Marios Pizzabar Projekt

Gruppe 7

Github Repository: https://github.com/JWillumsen89/MariosPizza

Martin Hansen - https://github.com/Cannabis2013

Morten Lundorff Kristensen - https://github.com/morten-l-k

Christian Rodrigo Perea Petersen - https://github.com/RodrigoPerea

Jonathan Willumsen - https://github.com/JWillumsen89

Domænemodel:
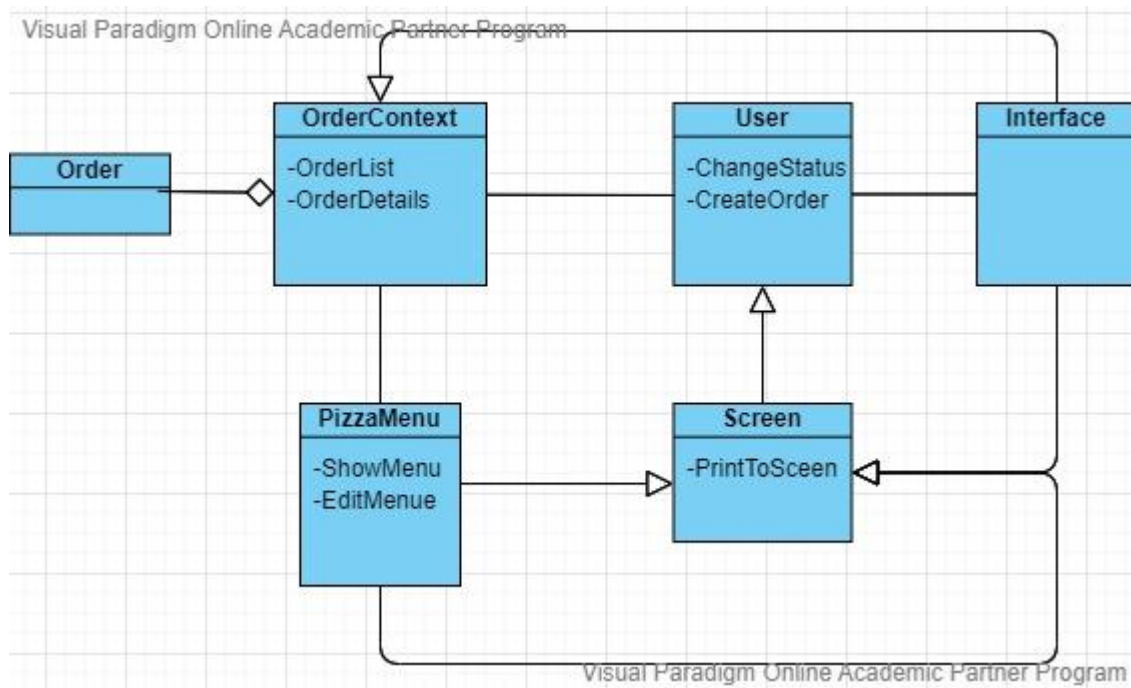
User stories:

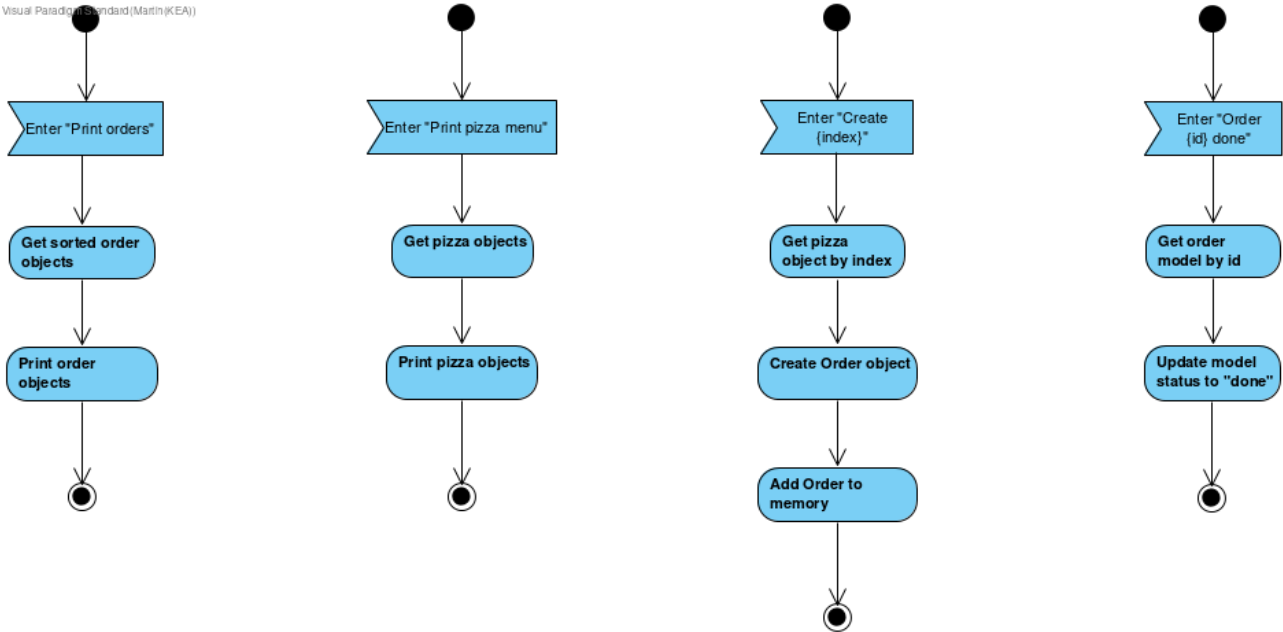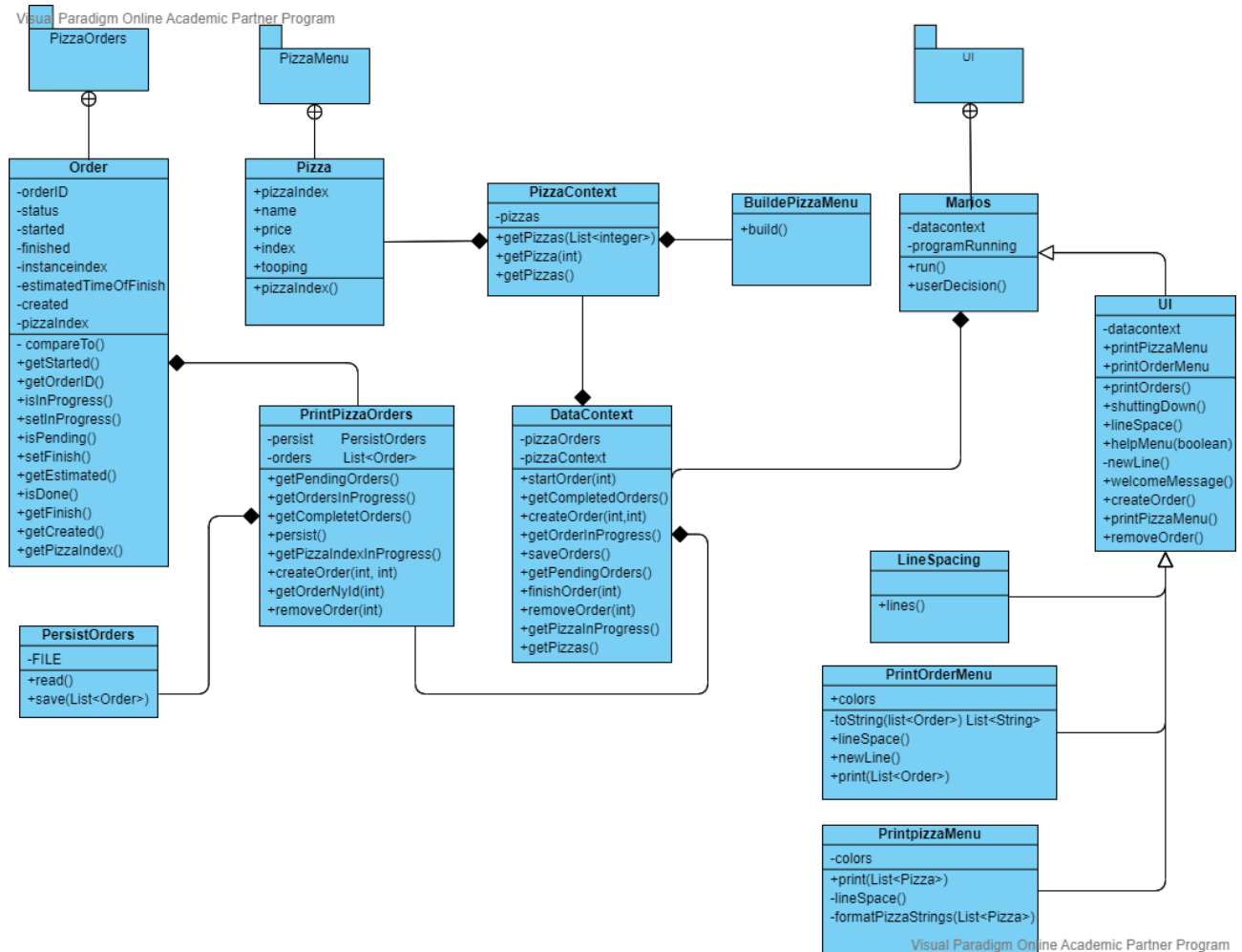| User story	issue
As a user, I should be able to see the full order list with pic up time, alle the time.	Should the order list appear in one side of the scene and be updated every time there's a new command, or should it just appear in the top when scene is changing?
As a user, I should be able to change my food menu.	
As a user, I should be able to create a new order, from the food menu.	
As a user, I should be able to set a pickup time to an order.	
As a user, I should be able to create an order without set a pickup time.	
As a user, I should be able to see my full food menu with a single command.	Should the menu appear in the bottom of the scene all the time?
As a user, I should be able to sort the order list, by pic up time.	Should the order list be sorted by newest (shorts time) in the top? And if there's no time, it will be placed in the top of the list?
As a user, I should be able to change status on an order.	Should the status change from in progress to done? Or just paced?
As a user, I should be able to get a full print of my turnover.	This will only work, if you can cancel an order. Should there be a price attached to every pizza in the menu?
As a user I should be able to see the user interface in the bottom scene all the time	
As a user, I should be able to cancel an order.	Should this just change to done, or should it be a cancellation so it will appear in statistics?
As a user, I should be able to update or add orders to the order menu.	

Aktivitets diagram:

Flow 1:
- Enter "Print orders"
- Get sorted order objects
- Print order objects
- (end)

Flow 2:
- Enter "Print pizza menu"
- Get pizza objects
- Print pizza objects
- (end)

Flow 3:
- Enter "Create {index}"
- Get pizza object by index
- Create Order object
- Add Order to memory
- (end)

Flow 4:
- Enter "Order {id} done"
- Get order model by id
- Update model status to "done"
- (end)

Klasse diagram:

PizzaOrders

PizzaMenu

UI

Order
- -orderID
- -status
- -started
- -finished
- -instanceindex
- -estimatedTimeOfFinish
- -created
- -pizzaIndex
- - compareTo()
- +getStarted()
- +getOrderID()
- +isInProgress()
- +setInProgress()
- +isPending()
- +setFinish()
- +getEstimated()
- +isDone()
- +getFinish()
- +getCreated()
- +getPizzaIndex()

Pizza
- +pizzaIndex
- +name
- +price
- +index
- +tooping
- +pizzaIndex()

PizzaContext
- -pizzas
- +getPizzas(List<integer>)
- +getPizza(int)
- +getPizzas()

BuildePizzaMenu
- +build()

Marios
- -datacontext
- -programRunning
- +run()
- +userDecision()

UI
- -datacontext
- +printPizzaMenu
- +printOrderMenu
- +printOrders()
- +shuttingDown()
- +lineSpace()
- +helpMenu(boolean)
- -newLine()
- +welcomeMessage()
- +createOrder()
- +printPizzaMenu()
- +removeOrder()

PrintPizzaOrders
- -persist   PersistOrders
- -orders   List<Order>
- +getPendingOrders()
- +getOrdersInProgress()
- +getCompletetOrders()
- +persist()
- +getPizzaIndexInProgress()
- +createOrder(int, int)
- +getOrderNyId(int)
- +removeOrder(int)

DataContext
- -pizzaOrders
- -pizzaContext
- +startOrder(int)
- +getCompletedOrders()
- +createOrder(int,int)
- +getOrderInProgress()
- +saveOrders()
- +getPendingOrders()
- +finishOrder(int)
- +removeOrder(int)
- +getPizzaInProgress()
- +getPizzas()

LineSpacing
- +lines()

PersistOrders
- -FILE
- +read()
- +save(List<Order>)

PrintOrderMenu
- +colors
- -toString(list<Order>) List<String>
- +lineSpace()
- +newLine()
- +print(List<Order>)

PrintpizzaMenu
- -colors
- +print(List<Pizza>)
- -lineSpace()
- -formatPizzaStrings(List<Pizza>)

Sekvens diagram:

Create order:

DataContext   PizzaOrders

1: createOrder(int,int)

1.1: createOrder(int,int)

Shut down:

DataContext   PizzaOrders   PersistOrders

1: saveOrders() : void

1.1: persist() : void

1.1.1: persist(List<Order>) : void
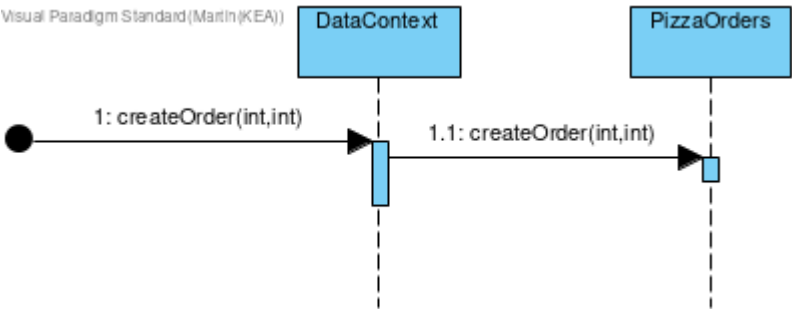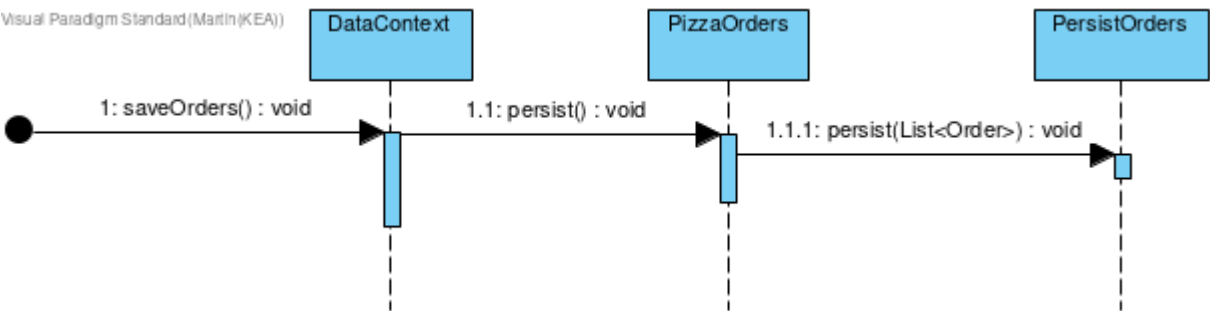
Interessent analyse:

Involverede \ Faser	Skal informeres	Skal konsulteres	Skal indgå	Har kontrol
Identifikation				Team
Planlægning		Alfonso	Mario	Team
Gennemførelse		Mario		Team
Afslutning	Alfonso Mario			Team