01/12/2022

MEETING 2

CHAIRMAN: Daan Panis

NOTES 😊

* Deadline tomorrow for Architecture Draft – upload on GitLab not Brightspace
* Recommended that we start coding from this point onward
* Keep microservices at 5-6 to avoid getting too crowded – (we have 6)
* We can use the Discussion tab to ask for elaboration on questions we might have.
* We will get a Document with more explanations after Tas have a meeting regarding specifications for our scenario today.

QUESTIONS/ANSWERS WE HAVE

* Customers NEED an account to order pizzas – without an account the customers do not exist to the store/system.
* The buy one get one free coupon: possibly two ways to implement it, we will get more information regarding this later today. You always get the cheapest pizza for free. You can only apply one coupon per order and thus even if you have 10 pizzas in the cart, only one becomes free by using this coupon.
* “Notification” – since there is no UI there’sno popup notification, or email. Since we just code the backend, a way to do it is to send a request with a string to the microservice that handles this functionality and have it recognize that as a “notification” for a placed order.
* Filtering : logic-wise we should be able to filter by multiple different allergies – but this is a very specific and “could-have” thing that we can choose how to implement. Bottom line: we can choose what is easiest for us. We might also get more elaboration on this after TAs discuss.
* Different types of Dough: we can include this in the customization of pizzas – we can also filter types of dough based on allergens eg. Gluten
* Non-Functional requirements are mostly technical requirements. Eg. Test coverage, language, etc. But it can also refer to deadlines etc.
* Use “the system” and not “administrator” to refer to processes that happen automatically – specifically referring to the coupon eligibility checks.
* Regional Manager account: it depends on our architecture, but it is most intuitive to have separate Regional Manager account and, if he wants to order as well, creates a user account just like any other customer would.
* All default pizzas are the same for every single store in the region. The regional manager is responsible for a number of stores (the ones in his region). It should be possible to have different default pizzas/topics for each region, meaning the Regional Manager can edit the default pizzas/topics and this change is applied to all stores in his region. Associating a Regional Manager to a store is enough. We can just map database to store and not to region of stores. We do not need to think about regions that much, it is not that deep.
* We do not need to include microservice interactions or UML in the Draft that is due tomorrow.
* We do not have to care about payments!
* Feedback on Microservice Diagram/Architecture Draft:
  + It’s good that we have separate Profile and Authentication microservices
  + Credentials are associated to a profile but are also used for Authentication – TA said keep them in Authentication and make a way to connect them to Profile
  + Wait until after TA meeting to make more informed changes
  + TA said it looks really good 😊 just don’t add more microservices
  + Move pickup store to an earlier point (not at checkout)
  + How does the microservice recognize if the User is a Customer or Regional Manager: it should be a different entity, we can create communication only between specific microservices. ie. There’s no functionality allowing Users to access any coupon bits. Include the credentials of the regional manager and make a check to the database to ensure this request is coming from the regional manager. If we can configure tokens then we can configure different tokens for User vs Regional Manager
* We don’t have to care too much about efficiency
* The point of the course is not about implementing as many features as possible – it is more about having proper structure, scalability, etc.
* For explaining why we made each microservice: What is the functionality and why did we decide it is big/important enough to be its own microservice.