough>

### Bibliography

### References

### J.D. Meier's Blog

<http://blogs.msdn.com/b/jmeier/archive/2008/11/24/application-architecture-diagrams-added-to-codeplex.aspx>

### AWW: A Web Whiteboard

<http://awwapp.com/>

### Blended Learning and the Reverse Classroom

<http://reverseclassroom.blogspot.com/>

### Khan Academy

<https://www.khanacademy.org/>

### Learn to Be

<http://www.learntobe.org/>

### Libraries

[doodle-js](https://github.com/lamberta/doodle-js)

[KineticJS](http://kineticjs.com/)

### Proprietary Software

[Pusher](http://pusher.com/)

[Union Platform](http://www.google.com/url?q=http%3A%2F%2Fwww.unionplatform.com%2F%3Fpage_id%3D2762&sa=D&sntz=1&usg=AFQjCNHNCtnJX4I3ujvfUnqniwUkO7VzBA)

### Related Tutorials

<http://www.html5canvastutorials.com/>

[Create a Drawing App with HTML5 Canvas and JavaScript](http://www.google.com/url?q=http%3A%2F%2Fwww.williammalone.com%2Farticles%2Fcreate-html5-canvas-javascript-drawing-app%2F&sa=D&sntz=1&usg=AFQjCNFw_r9DuRhYRCxfscd7YPahQZn_7g)

[Canvas Tutorial - Mozilla Developer Network](http://www.google.com/url?q=https%3A%2F%2Fdeveloper.mozilla.org%2Fen-US%2Fdocs%2FHTML%2FCanvas%2FTutorial%3Fredirectlocale%3Den-US%26redirectslug%3DCanvas_tutorial&sa=D&sntz=1&usg=AFQjCNFN2XpcXyN1FACH_gtaqNr8JBA-kg)

[HTML5 Canvas Tutorials](http://www.google.com/url?q=http%3A%2F%2Fwww.html5canvastutorials.com%2F&sa=D&sntz=1&usg=AFQjCNH4biRdcxs03an7dPO1tgq-BKlUsw)

[KineticJS – HTML5 Canvas Drawing Made Easy](http://thechangelog.com/kineticjs-html5-canvas-drawing-made-easy/)

[Real-time data exchange in HTML5 with WebSockets](http://www.google.com/url?q=http%3A%2F%2Fwww.adobe.com%2Fdevnet%2Fhtml5%2Farticles%2Freal-time-data-exchange-in-html5-with-websockets.html&sa=D&sntz=1&usg=AFQjCNG3vVxgvKwRBJlViiysgSDriQUYOQ)