

# TODO

## In Progress

### [API-0012] Edit Readme with updated information

No Description

### [API-0011] Add CI in Github actions

The branch <https://github.com/AstridCaballero/MembershipAPI/tree/actions/API-0011> uses Github actions through 'maven.yml' file to check the build is successful.

However, The idea was to check the tests as well for any future branches to pass them but It can't run the tests. The problem might be related to @quarkus test. If the tests don't have that annotation the tests will be found by surefire.

For this reason the branch is left and not merge to the main branch.

## Done

### [API-0010] Implement inactivity time out

The idle timeout of the system will ideally be managed by an UI but as I don't have that, I am creating a GET endpoint that will update a front end (Assumption).

**As** a system **I want** to log out user after 3 minutes of inactivity **So** Another user can login

**Acceptance criteria (1) Feature** Formula One employee is inactive for 3 minutes then it gets logout by the system

**(1.1) Scenario** Formula One employee has not click or perform any action on the system for 3 minutes then it gets logout

**Given** I am Login **When** I don't take any action during 3 minutes **Then** I expect to be logged out

### [API-0009] Implement Log out

**As** a Formula One employee **I want** to exit the system **So** I can go

**Acceptance criteria (1) Feature** Formula One employee clicks on Exit button and system asks for the card to be tapped.

**(1.1) Scenario** Formula One employee clicks on Exit button **Given** I am in the Welcome screen **When** I click on the Exit button and I tap the card to the sensitive PC **Then** I expect to leave the system **and** I expect to see a Goodbye Message.

### [API-0008] Add integration test for the endpoints implemented for the Order workflow

No Description

### [API-0007] Improve documentation

No Description

## [API-0006] Implement Get, Post, Put and Delete for Order workflow

**As** a Formula One employee **I want** to Make an order **So** that I can make a purchase

**Acceptance criteria (1) Feature** Formula One employee clicks on Order button and products are displayed.

**(1.1) Scenario** Formula One employee clicks on Order button and products are displayed **Given** I am in the Welcome screen **When** I click on the Order button **Then** I expect to see products displayed **and** I expect an order to be created **and** I expect the current total to pay, to be displayed

**(2) Feature** Formula One employee selects on a product and the current total updates

**(2.1) Scenario** Formula One employee selects by clicking on a product and current total to pay updates accordingly **Given** I can see the products displayed **When** I click on a product to select it **Then** I expect an orderProducts to be created **and** I expect the current total to be updated and displayed

**(2.2) Scenario** Formula One employee clicks on a plus button of a selected product and current total to pay updates accordingly **Given** I can see the products displayed **and** that there is a selected product **When** I click on a plus button of a selected product **Then** I expect an orderProducts to be updated (quantity and Price) **and** I expect the current total to be updated and displayed

**(2.3) Scenario** Formula One employee deselects a selected product and current total to pay updates accordingly **Given** I can see the products displayed **and** that there is a selected product **When** I click on a product to deselect it **Then** I expect an orderProducts to be removed **and** I expect the current total to be updated and displayed

**(3) Feature** Formula One employee clicks on payment to pay for the order

**(3.1) Scenario** Formula One employee clicks on payment and is successful **Given** I can see the payment button **and** I have a card's balance bigger or equal to the total to pay **When** I click on it **Then** I expect to see a 'Payment successful' message **and** I expect the card's balance to be updated

**(3.1) Scenario** Formula One employee clicks on payment and is unsuccessful **Given** I can see the payment button **and** I have a card's balance less to the total to pay **When** I click on it **Then** I expect to see a 'Insufficient funds' message

## Implement DB

Follow the repository pattern: -Entity class -EntityRepository -EntityService

## [API-0005] Implement integration testing WelcomeScreenResource

No Description

## [API-0004] Implement integration testing LoginResource

No Description

## [API-0003] Implement Post endpoint card TopUp

**As** a Formula One employee **I want** to top up card **So** that I can purchase

**Acceptance criteria (1) Feature** Formula One employee selects topUp option from welcome screen

**(1.1) Scenario** Formula One employee selects topUp option from welcome screen

**Given** I have logged in **and** I can see card balance **When** I select TopUp option **Then** I expect to be asked for amount to topUp **And** I expect to see a submission button that I can click **And** I expect to go back to welcome screen and see balance updated

## Update DB

Update PK for 'employee' table

## [API-0001] [API-0002]Implement Get and Post for Login

**As** a Formula One employee **I want** to Login into terminal Kiosk **So** that I can access the system

**Acceptance criteria (1) Feature** Formula One employee touches registered card on the Kiosk Touch sensitive PC and a four digit pin is prompted by the screen

**(1.1) Scenario** Formula One employee touches card on the Kiosk Touch sensitive PC and card is not registered **Given** I have a card **When** I touch the Kiosk Touch sensitive PC **Then** I expect to be asked for employee information

**(1.2) Scenario** Formula One employee touches card on the Kiosk Touch sensitive PC and card is registered **Given** I have a card **When** I touch the Kiosk Touch sensitive PC **Then** I expect to be asked for a four-digit pin number

## [API-0002] Implement Post endpoint for Login/registration

**As** a Formula One employee **I want** to Login into terminal Kiosk **So** that I can access the system

**Acceptance criteria (1) Feature** Formula One employee touches registered card on the Kiosk Touch sensitive PC and a four digit pin is prompted by the screen

**(1.1) Scenario** Formula One employee touches card on the Kiosk Touch sensitive PC and card is not registered **Given** I have a card **When** I touch the Kiosk Touch sensitive PC **Then** I expect to be asked for employee information

## [API-0001] Implement Get endpoint for Login

**As** a Formula One employee **I want** to Login into terminal Kiosk **So** that I can make a purchase

**Acceptance criteria (1) Feature** Formula One employee touches registered card on the Kiosk Touch sensitive PC and a four digit pin is prompted by the screen

**(1.2) Scenario** Formula One employee touches card on the Kiosk Touch sensitive PC and card is registered **Given** I have a card **When** I touch the Kiosk Touch sensitive PC **Then** I expect to be asked for a four-digit pin number

### COMMENTS

Astrid Caballero 2022-11-01

accessing H2 console: [Screenshot 2022-11-01 at 14.40.08.png](#)

## Researched about DB options

Which DB should I use for easy sharing?

### COMMENTS

Astrid Caballero 2022-11-01

Use H2 in-memory DB so I can initialise a script for the DB each time the application starts. **So H2 is good for prototyping.**

## Design DB

No Description

## Design System workflow

No Description

## Create Github repository

No Description

## Define commit message standard

following suggestions from: <https://cbea.ms/git-commit/>

## Create bones of Java API

**Java project uses** JDK 17.0.1 Maven 3.8.1 Quarkus 2.13.0.final

**Testing:** Jacoco quarkus-junit5 2.13.0.final rest-assured 5.2.0

**Documentation:** Swagger from quarkus-smallrye-openapi 2.13.0.final

# Better if

## Add validation of EmployeeID with FormulaOne Database

Add validation of EmployeeID with FormulaOne Database before persisting the data in line 63 (see screenshot)

## To validate data entered by user

No Description

## To hash the four-digit pin

No Description

## Github actions CI/CD pipeline

No Description

in tests to reduce the length of  
`.post("http://localhost:8100/api/r7jTG7dqBy5wGO4L/welcome/topUp")`

change: `.post("http://localhost:8100/api/r7jTG7dqBy5wGO4L/welcome/topUp")`

to: `.post("/r7jTG7dqBy5wGO4L/welcome/topUp")`

make swagger to display the right JSON format to fill

I have to delete the id every time I run this in swagger. See **attachment**

Solution: I added a description with an input example to be copied and used instead of using the default swagger json

# Blockers

## Lazy Initialization - hibernate

@get to consult a card and its related employee resulted in "LazyInitializationException". Problem explained: <https://vladmihalcea.com/the-best-way-to-handle-the-lazyinitializationexception/>

I used FetchMode.Join, got rid of fetch = FetchMode.LAZY and it works  
<https://www.baeldung.com/hibernate-fetchmode>

## MultipleBagFetchException due to using Join.Fetch and Lists

Quick fix is to use 'set' instead of 'List'.

However, the best solution is here: <https://stackoverflow.com/a/24676806>

here is another explanation of the issue and what bags are: <https://blog.eyallupu.com/2010/06/hibernate-exception-simultaneously.html>

## Mediatype - Swagger, could create 406 errors

mediaType = "application/text" ----> error 406 Not acceptable solution: mediaType = "text/plain"