

# J2EE CASE STUDY:

## Corporate Training Management system

### Objectives

On completion of the case study the participant should be able to:

- Implement J2EE API's to develop web based application
- Handle databases to store and manage data
- Implement coding standards as per code review tool

### Pre-requisites

- Database Programming- Oracle 10g SQL
- Technology programming – Core Java along with Java EE
- Knowledge of code quality & Testing tool like JUnit

### Duration

- 5 days.

### Scenario:

Knowledge provides you with a training program that is focused in its objective, comprehensive in its course material and yet takes the least amount of time possible. It has all the required infrastructure and excellent facilities for participants to pursue their courses. It has a solid growth over a decade and has their branches across all locations in country. They offer the trainings for all Basic technologies (c, c++, Java, J2ee etc.), Domain (Billing, CRM etc.) and database (Oracle, SQL Server etc.) streams. Currently training requests are managed manually on the register which causes lot of human errors. Also solid history in growth and expansion records concern in record maintenance. So the management decides to implement a system that could handle end to end functionality for all training requests. The proposed system is to avoid any kind of manual operations and for ease of operations.

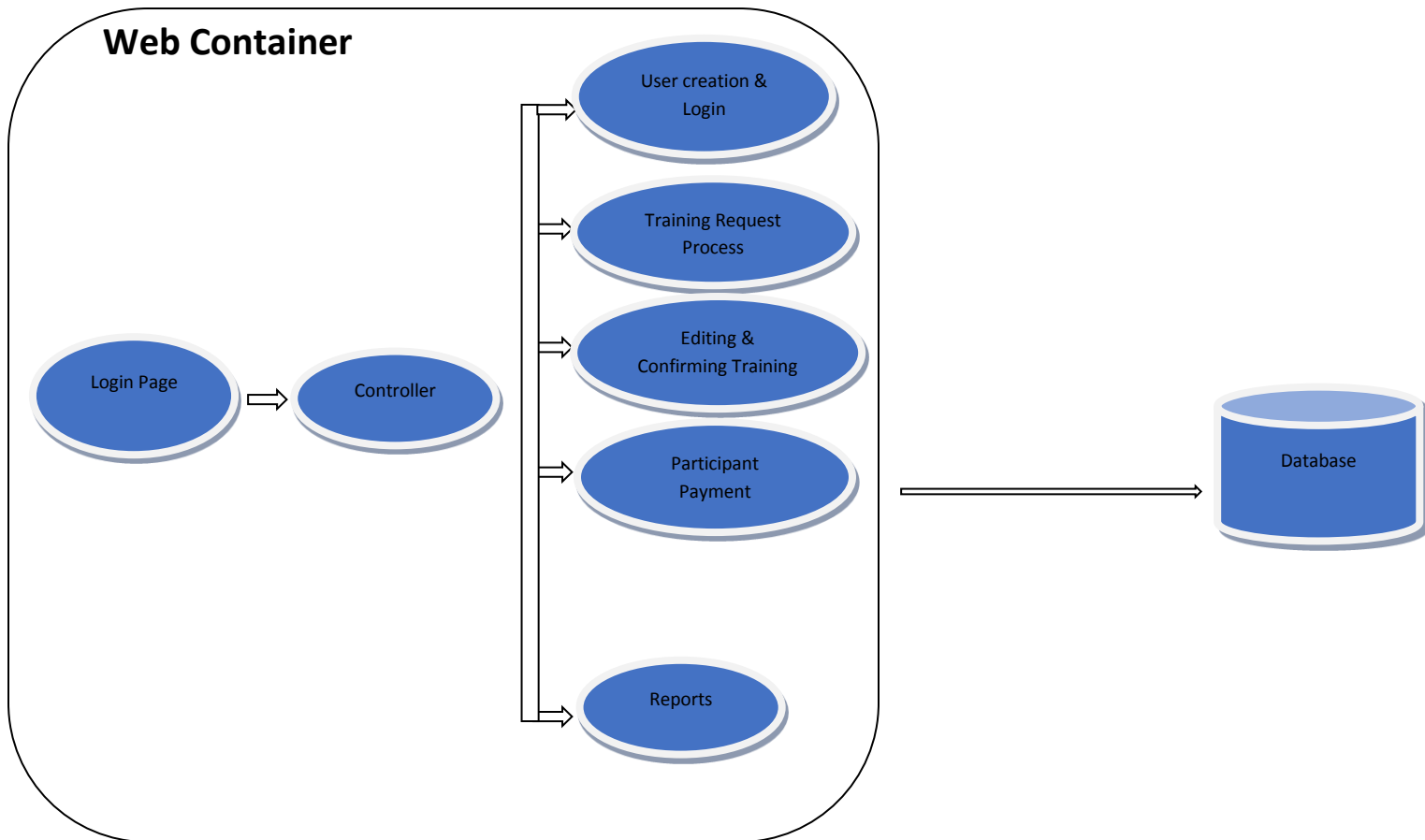
### Business Processes

- Registration Process (required only for users who is using system for first time. others can directly proceed to login process)

- INPUT: Participant information (For corporate user, company information is provided.)
- Validation / Business rules : every User must have unique & auto generated user id
- Output : generate unique training id and store details in the database
- Login Process
  - Input: login details which varies based on end user details
  - Validation / Business Rules: access control will be different based on the login, with 3 re-login attempts after that account gets locked.
  - Output: welcome screen
- Training request process
  - Input : training details
  - Validation / Business rules:
    - trainings will only be possible for future dates
    - Already planned trainings for month should be visible
    - Once booked request can changed by training executive only
    - faculty availability is checked on the training dates specified by requester
    - any training cancellations will be done only by training manager
    - the appropriate message is sent to the requester
    - training schedule date adjustment is possible a week prior to scheduled training.
  - Output : training confirmation
- Participant Payment process (this information can be filled by user or executive)
  - Input: the mode of payment
  - Validation / Business rules :
    - the payment amount should be as per the rules and compared with current system date
    - if the amount is not paid when expected, 2% charges are applicable

- training can be attended by the participant only if he has no dues
  - Output: the confirmation page to be displayed with Balance amount to be paid along with its expected due date
- Reports
  - Input: relevant input for specific reports
  - Output: Expected output is
    - trainer-wise delivered training details
    - technology-wise trainings delivered
    - company wise trainings attended
    - trainers availability in month
    - trainer wise planned trainings
    - trainer wise cancelled trainings

## Process Flow:



## Resources (hardware and software)

### Java and database technologies:

- Eclipse, Java 1.8, Java EE
- Oracle 10g database

### The hardware platform for deployment has to be minimum

- Pentium IV 1.2 GHz
- 1 GB RAM
- 40 GB hard disk

## Deliverables

- Implementation of database design
- Implementation of code to cover section “description”
- Submitted code follow java coding standards
- Code demonstrating Data dependent, independent and Exception handling as per the Field descriptions and business rules specified.
- The code submitted should have taken care of the common mistakes.

## Reflection upon learning

- Discussion and recording of common mistakes
- Lessons learnt

## References

- Java & Java EE Documentation