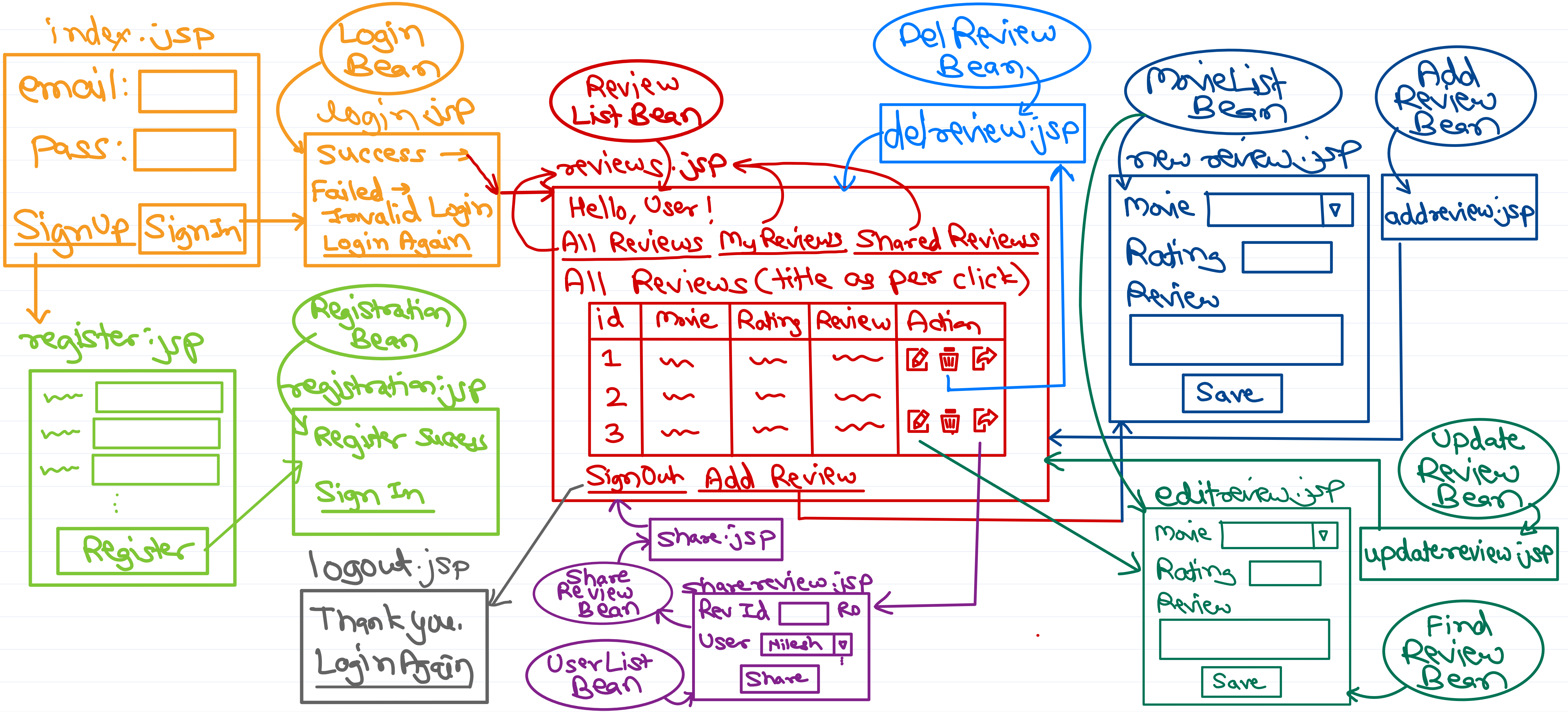
SUNBEAM

Java EE™

Advanced Java

Trainer: Nilesh Ghule

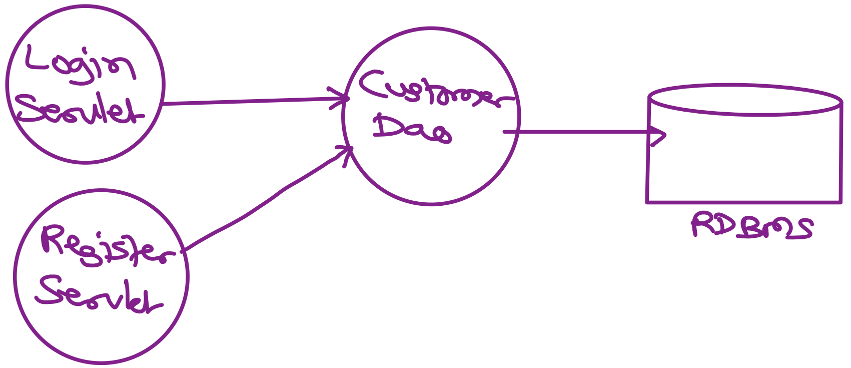
Movie Review JSP Assignment



Filters

AOP
Aspect Oriented Programming

- Filters is way of implementing AOP in Java EE applications. Filters are used to perform pre-processing, post-processing or both for each request.
- Multiple filters can be executed in a chain/stack before/after handling request.
- javax.servlet.Filter interface is used to implement Filters.
    - void init(FilterConfig filterConfig);
    - void doFilter(ServletRequest req, ServletResponse resp, FilterChain chain);
    - void destroy();
- Can be configured with @WebFilter or in web.xml (similar to servlets).



✓ AOP is implementation of cross cutting concerns without modifying business logic.

✓ Cross-cutting concerns = extra functionality
- logging → pre or post
- monitoring → post
- profiling → pre and post
  time − post = diff
- tx mgmt → pre and post
- security → pre

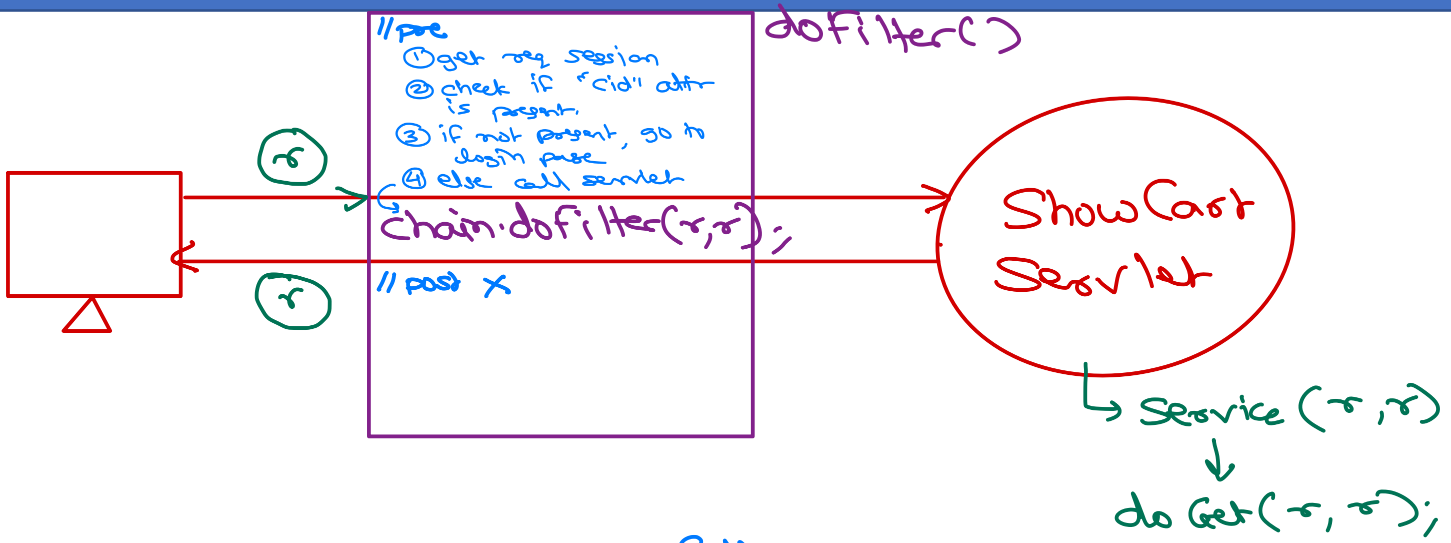
exception → rollback
success → commit()
pre → autoCommit(false)

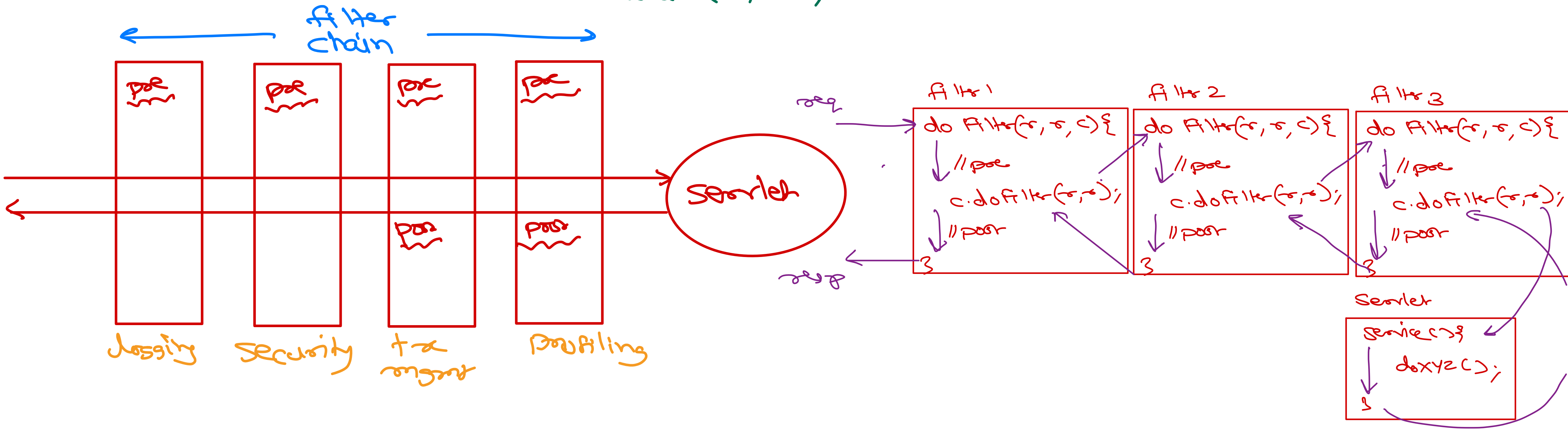
✓ CCC impl → Aspect → Advice
Aspect → class
Advice → method
Advice:
- → pre-processing
- → post-processing
- → pre & post

(Java) AOP
- → Java proxy
- → Aspect/J
- → Spring AOP
- → JavaEE filters

url-pattern = /*

Security Filter





Listeners

server-side

- Listeners are used to handle application level events.
- There are many listener interfaces.
  - ✓ ServletContextListener ✗
  - ✓ HttpSessionListener ✗
  - ✓ ServletRequestListener
  - ✓ ServletContextAttributeListener
  - ✓ HttpSessionActivationListener
  - ✓ HttpSessionAttributeListener
  - ✓ ServletRequestAttributeListener
- Listener class must implement one or more listener interface.
- Can be configured with @WebListener or in web.xml.

request begin
request end

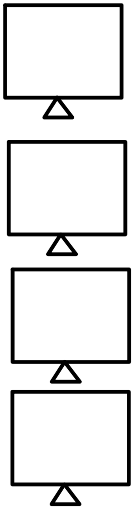
Context initialized
Context destroyed

Session Created
Session destroyed

attrib added
attrib removed
attrib modified

```
Context Initialized:
    ctx.setAttr("cnt", 0);
                        Integer
session Created:
  { cnt = ctx.getAttr("cnt");
    cnt++;
    ctx.setAttr("cnt", cnt);
session Destroy:
    cnt--;
```

Assign 4



keep track of num of online users at any moment.

```
<listener>
  <listener-class>pkg.MyListener</listener-class>
</listener>
```

JSP – ④ Custom tags

① JSP actions
② JSTL tags
③ third-party tags.

replace java code in jsp scriptlet

- Custom tags are used to combine presentation logic and business logic.
- There are two types of tags
  - ✓ Classic tags: inherited from Tag interface or TagSupport class.
  - ✓ Simple tags : inherited from SimpleTag interface or SimpleTagSupport class.
- SimpleTag implementation steps <> m </>
  - Decide tag name, attributes & body type. JSP empty
  - Implement tag handler class inherited from SimpleTagSupport
    - ✓ Constructor (param less)
    - ✓ Fields & getter/setter = attributes
    - ✓ setJspBody() = if there is body
    - ✓ doTag() = Logic implementation
  - Implement .tld file to define tag syntax. (.tld) tag library descriptor
  - In JSP, use <%@ taglib … %> & tag.

- SimpleTag life cycle
  - A new tag handler instance is created each time by the container using default constructor.
  - The setJspContext() and setParent()* methods are called.
  - The setters for each attribute defined for tag.
  - If a body exists, the setJspBody() method is called.
  - The doTag() method is called.
  - The doTag() method returns and all variables are synchronized.

SUNBEAM

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>