BURGER

Software Engineering Task

The Burgerama Burger Shop requires a simple Console Application to manage the sale and supply of their famous Burgers to any costumer in the hood. To gather all the requirements for this Application, interviews with different employees of the Burgerama Burger Shop were conducted. In the following, a summary of these interviews is provided.

The Burgerama Console Application should provide a main menu that allows any customer to either create a new account or to log into an existing account. For this selection, the main menu should provide two different options that can be selected like:

```
1) Register at Burgerama
2) Login

Please select you option…
```

The main menu should only accept valid options and should return an error message for any invalid option. If an invalid option was selected, the user should be allowed to choose another option.

For the registration the fields email, address and password are required. The login itself only requires the email and password of the user to authenticate. If an unregistered email address was entered, an error message should be shown to the user and the App should return to the main menu.

The customer management wants their customer data to be saved as an XML file on the disk to persist the user information for possible reuse in future. The password should be stored as a hash value to provide some basic security. The customer data is retrieved from the XML file during the Application startup and should be then available during runtime for customers to log in again.

The Burgerama Burger Shop sells different types of products that differ in price and delivery time. Besides their famous food they provide soft drinks and merchandise like shirts and snapbacks. In the following these products are listed:

Products				
Unique ID	**Category**	**Name**	**Price**	**Preparation Time**
130	Food	Cheeseburger	7.99 $	5 Minutes
89	Food	Classic Burger	9.90 $	12 Minutes
223	Drink	Coca Cola	2.99 $	2 Minutes
33	Drink	Sprite	2.99 $	2 Minutes
23	Food	Burgerama Special	16.99 $	16 Minutes
666	Food	Red Devil	12.99 $	14 Minutes
43	Drink	Red Bull	3.90 $	1 Minute
91	Merchandise (Clothing)	T Shift Burgerama	12.89 $	immediate
4	Merchandise (One Size)	Snapback	7.90 $	immediate
273	Merchandise (Clothing)	Hoodie	29.80 $	immediate
54	Food	French Fries	3.99 $	5 Minutes

387	Merchandise (One Size)	Stickers	0.99 $	immediate

The product manager wants to offer each product to customers after the login. The Burgerama Console App should therefore show an Order Menu after a customer logged in successfully. The Order Menu should provide different product options to the customer. The customer can select a product and a related amount.

The list of products should be loaded from a JSON file during Application startup, so that the sales management can adjust their offer to any time.

The product management stated some additional requirements for the ordering process:

- Each product ID has to be unique. The app should indicate an error if two products have the same ID.
- Products of category 'Drink' can be ordered with an option "on ice" that can be enabled or disabled by the customer after a Drink was selected. An exception to this option is the Red Bull which is supplied as a can.
- There are two types of merchandise. Clothing can be ordered in various sizes (S, M, L, XL, XXL). One Size merchandise – as its name suggests – does not provide this option.
- Merchandise has no additional preparation time as they can be added to the packet directly from the local shop's storage.

The customer can select a "Place Order" option in the Order Menu to finish the order. The App should print a summary of the order to the console screen. The summary should list different receipt entries. Each entry contains the product' name, the ordered amount, and the subtotal. The total price of the order is the sum of all subtotals. In addition, the US tax must be added on top of the total price. The tax in the US state California is 8.84 %. The tax should be also listed as a position on the receipt.

As a special service to customers, the App should also inform about estimated delivery prices. The Burgerama staff has experienced that the total preparation time for any order is equal to the longest preparation time of a sub item of the order. For example, if a customer orders a 'Burgerama Special' and 'French Fries', the longest preparation time of a sub item is 16 Minutes. So, the total preparation time of the order will also be 16 Minutes. In addition to the preparation time, the App should also consider the time that a driver will need to ship the order to the customer. The App will estimate this time at approximately 20 Minutes. The total delivery time should be shown as part of the receipt printed to the screen after an order was placed.

The Burgerama Shop has currently employed 3 drivers to ship orders to customers as fast as possible. As each driver uses different vehicles to ship orders, they can just take a maximum capacity of orders at once. If the maximum number of orders is exceeded, they have to do multiple delivery trips. As this additional time, the shipment time will increase by additional 15 Minutes. Below the drivers and their maximum capacities are listed.

Drivers	
Name	**Capacity**
Joe Pegg	3
Lara Croft	6
Nishal Bondeli	2

As the Burgerama Management is keen to hire additional drivers in the future, the driver configuration should be stored in a separate JSON file in order to adjust it

As soon as a new delivery was placed by a customer, the related shipment of this order should be assigned to the next driver that has some free capacity. If all driver capacities are exhausted, the order is assigned to the driver that has the least number of open orders.

To better manage their staff, the Burgerama foremen would like to have an overview of all open and closed orders. This overview should be updated for any new order that was received. The overview should contain the following information:

- Email of the user that placed the order
- Address of the user that placed the order
- Summary of the order that contains the ID of a product and the ordered amount
- Name of the driver that delivers the order
- Remaining preparation time in minutes
- Remaining shipment time in minutes
- State of the order. State is on of {Preparation, Delivery or Closed}

To process placed orders, the App needs another option for the Management to simulate time cycles. If this option is triggered from the main menu of the App, a cycle of 15 minutes is simulated. The simulation should affect the state and remaining time of all open orders.

Any new order is assigned a 'Preparation' state. As soon as a simulation cycle was triggered, the remaining preparation time should be reduced by 15 minutes. If the preparation time is less than 15 minutes, the remaining time difference should be used to reduce the shipment time. If the remaining preparation time has expired, the state of the order should be changed to `Delivery`. If another cycle is simulated, the remaining shipment time should be reduced by another 15 minutes. If the remaining delivery time has expired, the state of the order should be changed to `Closed`. The free capacity of a driver should be reduced by the number of closed procedures on his work list.

Example of how time cycle affects remaining times				
	Initial	Cycle 1	Cycle 2	Cycle 3
Remaining Preparation Time	12 Minutes	0 Minutes	0 Minutes	0 Minutes
Remaining Shipment Time	10 Minutes	7 Minutes	0 Minutes	0 Minutes
Total Delivery Time	22 Minutes	7 Minutes	0 Minutes	0 Minutes
State	Preparation	Delivery	Closed	Closed

The overview of open and closed orders for the Burgerama foremen should be updated for any new order that was placed or whenever a time cycle was simulated. It should be printed to an output file.

In order to stick with their quality standards, the Burgerama board members have also agreed that the App has to provide Unit Tests for each new feature that is implemented.

Have fun! Cheers!

Where to start from here?

- Install the latest .NET SDK on your computer
- Install Visual Studio / Rider
- Create a .NET console application and unit test project (XUnit Framework)
- Set up Git
- Focus on clean code & object-oriented programming!

Your Tasks

1. Create a Story Map for the Burgerama Console Application
2. Plan your releases on a weekly basis. Every release should also provide the required Unit Tests.
3. Commit yourself on features for a minimum viable product (MVP) at the end of your term.
4. Implement the MVP
5. The goal is not to implement all the requirements but to plan your approach your task in a systematical way and to learn some basic principles of Software Engineering as well as object-oriented programming.
6. Most important: Have fun while implementing!