# Part -3 Mini Project Development

# 20-Nov-2020

# Mini Project (curd operation)

# User Management

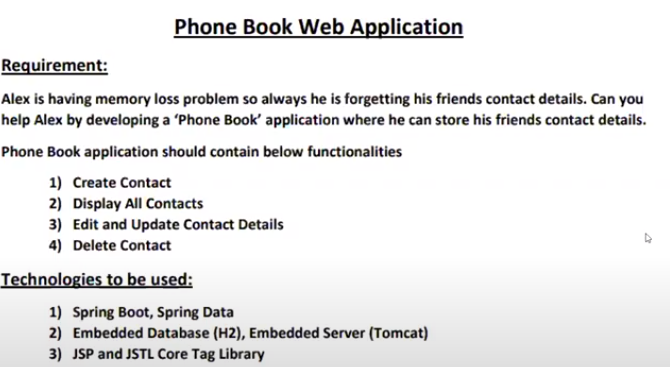
# Rest API (C2B And B2B)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

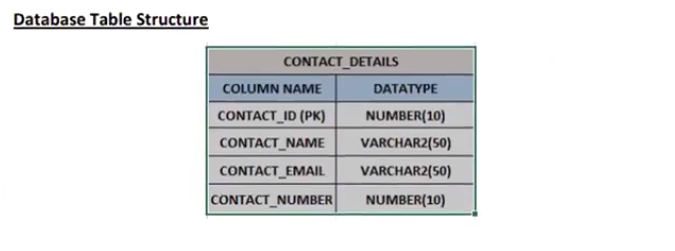
Yesterdays Session: Discussed on Initial stage of the project

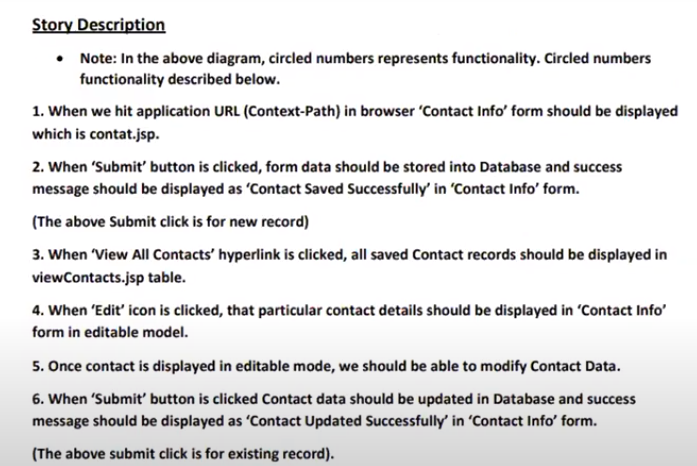
# 21st Nov Today’s session:

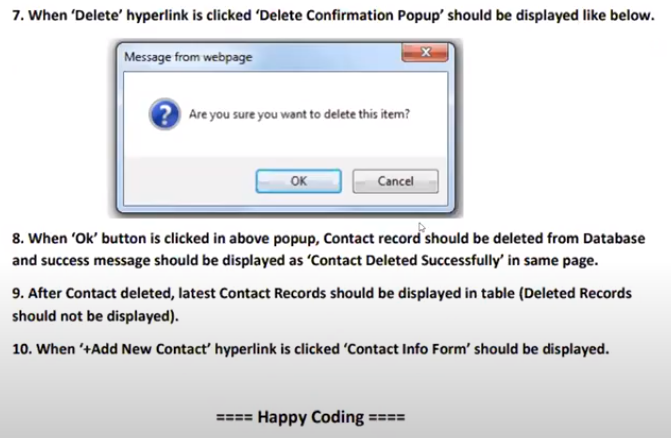
FFD document:







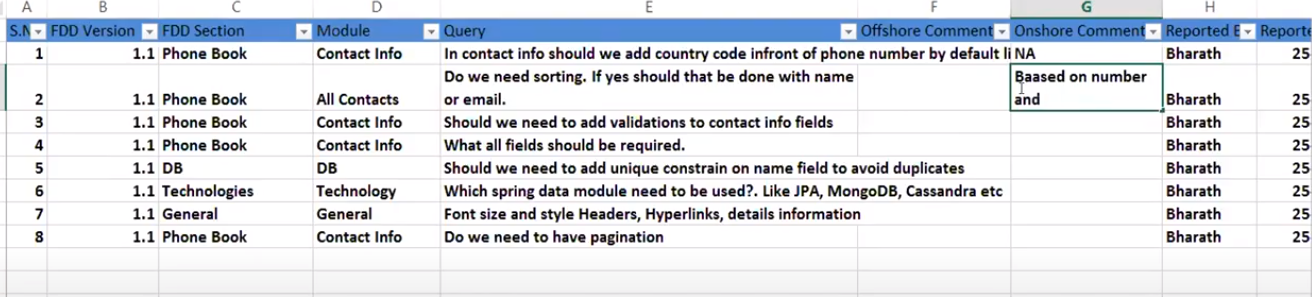




**Yesterday’s session: FFD Details**

# 22nd Nov

Todays session:



# Today’s session 23rd Nov

**Develop project structure, nothing else**

# Today’s Session 24th Nov

Steps to create 01-min project

1. Create Spring Boot application with below dependencies
2. Spring-boot-starter-web (to develop web application)
3. Spring-boot-starter-data-jpa
4. Project Lombok
5. Devtools
6. H2 database
7. Tomcat-embed-jasper (it is used to compile jsp in boot application)
8. Jstl
9. Configure below properties in application.properties file
10. Server port
11. Data source properties
12. View resolver
13. Create required packages in our application

Com.phonebook

Com.phonebook.entity

Com.phonebook.repository

Com.phonebook.service

Com.phonebook.controller

1. Create entity class and repository Interface
2. Create service interface and Implementation class
3. Create controller class with required methods and handler request
4. Create view files with presentation logic

Note: Optional 🡪 it used to avoid null pointer exception, whenever you are retrieving object from the DB, it may possible object will be there or may not be,

# Today’s session 25 Dec

Code

# Today’s Session 26th Dec

Code

# Today’s session 27th Dec

When user click on edit hyperlink that particular record should be display in Ediable mode then user should be able to update that record

* When user client on delete it should display confirmation popus window with ok and cancel buttons
* If we click on ok then record should be deleted
* If we click on cancel editable window should be close

Completed Edit and delete functionality

# Today’s session 27th Dec

* In every application delete operation will be available
* We can implement delete operation in 2 ways

1. Hard delete
2. Soft delete

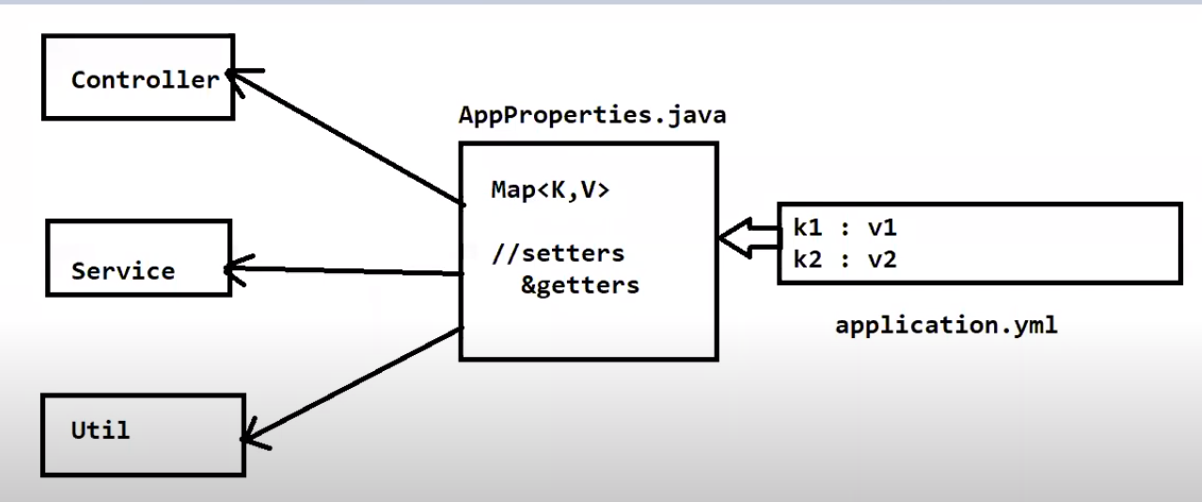
* Hard delete means deleting record permanently from the database using delete query,
* Soft delete means that making record as in active, it will be available in the database and when we want we will get that record
* Soft delete we can perform with help of update operation

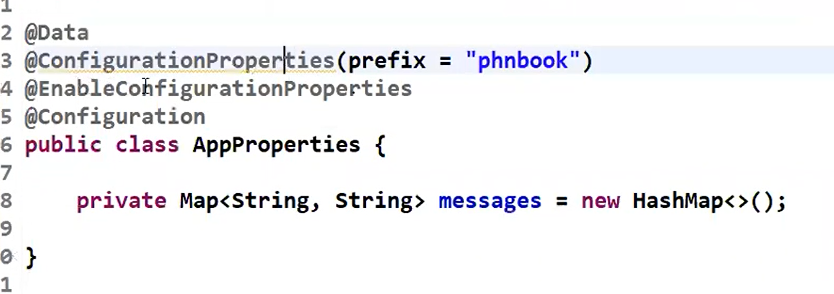
28th Completed all code:

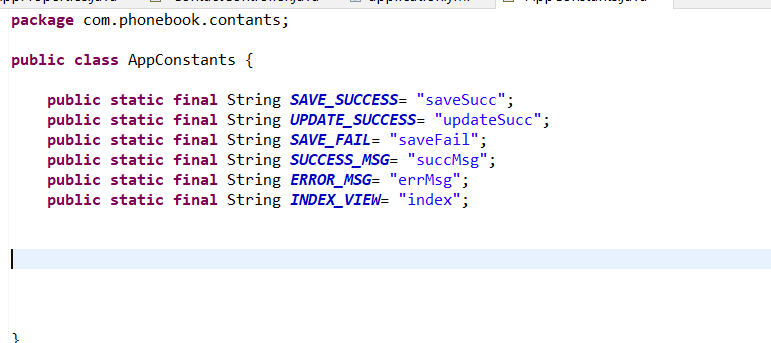
# Today’s session 29th Dec

* We cannot hardcode messages in the classes
* Write messages in the YML file

1. Create yml along with data and load that file to AppProperties.java
2. And use AppProperties in other classes







\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

Yesterday’s session: Dealing with application properties

* As part of application execution, we need to display few messages like success and failure msgs.
* It is not at all recommended to hard code success and failure message in source code
* To void hard coding we will configure the messages in application.properties or yml file
* In boot application, we can application.yml file msgs into java class using below annotation

@ConfigurationProperties

@EnableConfigurationProperties

* It is not recommended to write duplicate string Literals in our source code
* To avoid duplicate string literals we will use constant

Today’s Session : 30 Dec 07 : Code Review

* Code review is the process of verifying coding standards followed by developers in coding
* In Real time code Review will happen in 2 ways

1. Automated Review using tool
2. Peer Review

* We will use SonarQube server to perform code review, it will verify our source code and it will identify problems in our code
* SonarQube will not check the logic of the program is correct or not
* Peer review is the process of verifying the logic of the program

Junior developer logic will be verified by senior developers in the project

What is SonarQube?

SonarQube is developed by Java

Installations:

* Download sonarqube.org
* Start sonar sever by executing startsonar.bat

Location : sonar-> /bin/windows64/startsonar.bat

31s Dec : identify bug, code smell

32—Dec : Real time workflow ( get all the details from the notes)

Yesterdays : Code review process in Realtime

33: Dec : Logging :

* Login is the process of writing application execution details in runtime
* If application is running in local machine we can easily find the problem in debug
* If application functionality not working UTA then we can’t debug code, in that scenario loggiging is important

Logging components :

1. Logger: it is a class, it provide methods to perform logging
2. Layout: It is represent log message format, like what should be in log, like data, name of method
3. Appender : it represent destination of log file

We don’t

Log level : Trace, debug, info, warning, error, fatal

Debug : it shows all debug message along with info, warning, error and fatal

Info: it provide information

Warning:

Error:

Fatal

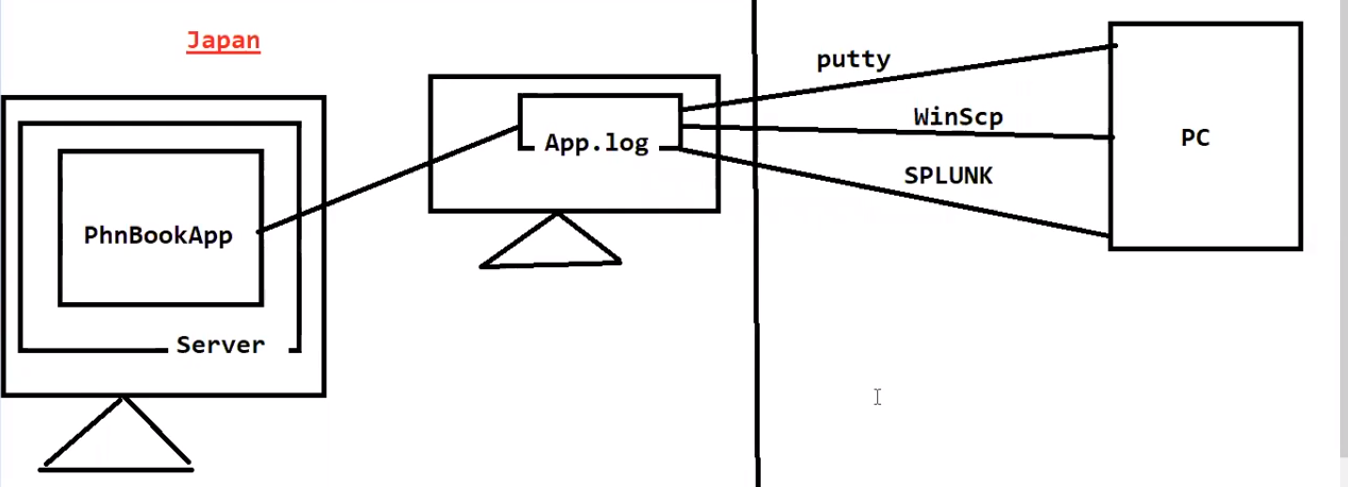
34 Dec :

Last session: discussed about logging

logging:

file:

path: PhoneBook.log



Splunk

35 Dec :