This document contains a description of all development tools utilized by Create-A-Page, as well as sequence diagrams, the entity-relationship diagram, and overall design decisions that our group found would be the best solution for this project.
Development Tools

Server-side tools

- **Apache** - We chose to use Apache because it is one, if not the most popular open source web servers used today. It is also pretty lightweight and fairly simple to install, configure, and maintain.

- **MySQL** - MySQL was the RDBMS of choice for the same reasons as Apache; it is open source and simple to setup.

- **PHP** - We decided to use PHP for its simple but yet robust API. Installation and configuration is also fairly simple.

Client-side tools

- **Dojo** - Dojo is an open source modular JavaScript library (or more specifically JavaScript toolkit) designed to ease the rapid development of cross-platform, JavaScript/Ajax based web applications and web sites. The toolkit provides widgets that are essentially prepackaged components of JavaScript code, HTML markup and CSS style declarations that can be used to enrich websites with various interactive features that work across browsers. Examples of widgets are menus, tabs, layouts, animation effects, drag-and-drop, and calendars, just to name a few. Another important feature of Dojo is that it provides an API for Ajax programming. Ajax provides asynchronous communication with the browser and server. The information is exchanged and the page's presentation is updated without a need for reloading the whole page. Traditionally, this is done with the JavaScript command XMLHttpRequest. Dojo provides an abstracted wrapper around various web browsers' implementations of XMLHttpRequest, thus providing its own API for performing the asynchronous server communication. For these reasons, we chose to use Dojo in our project because of its widgets and the increase in page performance, thus providing a more enjoyable user experience.

- **JSON** - JSON (JavaScript Object Notation) is another computer data interchange format. It is a text-based, human-readable format for representing simple data structures and associative arrays (called objects). We chose to use it over XML because it is lightweight and is heavily used in Ajax programming. It is also 100 times faster than its predecessor.

- **HTML** - HTML is the obvious markup language. Raw HTML was used in conjunction with the Dojo widgets.

- **CSS** - CSS is used extensively throughout the web application for creating the custom layouts.
Design Decisions

Introduction
When deciding on our overall design we considered a couple of similar applications to adopt some features, yet attempting to improve some features as well.

One of the major influences in our design was Google sites. Although it is Google, and it provides a very robust and customizable webpage creation, we found there were some features we wanted to improve. One of the major improvements we wanted to make was to minimize the effort in creating a page and navigating through pages to customize the settings and style of your webpage.

With Create-A-Page we have created the ability to create a page with little effort and customize the look and feel of your website in real-time. The goal of Create-A-Page was to have the ability to “edit as you view” and have all changes to your website viewed simultaneously by visitors to your website.

Client vs. Server
Create-A-Page was created to be service available to a group or community of user’s rather than a standalone client application. By hosting this service on a server it creates a “community” of pages accessible by the same URL “http://hostname/createapage/?username=username”) making it easier to remember.

Also, because Create-A-Page requires Apache, MySQL, and PHP, it is easier for the casual computer user to create a website without worry about configuring any software.

Flat files vs. Database
When deciding on how to save the user pages we had to consider flat files or a database. We found that advantages of a database were:

1. Platform-independent
2. Readily searchable due to indexing
3. Security – allowing only owners to view pages and/or images

Page redirect vs. Popup
One of, if not the major goal of this project was to provide everything up front for the administrator. We wanted to minimize having to navigate through pages to create a page and to design the overall look of
your website. When you consider popups you usually think of unnecessary clutter. We utilized Dojo to accomplish many of these popups. It allowed us to present information in a clean and unobtrusive way. Examples of this are editing the user page settings and also modifying the user account settings. On the server side, it also minimized the overall amount of PHP scripts needed to accomplish each task.

**Administrative Console**

We found it necessary to provide a super-administrator that would have complete control over user accounts. The super-administrator can edit and delete accounts, as well as reset passwords for possible locked out users.
Depending on the user, there are three ways to log into Create-A-Page. The three ways differ depending on the actor logging into the system. This sequence diagram shows the steps involved in logging into the system. NOTE: The order in which the actors logs in is irrelevant. They are listed separately only so the diagram is readable to others and the lines do not confuse the reader.
For the regular admin, the actor that creates and manages pages, he/she goes to the main site (currently [http://babyhuey.cis.temple.edu/createapage](http://babyhuey.cis.temple.edu/createapage)), which causes index.php to load. Index.php checks to see if a session was created by that computer. If not, then the user is directed to login.php, where it asks for login credentials. The admin sends the credentials, which are checked with the database (DB in diagram). The database sends confirm/deny to login.php, which informs index.php of results. Then index.php informs the admin if the login was successful or not. If login successful, index.php displays the main page of the admin, who can now begin working within his/her account.

For the super admin, the actor that manages regular admin accounts, he/she goes to the main site (currently [http://babyhuey.cis.temple.edu/createapage](http://babyhuey.cis.temple.edu/createapage)), which causes index.php to load. Index.php checks to see if a session was created by that computer. If not, then the user is directed to login.php, where it asks for login credentials. The admin sends the credentials, which are checked with the database (DB in diagram). The database sends confirm/deny to login.php. If unsuccessful, it informs index.php, which informs end user login credentials invalid. If successful, login.php sends the super admin to appadmin.php, which notifies super admin of successful login by displaying the super administrator’s main page, where the super admin can manage admin accounts.

For the regular user, the actor that can only view pages created by administrators, he/she goes to the main site (currently [http://babyhuey.cis.temple.edu/createapage/?username=username](http://babyhuey.cis.temple.edu/createapage/?username=username)), which causes index.php to load. Index.php checks to see if the username provided in the URL is a valid user. If not, then the user is directed to login.php, where it asks for login credentials. However, the regular user has no login credentials, and therefore cannot do anything unless it creates an account. If it was a valid username and the owner of this website has published his/her website, then the user will be allowed to navigate the website.

Create-A-Page: Creating a page
Only administrators can create pages in Create-A-Page. To do this, the admin clicks the “Create” tab in the admin bar at the top of the administrator’s main page (index.php). This makes action_content = create, which causes the page creation info to appear. This is done by attaching the WYSIWYG editor to a text area that opens so admin can add contents of new page. A text box opens as well for naming the new page. Index.php displays this info in the administrator’s browser. The admin then adds content desired in the new page. When complete, the admin saves the page contents. This calls the DB, which confirms the file saving. Index.php then displays the contents of the saved page to the admin in view mode as confirmation of the save.
When editing a page, the administrator selects the page he/she wants to edit from a dropdown menu in the admin bar. Once selected, index.php requests the page from the DB, which returns the page requested. Then, index.php opens the WYSIWYG editor, which attaches itself to a text area for editing the page. The page is then sent to the administrator’s browser so it can be edited as the action_content is set to “edit_content”. Once the administrator is done making changes to the page, he/she saves the changes, which are committed to the DB. When the save is confirmed, it is loaded to the administrator in view mode.
MySQL Database tables

4. **user_t**
   - This table contains all the information about a user. All of the fields self explanatory, but that last field “publish” tells us that this user’s page is published or not.

5. **user_content_t**
   - This table contains all user page information. The visible and position fields tells us if this page should be listed on a user’s navigation bar and in what order these pages should be displayed.
6. **user_page_settings_t**
   - This table contains all the user’s page settings. Each field represents a particular CSS element for each section of a user’s website.

7. **user_sec_questions_t**
   - This table contains all the user’s answers to the security questions used for identifying a user when using the password retrieval utility.

8. **sec_question_t**
   - This table contains a list of predefined security questions that a user may choose from when creating their account.

9. **user_images_t**
   - This table contains all the user’s images that were uploaded through the image upload utility. The images can be used for backgrounds as well as regular images used throughout a user’s webpage.

---

**PHP Docs**

*Please refer to PHP Docs folder included in this Documentation package.*