CSE 135

Discussion Section #3
MVC Design and Model Layer
SQL Schema (ddl.sql)

CREATE TABLE assignments (  
  assignment_id SERIAL PRIMARY KEY,  
  paper INTEGER REFERENCES papers(paper_id),  
  chair INTEGER REFERENCES chairs(chair_id),  
  reviewer INTEGER REFERENCES reviewers(reviewer_id),  
  due date
);

CREATE TABLE papers (  
  paper_id SERIAL PRIMARY KEY,  
  title TEXT
);

MVC Design (Struts Diagram)

welcome.jsp

showPapers.do

papers.jsp

PaperModel.getPapers()
**MVC Design (Struts Diagram)**

- **welcome.jsp**
  - Show Papers (hyperlink)
  - ShowPapers

- **papers.jsp**
  - Assignments
  - Reviewers
  - Show Papers (button)

- **showPapers.do**
  - PaperModel.getChairPapers()
  - AssignmentModel.getAssignments(…)
  - ReviewerModel.getReviewers()

- **insertAssignment.do**
  - AssignmentModel.insertAssignment(…)

- **DB**

**PaperModel.getChairPapers()**

```java
private static String selectChairPapers = "SELECT paper_id, title FROM papers 
ORDER BY paper_id";

public static CachedRowSet getChairPapers() {
    Connection conn = DbConnectionPool.getConnection();

    PreparedStatement pStmt = conn.prepareStatement(selectChairPapers);
    ResultSet chairPapers = pStmt.executeQuery();
    CachedRowSet crsChairPapers = new CachedRowSetImpl();
    crsChairPapers.populate(chairPapers);
    conn.commit();
    chairPapers.close();
    pStmt.close();
    conn.close();
    return crsChairPapers;
}
```
AssignmentModel.getAssignments()

private static String selectAssignments = "SELECT paper, chair, reviewer, due FROM assignments WHERE paper = ? ORDER BY reviewer";

public static List<CachedRowSet> getAssignments(List<Integer> paper_ids) {
    PreparedStatement pStmt = conn.prepareStatement(selectAssignments);
    List<CachedRowSet> crsListAssignments = new ArrayList<CachedRowSet>();
    for (int paper_id : paper_ids) {
        pStmt.setInt(1, paper_id);
        ResultSet assignments = pStmt.executeQuery();
        CachedRowSet crsAssignments = new CachedRowSetImpl();
        crsAssignments.populate(assignments);
        crsListAssignments.add(crsAssignments);
        assignments.close();
    }
    return crsListAssignments;
}

AssignmentModel.insertAssignment()

private static String insertAssignment = "INSERT INTO assignments (paper, chair, reviewer, due) VALUES (?, ?, ?, ?)";

public static void insertAssignment(AssignmentInsertForm assignment) {
    PreparedStatement pStmt = conn.prepareStatement(insertAssignment);
    pStmt.setInt(1, assignment.getPaperId());
    pStmt.setInt(2, assignment.getChairId());
    pStmt.setInt(3, assignment.getReviewerId());
    pStmt.setDate(4, assignment.getDue());
    pStmt.executeUpdate();
    conn.commit();
    ...
public class AssignmentInsertForm extends ActionForm {

    private int paperId;
    private int chairId;
    private int reviewerId;
    private Date due;

    public int getPaperId() { ... }
    public void setPaperId(int paperId) { ... }
    ...