Table of Contents

[A)General description: 2](#_Toc89620239)

[B) b) Language of Options, and Environment/Sensory information 2](#_Toc89620240)

[Options 2](#_Toc89620241)

[Environment/Sensory information: 2](#_Toc89620242)

[Defeasable: 2](#_Toc89620243)

[Scenarios: 2](#_Toc89620244)

[Panikos 2](#_Toc89620245)

[Cristian 3](#_Toc89620246)

[D)Architecture 3](#_Toc89620247)

A)General description:  
We have 2 students who have different ways of getting their foods.

Panikos prefers ordering delivery as much as possible then takeaway then cooking. Where as Cristian prefers cooking, then takeaway then lastly, delivery.

Firstly, there is a chance the restaurant does noDelivery or noTakeaway, moreover there is a chance we can’t cook (noCook) each contradicting all their respective options.  
Secondly, we can be in moodToCook so we cook despite preferences.  
Thirdly, we can haveHw so we prefer delivery/takeaway depending on the User because Uni>All.

Fourthly, there is a chance we have easyHw so we prefer to do what we wanted in the beginning and we allowed to.   
Latstly, if all options are contradicted, then we are left with no option than to cook.

# B) b) Language of Options, and Environment/Sensory information

Options:

* delivery(Method)
* takeAway(Method)
* cook(Method)

## Environment/Sensory information:

* noCook/0
* noDelivery/0
* noTakeaway/0
* noOptions/0 //If all above are true then we are forced to cook
* moodToCook/0
* haveHw/0
* easyHw/0

## Defeasable:

* prefersTakeaway #For Cristian who prefers takeaway than dleivery
* prefersDelivery #For Panikos when he prefers delivery than takeway

# Scenarios:

### Panikos

1. } , delivery(method)> # we prefer delivery always
2. noDelivery} , takeAway(method)> # we prefer takeaway after delivery is unavailable
3. moodToCook}, cook(method)> # we prefer to cook
4. haveHw}, delivery(method)> # we prefer delivery than takeAway (Cristian does not)
5. haveHw,easyHw}, delivery (method)> # we can still cook, but prefer delivery (Cristian does not)
6. moodToCook, haveHw , easyHw }, cook(method)> # we can do anything so we prefer to cook
7. moodToCook, noCook}, delivery(method)> # we cannot cook so we go with delivery
8. haveHw, noDelivery}, takeAway(method)> # we cant delivey like above so takeaway instead
9. haveHw,easyHw, noDelivery, noTakeaway}, cook(method)> # we cook even if last resort
10. moodToCook, haveHw , easyHw, noCook }, delivery(method)> # we delivery because we cant cook

### Cristian

1. } , cook(method)> # we prefer delivery always
2. noCook} , takeAway(method)> # we prefer takeaway after cooking is unavailable
3. moodToCook}, cook(method)> # we prefer to cook
4. haveHw}, takeAway(method)> # we prefer takeAway than delivery (Panikos does not)
5. haveHw,easyHw}, cook(method)> # we can still cook,so we do (Panikos does not)
6. moodToCook, haveHw , easyHw }, cook(method)> # we can do anything so we prefer to cook anyways
7. moodToCook, noCook}, takeAway(method)> # we cannot cook so we go with takeaway
8. haveHw, noTakeaway}, delivey(method)> # we can’t takeaway like we like so we delivery instead
9. haveHw, easyHw, noDelivery, noTakeaway}, cook(method)> # we cook because nothing else is allowed
10. moodToCook, haveHw , easyHw, noCook }, takeAway(method)> # we takeaway because we cant cook like we like

# D)Architecture

Asd

# E)Manual

Asd