4156 SQL + incremental schema design post mortem design spmt

10/6/16

reminders-

revised proposal
due tonight

practice tasks
due next Tuesday

cover at end of class

→ next team assignment due Oct 13

requirements and wireframes

define small set of high prority User stories for your 1st iteration

small enough that you will be able to demo for your team mentor with 2 weeks

expand these user stories into use cases with basic & alternate flows

sketch wireframes / story boards for each user story / use case

trello task board assigns to pairs

4156 10/6/16

do we need to use a framework? yes

why? to pass this course

AND because using a framework is almost always better, faster, cheaper

you could implement the full functionality of the system from scratch (you could also implement in assembly code r.t. use a compiler)

a framework provides a set of generic components that work together to accomplish common development tasks & functionalities

so developers do not need to reinvent or get independent libraries to play together nicely

there are frameworks for most programming & scripting languages for many different kinds of applications - we care about web development

4156 10/6/16

So what are these generic functionalities?

different frameworks build-in different functionalities & leave others to the developers

- standard interface between applications written using the framework & conventional web servers

- URL routing - maps incoming HTTP request to code that will handle it

- request & response objects - information to be received from or sent to browser

- template engine for separating application's logic from output HTML or other content produced

- development web server that updates itself as code changes so can test quickly / easily

- interface to external database(s)

4156 10/6/16

- various security

encode/sanitize inputs
cookies
session tokens
form tokens
password management
encryption (or hook to 3rd-party encryption)

- more general application support

resource pooling - objects, threads, DB connections
caching
MVC - model view controller
transaction processing
web services (SOAP) vs. REST
scalability (load balancing) replication

this is all server side
- some also support client side
e.g. AJAX toolkits

some frameworks are more closely tied to development
IDE plugins, build, testing

→ now back to next assignment