nodebucket

BUSINESS REQUIREMENTS DOCUMENT

Author	Professor Krasso
Last Updated	11/15/2019 1:15 PM
Version Number	1.0.0

Version History

Name	Date	Reason for Change	Version
Krasso, Richard	10/17/2019	New document format	1.0.0

<u>Table of Contents</u>

SECTION 1: OBJECTIVE / SCOPE

Overview

nodebucket, is task management software designed to simplify the way you create, track, edit, and delete time sensitive material. This is a full-stack MEAN (MongoDB, Express, Angular, and Node.js) application, utilizing the concepts that were taught in previous courses. MongoDB will be used for data persistence. Node.js will be used for manipulating and returning saved records, SoapUI for unit testing, and Angular for user interactions.

Schedule

The development of this application will encompass 3 Sprints and 5 days. The first 3 days will be for requirements gathering, project planning, and designing. The last 2 days will be for deployment and group presentations.

Required Tools/Frameworks

Name	Comments
MEAN Stack	MongoDB, Express, Angular, and Node.js
VS Code	Development IDE
SoapUI	API testing tool
MongoDB Compass	Database development IDE
Angular Material	https://material.angular.io/
Flex-layout	https://github.com/angular/flex-layout
primeNG	https://www.primefaces.org/primeng/#/
Courses GitHub repository	https://github.com/buwebdev
Slack	Collaboration
Zoom	Virtual meetings
Trello	Task management software
OneDrive	Document sharing

GitHub	Source control
Blackboard	LMS (Learning Management System)
Office 365	Email client
Git training	https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow
Regular expression cheat sheet	http://web.mit.edu/hackl/www/lab/turkshop/slides/regex-cheatsheet.pdf
Regular express training	https://regexr.com/

SECTION 2: BUSINESS REQUIREMENTS

Gradable Requirements

Requirement #	Requirement
1	Mongoose: Schema • item o text: {type: String}

2	Mongoose: Model
	
	 • Employee
	     o empId: {type: String, unique: true, dropDups: true}
	     o firstname: {type: String}
	     o lastname: {type: String}
	     o todo: [item]
	     o done: [item]

3	Node.js API:
	
	 • Employees
	     o FindEmployeeById: /api/employees/:empId
	     o FindAllTasks: /api/employees/:empId/tasks
	     o CreateTask: /api/employees/:empId/tasks
	     o UpdateTask: /api/employees/:empId/tasks
	     o DeleteTask: /api/employees/:empId/tasks/:taskId

4	SoapUI Unit Tests:
	
	• Employees
	     o FindEmployeeById
	     o FindAllTasks
	     o CreateTask
	     o UpdateTasks
	     o DeleteTask

5	Angular: • Sign in page o Material card containing a single form field (empId) and submit button

6	Angular:
	
	• Tasks management page:
	     o Two columns
	          ■ ToDo (column title)
	          ■ Done (column title)
	     o Create button
	          ■ Navigates users to the "Create new task" form/page
	          ■ New tasks are automatically placed under the ToDo column
	     o Drag-n-drop
	          ■ Users should be able to drag-n-drop the tasks between the different columns
	          ■ https://material.angular.io/cdk/drag-drop/examples
	     o Task card
	          ■ mat-menu (placed in the top-right hand corner of each card) to delete a task

7	Angular:
	
	• About Us
	     o Be creative, original, and showcase your styling skills
	     o Create your own back story (minimum of 250 words and must include images. Don't forget attribution)

8	Angular:
	
	• Custom 404 page
	     o Fully customized 404 page with image and textual content
	     o Be creative, original, and showcase your styling skills

9	Angular:

	• AuthGuard
	     o Users must be logged into the system to use the application
	     o The easiest solution is to install **ngx-cookie-service** and store a cookie in the user's browser during the sign-in process

10	Source control
	
	 • Each student will be expected to use two branches
	     o master
	     o dev
	
	 All development should be done on the dev branch. And, at the end of each Sprint, all code will be merged into the master branch. Grading will be based off the code merged to master.

11	9 Test Cases • Each week you will be required to fill-out three test cases based on the Sprints requirements. Refer to the "Test Case Template" for format/details
	Enhancements (enhancements are not required and should not be attempted before the gradable requirements are completed) • Add a middle column (between the ToDo and Done columns) and name it "Doing" • Provide users with the ability to drag-n-drop tasks between the three columns (ToDo, Doing, Done)

Blackboard Deliverables

Deliverable #	Requirement
1	Technical Design Document (including 6 test cases)
2	Link to your GitHub repository
3	Link to the application running on Heroku
6	Full solution, bundled in a .zip file (make sure you remove the node_modules directory before submitting the final solution)

SECTION 3: APPROVAL

Approvers	Approved (Y/N)	Date
Approved By (Client Acceptors, can be more than one)	Y	11/04/2019
Approved By (Solution Architect)	Y	11/04/2019
Approved By (Development Team)		