DESIGN REVIEW – WHAT TO EXPECT

CSE416 - Software Engineering

1

Format

- 30-minute team session
  - In class (2) – team receives a 5-point quality bonus. Review material is presented from student laptop on lectern
  - Not in class - Zoom session (all students are invited)
- Team selects the first diagram
- Instructor selects subsequent diagrams/material

You must be video enabled for your Zoom session

Date	Time Slot	Teams
4/16 (Thur.)	in class	1. Astros 2.
4/17 (Fri.)	9:00AM-10:00AM	1. 2.
4/17 (Fri.)	11:00AM-12:00PM	1. Padres 2.
4/17 (Fri.)	4:00PM-5:00PM	1. Giants 2. Braves
4/20 (Mon.)	9:00AM-10:00AM	1. Dodgers 2. Rockies
4/20 (Mon.)	2:00PM-3:00PM	1. Royals 2. Mets 3. Nationals
4/21 (Tues.)	in class	1. Tigers 2. Mariners
4/22 (Wed.)	9:00AM-10:00AM	1. Pirates 2. Yankees

2

Client/Server/Preprocessing

- Knowledge of relevant APIs - show an understanding of libraries and tools used in your design (e.g., Spring, pandas, shapely, etc.)
- Expected scope of design
  - Complete class diagram
  - Activity diagrams are labelled “AD” in master use case list
  - Sequence diagrams are labelled “SD” in master use case list
  - At least one sequence diagram should cover the interaction of objects in a request to your DB and any Spring service (e.g., conversion to JSON)

If you find you need to repeat logic in different sequence diagrams, just do it once and use a comment otherwise

3

Client/Server/Preprocessing Diagram Form

- Sequence diagram - use an electronic UML tool (e.g., LucidChart) and a set of notations consistent with the UML slide material and the UML components available with the tool
- Class diagram - Same as above
- Activity diagram – Same as above

You should be able to easily transition from one diagram to another

4

Logic of Sequence Diagrams

- Sequence diagrams should be logically consistent, for example:
  - a method call from an object is only shown if there was an immediately preceding method call to that object
  - a method call is made to an object only if the calling object has a reference to the called object
  - The method call should be a method of the class
- Logic of the sequence diagram should demonstrate that it will produce the desired results

Expect to show that methods in sequence diagram match class diagram

5

Activity Diagrams

- Remember that preprocessing use cases are just units of work so you should combine use cases in activity diagram
- Think of activity blocks as code you would aggregate in a function or module
- Be sure to show data flow with the name of the data on the arrow to a block
- One way to organize it is hierarchal
  - Top level activity diagram contains some activity blocks that are expanded in different activity diagrams.

6

DB Expectations

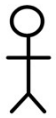
Expected scope of DB design: all required use cases

- Mongo DB
  - List of collections
  - Schema for each collection
  - Multiple collection aggregation strategy
  - Range of preprocessed data stored in DB
- Relational
  - 3rd Normal Form
  - Relational diagram
  - Understanding of JPA
  - Understanding of Spring interface to JPA (if using Spring)

7

Hints..

- You do not need to show direct access to your DB in your sequence diagrams. You only need to show a method call to the DB API
- Be sure to show the object (e.g., s:State) returned from a call
- For use cases beginning at the client, you do not need to include any Http client logic in your sequence diagrams other than the user action (e.g., click), and the call to the Axios object (or fetch API of the Window object)
- Show user trigger as the user icon with a message arrow (labelled click) to the DOM object



8

… Hints …

- Show the connection between the client and the server as 2 separate messages
  - Http method (with URL)
  - Controller method
- Be sure to specify the HTTP method in capital letters
- You can show the data passed to the server by
  - Following the Http method name with an identifier of the data (e.g., POST(state)) or
  - Showing the full URL as a comment above the HTTP method

9

…Hints

- Your class diagram will be used if there is a need to understand something we see in your sequence diagrams
- If any of your sequence diagrams are too large, you might consider decomposing them into separate sequence diagrams, with one diagram sending a message to the second diagram
- Be careful about showing loops in your sequence diagrams. In some cases, a nested loop might be better represented as a call to some method (that contains the loop)

10

Final Advice

- Relax
- The review should be fun and resemble a technical discussion you are having with colleagues in industry

Grading rubric on class Web site

11