BCS Digital Industries Apprenticeship Synoptic Project

Software Developer – Membership System

Version 1.4
August 2019

Change History

Any changes made to the project shall be clearly documented with a change history log. This shall include the latest version number, date of the amendment and changes made. The purpose is to identify quickly what changes have been made.

Version Number and Date	Changes Made
V1.0 August 2017	Document Created.
V1.1 July 2018	Submission email address amended.
V1.2 November 2018	Declaration template removed. To be supplied in a separate document.
V1.3 July 2019	Clarifications, corrections and improvements based on feedback.
V1.4 August 2019	Removal of virtual platform reference.

Project Overview and Objectives

Your customer, First Catering Ltd, is a catering business that supplies food and beverages to many large-scale businesses. They typically provide a contactless payment service with a card that employees of the businesses can use to register, top up and pay for food and beverages whilst onsite at the business.

First Catering Ltd have been awarded a new contract by Bows Formula One High Performance Cars. The contract specifies that Bows employees shall be able to use their existing employee cards to top up and purchase food at their existing kiosk terminals. First Catering now requires a web service API for these kiosk terminals to access to provide the services. This API will be fully documented and follow industry standards.

This project tasks you with designing and building a RESTful API membership card system.

You will need to:

1. **Design and build a RESTful web service API for the kiosk terminals to interface with.**
2. **Test that the API that you built meets its requirements.**
3. **Ensure that the system conforms to industry standards and best practices for design and implementation.**
4. **Provide appropriate documentation.**

Project Outputs and Deliverables

Once completed, to demonstrate completion of the tasks you will be asked to provide a series of outputs that should be submitted together with the synoptic project declaration.

Deliverable	Output	Evidence
Design	Create documentation to describe what the API will do and how it will work. This is likely to include: • Any assumptions made about the requirements • Any changes or additions made to the requirements • Textual and/or diagrammatic documentation of design elements, including use cases and/or sequence diagrams where appropriate • A data model	Word or PDF documents or similar

Construction	Create application code and associated work products to implement the API. • Your code should be logically structured and follow good coding and architectural practices.	Files containing program code and associated work products
Test	Create and execute a set of tests that demonstrate that the API meets its requirements. • For each test you should document its expected outcome and the actual result.	Use test template (Appendix C) or any other appropriate format
Document	Document the results of your work. • Discuss any limitations of your design and/or implementation. • Propose future improvements. • Create a user guide.	Word or PDF document or similar. A video might be suitable for a user guide.

Project Information and Equipment

To complete this project, you will need to review all the information specified below which can be found in the appendices. This will enable you to deliver the key outputs and deliverables for this project detailed in the table above:

- background information;
- business requirements;
- test template.

In addition, your training provider and / or employer will provide you with all the resources required to complete your project including:

- computer equipment with access to the Internet;
- an appropriate software development environment;
- suitable document preparation software.

Competencies and Knowledge Standards

Below is a list of competencies and knowledge standards covered by this project.

Knowledge Standards

- Understands and operates at all stages of the software development lifecycle.
- Understands and applies software design approaches and patterns and can interpret and implement a given design, compliant with security and maintainability requirements.
- Understands and applies the maths required to be a software developer (e.g. algorithms, logic and data structures).

Competency Standards

- Logic: writes good quality code (logic) with sound syntax in at least one language.
- Data: can effectively link code to the database / data sets.
- Test: can test code and analyse results to correct errors found using either V-model manual testing and / or using unit testing.
- Design: can create simple data models and software designs to effectively communicate understanding of the program, following best practices and standards.

Appendix A – Background Information

The following additional information has been provided to support you with the completion of your project.

Background information

- The kiosks contain a Touch Sensitive PC and a smoked black glass enclosure with a card reader behind the glass.
- Topping up with credit enables employees to purchase goods through the kiosk machines.
- The kiosks and the handling of money are all being looked after by the existing IT department at First Catering Ltd.
- The data cards hold a Card Id consisting of a unique sequence of 16 alphanumeric characters, e.g. **r7jTG7dqBy5wGO4L**.

Appendix B - Business Requirements

Business Requirements

- Your task is to create a RESTful web service for First Catering Ltd that will allow Bows Formula One employees to use their existing data cards in the kiosks to register and top up with money.
- If the card is not registered on the system, the card owner will be required to provide basic employee information:
    - unique employee ID;
    - name;
    - email,
    - mobile number.
- A four-digit pin number chosen by the employee should be used for further security.
- The data used for providing the services must be held in a database.
- When a card is presented to the system and the service finds that the card is already registered, the system will show a welcome message with the user's name associated with the card.
- If the user's card is not registered, then the system will respond requesting that the card needs to be registered.
- It is envisaged that when the user taps their card a second time the system informs the user and says "Goodbye".
- The application should timeout after a number of minutes of inactivity.
- You only need to provide the REST API, which should conform to industry standards.

Your documentation should include a discussion of how the endpoints of your API are intended to be used by the kiosk clients to satisfy these requirements.

In addition to the functional requirements you should consider non-functional requirements. Even if the time constraints of the project do not allow for you to address these in the code, you need to add some notes to explain the relevance and how you would improve the design and code to address these non-functional requirements.

Appendix C - Test Template

Test Number	Test Description	Test Tools	Expected Outcome	Results of Test	Pass / Fail

On completion, please upload documentation relating to the project deliverables and a completed project declaration (provided separately) to the relevant folder location as specified by your training provider. Alternatively, please send to epateam@bcs.uk