Bovelo project: Glossary and Lexicon

# User

* Needs to log in with a password to go further in the app
* Admin rights are given to special logins (for ex: Admin01, Admin02, …)
* Representative: the client, can see the catalog, configure a bike, add to cart and pass orders.   
  Methods:
  + setOrderBike() : creates a new order of bikes, from the cart. If cart is empty, does nothing.
  + addToCart(Bike,int) : adds the chosen bike to cart, with its quantity
  + getTimeBeforeShipping(): used for estimating when the bikes order will be ready to ship
  + getBikeInvoice(): export the order in txt format
* Admin: able to see its planning, manage orders (of the bike parts), see the catalog. It’s the “builder”  
  Methods:
  + getPlanning():returns the planning of the week
  + getBikePartInvoice(): export the order in txt format
  + validateBikePartOrder(): the missing parts are automatically ordered every week, an admin can check the orders and validate before payment.
* (The user can receive an invoice for its orders (export in text format for ex))
* Cart : Dictionary of bikes with the quantity of each bike

# Catalog

* City Bike: [basic frame; city wheels; city mudguard; luggage rack; light]
* Explorer Bike: (VTC) [basic frame; off-road wheels; explorer mudguard; luggage rack; light]
* Adventure Bike: (VTT) [reinforced frame; off-road wheels]
* Frame changes color and size
* Wheel changes size
* Methods:
  + (getDetails(): returns the details of each bike (price, size, options, etc))

# Bike

* Wheel : cityWheel or offRoadWheel
* Frame : basicFrame or reinforcedFrame
* Light
* Mudguard : cityMudguard or explorerMudguard
* LuggageRack
* totalTime: time to assemble the whole bike

# BikePart

* timeToAssemble: time to assemble the part to the bike
* location and quantity: where the part is stored and how many
* price: price of each part
* provider:

# OrderBike

Class that creates orders of bikes. A user may have multiple OrderBike’s. An OrderBike has at least 1 Bike in it.

* TimebeforeShipping is the estimated time at which the order should be sent
* isReadyToShip : if all the bikes are assembled, then it’s ready to ship
* orderId : the id of the order

# StockManagement

A class that takes all the OrderBike’s and will check the stock and order BikeParts if needed. Every week it will check the stock.

* Cart is a list of bikePart’s to pass orders
* manageStock() will get the stock of BikePart, the number of BikeParts for the week and set an order if out of stock

# IUpdatableComponent

An interface that gets a method to update each week.

# Planning

A class that takes all the OrderBike’s every week to create a planning, depending on the number of isAdmin (assembler or builder)

* planning: a list with three columns (one for each builder) with a list of all the bikes to assemble for the week
* weekId : the id of the planning (for ex : week1)
* bikeParts : a dictionary of all the bikePart needed for the week

# OrderBikePart

Class that creates orders of bikePart, it’s the stock management that can order (or “addToCart()”?) and the admins will verify and pass the order.