**Bob’s Repair Shop v2**

 - End-to-End Billing System -

TECHNICAL DESIGN DOCUMENT

By

Heather Peterson and Johnny Vanderhorst

Bellevue University, NE

 Fall 2018

**Purpose:**

The purpose of this document is to provide an overview of Bob’s Computer Repair Shop v2 application development. The application will provide end-to-end billing functionality and account management for Bob’s Computer Repair Shop. The application will be developed utilizing the MEAN stack tools (Mongo, Express, Angular, and Node.js)

Keys application requirements:

* Sign-in page
* Account registration
* Role management
* Forgot password page
* Security questions page
* Custom 404 and 500 page
* About Us page
* Contact us page
* Service repair page
* Invoice summary report
* Amount total by product graph
* Log files page

**Bob’s Repair Terminology**:

404 Page - Page Not Found error page

500 Page - Internal Server error page

Angular – a TypeScript-based open-source front-end web application platform led by the Angular Team at Google and other communities

AngularJS Material – is both a UI component framework and a reference implementation of Google’s Material Design Specification

APIs – (Application Programming Interface) is a set of routines, protocols, and tools for building software applications. Basically, an API specifies how software components should interact. Additionally, APIs are used when programming graphical user interface (GUI) components

Application Server - is a set of routines, protocols, and tools for building software applications. Basically, an API specifies how software components should interact. Additionally, APIs are used when programming graphical user interface (GUI) components

Express – or simply Express, is a web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs

Heroku - used for deployment; a platform as a service (PaaS) that enables developers to build, run, and operated applications entirely in the cloud

Log Files - a history of actions executed by a database management system and recorded to avoid losses in crashes or hardware failures; also known as a *transaction log*

MongoDB - is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents

mLab - Database-as-a-Service for MongoDB. mLab is a fully managed cloud database service featuring automated provisioning and scaling of MongoDB databases, backup and recovery, 24/7 monitoring and alerting, web-based management tools, and expert support. mLab runs on cloud providers Amazon, Google, and Microsoft Azure, and has partnered with platform-as-a-service providers

Node – (1) an individual computing device in a network; (2) a connection point in a data structure such as a linked list or tree

Node.js - is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model; npm

NoSQL– “non-relational” database that provides a mechanism for storing and retrieving data that is modeled in means other than the tabular relations used in relational databases

Relational Database – a database that consists of tables made up of rows and columns that are used to define relation between the contained data

TypeScript – is an open-source programming language designed for development of large applications and is a subset of JavaScript

**Bob’s Repair Shop User Personas**:



**User Stories:**

As a user, I need a service to that is fast and convenient, so I don’t spend a lot of time getting computer repairs.

As a user, I need an application that is responsive, so that it can be viewed on multiple devices.

As a user, I need to be able to create an account, so I don’t have to reenter my personal information every time I use the application.

As a user, I need to be able to edit my profile, so I can make changes in the future.

As a user, I need to be able to reset/recover my password, so I can regain access if I forget my password.

As a user, I need to be able to review an invoice, so I can confirm the services I selected.

As a user I need to be able to retrieve previous invoices, so I can view past purchases.

As a user, I need to easily find available services, so I don’t have to navigate through multiple pages to find what I am looking for.

As a user, I need the services clearly listed with associated cost, so I know what services are available and how much they cost.

As a user, I need an easy to use interface, so I don’t waste time figuring out how to navigate the site.

As a user, I need to be able to easily find the contact information, so I can call or email Bob’s Computer Repair Shop if I need to.

As an admin user, I need to be able manage roles, so I can manage user’s rights and access.

As an admin user, I need to be able to manage users, so I can create, read, update and delete users.

As an admin user, I need to be able manage security questions, so I can create, read, update and delete security questions.

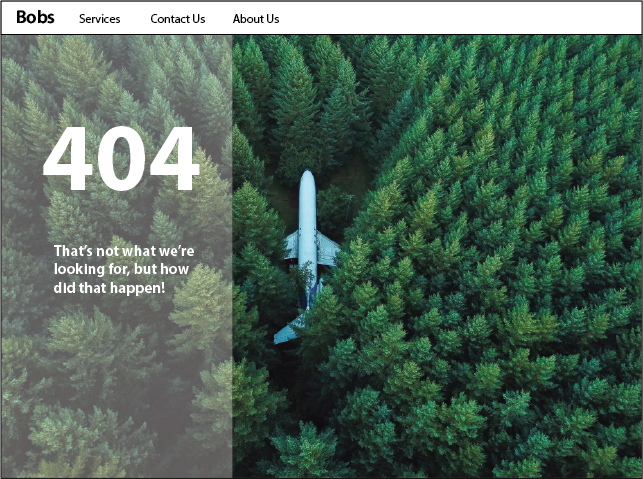
As an admin user, I need to be able to access log files, so I can review application activity.

**Time Estimations:**

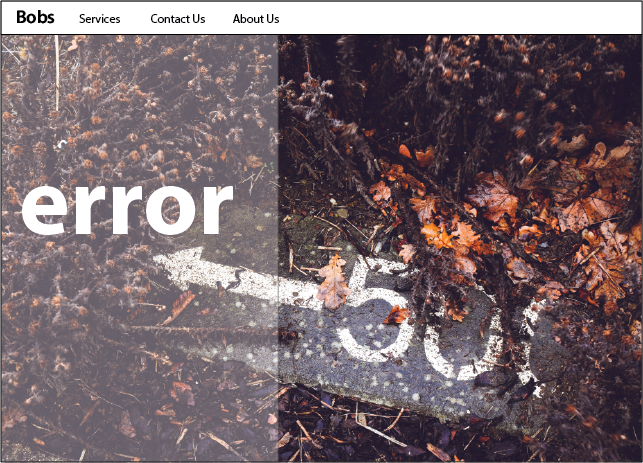
|  |  |  |
| --- | --- | --- |
| ***As a UI/UX web DEVELOPER, I need to create an online computer*** | ***Work*** | ***Points*** |
| ***repair site so new/old users can initiate end-to-end transactions*** | ***Hours*** |  |
| 1. Add functioning login page that allows users to sign-in or sign-up to access. | 3 | 4 |
| 2. Create a registration page for new users to sign-up and automatically creates and updates information. | 3 | 4 |
| 3. Add a role management page that incorporates CRUD (preconfigured roles: standard and admin). | 8 | 10 |
| 4. Add a user management page that incorporates CRUD. | 8 | 10 |
| 5. Developing a Security Questions page with an input of at least 3 questions/answers. | 5 | 6 |
| 6. Developing a Forgot Password page that retrieves username and security question data. | 6 | 8 |
| 7. Develop custom 404 and 500 pages. | 2 | 2 |
| 8. Add external About Us, Contact, Service Repair, and Invoice Summary pages. | 8 | 12 |
| 9. Create an Amount Total by-product graph. | 4 | 5 |
| 10. Create a Log files page. | 5 | 8 |
| TOTAL HOURS: | 52 | 69 |
|  |  |  |
| ***As a UI/UX web DESIGNER, I need to implement easy navigation*** | ***Work*** | ***Points*** |
| ***so that users/admin can easily use/manage the online repair shop*** | ***Hours*** |  |
|  |  |  |
| 1. Create a service that is fast and convenient for users/admin. | 3 | 6 |
| 2. Implement responsive design so site can be viewed on multiple devices. | 1 | 5 |
| 3. Apply browsing services so users can easily navigate when searching. | 3 | 5 |
| 4. Display price of services clearly so there is clear understanding of costs. | 2 | 5 |
| 5. Identifiable external links so users can contact and view history purchases. | 1 | 5 |
| 6. Apply a word doc and images to the About Us Page. | 1 | 2 |
| 7. Create a drop-down list for security questions/answers. | 4 | 7 |
| TOTAL HOURS: | 15 | 35 |
|  |  |  |

**Prototypes**

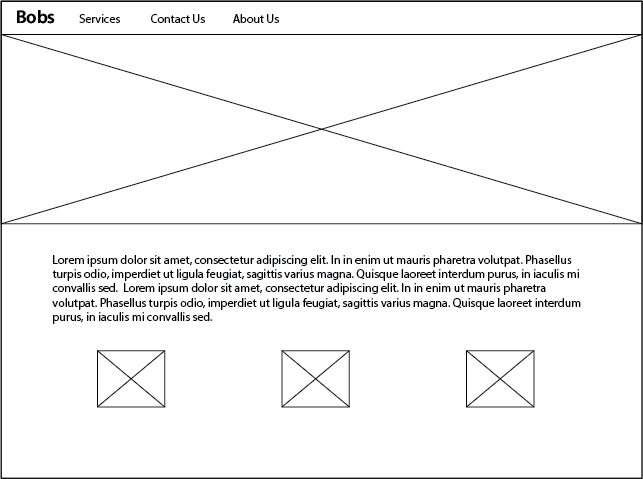
**404 Page**

****

**505 Page**

****

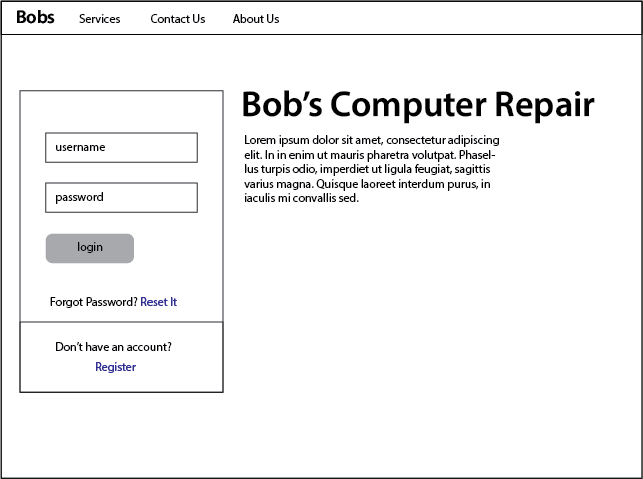
**About Us Page**

****

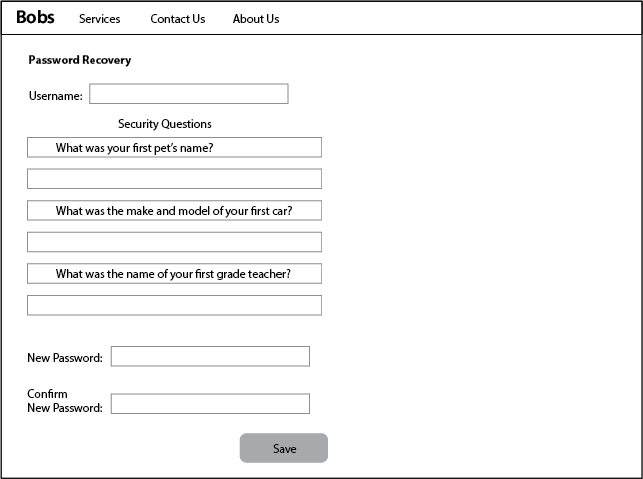
**Contact Page**

****

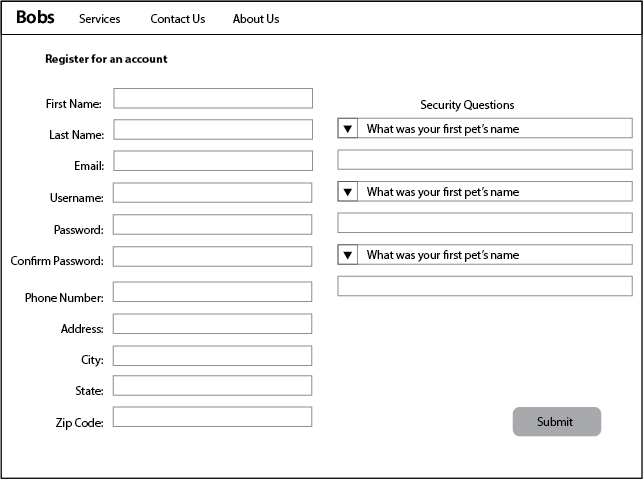
**Login Page**

****

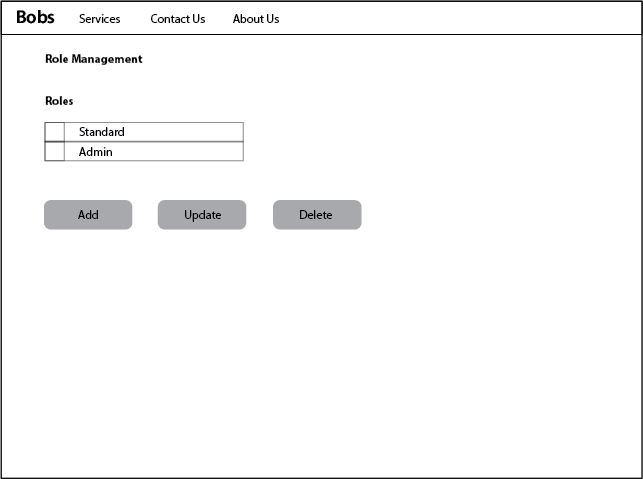
**Password Recovery**

****

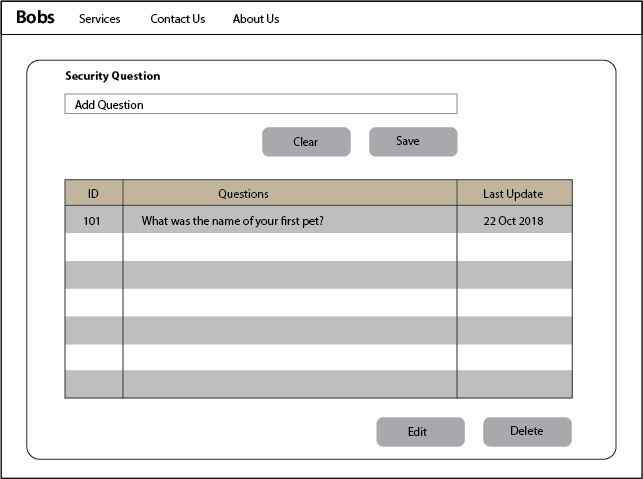
**Registration**

****

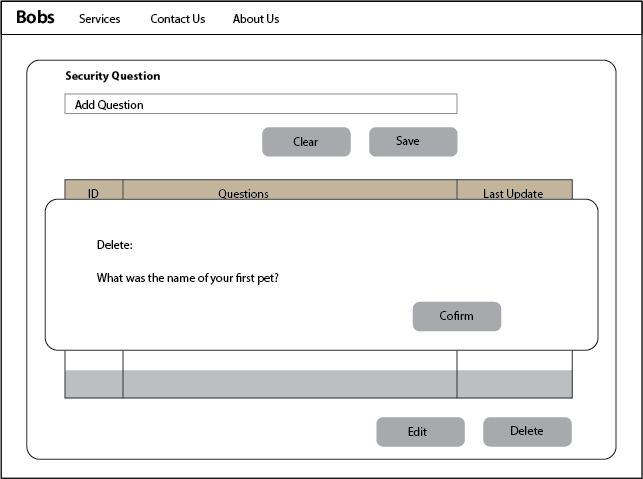
**Role-Management**

****

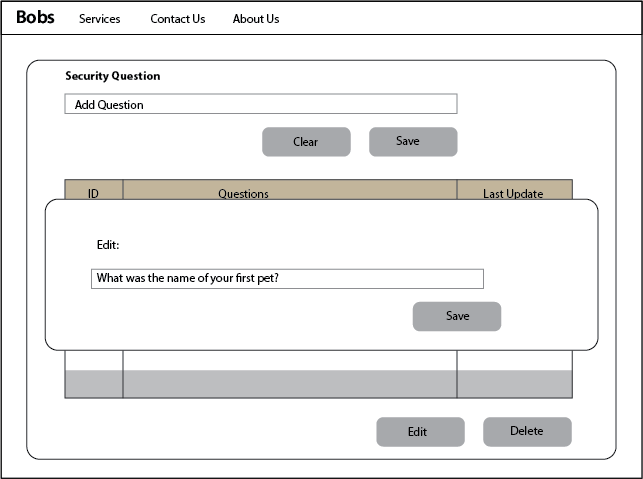
**Security Question Management**

****

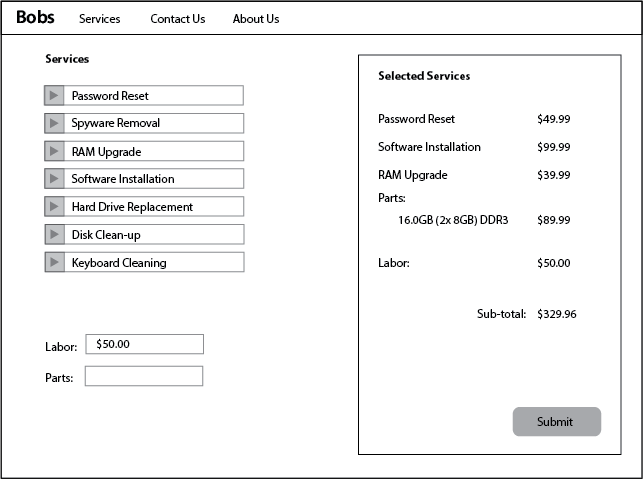
**Security Question Management (Delete)**

****

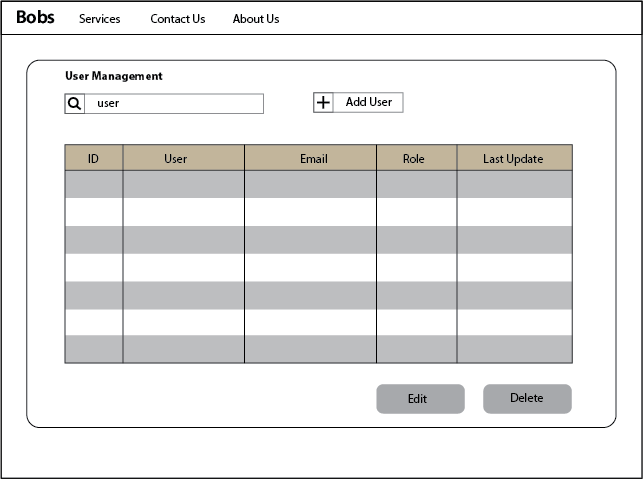
**Security Question Management (Edit)**

****

**Services**

****

**User Management**

****

**User Management Modal**

