

# Term Project

**(Weight 15%)**

Per the IS345 course plan, students are required to form teams and design and develop an Internet application using Java-based technologies (namely JSP and servlets).

The purpose of this project is to exercise concepts and techniques learned in the class and experience the full life-cycle of an Internet application development.

This document contains the rules that govern the project mechanics, the deadlines, and expected deliverables for this term project. In addition, a list of potential projects is accompanied with detailed requirements (any other project topic fulfilling the technical rules will be accepted).

## **Rules:**

1. Teams should be either 3 or 4 students.
2. Teams should not span different groups.
3. The project should embrace the following aspects:
  - The main pages must be built using JSP and servlets.
  - Using a database connection.
  - Making use of session, and cookies.
  - Perform client side input validation using JavaScript.
4. It is recommended to use NetBeans, but this is not mandatory.
5. The deadlines will be enforced strictly.

### Time Table:

<b>Week No.</b>	<b>Start Date</b>	<b>Deliverable</b>
7	Nov-07	Teams Members Names + Selected idea
10	Nov-28	Design document: ERD + site map (screen transition diagram) + pages mock ups
12	Dec-27	Implementation: Final ERD + JSP + servlets + user manual

### Potential Projects:

University Library Management (Ex. <a href="http://www-sul.stanford.edu">http://www-sul.stanford.edu</a> )
Required minimum functionalities: <ol style="list-style-type: none"><li>1. Signing up a new user (e.g. admin, student)</li><li>2. Updating a user details</li><li>3. Adding a book (by admin)</li><li>4. Updating a book details (by admin)</li><li>5. Browsing books ((by admin and students)</li><li>6. Showing a list of books that satisfy certain criteria (e.g., ISBN, publication year, author...)</li><li>7. Sending emails to late borrowers</li><li>8. Performing operations on books: borrowing, returning, extending borrowing period.</li></ol>

Domestic flight reservation (Ex. <a href="http://www.egyptair.com">http://www.egyptair.com</a> )
--

Required minimum functionalities:

1. Signing up a new user (e.g. admin, customer)
2. Updating a user details
3. Adding an aircraft (by admin)
4. Updating an aircraft details (by admin)
5. Adding a flight (by admin)
6. Updating a flight details (by admin)
7. Showing a list of available flights that satisfy certain criteria (e.g. date, source, destination, required number of seats...)
8. Performing operations on flights: booking, canceling, changing class, accepting payment
9. Sending booking/cancelation confirmation emails.

Train booking (Ex. <http://www.amtrak.com>)

Required minimum functionalities:

1. Signing up a new user (e.g. admin, customer)
2. Updating a user details
3. Adding an train (by admin)
4. Updating a train details (by admin)
5. Adding a trip (by admin)
6. Updating a trip details (by admin)
7. Showing a list of available seats that satisfy certain criteria (e.g. date, time, source, destination, required number of seats...)
8. Performing operations on trips: booking, canceling, and accepting payment

9. Sending booking/cancelation confirmation emails.

Basic Email (Ex. <http://pawpawmail.com>)

Required minimum functionalities:

1. Signing up a new user
2. Updating a user details
3. Sending an email
4. Showing a list of received emails
5. Showing a list of emails that satisfy certain criteria  
(e.g. from, to, date range...)
6. Performing operations on emails: view, delete, archive, reply, forward.

Note:

You should not use the JavaMail API in this project.

Movie Rental (Ex. <http://www.familyvideo.com/>)

Required minimum functionalities:

1. Signing up a new user (e.g. admin, customer)
2. Updating a user details
3. Adding a movie (by admin)
4. Updating a movie details (by admin)
5. Browsing movies
6. Showing a list of movies that satisfy certain criteria (e.g. year, main actors...)
7. Performing operations on movies: renting, returning, extending rental, payment acceptance.
8. Sending emails to late clients

Online Recruitment (Ex. <http://www.recruitegypt.net/>)

Required minimum functionalities:

1. Signing up a new user (e.g. job seeker, employer)
2. Updating a user details
3. Adding/updating a vacancy (by employer)
4. Showing a list of vacancies that satisfy certain criteria (e.g. industry, location, required experience...)
5. Showing a list of job seekers that satisfy certain criteria (e.g. industry, location, experience...)
6. Performing operations on vacancies: apply and save (by job seeker)
7. Performing operations on vacancies: hide (by employer)

Concerning the grading schema for the project documentaion

I suggest 1.5 for ERD

1.5 for site map

2 for Mockups

the criteria is to check whether ERD, sitemap, and mockups  
at least are fulfilling the stated requirements

Final phase

For all projects:

- Sign-up/login: 1 grade
- Update user details: 1 grade
- Email: 1 grade (except Basic Email)
- Search: 1.5 grades (except Basic Email)
- Add/Update: 2 grades (for all items per project) (except Basic Email)

Bonus:

Ajax (at least one function) 1 grade

CSS (on all pages) 1 grade

Library: (total 10 grades)

1. List of books: 1.5 grades
2. Perform operations: 2 grades

Domestic flight reservation:

1. Show available flights == search
2. Perform operations: 2 grades
3. Booking and canceled operations are graded alone (on DB): 1.5 grades

Train Booking:

1. Show available seats == search
2. Perform operations: 2 grades
3. Booking and canceled operations are graded alone (on DB): 1.5 grades

Movie Rental:

1. Show available seats == search
2. Perform operations: 3.5 grades

Basic Email:

- Send email: 1 grade
- Show list of received email: 1.5 grades
- Show list of emails with certain criteria: from, to, date range,...: 1.5 grade
- Perform operations (view, delete, archive, reply/forward)-- a grade per each: 4 grades

On-line recruitment: [from points 4-7 (1.5 grades/each)]

- Show list of vacancies with certain criteria (industry, location, required experience,..)
- Show list of job seekers with certain criteria (industry, location, experience,..)
- Perform operations by job seekers: (Apply, Save)
- Perform operations on vacancies