Dragon Toolkit

Software Requirements

Michael Bergonzoni
Tiffany Godette
John Kim
Tim Paczkowski
Brandon Schow
## Contents

1 Abstract .................................................................................................................. 4
2 Introduction ............................................................................................................. 4
   2.1 Purpose ........................................................................................................... 4
   2.2 Scope ............................................................................................................... 4
   2.3 Definitions ...................................................................................................... 4
3 General Description .................................................................................................. 5
   3.1 Functional Overview ..................................................................................... 5
      3.1.1 Website Overview ............................................................................... 5
      3.1.2 Tools Overview ............................................................................... 5
4 Functional Requirements .......................................................................................... 5
   4.1 Website ........................................................................................................... 5
      4.1.1 Navigation ............................................................................................. 5
      4.1.2 Page Body ............................................................................................. 6
      4.1.3 Account Features ............................................................................... 6
      4.1.4 Help and Documentation ..................................................................... 6
      4.1.5 Contact ................................................................................................. 6
   4.2 Tool Wizards ................................................................................................... 6
      4.2.1 Purpose ................................................................................................. 6
      4.2.2 Interface .............................................................................................. 6
   4.3 Tools ................................................................................................................ 8
      4.3.1 DBanner ............................................................................................... 8
      4.3.2 DBannerZoom ................................................................................... 10
      4.3.3 DBreadCrumbs ................................................................................. 12
      4.3.4 DLearn ................................................................................................. 13
      4.3.5 DFormBuilder ................................................................................... 15
      4.3.6 DHangman ......................................................................................... 17
      4.3.7 DImageNav ........................................................................................ 19
      4.3.8 DLinkDetective ............................................................................... 21
      4.3.9 DMaps ................................................................................................. 23
      4.3.10 DSurvey ............................................................................................. 25
      4.3.11 DWordSearch ................................................................................... 27
5 Non-Functional Requirements ................................................................................ 29
   5.1 Website Layout ............................................................................................... 29
      5.1.1 Header ................................................................................................. 29
      5.1.2 Footer ................................................................................................. 29
      5.1.3 Navigation ........................................................................................... 29
      5.1.4 Page Content ...................................................................................... 29
   5.2 Tool Wizards ................................................................................................... 29
      5.2.1 Position ............................................................................................... 29
      5.2.2 Template Layout ............................................................................... 29
   5.3 Security ........................................................................................................... 29
1 Abstract

Many web developers do not have the programming experience or the time to develop some of the more advanced components of a website. DragonToolkit is a collection of easy-to-use wizards that facilitate the creation of web components. A webpage will provide links to various tool wizards, where the user can create components customized for their specific needs. The wizard generates cross-platform code that can be implemented in the user's specific domain. This is intended for users with some experience with web design languages such as HTML and CSS or with a WYSIWYG editor.

2 Introduction

2.1 Purpose

The purpose of this document is to outline the interface for DragonToolkit. This includes the general website as well as the wizards for generating tool code.

Functional requirements are listed that describe the user's interactions with the website and wizards.

Non-functional requirements describe the layout of the interface.

2.2 Scope

This document lists the requirements to build the software. It is intended as a guide for developers to correctly implement the system.

2.3 Definitions

- **Website**: collection of pages allowing the user to access the tool wizards and other user options
- **Wizard**: guided tool to help user to generate customized component code
- **Browser**: a software application for retrieving, presenting, and traversing information resources on the World Wide Web (WWW)
- **Code**: the code is the set of instructions for a computer governed by a particular language
- **Cross platform code**: the set of instructions for computer that is portable and can be ran on most if not all operating systems; this means that a particular file or application (or even certain hardware) can be used on more than one computer platform
- **CSS**: cascading style sheets are declarations that describe how a document should be presented on the Web; CSS can be written for different media, can be part of an HTML document, or can be a separate file that is applied to multiple HTML documents
- **Frame**: a frame is a method of displaying multiple HTML documents on one page
- **Mark up Language**: a markup language is a system for annotating a text in a way which is syntactically distinguishable from that text
- **HTML (HyperText Markup Language)**: predominant markup language for web pages that provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, etc as well as for links, quotes, and other items

- **WYSIWYG (What You See Is What You Get)**: term used in computing to describe a system in which content displayed during editing appears very similar to the final output

- **Web Application**: a computer software application that is hosted in a browser-controlled environment or coded in a browser-supported language and reliant on a common web browser to render the application executable.

- **URL (Uniform Resource Locator)**: global address of documents and other resources on the World Wide Web.

### 3 General Description

#### 3.1 Functional Overview

The primary function of the application is to allow users to access and run wizards which generate usable code. The web application provides additional functionality to help achieve this goal.

#### 3.1.1 Website Overview

The website provides an interface for accessing individual tools, as well as other website functions.

#### 3.1.2 Tools Overview

Each tool provides a wizard for allowing the user to customize the component. The output of the wizard is usable code to add the component to a website.

### 4 Functional Requirements

#### 4.1 Website

**4.1.1 Navigation**

4.1.1.1 General

4.1.1.1.1 Provide a link to home page

4.1.1.2 Tools

4.1.1.2.1 Opens a window to allow the user to browse tool wizards.

4.1.1.3 Account

4.1.1.3.1 Provide link to register new account

4.1.1.3.2 Provide link to log in to existing account

4.1.1.3.3 Provide link to log out of account
4.1.1.3.4 Provide link to account page
4.1.1.4 Help and Documentation
4.1.1.4.1 Provide link to help and documentation
4.1.1.5 Contact
4.1.1.5.1 Provide link to contact information

4.1.2 Page Body
4.1.2.1 Contains current content viewed by user

4.1.3 Account Features
4.1.3.1 Account login and registration
4.1.3.1.1 Allow user to register new account. Requires user to input login name, password, and email address
4.1.3.1.2 Allow user to log in to account. Requires user to input login name and password
4.1.3.1.3 Allow user to log out
4.1.3.2 Functions available to logged in users
4.1.3.2.1 Allow user to save list of favorite wizards
4.1.3.2.2 Allow user to save wizard sessions
4.1.3.2.3 Allow user to continue saved wizard sessions

4.1.4 Help and Documentation
4.1.4.1 Allow user to access documentation pages

4.1.5 Contact
4.1.5.1 Allow user to access contact information
4.2 Tool Wizards

4.2.1 Purpose

4.2.1.1 Guide user through input collection to final code output
4.2.1.2 Collect input to customize tool
4.2.1.3 Allow intermediate functions Save, Cancel
4.2.1.4 Generate code to create customized tool

4.2.2 Interface

4.2.2.1 Wizard pages presented sequentially to gather input (see state diagrams for individual tools listed below)
4.2.2.1.1 Contains title of tool
4.2.2.1.2 Contains input fields specified in state diagram
4.2.2.1.3 Contains navigation and function buttons specified in state diagram
4.2.2.1.3.1 “Back” moves to previous window
4.2.2.1.3.2 “Next” moves to next window
4.2.2.1.3.3 “Save” saves the state of the wizard for logged in users
4.2.2.1.3.4 “Generate Code” sends wizard input to generator and creates output
4.2.2.1.3.5 “Add/Modify” alter previously inputted fields
4.2.3 Output

4.2.3.1 Customized code meeting specifications of the user through the wizard is generated and provided after completing the wizard.

4.2.3.1.1 Final page after wizard displays a text field with generated code. Code can be copied by user.
4.3 Tools

4.3.1 DBanner

4.3.1.1 Sample Output

Rotating banner that displays multiple (linked) html pages. Page timing can be user defined. Playback buttons will also allow users to play/pause or seek to specific html pages.

4.3.1.3 Wizard Input Fields

4.3.1.3.1 Width- integer for the width, in pixels
4.3.1.3.2 Height- integer for height, in pixels
4.3.1.3.3 Display Time- integer to display each page, in seconds
4.3.1.3.4 Location- a list of the pages to be added to the banner
4.3.1.3.5 DBanner Page 1…n- string for the page location, as a path

4.3.1.4 Wizard Interface
4.3.2 DBannerZoom

4.3.2.1 Sample Output

![Sample Output Image](image-url)

4.3.2.2 Purpose

Provides a magnified viewing window to display images. The user moves the window over the image using the mouse cursor, which magnifies the image at that point and displays the zoomed window at the cursor location.

4.3.2.3 Wizard Input Fields

- 4.3.2.3.1 Image URL- URL consisting of a string specifying the image location
- 4.3.2.3.2 Percentage- Integer decreasing the image size to a specific percentage
- 4.3.2.3.3 Width- integer for the width, in pixels
- 4.3.2.3.4 Height- integer for the height, in pixels

4.3.2.4 Wizard Interface
Specify the zoom window size

Image URL:

Next
Previous
Save
Generate Code

Legend:
- Input Box
- Disabled Input Box
- Pull Down Menu
- Radio Button

Percentage: [ ] OR Width: [ ] (pixels)
Height: [ ] (pixels)

Percentage: [ ] OR Width: [ ] (pixels)
Height: [ ] (pixels)
**4.3.3 DBreadCrumbs**

**4.3.3.1 Sample Output**

A trail of hyperlinks is automatically generated that allows a user to track their current location inside a website. Also, the user may jump to any page previously visited within that site.

**4.3.3.3 Wizard Interface**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>string for the URL of a site, as a path</td>
</tr>
<tr>
<td>Style</td>
<td>the appearance of the arrow showing the path</td>
</tr>
<tr>
<td>Color</td>
<td>hex code for the text color between the arrows</td>
</tr>
</tbody>
</table>

**Legend**

- Input Box
- Radio Button
4.3.4 DLearn

4.3.4.1 Sample Output

Create customized flash cards that are presented on a web page. Flashcards may contain text and/or pictures. Various viewing layouts may be set by the user.

4.3.4.2 Purpose

4.3.4.3 Wizard Input Fields

4.3.4.3.1 Title- string for the title of the flash card
4.3.4.3.2 Description/Instructions- string for how to use/operate flash cards
4.3.4.3.3 Front- string description for the front of the flash card
4.3.4.3.4 Back- string description for the back of the flash card
4.3.4.3.5 Title 1…n- string for the title of the flash card
4.3.4.4 Wizard Interface
4.3.5 DFormBuilder

4.3.5.1  Sample Output

Feedback Form

Name

Email

Feedback Type [Suggestion]

Comment

Submit

4.3.5.2  Purpose

Add forms with customized input fields to a page. Validation options will be included to filter content for malicious inputs and to verify the integrity of the data.

4.3.5.3  Wizard Input Fields

4.3.5.3.1  Name- A string containing the name of the form
4.3.5.3.2  Submit Action- a String containing the action of the form
4.3.5.3.3  Field List- A list of fields to be added to the form
4.3.5.3.4  Field Name- The name of each field
4.3.5.3.5  Field Type- The type of each field

4.3.5.4  Wizard Interface
**4.3.6 **DHangman

4.3.6.1 Sample Output

- Tries: 5
- Letters Used: O B A D Z
- Hints: Greatest NFL team of all time

---

### Sample Output Image

---

4.3.6.2 Purpose

Generates a hangman game using a given word list from the webmaster. The user will then be prompted to enter letters; every wrong guess causes a new part of the 'victim' to appear on the scaffold. A successful guess will cause the letter to appear in the appropriate position on-screen. The game ends when the word has been filled in or the 'victim' dangling on the noose has been completed.

The user is able to restart the game with a new word.

The word list is automatically generated from the website.

4.3.6.3 Wizard Input Fields

- **Word/Phrase** - A string consisting of the word to be used in the hangman puzzle
- **Hints** - A list of strings for hints on the mystery word in the puzzle
- **Strikes Allowed** - An integer from a list allowing how many guesses a user can make before the game ends
- **Hangman Image** - An optional field allowing the user to provide a path for a hangman image
- **Background color** - An optional field that changes the background color of the hangman game
- **Time Limit** - An integer that limits the amount of time a user has to play the game. By default, there is no time limit
- **Generate / Enter Word or Phrase** – Choose to enter a string or have words generated from the website entered
- **Location** – A URL for a website that will be used to generate a list of words
- **Words Generated** – A list of words generated from the website entered
Software Requirements

4.3.6.4 Wizard Interface

DHangMan

- Enter Word/Phrase
- Generate Word/Phrase

*(If no selection is made, generate word/phrase will be used.)*

Strikes Allowed: 1, 2, 3, 4, 5, 6 (default)

Next

DHangMan

Hang Image: {path to file}

*(If no image is specified, the default image will be used.)*

Background Color: FFFFFF (default)

Time Limit: 0 (default) (minutes)

* A time limit of 0 implies no limit. *

Previous Save Generate Code

DHangMan

Location: {path to website}

Words Generated: Word 1, Word 2, Word 3, ...

Hints:

Previous Next

DHangMan

Word/Phrase: (100 characters max)

Hints:

Previous Next

Legend
- Delete Button
- Input Box
- Pull Down Menu
- Radio Button
4.3.7 DImageNav

4.3.7.1 Sample Output

![Sample Output Image]

4.3.7.2 Purpose

Creates graphical buttons from an image. The image is split into pieces, each of which is used as the background for a button. This allows the user to easily create a menu bar with a consistent theme.

4.3.7.3 Wizard Input Fields

4.3.7.3.1 URL - a string containing the address that the original picture is located at.
4.3.7.3.2 Link 1...n - a string containing the URL that each button will point to
4.3.7.3.3 Label 1...n - a string containing the label that will be displayed on each button

4.3.7.4 Wizard Interface
Software Requirements

Legend
- Delete Button
- Input Box
- Pull Down Menu
- Radio Button

DImageNav

Location: [path to image file]

Number of Buttons:
1.
2.
3.
4.

Next

DragonToolkit

DImageNav

Preview

Previous
Save
Generate Code
4.3.8 DLinkDetective

4.3.8.1 Sample Output

![DLinkDetective Log for racewalk.com](image)

Currently testing for:
- Hyperlinks

Total Hyperlinks: 43
Dead/Invalid Hyperlinks: 2
Percentage of valid HyperLinks: 95%

<table>
<thead>
<tr>
<th>URL</th>
<th>VALID?</th>
</tr>
</thead>
<tbody>
<tr>
<td>yahoo.com</td>
<td>PASS</td>
</tr>
<tr>
<td>one.drexel.edu</td>
<td>FAIL</td>
</tr>
<tr>
<td>ebay.com</td>
<td>PASS</td>
</tr>
<tr>
<td>....:</td>
<td>......</td>
</tr>
</tbody>
</table>

Total time to crawl domain: 1 minute 4 seconds.

4.3.8.2 Purpose

Automatically check for dead/invalid links by traversing to each individual link and returning a detailed report of any issues to the developer.

4.3.8.3 Wizard Input Fields

4.3.8.3.1 Domain- The path to the domain where the DLinkDetective will scan
4.3.8.3.2 Image Locations- Checkbox that allows for checking images on a webpage
4.3.8.3.3 HyperLinks- Checkbox that allows for checking hyperlinks on a webpage
4.3.8.4 Wizard Interface
DLinkDetective

Domain: [path to domain]

Check for:
- Image locations
- HyperLinks
- ....

Legend
- Check Box
- Input Box
4.3.9 DMaps

4.3.9.1 Sample Output

A component that provides a map with directions to a business or other location. Along with the map, the developer may also choose to add total travelled distance and time. Since routes may vary between different mapping services, the developer may choose between Google, Yahoo, and Microsoft (Bing) Maps.

4.3.9.2 Purpose

4.3.9.3 Wizard Input Fields

4.3.9.3.1 Address- string of the address for the current location, set by the webmaster
4.3.9.3.2 Search Engine- Drop-down list of the available map engines for DMap.
4.3.9.3.3 Travel Time- Checkmark to enable total travel time from the webmaster’s address to the user’s address
4.3.9.3.4 Travel Distance- Checkmark to enable travel distance from webmaster’s address to the user’s address

4.3.9.4 Wizard Interface
4.3.10 DSurvey

4.3.10.1 Sample Output

Class Evaluation

Question 1:
I enjoyed this class.
Disagree ○ ○ ○ ○ ○ ○ Agree

Question 2:
I enjoyed this class.
○ True
○ False

Question 3:
What improvements would you suggest?

4.3.10.2 Purpose
Web-based component that enables the creation of surveys with customized questions. The wizard will guide the user during the creation process allowing them to select question types. The application will provide templates for creating effective surveys.

When filled out and submitted, the survey generates an XML file that may be used by the webmaster.

4.3.10.3 Wizard Input Fields
4.3.10.3.1 Title- A string containing the title of the survey
4.3.10.3.2 Descriptions/Instructions- a string containing a descriptions or instructions for the survey
4.3.10.3.3 Question List- A list of questions to be added to the survey
4.3.10.3.4 Question- a string containing the question
4.3.10.3.5 Type- The question type
4.3.10.4 Wizard Interface
4.3.11 DWordSearch

4.3.11.1 Sample Output

<table>
<thead>
<tr>
<th>Time Remaining: Unlimited</th>
<th>Difficulty Level: Easy</th>
</tr>
</thead>
</table>

Generates a word search from a list of words provided by the webmaster. The program will automatically place the words on the playing field, introduce random letters to camouflage them, and then allow the user to select words by highlighting them with the mouse. The game ends when the user selects all the words in the list.

The user is able to restart the game with a new word set.

The word list is automatically generated from the website, where the webmaster may choose words from the generated list and create their own list to use in the word search.

4.3.11.2 Purpose

4.3.11.3 Wizard Input Fields

4.3.11.3.1 Title- A string containing the title for the word search
4.3.11.3.2 Time Limit- The time limit for the game, in minutes
4.3.11.3.3 Difficulty- The difficulty level of the word search
4.3.11.3.4 Word List- A list of words to be used in the word search
4.3.11.3.5 Location- A URL for a website that will be used to generate a list of words
4.3.11.3.6 Words Generated – A list of words generated from the website entered.
### 4.3.11.4  Wizard Interface

**DWordSearch**

- **Title:**

- **Time Limit:** 0 (default) (minutes)

* A time limit of 0 implies no limit. *

- **Difficulty:** Easy, Standard, Hard, Stump Me

---

**Legend**

- Input Box
- Pull Down Menu
- Radio Button

**DWordSearch**

- **Word List**
  - Word 1
  - Word 2
  - Word 3
  - ...

- **Location:**  [path to website ]

- **Words Generated:**
  1. Word 1
  2. Word 2
  3. Word 3
  ...

---

Previous  Save  Generate Code

Previous  Next
5 Non-Functional Requirements

5.1 Website Layout

5.1.1 Header
5.1.1.1 Banner appears consistently across pages
5.1.1.2 Header contains Dragon Toolkit title

5.1.2 Footer
5.1.2.1 Footer appears consistently across pages

5.1.3 Navigation
5.1.3.1 Navigation appears consistently underneath the header

5.1.4 Page Content
5.1.4.1 Page content appears below navigation bar
5.1.4.2 Contains pages navigated to by user

5.2 Tool Wizards

5.2.1 Position
5.2.1.1 Currently selected tool wizard appears in content pane

5.2.2 Template Layout
5.2.2.1 First page of wizard contains title and description
5.2.2.2 Each page contains input fields specific to tool for customized output
5.2.2.3 Wizard pages contain buttons to move forward and back
5.2.2.4 Wizard pages contain cancel button
5.2.2.5 Generate button to output code