Web 450 – nodebucket Project

TECHNICAL DESIGN DOCUMENT

|  |  |
| --- | --- |
| **Author** | Jamal Eddine Damir |
| **Last Updated** | 4/17/2023 3:19 AM |
| **Version Number** | 1.0.0 |

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason for Change | Version |
| Jamal Eddine Damir | 04/04/2023 | Sprint one requirements | 1.0.1 |
| Jamal Eddine Damir | 04/14/2023 | Sprint two requirements | 1.0.2 |
| Jamal Eddine Damir | 04/17/2023 | Spring Three requirements | 1.0.3 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[SECTION 1: INTRODUCTION 3](#_Toc32226123)

[1.1. Purpose 3](#_Toc32226124)

[1.2. Terminology 3](#_Toc32226125)

[1.3. User Personas 3](#_Toc32226126)

[1.4. User Stories 3](#_Toc32226127)

[1.5. Time Estimations 3](#_Toc32226128)

[SECTION 2: PROCESS DESIGN 4](#_Toc32226129)

[2.1. Prototypes 4](#_Toc32226130)

[2.2. ORD (Object Relational Diagram(s) 4](#_Toc32226131)

[2.3. NoSQL Document Diagram 4](#_Toc32226132)

[2.4. NoSQL Data Structure 4](#_Toc32226133)

[SECTION 3: QA TESTING 4](#_Toc32226134)

[3.1. QA Test Plan 4](#_Toc32226135)

[3.2. Unit Tests 4](#_Toc32226136)

[SECTION 4: REFERENCES 5](#_Toc32226137)

# SECTION 1: INTRODUCTION

## Purpose

Nodebucket is a task management app designed to help users efficiently create, delete and organize tasks.

## Terminology

|  |  |
| --- | --- |
| **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/*](https://material.angular.io/) |
| *Flex-layout* | [*https://github.com/angular/flex-layout*](https://github.com/angular/flex-layout) |

## User Personas

# Persona 1: Rebecca Turner

Age: 31

Gender: Female

Job Title: Accountant

Marital Status: Single

Bio: Rebecca is a 31-year-old accountant who specializes in tax preparation and financial analysis. She is detail-oriented and enjoys problem-solving. Her work often involves managing multiple tasks. In her free time, Rebecca enjoys reading books on personal finance and taking long walks in the park with her dog.

# Persona 2: Marge Simpson

Age: 42

Gender: Female

Job Title: Housewife

Marital Status: Married

Short Bio: Marge is a 42-year-old housewife who manages a variety of tasks daily, including meal planning, household chores, and caring for her children. She is highly organized and is a looking for a task management app to stay on top of her responsibilities. Marge enjoys practicing yoga and reading self-help books to improve her time management skills.

# Persona 3: Michael Dickerson

Age: 41

Gender: M

Job Title: Crypto Trader

Marital Status: Divorced

Short Bio: Michael is a 41-year-old crypto-trader. His job involves researching investment opportunities, analyzing market trends, and managing his investment portfolio. He is highly analytical and enjoys taking calculated risks to maximize his returns. In his free time, Michael enjoys reading about new developments in the cryptocurrency industry and attending industry conferences to network with other traders.

## User Stories

|  |  |
| --- | --- |
| **Story** | **Points** |
| As an accountant, I need to see a todo list to prioritize my time | 3 |
| As an accountant, I need to be able to delete tasks if the client cancels | 1 |
| As an accountant, I need to make a list of financial products to buy, so that I can increase my client net worth | 3 |
| As an accountant I need the ability to log in to see my to-do list to keep them secure | 5 |
| As an accountant I need this todo list for my team to stay on track | 3 |

|  |  |
| --- | --- |
| **Story** | **Points** |
| As a house wife, I need to access the application on a mobile device so I can manage my list on the go | 5 |
| As a house wife I need to be able to move my done to-do items to a done column so I can see what I've accomplished | 3 |
| As a house wife, I need a to do list to get chores done around the house, so they are all completed on time | 3 |
| As a housewife, I need my husband to see the updated list so he doesn't forget todo things | 3 |
| As a house wife, I need a todo list so my children can do their chores. | 3 |

|  |  |
| --- | --- |
| **Story** | **Points** |
| As a crypto trader, I need to be able to reorganize my to-do list to I can see what is more important to me. | 5 |
| As a crypto trader, I need to be able to log in so my trading todo list is secure | 5 |
| As a crypto trader I need to see what is done, so I can keep track of my clients | 3 |
| As a crypto trader, I need to bold important items so I don't forget to trade crypto | 1 |
| As a crypto trader, I need to sign-out so my information is secure. | 3 |

## Time Estimations

See Above for points estimation.

# SECTION 2: PROCESS DESIGN

## 2.1. Prototypes

* Link to functional prototype: <https://marvelapp.com/prototype/76a9c45/screen/91027544>
* Screenshots

*Graphical user interface, text

Description automatically generated*

*Graphical user interface, website

Description automatically generated*

*Graphical user interface, text

Description automatically generated*

*Graphical user interface

Description automatically generated*

*Text

Description automatically generated*

*Graphical user interface, text, website

Description automatically generated*

*Graphical user interface, website

Description automatically generated*

*Graphical user interface

Description automatically generated*

## 2.2. ORD (Object Relational Diagram(s)

*Diagram

Description automatically generated*

## 2.3. NoSQL Document Diagram

*[Covert the above ORD into a NoSQL Document Diagram]*

## 2.4. NoSQL Data Structure

*Text

Description automatically generated*

# SECTION 3: QA TESTING

## 3.1. QA Test Plan

|  |  |
| --- | --- |
| **Test Executed by**: Jamal Damir | **Test Execution date:** 04/04/2023 |
| **Description: Nodebucket sprint-1 tests** |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Navigate to:  http://localhost:3000 | User should be land on the login page. The input field should be clear, and the login button should be in a disactivated state. | The login page was displayed, the input field was clear, and the login button was in a disactivated state. |  |  |
| 2 | Enter one of the predefined user IDs | Login button is activated, and its color changed from gray to yellow. | Upon entering a valid ID, the login button was switched to activated and its color changed yellow. |  |  |
| 3 | Click on the login button. | Login should complete successfully, and the user should be redirected to the home page. | Login completed successfully and the user was redirected to the home page. |  |  |

Comments: The above test-case goes through the display and the actions taken on Nodebucket’s login page. The tests above check for the proper display of the login page when run locally on both localhosts 3000 and 4200. The login button and the input filed were tested and the desirable outcome was achieved.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Enter an invalid numerical user ID. | Login button should activate, and its color changed to yellow. | Login button activated and color changed to yellow. |  |  |
| 2 | Click login button | Input field is reset, login button is disactivated, and error message requiring a valid ID is displayed. Login should fail. | Login failed, the input button went back to disactivated state, and the input filed was reset. An error message requiring a valid user ID was displayed and access to the home page wasn’t allowed. |  |  |
| 3 | Enter a non-numerical user ID. | Login button should activate, and its color changed to yellow. | Login button activated and color changed to yellow. |  |  |
| 4 | Click login button | Input field is reset, login button is disactivated, and error message requiring a numerical user ID is displayed. Login should fail. | Login failed, the input button went back to disactivated state, and the input filed was reset. An error message requiring a numerical user ID was displayed and access to the home page wasn’t allowed. |  |  |

Comments: The above steps were used to test the login page’s behavior when either an invalid or a non-numerical user ID is entered in the input field. In both cases, login to the application failed, the input filed was reset and the login button was deactivated. The proper error messages were displayed depending on the action taken.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Leaving the input field empty | Button should stay disactivated. | Login button stayed disactivated. |  |  |
| 2 | Clicking outside the input filed. | “Employee ID is required field” error message should show-up under the input field. | “Employee ID is required field” error message was displayed in the expected area. |  |  |
| 3 | Enter a valid user ID | “Employee ID is required field” error message should disappear, and the login button should change to activated state. | “Employee ID is required field” error message disappeared, and the login button changed to activated state. |  |  |
| 4 | Click login button | Login should complete successfully, and the user is redirected to the home page. | Login was successfully completed, and the user was redirected to the home page |  |  |

Comments: The above steps were used to test the login page’s behavior when the input filed was left empty. The expected outcomes were achieved, and the login page behaved as expected after the user entered a valid ID.

|  |  |
| --- | --- |
| **Test Executed by**: Jamal Damir | **Test Execution date:** 04/14/2023 |
| **Description: Nodebucket sprint-2 tests** |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Navigate to:  http://localhost:3000 | User should be land on the login page. The input field should be clear, and the login button should be in a disactivated state. | The login page was displayed, the input field was clear, and the login button was in a disactivated state. |  |  |
| 2 | Enter one of the predefined user IDs | Login button is activated, and its color changed from gray to yellow. | Upon entering a valid ID, the login button was switched to activated and its color changed yellow. |  |  |
| 3 | Click on the login button. | Login should complete successfully, and the user should be redirected to the home page. | Login completed successfully and the user was redirected to the home page. |  |  |
| 4 | Enter valid task name in the input are | Plus sign (create task button) should activate, and no error messages should be displayed. | Plus sign (create task button) activated, and no error messages were displayed. |  |  |
| 5 | Click plus sign | Task should be added as new tile in the toto column. Success message should display on top of the input area. | Task successfully added as new tile in the toto column. A success message was displayed on top of the input area. |  |  |

Comments: The above steps were used to test task creation part of the application. The expected outcomes were achieved, and the task creation function behaved as expected after the user entered a valid task input.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Navigate to:  http://localhost:3000 | User should be land on the login page. The input field should be clear, and the login button should be in a disactivated state. | The login page was displayed, the input field was clear, and the login button was in a disactivated state. |  |  |
| 2 | Enter one of the predefined user IDs | Login button is activated, and its color changed from gray to yellow. | Upon entering a valid ID, the login button was switched to activated and its color changed yellow. |  |  |
| 3 | Click on the login button. | Login should complete successfully, and the user should be redirected to the home page. | Login completed successfully and the user was redirected to the home page. |  |  |
| 4 | Enter valid task name of less than 3 characters in length | Plus sign (create task button) should stay disabled, and task length error message should be displayed. | Plus sign (create task button) stayed disabled, and task length error message should was displayed. |  |  |
| 5 | Click inside then outside input. | Plus sign (create task button) should stay disabled, and task required error message should be displayed. | Plus sign (create task button) stayed disabled, and task required error message was displayed. |  |  |
| 6 | Enter a task name of more than 35 characters. | Plus sign (create task button) should stay disabled, and task length error message should be displayed. | Plus sign (create task button) stayed disabled, and task length error message should was displayed. |  |  |

Comments: The above steps were used to test the validations for. The task creation part of the application. The expected outcomes were achieved, and the validation errors were displayed as expected after the user entered a invalid task input.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Navigate to:  http://localhost:3000 | User should be land on the login page. The input field should be clear, and the login button should be in a disactivated state. | The login page was displayed, the input field was clear, and the login button was in a disactivated state. |  |  |
| 2 | Enter one of the predefined user IDs | Login button is activated, and its color changed from gray to yellow. | Upon entering a valid ID, the login button was switched to activated and its color changed yellow. |  |  |
| 3 | Click on the login button. | Login should complete successfully, and the user should be redirected to the home page. | Login completed successfully and the user was redirected to the home page. |  |  |
| 4 | Click on the Contact button on the navigation bar. | User should be redirected to the contact us page. | User was redirected to the contact us page. |  |  |

Comments: The above steps were used to test the contact section of the app. The expected outcomes were achieved, and user was redirected to the right page.

|  |  |
| --- | --- |
| **Test Executed by**: Jamal Damir | **Test Execution date:** 04/14/2023 |
| **Description: Nodebucket sprint-2 tests** |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Navigate to:  http://localhost:3000 | User should be land on the login page. The input field should be clear, and the login button should be in a disactivated state. | The login page was displayed, the input field was clear, and the login button was in a disactivated state. |  |  |
| 2 | Enter one of the predefined user IDs | Login button is activated, and its color changed from gray to yellow. | Upon entering a valid ID, the login button was switched to activated and its color changed yellow. |  |  |
| 3 | Click on the login button. | Login should complete successfully, and the user should be redirected to the home page. | Login completed successfully and the user was redirected to the home page. |  |  |
| 4 | Drag task from todo and drop it in done list. | Task should be removed from todo list and added to done list. | Task was removed from todo list and successfully added to done list. |  |  |
| 5 | Drag task from done and drop it in todo list. | Task should be removed from done list and added to todo list. | Task was removed from done list and successfully added to todo list. |  |  |

Comments: The above steps were used to test the drag and drop functions of the nodebucket app. The expected outcomes were achieved, and user was able to drag and drop tasks back and forth between the todo and done columns.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Navigate to:  http://localhost:3000 | User should be land on the login page. The input field should be clear, and the login button should be in a disactivated state. | The login page was displayed, the input field was clear, and the login button was in a disactivated state. |  |  |
| 2 | Enter one of the predefined user IDs | Login button is activated, and its color changed from gray to yellow. | Upon entering a valid ID, the login button was switched to activated and its color changed yellow. |  |  |
| 3 | Click on the login button. | Login should complete successfully, and the user should be redirected to the home page. | Login completed successfully and the user was redirected to the home page. |  |  |
| 4 | Click on task deletion icon. | Pop-up message should be displayed asked user to confirm or cancel action. | Pop-up message was displayed with options to confirm or cancel action. |  |  |
| 5 | Click confirm button | Task should be deleted from the proper column and a success message should be displayed on top of input area. | Task was deleted from the proper column and a success message was displayed on top of input area. |  |  |
| 6 | Click “No thanks” button. | Deletion should be canceled; item remains in its column and a “Deletion cancelled” message should be displayed on top of input area. | Deletion was canceled, item remained in its column and a “Deletion cancelled” message was displayed on top of input area. |  |  |

Comments: The above steps were used to test the delete task function of the nodebucket app. The expected outcomes were achieved, and user was able to delete tasks back from both the todo and done columns. A confirmation dialog was displayed before to confirm or cancel deletion.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Expected** | **Actual Results** | **(Pass/Fail)** | **Notes** |
| 1 | Navigate to:  http://localhost:3000 | User should be land on the login page. The input field should be clear, and the login button should be in a disactivated state. | The login page was displayed, the input field was clear, and the login button was in a disactivated state. |  |  |
| 2 | Enter one of the predefined user IDs | Login button is activated, and its color changed from gray to yellow. | Upon entering a valid ID, the login button was switched to activated and its color changed yellow. |  |  |
| 3 | Click on the login button. | Login should complete successfully, and the user should be redirected to the home page. | Login completed successfully and the user was redirected to the home page. |  |  |
| 4 | Click on the About button on the navigation bar. | User should be redirected to the about us page. | User was redirected to the about us page. |  |  |
| 5 | Navigate to non-existing route | User should be redirected to custom “404 not-found” page. | User was redirected to custom “404 not-found” page. |  |  |

Comments: The above steps were used to test the “About-us” and the “404 not-found” sections of the app. The expected outcomes were achieved, and user was redirected to the right page.

## 3.2. Unit Tests

Test for findEmployeeById API:

Graphical user interface, text, application

Description automatically generated

*Graphical user interface, text, application

Description automatically generated*

Tests for findAllTasks:

Graphical user interface, application, Word

Description automatically generated

findAllTasks wrong employee ID

Graphical user interface, application, Word

Description automatically generated

findAllTasks wrong foo instead of tasks

Graphical user interface, application, Word

Description automatically generated

# 

Test for creatTask:

Graphical user interface, application, Word

Description automatically generated

createTask Validation error:

Graphical user interface, application, Word

Description automatically generated

# 

createTask wrong employee ID:

Graphical user interface, application

Description automatically generated

updateTaks Test Cases:

Success:

Graphical user interface, application

Description automatically generated

Employee not found:

Graphical user interface, application

Description automatically generated

Schema validation:

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

deleteTask:

Success:

Graphical user interface, application

Description automatically generated

Task not found:

Graphical user interface, text, application

Description automatically generated

Employee not found:

Graphical user interface, application

Description automatically generated

# SECTION 4: REFERENCES

*[APA Reference List]*