

Office Hours 10/3 BRD HLD

Monday, October 3, 2022 7:50 PM

Error handling

Exit process as early as possible

End of milestone 1

Grade is based off content

This feature is low feature, this is high feature

Let's you know what you want to do next semester

Right now its just anybody assigned to whatever

This will change over time

By milestone 2 you will know requirements

Will have to alter BRD, might have to alter project plan

Getting things in order this semester for next semester

Microservice section HLD

You're not detailing services

1. Based on microservice design, do I want to have multiple services?
Self-contained? Cross-cutting concerns?
2. Do I want to deploy them in different environments or all at once?

On BRD what else should we add?

What needs to be in place for you to even perform this operation

Success and failure outcomes

Incoming operations: this needs to happen for it to be successful

All 5 of these need to happen or any 5 of these

Failure outcome: any one of these fail all of it fails

Biggest issue for success outcomes MUST ADD

Do any of them need to happen or all of them need to happen

Some functional requirements contradict non-functional requirements

Functional requirements mention user must see 100% of data points

Non-functional requirements say user must see 75% of data points

We don't know requirements specific to core components yet so just do it as best as we can

We will flesh out BRD later once we know more

HLD

Where to handle errors?

Which errors do you want to handle?

Where do you want to handle security?

What type of security?

UI: authentication, authorization, or both
Data access: authentication, authorization, or both

Activities diagram CAN be used

Every step of flow = low level

Use-case diagrams = low level

Data flow diagram = low level

How should system behave REGARDLESS of feature
AGNOSTIC of a feature

Any material you use must be referenced (including his own notes/diagrams)

Business rule

Only 13+ can use our app
Arbitrary rules

TEAM REVIEW

Daily standup

Retrospective

All of these are meetings

We need to have PROOF of these meetings

Biggest issue is commits

Copy of everything needs to be in github

Committed just now

Questions:

You are only doing something one at a time

You typically don't assign people different things in one sprint

Stagger them

- **SPRINT PROCESS**

Every task should be 8-16 hours

4-6 hours a task for student

Some teams even go down to 1-2 hours

Breaking down to smaller subtasks

- **Story item is broken down from project backlog into spring**

40 hour item into 4 hour task

- Risk mitigation is only in internal project plan
- Microservice architecture is too service specific atm
 - Not agnostic design
- Willing to take on the cost?
- Can use technology pending approval and inclusion in bill of materials
 - Very rare to need another type of database
 - 80% of all consumer grade apps are relational
- Can have separate databases but their bounded context is encompassing of that
 - SHOULD NOT have jumps to each other database
- Bounded context is not properly sized originally, need to expand context
 - Microservice principle: if one service breaks another is not affected
 - Major benefit
- Low level design due for milestone 2 is only registration and log-in
- Site map is technically later, need to understand how to build solution first before we can decide how many pages to implement
 - After high-level design is vetted + approved but before specific features
- Do not decide on technology at this point
- There are always people who forget features
- Fixing stuff = create new work item
 - Hours get lost
 - If it's one task then just whatever but it's a big task put it into the backlog