

HYSEA
Organizes
**STUDENTS COMPETITION ON
SOFTWARE DESIGN**
January 2008

Topic : Design for Web based movie reservation system

Conducted by:

Web based movie reservation system

1.1 Introduction

Ramesh wants to take his family to a movie tomorrow. Like most senior executives, he has no time to get out of office during working hours, stand in a queue and book tickets. He asks his help-desk to book the tickets for him. Lack of privacy when using the help desk bothers him. But what took the cake was that, the last time he asked some one from help desk to book tickets for him, they booked in the first row! He hates the fact that he is not in control of the decisions being made at the time of booking.

What if, his favorite movie theatre let him book tickets online! Ramesh sees the opportunity there and decides to invest in a product that would allow people like him book movie tickets on the web. It would let them look at the available choice of seats and choose the best option for their families, in the privacy of their homes, at any time of the day or night.

You and your team mates have been chosen to design this product.

1.2 Problem Definition

Some key considerations for you design

- It has to be a web based solution
- You can assume your users will use browsers which support java script
 - However they might not have flash or java or other plug-ins installed
- The different parties involved are:
 - The vendor: people writing/selling the software
 - The theater: people who bought the software from the vendor
 - The administrator: An administrator running/managing this software on behalf of the theater
 - The movie-goer: Person using the software to buy tickets for the theater
- The solution should support a multiplex scenario, where there are multiple screens in the same theatre.
- Each theatre might have a different seating layout, the administrator should be able to configure seating for each screen.
- There are tickets available for multiple prices for the same show. The price decides the set of seats to choose from and vice-versa. The administrator should be able to configure this.
- The administrator should be able to configure the movies, shows that are going to play in the upcoming few weeks and assign the screen for the shows.
- Identify different type of mechanisms for paying tickets (Credit card, debit card etc.,)
- You may assume that you can integrate with an external payment gateway software to process credit cards

- There will be multiple movie-goers booking tickets at the same time
- Software should handle unblocking of seats blocked for a movie-goer, if the payment is not successful / completed.
- Some movie-goers might want to choose seats from the set of available seats and other might just want the software to choose the best seats available. Design an algorithm that can allot the best seats to the users automatically.
- Different users will have different reasons for visiting the site.
 - Some might be looking to while away the time before the train leaves so they might want to know which movies fill the available time slot.
 - Some other users might want to watch a movie and might want to know for which shows the tickets are available.
 - Imagine a party of 10/15/20 decide to go for the movie, Wouldn't they like to know on which of the days continuous block of seats are available? Let's say a movie is very popular and they all are planning to go any day of the week, provided you get all 10/15/20/ together....
You are required to come up with such use cases and design functionality which would enable such uses.
- The design should be modular.
 - Identify which modules can be replaced with existing open-source solutions to reduce cost of development
- As the architect of the solution, you are required to add and document additional design considerations not listed here.

1.3 Evaluation Criteria

Your design will be evaluated on three aspects: Technical merit, Functionality and User Experience (UX). The success of the project depends on all the three aspects and the design will be graded using the formula:

$$\text{grade} = (\text{technical-merit})^{0.85} * (\text{functionality})^{0.85} * (\text{user-experience})^{0.95}$$

The implications of this formula are listed in the table below:

Technical	Functional	UX	Overall
100%	100%	100%	100%
100%	80%	100%	83%
100%	100%	80%	81%
80%	80%	80%	55%
100%	60%	100%	65%
100%	100%	60%	62%
75%	75%	75%	47%
80%	60%	80%	43%
60%	60%	60%	26%
100%	100%	0%	0%

In addition to the above, some of the common aspects that will be looked at for evaluation would be:

- Completeness/Quality
- Simplicity
- Innovation / Originality
- Features/substance
- Feasibility
- Scalability
- Technology selection / Architecture

Hint:

1. *DO not reinvent the wheel. You may refer to the commercial off the shelf products available on the net for some ideas and improve on them.*
2. *To see a live version of a web based movie reservation system, you can visit these sites. (However, you will notice the requirements for your application are different).*
 - a. <http://www.prasadz.com/>
 - b. <http://www.pvrcinemas.com/>
3. *To know more about documenting screen-flows visit the site:*
 - a. <http://www.agilemodeling.com/artifacts/uiFlowDiagram.htm>
4. *Please go through online help material to get more insights into concepts and ideas of design. Some of the websites you can search are*
 - a. [Campusconnect.infosys.com](http://campusconnect.infosys.com) (id: demo@infosys.com ; Passwd: infosys)
 - b. www.google.com (very commonly used search portal)
 - c. en.wikipedia.org (free online encyclopedia)
 - d. www.usabilitymatters.org

Attractive Prizes for Winners!!!

- | | |
|---------------------------------------|------------------|
| First prize (Max Team Size of Three) | - Laptops* |
| Second prize (Max Team Size of Three) | - iPods* |
| Third prize (Max Team Size of Three) | - Mobile phones* |

* All Team members will be awarded individually

KEY MILESTONE DATES

- | | |
|----------------------------|--------------------------------------|
| 16 th Dec. 2007 | Announcement of the Competition |
| 4 th Jan. 2008 | Last day for submission of Abstract |
| 11 th Jan. 2008 | Last date for seeking clarifications |



25th Jan. 2008
8th Feb. 2008
15th Feb. 2008

Last day for submission of Designs
Announcement of short-listed designs
Prize distribution to winners

Contact info:

For information or clarification, please send a mail to hysea.sdc2008@gmail.com

1.4 Submission Format

- Final submission should be a document in word/pdf format
- Document your assumptions if any
- Document your choices, and highlight the advantages of the choices you made
- Supplement the design document with component diagrams.
- While documenting algorithms, add pseudo code and discuss the advantages of your algorithm
- Document use-cases using use-case diagrams
- Provide rough sketches of various web-pages as seen by your users in each flow (registration, booking etc). The sketches could be made in power point / visio and attached as images in the word file.

COMPETITION INFORMATION AND GUIDELINES

1. Each team should comprise to a maximum of three (3) members only
2. Each college can have maximum of two (2) teams participating in the competition
3. One person can be part of only one team.
4. Finalists will be notified individually and to their college.
5. Any document that does not follow the format mentioned below will automatically be disqualified for the competition
6. All submissions become property of HYSEA.

<<Title of the Paper>>

Author(s) Names:

Some general guidelines:

. Focus your paper on the core idea(s). At least 50% of the paper should be devoted to the topic at hand this is not a tutorial; do not explain background, introduction or basic concepts at length;

The average size of the paper should be 8-10 pages of MS Word

. Do not copy abstract to the conclusion section. Abstract should have what the paper is focused on; while conclusion should include specific, measurable outcomes of using the said approach

. Use tables, diagrams and pictures. Refer to these in the text. In case there is no discussion of these exhibits in your text, remove the exhibit

. Do not use adjectives. Qualify statements with metrics (e.g. large project vs. 350 man-month project)

. Acknowledge source of information (e.g. data from Gartner paper xxx pages 5-7 or connect to one of the references)

. Use standard terminology

. Use simple language and short sentences.

. Use the third person (Avoid usage of .I. and .You.)

. Avoid jargon. When you use technical terms/ jargon/ acronyms/ abbreviations, please explain them in brief, the first time they appear in the paper.

. Adhere to the sequence of the paper as suggested in the sample below

Introduction

(Heading 1. Arial, 12 size, bold, paragraph setting before 0, after 6 pts, line spacing single)

It contains the broad overview of the paper (no details); set the context for the paper.

Main Body of the Paper

It should include headings, sub-headings, illustrations (to demonstrate results), More than 90% of the paper should be main body.

The main body of the paper should cover the following:

. Problem definition

. Proposed solution

. Architecture

. Environment

. Performance Attributes

. Functional overview / Flow diagram

. Database relationship / ER diagram

. Constraints / Exceptions

(Heading 2. Arial, 11, bold, paragraph setting before 0, after 3 pts, line spacing single)

(Heading 3. Arial, 10, bold and italics, paragraph setting before 0, after 3 pts, line spacing single)

Please note that the figure number should be referred as: Figure 1, Figure 2, Figure 3 etc instead of .refer figure below. Or .Figure shown above...

Conclusion

This is the final section of the main body of the manuscript and it is mandatory. This section must be well connected to the abstract. It must not be a restatement of the abstract. It must draw points from the discussion presented in the body of the manuscript. It must be written in a brief and logical manner.

References

All sources of information (artwork or otherwise) including full name, title (if applicable), address and phone number/Web page must be listed.

At the end of the article, references must be arranged in alphabetical order of authors. Surnames and chronologically for each author. The author's surname is placed first, followed by the year of publication in parentheses. For more details, refer MLA Handbook for Writing Research Papers

Sample:

[1] A.B. Smith, C.D. Jones, and E.F. Roberts, .Article Title., Journal, Publisher, Location, Date, pp. 1-10.

[2] Jones, C.D., A.B. Smith, and E.F. Roberts, Book Title, Publisher, Location, Date.

Abstract for HYSEA Students Competition

Name of the team

(how do you want your team to be called)

Authors Name(s)

(indicate the primary author within bracket)

Name of the College and address

Tel number and e-mail of the primary author

Abstract Title

Body of the abstract (not exceeding 250 words)

Please submit the abstract to hysea.sdc2008@gmail.com



Author Declaration Form

**Name of the team
(how do you want your team to team called)**

Name of Primary Author

Name(s) of other Author(s)

Class / Discipline

Name of the College

Address

City / State

PIN/Zip

Phone / mobile

Email

Paper Title

I hereby certify the following:

- . The content / structure that forms part of the above paper is original and has not been copied from an existing source of content
- . The paper is not a reconstruction of the whole or part of an existing source of content.
- . The sentences and key phrases in the paper have not been copied from an existing source of content.
- . All the reference material used in the development of the paper is listed as an annexure to the submission for the paper.
- . Papers / designs submitted will be the property of HYSEA.

Signature:

Date:



<<Title>>

Design Document

Name of the team:

Name of the Author(s):

Name of the college:

REVISION CONTROL CHART

© Copyright 2008; this document is the property of Hysea

REVISION CONTROL CHART					
Revision	Date	Changes	Affected pages	Prepared by & date	Reviewed by & date
0.1	15-12-2007	Initial version	All		

Table of Contents

The sections mentioned are indicative and any additional topics can be added at appropriate sections.

1. Introduction
2. Definitions and Acronyms
3. Requirements
 - 3.1 Problem Definition
 - 3.2 User Interface requirements
 - 3.3 Performance requirements
 - 3.4 Constraints
4. Solution Proposal
 - 4.1 Proposed Solution
 - 4.2 Architecture
 - 4.3 Environment
 - 4.4 Performance Attributes
 - 4.5 Functional overview / Flow diagram
 - 4.6 Database relationship / ER diagram
 - 4.7 Constraints / Exceptions
5. Conclusion
6. References

Glossary

Usability - Ease with which a user can learn to operate, prepares inputs for, and interprets outputs of a system or component. The system with good usability will be treated as 'user friendly' system.

Security - The ability of a system to manage, protects, and distributes sensitive information. Security of the system should prevent the unauthorized access to the system and its data.

Maintainability - The ease with which a software system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment.

Coding Standards - Programming or coding conventions for writing source code in a certain programming language. Coding standards are useful for ease of maintenance; ease of understanding and to get good performance.

Typically, in software industry each company may have it's own coding standards for each programming language. These standards will be followed by all the developers across the organization in various projects.

Webex - It's a kind of tool to support online meetings, web conferencing and video conferencing services

Metrics - A quantitative measure of the degree to which a system, component, or process possesses a given attribute. Software metric is a measure of some property of a piece of software or its specification. e.g number of defects per lines of code;

Annexure: Students Kit

Given below are the templates (with some example entries) for the documents related to a sample project. These are just guidelines only. This document acts like reference copy only.

Requirements Specification (RS)

Following is a template for the RS document. Some example requirements are entered in it to show how to use the template. Make sure that you enter even the smallest/most trivial requirements also.

No.	Requirement	Essential or Desirable	Description of the Requirement	Remarks
RS1	The system should have a login	Essential	A login box should appear when the system is invoked.	The logins are assigned by the mail-admin
RS2	The system should have help screens	Essential	Help about the various features of the system should be provided in sufficient detail in a Q&A format.	For example help should be provided for : How to create a defect. How to modify a defect etc.
RS3	The system should 'lock' the login id if wrong password is entered 3 times in a row	Desirable	This feature will improve the robustness of the application	Since the application is going to be used only by the employees of the organization, this feature is not essential. However, if time is there, this will be implemented.
RS4				

Database Fields Specification

Following tables may be created to serve the application:

User table – Contains details about the users of the system (their name, role, access privileges etc)

Defects table – Contains details about the defects created by the users of the system (defect id, creator’s id, status of the defect, owner id, date/time of creation/last modification etc)

User table is taken as the example for specifying the field details. Some example entries are filled in.

No.	Field Name	Range of valid values for the field	Remarks
1	User ID	To be defined by the team.	This is the key field of the database as it is unique for a user. This will also serve as the login for the system.
2	Name	Up to 30 characters in length.	Special characters like underscore are not allowed.
3	Access Details	Application Admin / User	
4	Email ID		Should be a valid email ID.

High Level/Detailed Design (HLD/DD)

Overview of the system

Provide a high-level block diagram depicting where the database will be located, where the application will run etc.

Detailed Design

Split the system into its design components. In this case, the components would be user-verification, defect creation, mail notification, component maintenance, report generation etc. For each of the components, provide information in the following format. User-verification component is taken as the example.

Component one

User-verification

Purpose

This component will verify if the user who is trying to access the system is a valid user.

Component two

Component three....and so on....