Development of a feature rich, practical online Tickets reservation system for Cinema halls.

Contributed by
Sunday, 24 February 2008
Last Updated Monday, 10 March 2008

{mosgoogle} Title of the project Development of a feature rich, practical online Tickets reservation system for Cinema halls. Abstract of the project This project is aimed at developing an online ticket reservation system for Cinema Halls. The Ticket Reservation System is an Internet based application that can be accesses throughout the Net and can be accessed by any one who has a net connection. This application will automate the reservation of tickets and Enquiries about availability of the tickets. This application includes email confirmation for the tickets. Keywords Generic Technology keywords Databases, Network and middleware, Programming Specific Technology keywords MS-SQL server, HTML, Active Server Pages UNIX, Shell, C, Oracle Project type keywords Analysis, Design, Implementation, Testing, User Interface Functional components of the project Following is a list of functionalities of the system. More functionality that you find appropriate can be added to this list. And, in places where the description of functionality is not adequate, you can make appropriate assumptions and proceed. {mosgoogle} The Cinema hall has a web site and any user of internet can access this. The cinema hall is a multiplex which has 5 screens. Each screen has 3 different types of seats/classes. Only 50% of the seats are available for online reservation. A person should be able to login to the system through the first page of the application change the password after logging into the system Should be able to create a new login for the accessing the reservation facility. Query the films on show for two weeks (Only two weeks advance reservation is available) should be there. No reservation before two days can be done. See his current reservations on different movies along with the details. Able to choose the seats which can are available for a certain class. Can select seats from different classes as well for same show and screen also. Give details about the details about the credit card details. Able to select the mode of transfer of tickets whether through the courier or collection at the counter (as per that fare will be charged). A mail should be send to the concerned person about the confirmation of the ticket to the specified email address. The login Id and password should be sent to the mentioned email address if a new account is created. A calendar should be there which helps the person to select dates. It should also show the public and nation holidays. {mosgoogle} The system should automatically show the fare for the corresponding shows and amount of money needs to be pay for selected seats.

Steps to start-off the project There are couples of alternatives to implement such a system. A. Microsoft platform: The system is developed using Active Server Pages as the Front end and SQL Server as the back end. B. Unix-based platform: HTML or even Shell scripting, C programming, any Relational database (e.g. Postgress or Oracle or even flat files), and tools in UNIX The following steps will be helpful to start off the project. Study and be comfortable with technologies such as Active Server Pages/HTML and SQL server. UNIX commands, Shell programming, C Programming, Tools like AWK etc. Some links to these technologies are given in the ‘Guidelines and References’ Section of this document Decide on a reservation policy and other related aspects like how many screens that multiplex has and what is the fare structure of different types of tickets. Then the number of daily show and timings. Make a database of people whosoever login. Decide on the various details of the people that would be stored in the database (like name, age group, address, location, system-login, password in cryptic form, etc). Since the real-time project needs to be tested in real-time, you can take ‘hours’ as ‘days’ for testing the system. However, the display will still be in ‘days’ only. Create the front-page of the reservation system giving a brief description about the system and a login box. Create the help-pages of the system in the form of Q&A. This will help you also when implementing the system. Create other sub-systems like automatic notification, screens for various functions (like login,calendar,reservation etc) The data’s Encryption should be done as it deals with the credit cards and related things. Requirements Hardware requirements Number Description Alternatives (If available) 1 PC with 2 GB hard-disk and 256 MB RAM Not-Applicable 2 Software requirements Number Description Alternatives (If available) 1 Windows 95/98/XP with MS-office Not Applicable 2 MS-SQL server MS-Access 3 Linux Not Applicable 4 Oracle database system POSTgres Manpower requirements 2 to 3 students can complete this in 4 – 6 months if they work fulltime on it. {mosgoogle} Milestones and Timelines Number Milestone Name Milestone Description Timeline Week no. from the start of the project Remarks 1 Requirements Specification Complete specification of the system (with appropriate assumptions) including the framing of reservation policy etc constitutes this milestone. A document detailing the same should be written and a presentation on that be made. 2-3 Attempt should be made to add some more relevant functionality other than those that are listed in this document. 2 Technology familiarization Understanding of the technology needed to implement the project. 4-5 The presentation should be from the point of view of being able to apply it to the project, rather than from a theoretical perspective. 3 Database creation A database should be created. As per the rules taken for the purpose of maintenance of the records. 5-7 It is important to finalize on the database at this stage itself so that development and testing can proceed with the actual database itself. 4 High-level and Detailed Design Listing down all possible scenarios (like reservation application, cancellation, search automatic money crediting etc) and then coming up with flow-charts or pseudocode to handle the scenario. 7-9 The scenarios should map to the requirement specification (i.e. for each requirement that is specified, a corresponding scenario should be there). 5 Implementation of the front-end

of the system Implementation of the main screen giving the login, screen that follows the login giving various options, screens for each of the options (application form, cancellation form etc). 10-12 During this milestone period, it would be a good idea for the team (or one person from the team) to start working on a test-plan for the entire system. This test-plan can be updated as and when new scenarios come to mind. 6 Integrating the front-end with the database The front-end developed in the earlier milestone will now be able to update the database. Other features like mail notification etc should be functional at this stage. In short, the system should be ready for integration testing. 12-13 7 Integration Testing The system should be thoroughly tested by running all the test cases written for the system (from milestone 5). 14-15 Another 2 weeks should be there to handle any issues found during testing of the system. After that, the final demo can be arranged. 8 Final Review Issues found during the previous milestone are fixed and the system is ready for the final review. 16-18 During the final review of the project, it should be checked that all the requirements specified during milestone number 1 are fulfilled (or appropriate reasons given for not fulfilling the same) Guidelines and References http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnasp/html/asptutorial.asp (ASP tutorial) http://www.functionx.com/sqlserver/ (SQL-server tutorial) http://heather.cs.ucdavis.edu/~matloff/UnixAndC/Unix/CShellII.html (Shell script introduction) Students Kit Objective These guidelines are for the student to adopt to make progress in the project. Given below are the templates for the documents related to the project. These are just guidelines only. These can be improved by the team. Requirements Specification (RS) Following is a template for the RS document. Some example requirements are entered in to it to show how to use the template. Make sure that you enter even the smallest/most trivial requirements also. That would help in validating the system during testing. No. Requirement Essentialor 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. The leave policy should also be part of the help. 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 Listing of movies theater-wise and time-wise should be there. Essential The listing is because we don’t want to show all the details in one screen and confuse the user. Admin interface should be given to update the lists daily. RS5 Database Fields Specification A database of users should be maintained for the site. This will help in tracing the patterns of the audience (like which kind of movie people like in the evening times, etc). User Id is the key of the database. The range of valid values entered below as examples need not be taken as such. They can be modified by the team. No. Field Name Range of valid values for the field Remarks 1 User Id Up to 20 characters in length (including special characters) This is selected by the user while registering himself. 2 Name Up to 30 characters in length. Special characters like underscore are not allowed. 3 Email Address Up to 60 characters in length (including the domain name) This detail is to send mail notifications to the user 4 Password Up to 20 characters This field should be encrypted in different algorithm than the login. With other restrictions as special characters of certain kind should be there in the application. 5 History This is a pointer to an array containing the past movies that the user had seen. This history will be used to send him mails if his kind of movie comes up in the theater 6 High Level/Detailed Design (HLD/DD) {mosgoogle} Overview of the system Provide a block diagram depicting where the database will be located, where the application will run etc. Also, provide details about the database server that is going to be used etc. Design Components Split the system into its design components. In this case, the components would be user-verification, mail notification, report generation, reservation, cancellation, approval 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. Pseudocode Pseudocode is written to get more clarity on the component so that the actual implementation is made easier. For the user-verification component: Bool verify user (login_id, password1) { % gets the login_id (which is the login) and the password from the user. Get_login_and_password (); % access the database entry for this If get_database_entry (login_id, database_entry) { % gets the encrypted password. Get_encrypted_password (login_id, password2); % decrypts the password. T decrypted password is password3. Decrypt_password(password2, password3); % compare the passwords. If compare_passwords (password1, password3) { % enter in to the system. Enter_system (); } else % pass comparison failed. Report_error(‘incorrect password. Try again.’); } else % unable to get the database entry report_error (‘invalid login’); } Component two Component three .. Test-Plan (TP) The test-plan is basically a list of testcases that need to be run on the system. Some of the testcases can be run independently for some components (report generation from the database, for example, can be tested independently) and some of the testcases require the whole system to be ready for their execution. It is better to test each component as and when it is ready before integrating the components. It is important to note that the testcases cover all the aspects of the system (ie, all the requirements stated in the RS document). No. Test case Title Description Expected Outcome The requirement in RS that is being tested Result 1 Successful User Verification The login to the system should be tried with the login assigned by the admin and the correct password Login should be successful and the user should enter in to the system RS1 Passed 2 Unsuccessful User Verification due to wrong password Login to the system with a wrong password Login should fail with an error ‘Invalid Password’ RS1 Passed 3 Unsuccessful User

Verification due to invalid login id Login to the system with a invalid login id Login should fail with an error ‘Invalid user id’ RS1 Passed 4 Listing Listing of movies theater-wise and show-wise When a particular theater is selected, the movies on show in that theater should show up. RS4 Passed 5 {mosgoogle}