Bob’s Computer Repair Shop

SPRINT SCHEDULE

Author	Professor Krasso
Last Updated	9/4/2023 9:46 AM
Version Number	1.0.0

Version History

Name	Date	Reason for Change	Version
Krasso, Richard	9/3/2023	New Document	1.0.0

Table of Contents

BCRS: SPRINT 1

Gradable Requirements:

Requirement #	Requirement
1	MongoDB database design • Business rules: ○ a USER can have one ROLE. ○ a USER can have many SECURITY_QUESTION. ○ a SECURITY_QUESTION has an ANSWER. ○ a USER generates many INVOICE ○ an INVOICE has many LINEITEM • Create an ORD based on the above business rules. • Convert the ORD into a NoSQL data structure. • Create a new database named bcrsDB with the appropriate collections. Use CamelCase for the database collection names.
2	User: • email • password • firstName • lastName • phoneNumber • address • isDisabled • role • selectedSecurityQuestions Role: • name Selected Security Questions: • questionText • answerText

	Invoice: • email • fullName • lineItems • partsAmount • laborAmount • lineItemTotal • invoiceTotal • orderDate Line Items: • name • price **Special note**. The design of the database is a team decision, but at minimum, the above fields must be included in the collections. By default, all users are assigned the role: standard.
3	Node.js API: • Users: o findAll: ▪ Verb: GET ▪ Route: localhost:3000/api/users ▪ Status: 200 – OK ▪ Error Handling: • 400 – Bad Request • 404 – Not Found • 500 – Internal Server Error o findById: ▪ Verb: GET ▪ Route: localhost:3000/api/users/:userId ▪ Status: 200 – OK ▪ Error Handling: • 400 – Bad Request • 404 – Not Found • 500 – Internal Server Error o createUser: ▪ Verb: POST ▪ Route: localhost:3000/api/users

	▪ Status: 201 – Created ▪ Error Handling: • 400 – Bad Request • 404 – Not Found o updateUser ▪ Verb: PUT ▪ Route: localhost:3000/api/users/:userId ▪ Status: 204 – No Content ▪ Error Handling: • 400 – Bad Request • 404 – Not Found • 500 – Internal Server Error o deleteUser: ▪ Verb: DELETE ▪ Route: localhost:3000/api/users/:userId ▪ Status: 204 – No Content ▪ Error Handling: • 400 – Bad Request • 404 – Not Found • 500 – Internal Server Error • Security: o signin ▪ Verb: POST ▪ Route: localhot:3000/api/security/signin ▪ Status: 200 – OK ▪ Error Handling: • 400 – Bad Request • 404 – Not Found • 500 – Internal Server Error **Special note**: The deleteUser API does not actually remove a document from the collection. Instead, you are setting a property named “isDisabled” to true. For the signin API use the npm package bcryptjs to compare the hashed password against the passed-in plain text password. Refer back to the solution you built in WEB 420 for assistance on how to verify passwords. Swagger is required for all APIs in this project.
4	SoapUI Unit Tests: • Users: o findAll o findById o createUser o updateUser

	o deleteUser • Security o signin
5	Angular: • Home/landing page o Build a landing page for the BCRS project. This is the main page a user will see when they visit your website. • Sign in page o Build a sign in page for the BCRS project. This page should have input fields for email and password and a submit button. Email and password are required fields. Add data validation for all use cases. o Use the npm package ngx-cookie-service for authentication (just like we did in the Nodebucket project). Save the employees full name, email address, and role to the browser. • Admin: o User configuration page (CRUD) operations: ▪ View a list of users. ▪ Create a new user. ▪ Edit a user record. ▪ Delete a user record. • 404: o Design a 404 page that is consistent with your selected house theme. The 404 page should be fully customized with images and textual content. Be creative, original, and showcase your teams styling skills. • AuthGuard o Users must be signed in to access the website. Use the cookie from the signin process to validate their role. • RoleGuard o Users must have an admin role to view the User Configuration Pages. Use the cookie from the signin process to validate their role.
6	3 functional test cases per team member (use the test case template document). For example, if there are three team members there should be a total of nine functional test cases.

BCRS: SPRINT 2

Gradable Requirements:

Requirement #	**Requirement**
1	Node.js API: • Security: o register ▪ Verb: POST ▪ Route: localhost:3000/api/security/register ▪ Status: 200 – OK ▪ Error Handling: • 400 – Bad Request • 404 – Not Found • 500 – Internal Server Error o verifyUser: ▪ Verb: POST ▪ Route: localhost:3000/api/security/verify/users/:username ▪ Status: 200 - OK ▪ Error Handling: • 400 – Bad Request • 404 – Not Found • 500 – Internal Server Error o verifySecurityQuestions: ▪ Verb: POST ▪ Route: localhost:3000/api/security/verify/users/:username/security-questions ▪ Status: 200 – OK ▪ Error Handling: • 400 – Bad Request • 404 – Not Found • 500 – Internal Server Error o resetPassword: ▪ Verb: POST ▪ Route: localhost:3000/api/security/users/:username/reset-password ▪ Status: 200 – OK ▪ Error Handling: • 400 – Bad Request • 404 – Not Found • 500 – Internal Server Error • Users:

	o findSelectedSecurityQuestions: ▪ Verb: POST ▪ Route: localhost:3000/api/users/:username/security-questions ▪ Status: 200 – OK ▪ Error Handling: • 400 – Bad Request • 404 – Not Found • 500 – Internal Server Error **Special note**. Swagger is required for all APIs in this project.
2	SoapUI Unit Tests: • Security: o register o verifyUser o verifySecurityQuestions o resetPassword • Users: o findSelectedSecurityQuestions
3	Angular: • Register: o Build the account registration page. • Forgot Password: o Build the forgot password process. • Employee Directory: o Design an employee directory page with a list of employees who work for BCRS (minimum of six employees – including Bob). Include images for each employee. Display 3 images per row for large and extra-large view ports. Display 2 images per row for medium and small view ports and display 1 image per row for mobile devices. • FAQ: o Design an FAQ page for BCRS. This page should include an explanation on how to use the website. Group information by topic. For example: ▪ User configuration. Include images and instructions. ▪ Account registration. Include images and instructions. ▪ Reset password. Include images and instructions. ▪ Service repair request. Include images and instructions. ▪ Invoice summary. Include images and instructions. ▪ Purchases by service. Include images and instructions.

4	3 functional test cases per team member (use the test case template document). For example, if there are three team members there should be a total of nine functional test cases.

BCRS: SPRINT 3

Gradable Requirements:

Requirement #	**Requirement**
1	Node.js API: • Invoices: o createInvoice: ▪ Verb: POST ▪ Route: localhost:3000/api/invoices/:username ▪ Status: 200 – OK ▪ Error Handling: • 400 – Bad Request • 404 – Not Found • 500 – Internal Server Error o findPurchasesByService: ▪ Verb: GET ▪ Route: localhost:3000/api/invoices/purchases-graph ▪ Status: 200 – OK ▪ Error Handling: • 400 – Bad Request • 500 – Internal Server Error **Special note**. Swagger is required and must be used to document the API(s).

2	SoapUI Unit Tests:
	• Employees
	    o createInvoice
	    o findPurchasesByService

3	Angular: • Service Repair page: o Design the service repair page. • Invoice Summary page: o Design the invoice summary page. Provide users with an option to print the invoice.

	• Purchases by Service Graph: ○ Design the purchases by service graph page. • My Profile page: ○ Design my profile page. Provide users with the ability to update their address and phone number.
4	3 functional test cases per team member (use the test case template document). For example, if there are three team members there should be a total of nine functional test cases.

BCRS: Group Presentation

Overview:

Requirement #	**Requirement**
N/A	Group presentations are held on days 44 and 45 (Thursday and Friday) of the last week of class. Presentations should last between 30-45 minutes and cover the following topics: • A live end-to-end demo • Individual roadblocks • Lessons learned • What you liked the most about the project • What you liked the least about the project • If you could do it all over again, what would you do differently and why
N/A	Grading criteria: • All or nothing • If you participate in the presentation, you will receive full credit • If you cannot attend class or participate in the presentation you will not receive credit • I do not need an explanation of why you cannot make class/participate