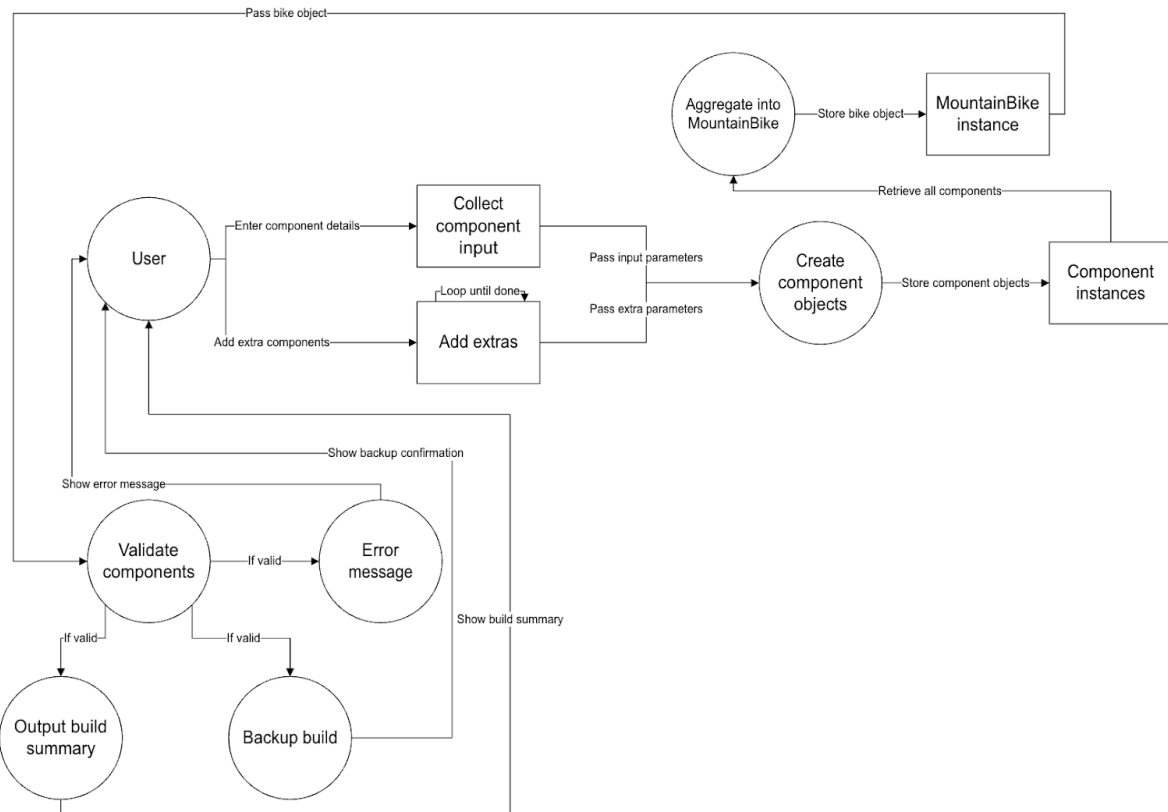


Designing:

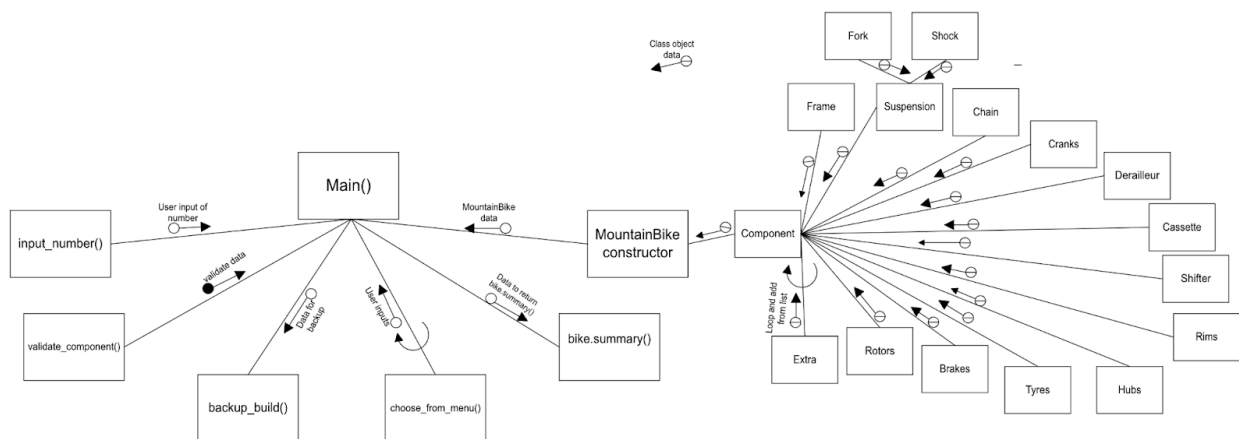
Data flow diagram:

DFD of the full code (couldn't do curved arrows)



Structure chart:

Structure chart of the code



Data dictionary:

Variable Name	Data Type	Description	Example	Validation
name	string	Component type name	Fork	Non-empty string
brand	string	Brand of component	Fox	Selected or user input
price	float	Component price	799.99	Positive number
material	string	Frame material	Carbon	Predefined list or user input
rear_travel	integer	Frame rear travel (mm)	150	Positive integer
travel_mm	integer	Fork travel (mm)	160	Positive integer
stanchion_width	integer	Fork stanchion width (mm)	36	Positive integer
shock_type	string	Shock type	Coil	Predefined list or user input
front_size	string	Front wheel size	29"	Predefined list or user input
rear_size	string	Rear wheel size	27.5"	Predefined list or user input
frame	object of class Frame	Frame object	-	Created from input
fork	object of class Fork	Fork object	-	Created from input
shock	object of class Shock	Shock object	-	Created from input
chain	object of class Chain	Chain object	-	Created from input
cranks	object of class Cranks	Cranks object	-	Created from input
derailleur	object of class Derailleur	Derailleur object	-	Created from input
cassette	object of class Cassette	Cassette object	-	Created from input

shifter	object of class Shifter	Shifter object	-	Created from input
rims	object of class Rims	Rims object with sizes	-	Created from input
hubs	object of class Hubs	Hubs object	-	Created from input
tyres	object of class Tyres	Tyres object	-	Created from input
brakes	object of class Brakes	Brakes object	-	Created from input
rotors	object of class Rotors	Rotors object	-	Created from input
extras	list[Extra]	List of extra components	['Stem']	Entered until 'done'
value	integer/float	Temp numeric input	299.99	Valid number or re-prompt
choice	integer	Menu selection index	2	Valid menu index
title	string	Menu title string	Choose fork brand	Non-empty string
options	list[string]	List of menu options	['Fox', 'RockShox']	List of strings

Pseudocode:

Pseudocode for the total_price() method

None ▼

```
BEGIN total_price
```

```
    SET drivetrain_total TO 0
```

```
    SET wheelset_total TO 0
```

```
    SET other_total TO 0
```

```
    FOR EACH component IN [chain, cranks, derailleur, cassette,
shifter]
        ADD component.price TO drivetrain_total
    NEXT component

    FOR EACH component IN [rims, hubs, tyres]
        ADD component.price TO wheelset_total
    NEXT component

    FOR EACH component IN [frame, fork, shock, brakes, rotors]
        ADD component.price TO other_total
    NEXT component

    FOR EACH extra IN extras
        ADD extra.price TO other_total
    NEXT extra

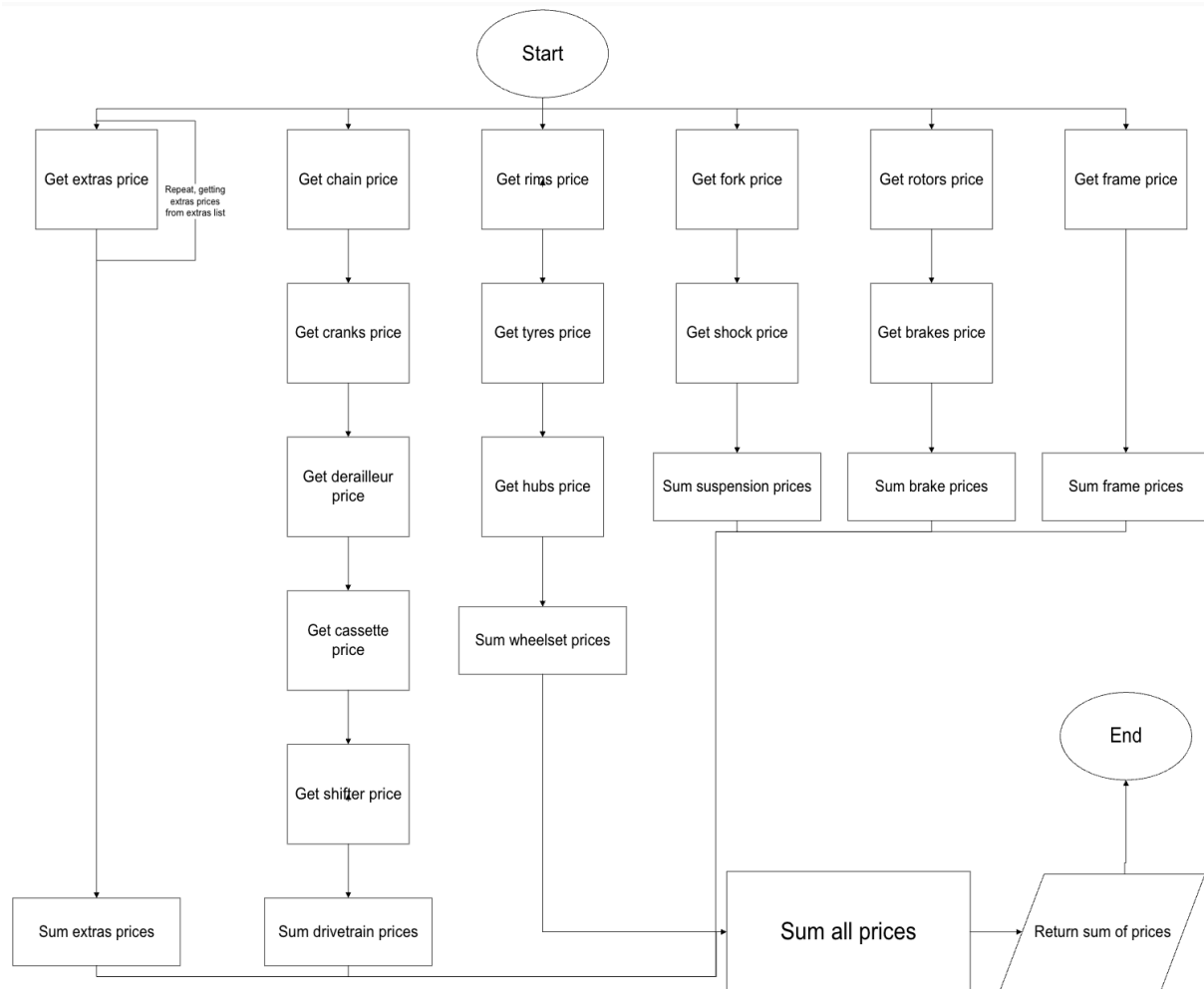
    RETURN drivetrain_total + wheelset_total + other_total

END total_price
```

Flowchart:

Flowchart for the total_price() method

Every “get ____ price” is meant to be the same shape as the “Return sum of prices” instead of a rectangle.



Desk check:

Desk check table of all major function or methods

Section	Step	Variable / Input	Value	Description / Action	Output / Result
total_price()	1	drivetrain_total	50+120+110+90+80 = 450.0	Sum of 5 drivetrain component prices	450.0
	2	wheelset_total	300+250+200 = 750.0	Sum of rims, hubs, tyres	750.0
	3	other_total (base)	1000+750+450 +350+100 =	Frame, fork, shock,	2650.0

			2650.0	brakes, rotors	
	4	extras_total	70.0 + 30.0 = 100.0	Sum of all extras in list	100.0
	5	final total_price	450.0 + 750.0 + 2750.0 = 3950.0	Sum of all component group totals	3950.0
input_number() (allow_float = False)	1	input	"abc"	Invalid input - triggers retry loop	-
	2	input	"300"	Converted to int(300)	300
input_number() (allow_float = True)	1	input	"299.99"	Converted to float(299.99)	299.99
choose_from_menu()	1	title	"Choose shock brand"	Menu displayed with numbered options	-
	2	input	2	Chooses options[1] = "Fox"	"Fox"
	3	input	4	Triggers custom input	-
	4	Custom input	"Öhlins"	Accepted as custom string	"Öhlins"
Fork Constructor	1	brand	"Fox"	Passed to super().__init__()	Stored in self.brand
	2	price	999.99	Converted to float, stored	Stored in self.price
	3	travel_mm	170	Converted to int, stored	self.travel_mm = 170
	4	stanchion_width	38	Converted to int, stored	self.stanchion_width = 38
validate_component()	1	Any component (e.g. Fork)	(any object)	Stub always returns True	True

Test data:

Test Data Table – Mountain Bike Builder Program

choose_from_menu(title, options)

Test Case	Input	Expected Output	Reason for Inclusion
TC1	2 (menu = ["Fox", "RockShox", "Öhlins"])	"RockShox"	Normal case – valid selection from list
TC2	4 then "Cane Creek"	"Cane Creek"	Custom input – test "Other" path
TC3	"abc"	Re-prompt	Invalid string input
TC4	0	Re-prompt	Out-of-range (too low)
TC5	99	Re-prompt	Out-of-range (too high)

input_number(prompt, allow_float=False)

Test Case	Input	Expected Output	Reason for Inclusion
TC1	"200"	200 (int)	Normal integer input
TC2	"299.99"	Re-prompt	Float when only int allowed
TC3	"abc"	Re-prompt	Invalid string input
TC4	"149.99"	149.99 (float)	Valid float with allow_float=True
TC5	"-50"	-50.0	Negative number (edge case)

total_price() Method

Test Case	Input Prices (key components)	Expected Output	Reason for Inclusion
TC1	Frame:1000, Fork:600, etc.	~3500.0	Normal total price calculation
TC2	All components: 0	0.0	Test zero total

TC3	shock = None	Crash/Error	Handling of NoneType (invalid state)
TC4	Extras = [999.99]	Higher total	Tests extras list handling

Fork Constructor

Test Case	Input	Expected Result	Reason for Inclusion
TC1	"Fox", 999.99, 160, 36	Fork object created	Normal object creation
TC2	Strings: "850", "150", "35"	Converted correctly	Test string conversion to int/float
TC3	"Fox", -200, 160, 36	Negative price	Boundary case – validation needed later
TC4	"" (blank brand)	Allowed	Test blank brand (no validation)

validate_component(component)

Test Case	Input	Expected Output	Reason for Inclusion
TC1	Valid Fork object	True	Normal case – placeholder logic
TC2	None	True	Always returns True (shows weakness)

Class diagram:

