chronoview

Design Document

2/27/2012

Rev 2.0

Shea Orr

Mark Brownell

Adam Richards

Scott Seward

Table of Contents

1. Introduction

Chronoview's aim is to provide quality timeline creation software aimed towards instructors, students, and layman. Rather than creating a timeline in software that has been tailored for a specific domain, Chronoview will be able to accomplish any timeline-related task. Users will have the ability to save their timelines to our server, browse and chose from previously entered events, and share their creations across the web.

As a whole, Chronoview will give users the ability to:

- Upload interval and event data
- Generate timeline from this data
- Interact with the generated intervals, allowing them to shift interval positions, change colors, and group intervals into categories
- Add events to any given interval
- Save the timeline for later viewing or editing
- Export their timeline as an image

2. Architecture Overview

2.1. Website & Parser

When the user decides to investigate our software, the first thing he or she will encounter is our webpage. The home page will introduce the user to Chronoview and its features. It will also have a small FAQ. From here, the user will be able to click a "Get Started!" button which will redirect them to a registration page if not logged in, or directly to the timeline creation page if so. The registration page allows the user to register with the site. They are immediately redirected to the timeline creation page.

From a link on the timeline creation page, users will be able to browse to a timeline file that they have created, and upload it to the site. The built-in file parser will parse the file and interpret it in to timeline data. The uploaded timeline will then be displayed. If the user does not have a timeline file to upload, they may begin immediately inputting data and designing their timeline.

Every web page will contain a toolbar at the top of the web page which allows users to logon, logoff, and view their account.

The web page will be implemented using the Twitter Bootstrap toolset; a flexible toolkit of CSS, HTML, and Javascript.

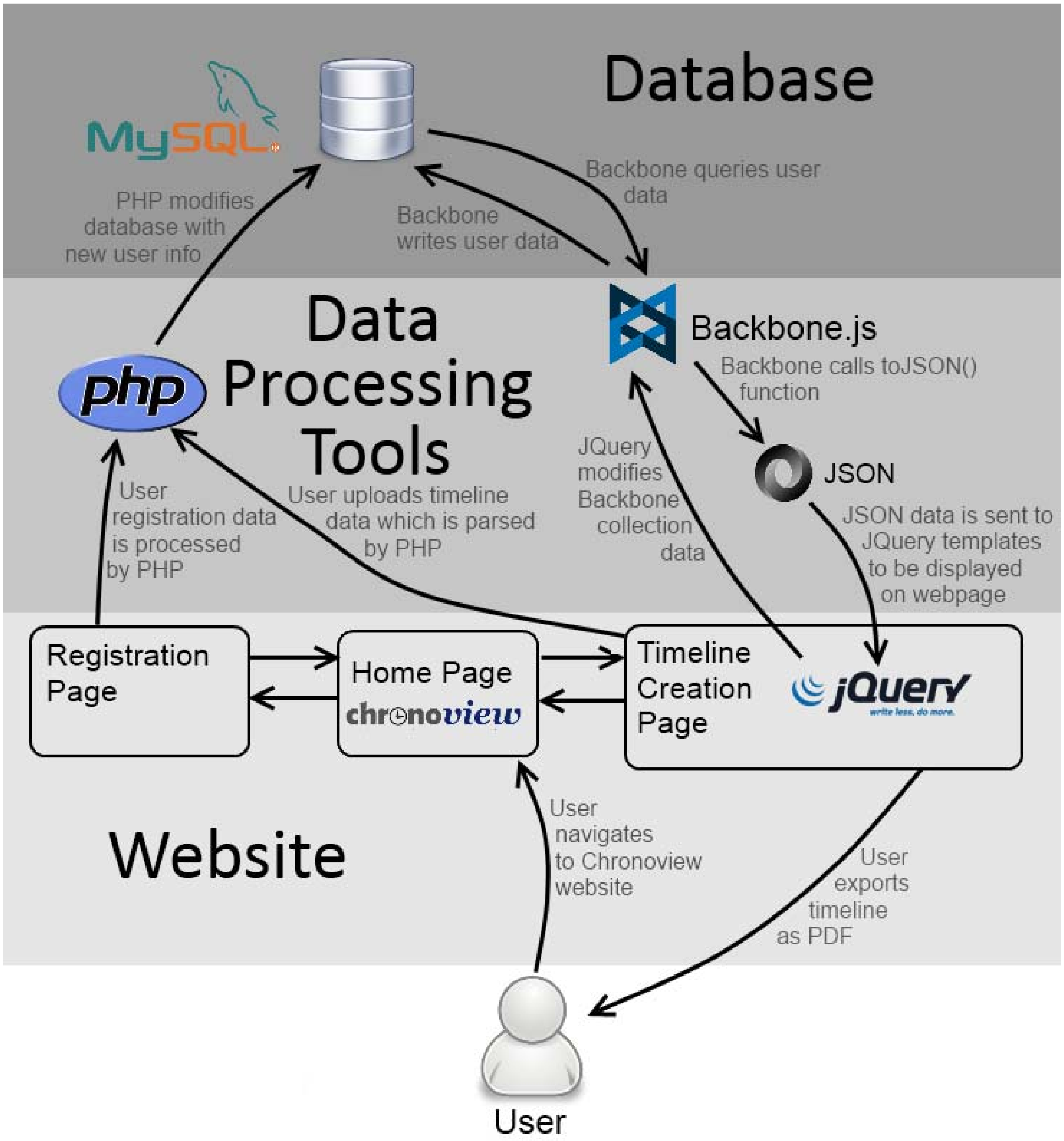
2.2. Database

The database backend will store all user and timeline information. The database will be accessed when a new user registers, tries to login, logs off, uploads a timeline, saves a timeline, etc. It will be implemented using MySQL.

2.3. Timeline Creation Interface

The timeline creation interface is the main bulk of this project. Once users have logged in and navigated to this page, they will be able to begin creating a timeline. The timeline creation interface allows users to add intervals events and descriptions, export their timeline into PDF format, or generate a link to their timelines in order to share with friends. Finally, they can save their timeline for later viewing or editing.

2.4. Visualization



Database
MySQL
PHP modifies database with new user info
Backbone writes user data
Backbone queries user data
Data Processing Tools
php
Backbone.js
Backbone calls toJSON() function
JSON
JQuery modifies Backbone collection data
JSON data is sent to JQuery templates to be displayed on webpage
User registration data is processed by PHP
User uploads timeline data which is parsed by PHP
Registration Page
Home Page
chronoview
Timeline Creation Page
jQuery
write less, do more.
Website
User navigates to Chronoview website
User exports timeline as PDF
User

3. Detailed Module Descriptions

3.1. Website & Parser

The website with parser will provide for the main interaction with the timeline software. The website will give users a clear and easy to use graphical interface that will allow a user to create and modify timelines of their choice. The parser will provide for extra functionality so that a user can upload an external file that defines a timeline. This will be most useful when a user has an existing data file that they wish to easily translate to a timeline.

The goal of the web site is to provide users with an easy way to create and share their information. Users should be able to have personal accounts and be able to share their work with other collaborators.

3.1.1. Site Map

The web page will contain a basic home page that every user will see upon their initial visit, a registration page on which to register a new account, and a timeline creation page.

3.1.1.1. Home Page

The home page will include basic information about the software. This information will relate to what the software is, what it's used for, and why someone should use it. The home page will also contain an example timeline that a user can interact with to further understand the capabilities of the software. An example timeline should demonstrate the basic features that our timelines can possess, such as multiple intervals, nested intervals, and events. The main page will link further to a frequently asked questions (FAQ) section, for answers to common questions about the software that may not be apparent or provided on the home page.

3.1.1.2. Registration Page

The registration page will permit new users to register for accounts. A user must register for an account before they are able to edit any existing timelines or create and save any timelines. The registration page will ask for basic identifying information about the user, such as desired user name, password, and email address. There may be further optional information provided such as place of employment, current city, and a short user bio.

3.1.1.3. User Profile Page

When a user is registered and logged into the web page they will be allowed to view a personal profile page that will display information about their account. This information includes, which timelines they have created and which timelines they have been given privilege to edit. They also have the ability to further customize their personal information such as their bio, current city etc. The user can also change their email address and password from this page and can even delete their account if needed.

3.1.1.4. Timeline Creation Page

The timeline creation page provides direct access to all timeline creation tools. On this page a user will be able to upload an external document and have it converted to a timeline. They will also be able to create a timeline from scratch. Buttons will be provided to the user to add new intervals and events and modify existing intervals and events on their timeline. Users will also be able to load and edit any timelines that they have saved previously. Users will be able to add and change permissions to their saved timelines to allow other collaborators access to edit their timelines. A detailed description of this module can be found in section 3.3: Timeline Creation Interface.

3.1.2. Login/Logout Functionality

The home and registration pages will also give a registered user the ability to login to & logout from the system. A toolbar will be visible on the top of the web page and will allow a user to log in, or navigate to the registration page. When a user is logged into the system they are able to create new timelines, or view existing public or shared timelines. These abilities will be accomplished through a timeline creation and editing page.

3.1.3. Parser

To upload an existing file that describes a timeline the user will need to use the built in file loader to locate the file on their local machine and upload it to the server. When a file is uploaded it will need to be parsed through the use of PHP. A PHP function will read a given file and separate out all relevant information that pertains to timeline elements. This parser will enter in to the database all parsed elements such as when intervals begin and end, which events belong in an interval, and the detailed information of each interval.

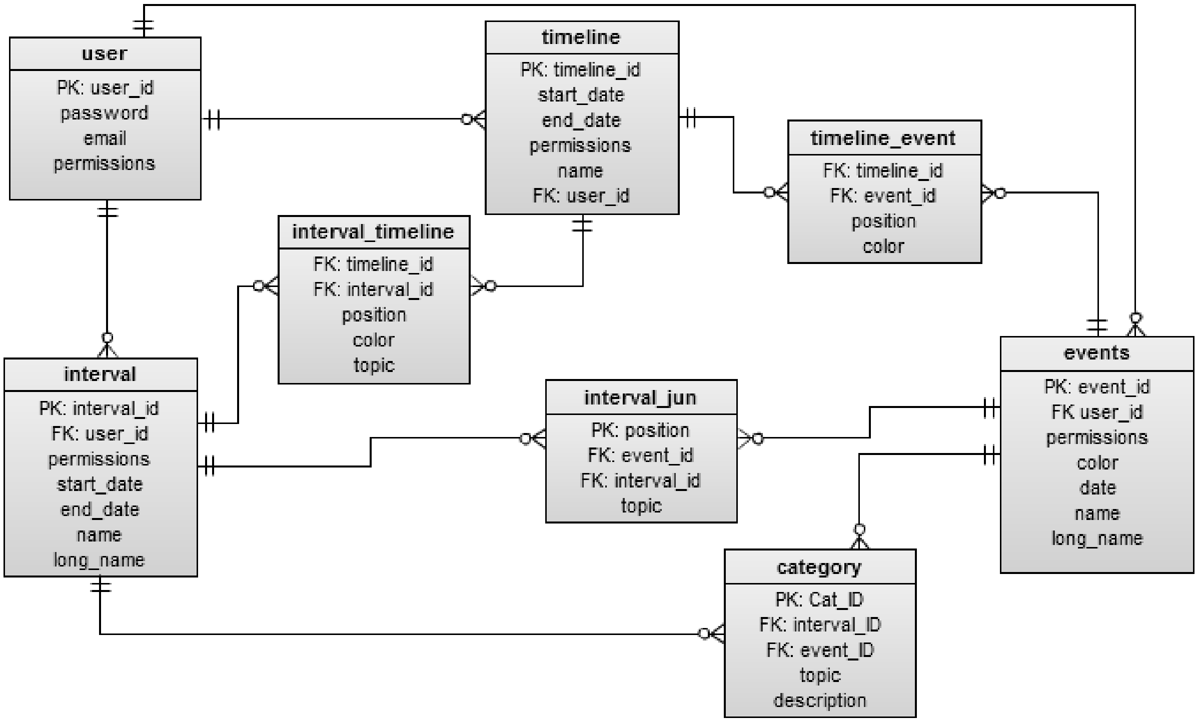
3.1.4. Tools

The main web page will be constructed mostly of HTML. This HTML will be organized and stylized by CSS through the use of Twitter's Bootstrap kit. Some PHP can also be included to ease the design and layout of the main page, but overall it will be relatively static. The only non-static and interactive portion will be the use of an interactive timeline for example and demonstration purposes.

The login/logout toolbar will interact with the database through the use of PHP MySQL queries. When a user attempts to log into the page, a PHP MySQL query will be sent to the database and the input will be checked against known users. If the user is present and has the correct password then the client's web browser will begin to store session information related to the state of their login, primarily that they are correctly logged in.

3.2. Database

3.2.1. Visualization



3.2.2. Table Map

3.2.2.1. User

The user table is a list of user and user specific information. While it doesn't hold any values linking events, timelines or intervals. The Primary key is used in other tables to link the user to their created items.

- Primary key: user_id(varchar)- is the username for each specific user. Each username is unique for that user.
- password(varchar) – password with a minimum of 8 characters
- email(varchar) – email relating to the user, each email can only be used once in the system.
- permissions(int) – contains an integer specifying what kind of a user they are. Administrator – 2, registered user – 1, banned user -3, unregistered user – 0. (registered and unregistered pertains to whether or not the users email has been verified)

3.2.2.2. Events

The events table holds the list of all the events used on the site. This is used mainly between timelines and intervals, but can also be linked to User table because of who creates them.

- Primary Key event_id(int) – unique identifier for each event, this is system created.
- Foreign Key user_id(varhar)- user_id from the user table.
- Permissions (int)- contains the value of who will be allowed to edit the event.
  - Private* – 0, no one else can edit it or see it.
  - private-public* – 1, others can see it but not edit it.
  - Public – 2, everyone can see and edit it.
    * Administrators can view/edit all created event
- Color(varchar) – specific color that is default for that event.
- Date(date)- Date the event occurred.
- Name(varchar) – Short name of the event, if you had a book it would just be the books title.
- Long_name(varchar) – longer name for an event, if you had a book not only did it list the book but who the book belonged to.

3.2.2.3. Interval_jun

Junction table between the intervals and events, this is needed for a many to many relationship. It basically links the events to a specific interval.

- Junction Table – used to create a many to many relationship
- Foreign Key event_id(varchar) – same from timeline table.
- Foreigh Key interval_id(varchar) – same for interval table.
- Primary Key Position(varchar) – the position on the interval for the event
- Topic(varchar) – contains a topic for each interval/ event, like Political Science or Philosophy.

3.2.2.4. Interval_timeline

Junction table between interval and timeline tables this is need for the many to many relationship. It basically creates the link between these two tables without repeating any unneeded data.

- Junction Table – used to create a many to many relationship
- Foreign Key timeline_id(varchar) – same from timeline table.
- Foreigh Key interval_id(varchar) – same for event table.
- Position(varchar) – the position on the timeline for the event
- Color- the color the event should be on the timeline.
- Topic(varchar) – contains a topic for each interval/ event, like Political Science or Philosophy.

3.2.2.5. Interval

The interval table holds the basic data for the list of intervals, this is linked to the timeline, events, and user tables through various junction tables and foreign keys.

- Primary Key interval_id (int) – unique id for the interval created.
- Forgein Key user_id(varchar) – from the user table specifying who created the interval.
- Permissions (int)- contains the value of who will be allowed to edit the interval.
- Private* – 0, no one else can edit it or see it.
- private-public* – 1, others can see it but not edit it.
- Public – 2, everyone can see and edit it.
  * Administrators can view/edit all created interval
- Start_date(date) – the starting date for the interval
- End_date (date)– ending date for the interval
- Name(varchar) – user given name of the interval, like Isaac Newton.
- Long_Name – Longer name for the interval, like Sir Isaac Newton's Life.

3.2.2.6. Timeline

The timeline table holds a list of timelines or their basic data this is linked to the events, intervals, and user tables through various junction tables and foreign keys.

- Primary Key: timeline_id (int)– unique identifier for each event, this is system created.

- Name - timeline_id (varchar)– each timeline will contain aid specified for the user at timeline creation.

- Foreign Key: user_id(varchar) – contains the original creator of the timeline, which gives them extra option for each timeline, depending on permissions and editing that timeline, or sending a link to edit that timeline.

- Start_date(date) – the starting date for the timeline

- End_date (date)– ending date for the timeline

- Permissions (int)- contains the value of who will be allowed to edit the timeline.

  Private* – 0, no one else can edit it or see it.

  Private-Public* – 1, others can see it but not edit it.

  Public – 2, everyone can see and edit it.
  * Administrators can view/edit all created timelines

3.2.2.7. Category

Category holds the specific information for each event or interval, you can have many section in this table for the same event or interval depending on the topic used for that event or interval.

- Primary Key cat_id(int) – a unique id for the category generated by system.

- Foreigh Key interval_id(varchar) – same for interval table.

- Foreign Key event_id(varchar) – same from timeline table.

- Topic(varchar) – contains a topic for each interval/ event, like Political Science or Philosophy.

- Description (var char) – In depth description of interval or event. This could be just text, images or imbed video.

3.2.3. Tools

3.2.3.1. MySQL

The database will be implemented using MySQL.

3.2.4. Detailed Database Functionality

3.2.4.1. User Deletion

When a user is deleted not everything by that user will be deleted, deletion will start with timelines checking to see if permissions are full private for that timeline(once a timeline is set to public it cannot be set to private again or deleted by the user). Then the intervals are looked for deletion not only do they have to be set to private but they must also never be used in a timeline(by removing the timelines first this removes conflicts within the timeline table). Followed by events which must be full private and never used in another users timeline or intervals.

3.2.4.2. Interval Deletion

An interval can only be deleted here if it is not linked another users timeline. Administrators can remove intervals regardless of the timelines it is in.

3.2.4.3. Event Deletion

An event can be deleted as long as it has never been used in another users intervals or timelines, unless done by the Administrator.

3.2.4.4. Timeline Deletion

A timeline can only be deleted by the user created it when the timeline is marked as private, or by the Administrator.

3.2.4.5. Permissions Change

Timelines can only move up with the permissions from private to public, since there is no other way to ensure that someone else might not be using that timeline for something, unless set by the administrator themselves.

3.2.4.6. Duplicate Events

To ensure each event in unique Category table was created, by allowing different topics for each event the user instead of having to create a special event for just a different description they can add a topic instead. This is looked at when events are going to be added, name and long_name variables in the event are compared with what the user adds and checks for correlation.

3.2.4.7. Duplicate Named Intervals/Timelines

Name and long_name variables are looked at and then used to compare with the two tables finding a public timeline or interval already there these are then used as a suggestion, which would allow the user to create a new table with much of the same information, or slightly modify an existing table.

3.3. Timeline Creation Interface

3.3.1. Timeline Generation

The timeline generation module is responsible for presenting the user interface to create new timelines, as well as the actual programming that implements and stores the entered timeline information.

This module will be a separate area of the ChronoView public website. Users must first register in order to view the timeline creation interface. Once logged in, a user will be able to view a list of already created timelines, as well as create a new timeline. Editing an existing timeline or creating new will bring the user to the timeline creation interface.

The timeline creation interface will display the title of the timeline currently being worked on at the top. There will be menus which will allow users to manage event and category data, collaborate with others, preview and export the timeline. Drag and drop functionality will be available to move intervals to the desired location, as well as drag events to their appropriate intervals and bind them together.

The timeline creation module will rely heavily of client side functionality. Due to this, the module will be implemented using an MVC (Model View Controller) called Backbone.js in order to efficiently manage the timeline interface, as well as separate the interface from the back end functionality.

3.3.1.1. Backbone

Backbone.js provides a necessary structure to the ChronoView project. With multiple developers working on the same implementation, it is important to use a set of standards and impose a structure on the data models and implementation techniques. Backbone will give the ChronoView team a unified method to model data, and convert these data models into JSON format, to be interpreted using jQuery templates (*see section 3.3.1.2 for more details*)

3.3.1.1.1. Models

Using Backbone, a model is a representation of a set of data. Within each timeline, individual intervals and events will be modeled using Backbone. Associated information including event name, dates, description, category and more will be attached to each model. Modeling our data in this fashion enables users to make changes or additions and have them quickly and accurately reflected in the visual interface, with little interaction on behalf of the developers. Event handlers, which recognize that a change has been made, will automatically reflect that change in the database using AJAX calls.

3.3.1.1.2. Collections

A collection is a set of models. Within a timeline, a collection represents all of the associated intervals and events. Each interval or event is represented by a separate model. Backbone allows ChronoView developers to set up a model class, of which other models can be instantiated. Implementing this way makes every interval or event behave in the same fashion, reacting the same way when updated, without duplicating any code.

3.3.1.2. JQuery Templates

jQuery templates will be used to parse JSON formatted (often termed JSON-ified) data output using Backbone models into HTML format. These templates are constructed within script tags on the timeline creation interface. jQuery templates are capable of if statements, for loops, and specific display functionality which we believe will be able to parse JSON timeline data into a visual HTML timeline display. These templates will separate logic from the user interface, and provide a consistent way to develop timeline displays using various data and style elements.

3.3.1.3. CSS

This HTML format generated using the jQuery templates will rely on custom CSS classes that are either previously existing or created dynamically by the software users. Users are able to create custom CSS elements indirectly by specifying the color and priority of categories. A number of CSS classes will be provided by default to allow categories of standard colors.

3.3.2. Interaction

Once the timeline data is modeled, stored and visually displayed, the user must be able to interact with and manipulate the visual characteristics of the timeline. The user may decide to reposition interval placement within the timeline (mostly vertically) or attach (bind) certain events to their respective intervals. The user also will have the ability to attach images or files to intervals or events within the timeline. Images may also be embedded and positioned statically within the timeline for decoration purposes. Once the timeline display is completely set up, the user will be able to export the timeline to JPEG or PDF format. All of these user interaction functions will be accomplished through a sophisticated jQuery user interface.

3.3.2.1. JQuery User Interface

The interface providing timeline manipulation will be implemented using jQuery. All interaction with generated timelines will occur on the client side, and will send information back to the database when necessary. The backbone models will be responsible for detecting updates and reporting information back to the database. The jQuery interface will update the Backbone models.

3.3.3. Exportation

After visually manipulating the timeline to desired specifications, the user will be able to easily export the timeline to either JPEG or PDF format for inclusion in external documents such as reports, presentations etc. Exportation will be accomplished by capturing a screenshot of the iFrame containing the timeline, and saving it to the specific format.

3.3.4. Social

An important aspect of the ChronoView web-based timeline software is the fact that users can collaborate, share timelines and discuss various aspects. When creating a timeline, users will be able to view similar events that others have already input for the time period they are working on. Once a substantial amount of data is accumulated within the database, it will become easier for users to create timelines based on existing data.

3.3.4.1. Share on Facebook

Specific timelines can be shared on Facebook, to receive reaction from the community associated with the user that created it. This option will be easily accessible through the "Collaboration" menu option described in section 3.3.3.2.

3.3.4.2. Share via E-mail

Another option for sharing created timelines will be an e-mail invite tool. Filling out how many friends a user would like to e-mail, and entering their e-mail addresses will send out e-mail invitations (good for those users only) to view the timeline and post comments.

4. Implementation Plan

4.1. Milestones

Feb 25, 2012 - Begin Prototyping Interface
This coming weekend, Adam will begin prototyping the ChronoView timeline interface. This will include testing selected technologies including Backbone.js, JSON notation, and jQuery interaction functionality. Interface mock-up will be completed to guide prototype development.

Feb 27, 2012 - Database Implemented
The database schema, including all tables, will be created by Mark on the Mac server in room 104, with PHPMyAdmin installed so other team members can view tables, attributes and relationships for development purposes.

Mar 5, 2012 - Website Setup Complete
A basic, functional front-end visitor website will be set up. This will allow visitors to

ChronoView to be able to learn what the site is all about, and sign up for accounts in order to create their own timelines (*which comes later*). The website will be implemented by Shea and Scott.

Mar 12, 2012 – Excel / Spreadsheet Parser Complete
The parser that will handle uploading event data in Excel workbook, CSV, or Tab-delimited files will be complete by March 12th. This upload tool will notify the users of required file format and column arrangement for upload. This tool will parse event data and store in the database for later retrieval by the software.

Mar 12, 2012 – Timeline Viewable
This milestone describes the main functionality of the ChronoView website. The timeline software must be able to visually generate a timeline based on either uploaded or manually entered event data. (*Manipulation comes later*)

Apr 5, 2012 – Working Interface Prototype
On this date, we must have a working timeline creation / manipulation interface complete. Users will be able to interact with timeline elements, attach pictures & files, export, and all functionality listed in the requirements document.