**DING DOG SWITCH**

**Liz Kerber**

**August 1, 2016 // Scope of Work, Version 1**

**DOCUMENT OBJECTIVES**

The purpose of this document is to provide clear and detailed information about the work that Liz Kerber will perform, the requirements and specifications of the work, and the final product of the work. By accepting this document you acknowledge your understanding and agreement of this scope of work. Any requirement falling outside the specifications of this document will be considered “Out of Scope” and may require significant changes to the timeline established for this project.

This document supersedes previous estimates generated and any other documentation provided regarding the work to be performed for this project by Liz Kerber.

**SCOPE OF WORK DETAILS**

Liz Kerber will be creating a physical doorbell-like object that will be animal-activated and that will send data via WiFi to a web-based single-page application. The application will send a text message to the animal’s owner to notify them the doorbell has been activated. The application will also allow for personalization of the text messages, store input from the doorbell in a database, and display previous doorbell input.

**COMPONENT**

Doorbell



Web application

Login view

* Site name
* Login
  + Username text input with placeholder “Username”
  + Password input with placeholder “Password”
  + “Go” button to go to main view or login failure view

Login failure view

* Site name
* “Uh oh” header
  + Description of possible issues
  + Link to go back to login view

Register view

* Site name
* Register
  + Username text input with placeholder “Username”
  + Password input with placeholder “Password”
  + Confirm password input with placeholder “Confirm password”
  + “Go” button to go to login view or register failure view

Register failure view

* Site name
* “Uh oh” header
  + Description of possible issues
  + Link to go back to register view

Main view

* Site name
* “Log out” button
* “History” header
  + List of 15 most recent notifications, including date, time, and text content
  + “View all” button to go to notification history view
* “Change the content of your texts” header
  + Label for text input
  + Text input limited to 140 characters
  + “Update” button

Notification history view

* Site name
* “Log out” button
* “History” header
  + List of all notifications, including date, time, and text content

**SUBSECTION**

The doorbell-like object will include a power source, a button, a WiFi transmitter, and a small box that encloses all of the components. The power source will be compatible with either a 9V alkaline battery or USB power. The button will be approximately ½ inch in diameter, with a larger cardboard circle affixed to it for ease of activation; the cardboard circle will be approximately two inches in diameter. The WiFi transmitter will send an alert to the application when the button is pressed using the littleBits Cloud HTTP API. The application will then make an HTTP POST to Twilio, which will send the SMS messages (“text messages”) using their REST API.

Express routes will be used to switch between views on the web application. From the opening view there will be the option of logging in or registering (using Passport). If registration or login is unsuccessful there will be failure views with links back to the registration or login views, respectively.

After successfully logging in the user will be brought to the main view, which will display a history of the most recent text messages, and a text input field that will allow for changing the content of the text messages the user is receiving. All text messages’ content, date, and time sent will be stored in an SQL database. The 15 most recent text messages’ content will be displayed on the DOM, as well as the date and time at which they were sent. The text input for changing text messages’ content will be limited to 140 characters. There will be a button next to the input, and when clicked the value of the input will be saved and used for text messages’ content until changed again.

The notification history view will retrieve all text messages from the SQL database and display them on the DOM, with the most recent on top.

The SQL database will be managed through PostgreSQL, and will store users and hashed passwords, as well as each user’s text message history.

**DEPLOYMENT**

This project will be using the following electrical components:

* littleBits USB power/littleBits battery power
* littleBits button
* littleBits cloudBit

This project will be using the following frameworks and technologies:

* NodeJS, version 6.2.2
* AngularJS, version 1.5.8
* PostgreSQL, version 6.0.3
* Express, version 4.14.0
* Express-session, version 1.14.0
* Passport, version 0.3.2
* Body-parser, version 1.15.2
* Twilio
* Twilio’s REST API
* littleBits Cloud HTTP API, version 2

Git will be used for version control. Heroku will be used for deployment. The application will be deployed upon approval of Ryan Mulcahy and Joel Miller.

**PROJECT MILESTONES**

|  |  |  |
| --- | --- | --- |
| **Title** | **Estimated Date of Completion** | **Actual Date Completed** |
| Scope of work approved | Tues, 08/02/16 | 08/02/16 |
| Doorbell apparatus built | Weds, 08/03/16 | v. 1: 08/02/16  v.2: 08/06/16 |
| Database set up – users, text message history | Thurs, 08/04/16 | 08/06/16 |
| Button presses recorded in database | Thurs, 08/04/16 | 08/05/16 |
| Text messages sent upon button press | Fri, 08/05/16 |  |
| Notification history view functionality | Sat, 08/06/16 | 08/07/16 |
| Main app page functionality | Sun, 08/07/16 |  |
| Passport functionality | Mon, 08/08/16 | 08/04/16 |
| Web app testing | Tues, 08/09/16 |  |
| Deployment | Thurs, 08/11/16 |  |

**BROWSERS SUPPORTED**

Google Chrome’s desktop version 51.0.2704.103 will be fully supported.

**RISKS**

The web application will only be tested on the Google Chrome browser. The author is working under the assumption that the APIs required will be functioning while the application is being built and after it has been deployed. Additionally, the author may encounter unforeseen difficulties with the frameworks being used.

**MOCKUPS**











