

Guidelines for MCA students for their Final Year Project

STUDENT Project Kit

(The project from functional point of view)

SPK—Page 1

Title of the project

Development of a feature-rich, practical online leave management system (LMS)

Abstract of the project

This project is aimed at developing an online leave management system that is of importance to either an organization or a college . The Leave Management System (LMS) is an Intranet based application that can be accessed throughout the organization or a specified group/Dept. This system can be used to automate the workflow of leave applications and their approvals. The periodic crediting of leave is also automated. There are features like email notifications, cancellation of leave, automatic approval of leave, report generators etc in this system.

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 functionalities that you find appropriate can be added to this list. And, in places where the description of a functionality is not adequate, you can make appropriate assumptions and proceed.

There are registered people in the system. Some are approvers. An approver can also be a requestor. In an organization, the hierarchy could be Engineers/Managers/Business Managers/Managing Director etc. In a college, it could be Lecturer/Professor/Head of the Department/Dean/Principal etc.

Guidelines for MCA students for their Final Year Project

STUDENT Project Kit

(The project from functional point of view)

SPK—Page 2

1. 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
 - see his/her eligibility details (like how many days of leave he/she is eligible for etc)
 - query the leave balance
 - see his/her leave history since the time he/she joined the company/college
 - apply for leave, specifying the from and to dates, reason for taking leave, address for communication while on leave and his/her superior's email id
 - see his/her current leave applications and the leave applications that are submitted to him/her for approval or cancellation
 - approve/reject the leave applications that are submitted to him/her
 - withdraw his/her leave application (which has not been approved yet)
 - cancel his/her leave (which has been already approved). This will need to be approved by his/her Superior
 - get help about the leave system on how to use the different features of the system
2. As soon as a leave application /cancellation request /withdrawal /approval /rejection /password-change is made by the person, an automatic email should be sent to the person and his superior giving details about the action
3. The number of days of leave (as per the assumed leave policy) should be automatically credited to everybody and a notification regarding the same be sent to them automatically

Guidelines for MCA students for their Final Year Project

STUDENT Project Kit

(The project from functional point of view)

SPK—Page 3

4. An automatic leave-approval facility for leave applications which are older than 2 weeks should be there. Notification about the automatic leave approval should be sent to the person as well as his superior
5. A summary report of the leave details of his/her sub-ordinates should be sent to every manager periodically
6. A calender giving the public holidays of the organization/college should be available on the system

Steps to start-off the project

There are couple 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 (eg Postgress or Oracle or even flat files) , and tools in Unix

The following steps will be helpful to start off the project.

1. Study and be comfortable with technologies such as
 - a. Active Server Pages/HTML and SQL server.
 - b. 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

2. Decide on a leave policy (ie, the different types of leave such as earned leave, medical leave etc, the number of days of leave that the employees at different levels are eligible to, etc) and define it formally.
3. Make a database of people at different levels with their roles and form a hirearchy of them, like which role reports to which particular role. Decide on the various details of the people and their roles that would be stored in the database (like employee/registration-number, name, grade, location, system-login, password in cryptic form, etc)

Guidelines for MCA students for their Final Year Project

STUDENT Project Kit

(*The project from functional point of view*)

SPK—Page 4

4. Assign a mail-admin who will create mail-ids for the people in the intranet of your lab or in the internet. These mail-ids will be used for sending automatic notifications and reports. The mail-admin will also take care of assigning the logins to the users of the leave system
5. 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.
6. Create the front-page of the leave system giving a brief description about the system and a login box
7. Create the help-pages of the system in the form of Q&A. This will help you also when implementing the system
8. Create other sub-systems like automatic notification, screens for various functions (like apply, reject, cancel, withdraw etc)

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.

Guidelines for MCA students for their Final Year Project

STUDENT Project Kit

(The project from functional point of view)

SPK—Page 5

Milestones and Timelines

(The Timeline is given in week number from the start of the project.)

No.	Milestone Name	Milestone Description	Timeline Week no.	Remarks
1	Requirements Specification	Complete specification of the system (with appropriate assumptions) including the framing of leave 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 functionalities 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 of atleast 100 entries of employees of all grades should be created. The number of mail-ids to be created need not be 100. It can be around 10 to 20.	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 leave application, approval, rejection, cancellation, automatic credit etc) and then coming up with flow-charts or pseudocode to handle the scenario.	7-9	The scenarios should map to the requirement specification (ie, 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	10-12	During this milestone period, it would be a good idea for the team (or one

Guidelines for MCA students for their Final Year Project

		the login giving various options, screens for each of the options (application form, cancellation form etc).		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 employee leave 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 testcases 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/CShellIII.html> (Shell script introduction)

-----Last Page (Page # 6) of the Student project Kit-----

Guidelines for MCA students for their Final Year Project

Faculty Kit

(The project from faculty point of view: Determining the evaluation process & deadlines)

FK-Page 1

The faculty kit contains the evaluation strategy for the different milestones of the project and any other documents/links that may aid in the evaluation process (like sample quizzes on technologies etc)

Evaluation Strategy/Tips for the different milestones of the project

Objective

These guidelines are meant for the assessor of the project to check the progress and health of the project. Given below are some tips for evaluation at each of the important milestones of the project. These are in addition to the evaluation process that you may have in assessing the project.

Requirements Specification

Some key points that need to be evaluated on the Requirements Specification document are:

1. Clarity of the specification document
2. Validity of the assumptions made
3. Understanding of the same by all the team-members
4. Quality of the presentation

Technology Familiarization

The evaluation of this milestone can be done through a presentation by the team on the technologies that are going to be used in implementing this project. This can be followed by a quiz on those topics to evaluate the team's understanding.

Database Creation

A small write-up on the database, giving the fields, explaining each field etc should be written. This write-up can be evaluated on the following points.

1. Clarity and conciseness of the database design. Like, whether key is defined, whether any redundant fields are there etc.
2. Whether data-storage calculations are made, and if so, whether they are done properly etc.
3. Whether any data-backup/recovery mechanism is discussed or being thought of.

Guidelines for MCA students for their Final Year Project

Faculty Kit

(The project from faculty point of view: Determining the evaluation process & deadlines)

FK-Page 2

High-Level and Detailed Design

A presentation on the design followed by a viva on the same or the submission of the design document can be used for evaluating this milestone. Some points to look for in the design are:

1. Whether the design covers all the requirements stated in the Requirements-Specification document. And, if not, whether any valid reasons are given for not covering some requirements.
2. Whether the pseudocode or the flow-charts given are detailed enough to proceed with implementation.
3. Whether any alternative designs are being discussed.
4. Whether the different error messages/notification formats are given in a precise form.
5. Robustness of the design (like, what would happen if the database server goes down when a user is entering a leave application)

Front-end implementation

A demo of the front-end can be given by the team for evaluation. Features like mail notification, actual updation of the database etc come in the next stage. This stage is for the front-end only.

Evaluation can be based on parameters like,

1. The look-and-feel of the user interface, like whether there are too many boxes/links in one page, whether any sound-effects are there (for successful/unsuccessful logins, for example) etc
2. The quality of help available for the user, like whether the help instructions match the actual implementation, whether they are simple to understand etc
3. The intuitive-ness of the design, like whether you are able to use the application without spending too much time in reading help screens etc
4. Appropriate, meaningful and non-confusing error messages where applicable etc

Integrating the front-end with the database

A full demo showing all the features of the system should be given at this stage. The stability and robustness of the application should be evaluated at this stage.

Guidelines for MCA students for their Final Year Project

Faculty Kit

(The project from faculty point of view: Determining the evaluation process & deadlines)

FK-Page 3

Some points to look for at this stage are,

1. Whether the demo progresses without any screen/database crashing down or hanging for ever
2. Whether all the major features are demonstrated convincingly and without any major confusion
3. Whether the team is able to create a new login for you and allow you to do operations like logging in, applying for leave, etc
4. Whether the team is able to remove your login and demonstrate that it is indeed removed from the database

Test-plan review

The test-plan document will be reviewed for its completeness/correctness and clarity. Some points to look for are,

1. Whether all the requirements specified in the RS are getting tested for their functionality
2. Whether all the testcases have a description on how exactly to execute the testcase
3. Whether all the possible error scenarios (like, invalid password, etc) are getting tested
4. Whether all the possible exception scenarios (like, the database going for a shut-down operation, the leave-application screen closing down in the middle of the operation, etc) are getting tested

Final review

Final demo and the project report are used for evaluation. The intermediate presentations, documents and write-ups can also be submitted for evaluation.

Documents/References that may aid the process of evaluation

1. www.w3schools.com/asp/asp_quiz.asp (Quiz on ASP technology)
2. http://www.sql-server-performance.com/articles_performance.asp (Quiz and articles on SQL server)

Guidelines for MCA students for their Final Year Project

Students Kit

(The project from students point of view: Execution of the project)

SK- Page 1

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 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	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.	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.

Guidelines for MCA students for their Final Year Project

Students Kit

(The project from students point of view: Execution of the project)

SK- Page 2

Database Fields Specification

Employee Number/Registration Number 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	Employee Number/Registration Number	1 to 1000	This is the key field of the database as it is unique for an employee in the organization. This will also serve as the login for the system.
2	Name	Up to 15 characters in length.	Special characters like underscore are not allowed.
3	Role	Pre-defined set (like engineers/managers/etc)	The reporting hierarchy is based on the role of the person. For example, an engineer reports to a manager, a manager reports to a business manager etc
4	Email Id	Up to 25 characters in length (including the domain name)	This field should also be unique for a person because no two employees in an organization can have the same email id.
5	Superior's Employee Number/Registration Number	1 to 1000	This is the employee number/registration number of the superior of this employee. Other details about the manager can be found in this same database by using the employee number as the key. If this field is zero, it means the employee is at the highest level in the organization (MD/CEO etc)

Guidelines for MCA students for their Final Year Project

Students Kit

(The project from students point of view: Execution of the project)

SK- Page 3

High Level/Detailed Design (HLD/DD)

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, application, 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 (emp_no, password1)
{
    % get the emp_no (which is the login) and the password from the user.
    Get_login_and_password();

    % verify if this is a valid login (ie, from 1 to 1000).
    If login_id_valid(emp_no)
    {
        report_error('invalid login id');
        return false;
    };

    % access the database entry for this
    if get_database_entry(emp_no, database_entry)
```

Guidelines for MCA students for their Final Year Project

```
{
% get the encrypted password.
Get_encrypted_password(emp_no, password2);

% decrypt the password. The decrypted password is password3.
Decrypt_password(password2, password3);
% compare the passwords.
If compare_passwords (password1, password3)
{
    % enter in to the system.
    Enter_system();
}
else % password comparison failed.
    Report_error('incorrect password. Try again.');
```

```
}
else % unable to get the database entry
    report_error ('invalid login');
}
```

Component two

Component three

Guidelines for MCA students for their Final Year Project

Students Kit

(The project from students point of view: Execution of the project)

SK- Page 5

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.	Testcase 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