SOFTWARE REQUIREMENTS SPECIFICATION

FOR EMPLOYEE MANAGEMENT SYSTEM

Introduction

Every organization, whether big or small, has human resource challenges to overcome. every organization has different employee management needs; therefore, we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with the right level of human resources for your future goals. Also, for those busy executive who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all times.  These systems will ultimately allow you to better manage resources. One of the main features in employee management system is time tracking for employees. Effective time tracking mechanism saves both time and money for the organization.

Purpose Of the System

As we know in any technical educational office, the manager of office used to spare lot of time even after the normal office hours either at home or office for preparation of daily/weekly report and other necessary record. Now with the help of this system, the manager has the information on his finger tips and can easily prepare a record based on their requirements apart from daily/weekly report. Finally, we can say that this system will not only auto-mate the process but save the valuable time of the office manager, which can be well utilized by this institute. This will be an additional advantage and management of manpower based on their free time from his normal duty.

Benefits Of This System

* This system will reduce the complexity of employee management. By using this system, we can easily maintain all the records about” ON EMPLOYEES” or “OFF EMPLOYEES”.
* It will reduce searching time.
* It can be easily handled by the person who have elementary knowledge of computer because it provides a user-friendly environment.
* It’s hardware and software configuration is not very costly that means the hardware and software requirement for this software/project are not very costly.

Tools, Platform And Languages To Be Used

Server-side technologies

Spring Boot - 2.0.5.RELEASE

JDK - 1.8 or later

Spring Framework - 5.0.8 RELEASE

Hibernate - 5.2.17.Final

Spring Data JPA - 2+

Front end technologies

REACT JS

Bootstrap 4

npm- 6.9.0

JQuery

Node js 10+

Tools

Maven -

IDE - Eclipse or Spring Tool Suite (STS) // Spring boot API development

Visual Studio 2022 // Angular App development

The feasibility study of this project has revealed the project as follows: -

ECONOMIC FEASIBILITY

The project has shown the economic feasibility by the study of the fact that by using this software the increased number of the consumers can be given service effectively and efficiently and can save a lot time and saving time means saving money. The cost and benefit analysis has shown that cost that have incurred in developing the project is less than the benefits that the project is going to provide once it is developed, so this project has passed the feasibility test.

BEHAVIOURAL FEASIBILITY

The working staff members are also interested in this project, as it will help them to do work with ease and efficiently without complexity, so they supported the development of this project with full enthusiasm. This shows the behavioral feasibility of the project.

TECHNICAL FEASIBILITY

Technical feasibility centers on the existing computer system (Hardware, Software etc) and to what extent it supports the existing system. As the existing system computer system is viable so there is no matter of technical feasibility that is the system is technically feasible.

TIME FEASIBILITY

It is the determination of whether a proposed project can be implemented fully within stipulated time frame. The project was decided to be done in three months and was thought to be feasible enough.

Requirement Analysis

The aim of requirement analysis is to understand the exact requirement of the customer and to document and to document them properly. Requirement analysis involves obtaining a clear and thorough understanding of the product to be developing with a view to remove all ambiguities and inconsistencies from the initial customer perception the problem.

The question arising during the requirement analysis phases is: -

What is the problem?

Why is it important to solve the problem?

What are the possible solutions to the problem?

What exactly are the data inputs and data outputs by system?

What are the likely the complex cities that might arise while solving the problem?

During requirement analysis there exist mainly two activities.

Requirement gathering

Analysis of gathered requirements.

Requirement gathering: - This involves interviewing the end user and customers to collect all possible information regarding the bank.

Analysis of gathered requirement: - The main purchase of analysis is to collect information to clearly understand the exact requirement of customer and resolve anomalies, conflicts and inconsistencies in the gathered requirement.

Software Life Cycle Adopted

In order to develop the project “Employee Management” we have adopted the iterative enhancement model. This model removes the shortcoming of waterfall model. Since many facts of this system are already known. It is not a new concept and hence no research is required. A working version can be easily created and hence the system can start working. Rest of the functionalities can be implemented in the next iteration and can be delivered later. As the requirement analysis is also not required. It not being a new technology risk involved is also less. So, one need not perform detailed risk analysis. If redevelopment staff is less than development can be started with a smaller number of people and in next increments others can be involved. As this model combines the advantage of waterfall model and prototyping, clients are always aware of the product being delivered and can always suggest changes and enhancements and can get them implemented. As less amount of customer communication is required one need not apply spiral model in which all types of analysis is done in detail. As the deadline is affordable one need not to for Rapid Application Development model. Iterative enhancement model is useful when less manpower is available for software development and the release deadlines are specified. It is best suited for in house product development, where it is ensured that the user has something to start with. The complete product is divided into releases and the developer delivers the product release by release.

* Requirement Analysis
* Coding
* Testing
* Maintenance
* Design

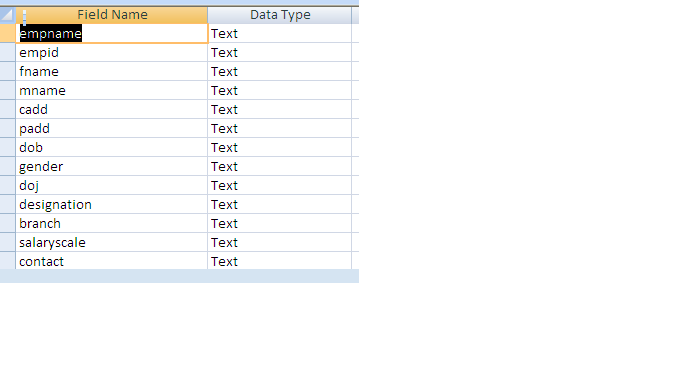
Delivery of first increment

Thus, increments are developed till final product is not obtained.

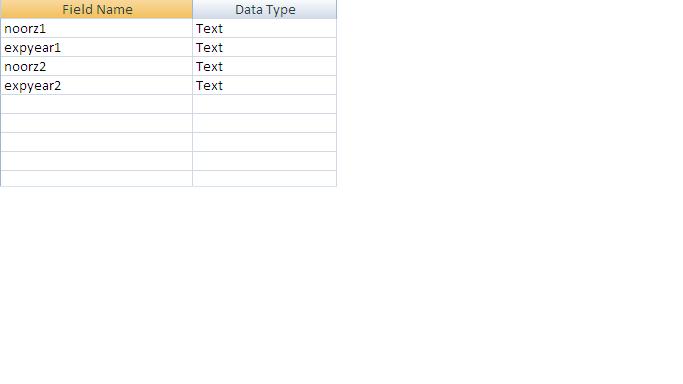
Database

Data Base Design

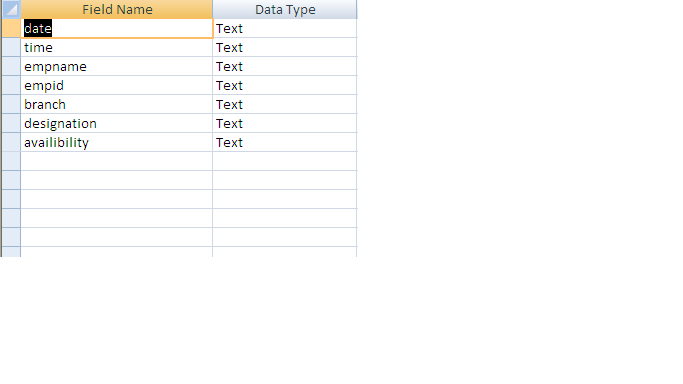
Table1



Attendance table :-



Attendance Table:



About Pages Used In The Project

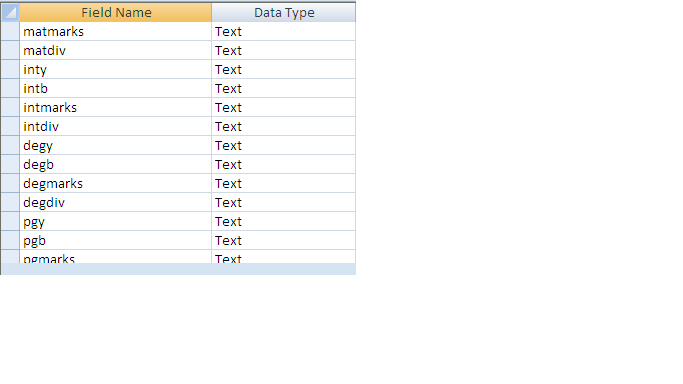
HOME PAGE:- It is the front page where all the buttons are present, They are:

Record entry

View

Daily attendance

Exit



SELECTION PAGE:-It is the page where we have to select, Whom record we want to view ON emp record or OFF emp record.

RECORD PAGE:- It is the page where records of all the employees are present.if it is ON employee record page it contains record about ON employees and if it is OFF employee page it contains record about OFF employees.

SEARCH FRAME:- It is the frame where we have to enter emp name and empid for get more information about a particular employee. After entrying name and id, we can know-

* Employee permanent record
* Employee salary details
* Employee attendance record

DAILY ATTENDENCE PAGE: - It is the page by which we enter daily attendance of all working employees.

EMPLOYEE PERMANENT RECORD PAGE: - It is the page where all the permanent records about an employee is present.such as- Emp name ,Qualification,Experience etc.

EMPLOYEE SALARY DETAILS: - I t is the page where salary details about a particular employee is present. Such as salary scale,Basic,H.R.A.,D.A etc.

EMPLOYEE ATTENDENCE RECORD: - It is the page where attendence record of a particular employee is present. Such as- no. of attendance in current month.

Testing

This phase determines the error in the project. If there is any error then it must be removed before delivery of the project. For determining errors various types of test action are performed.

Unit Testing: -

Unit testing focuses verification effort on the smallest unit of software design – the module. Using the detail design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and the errors detected as a result is limited by the constrained scope established for unit testing. The unit test is always white box oriented, and the step can be conducted in parallel for multiple modules.

Unit testing is normally considered an adjunct to the coding step. After source level code has been developed, reviewed, and verified for correct syntax, unit test case design begins. A review of design information provides guidance for establishing test cases that are likely to uncover errors. Each test case should be coupled with a asset of expected results.

Because a module is not a stand-alone program, driver and/or stub software must be developed for each unit test. In most applications a driver is nothing more than a main program that accepts test case data passes such data to the module (to be tested), and prints the relevant results. Stubs serve to replace modules that are subordinate (called by) the module to be tested. Stub or “dummy subprogram” users the subordinate module’s interface, may do minimal data manipulation, prints verification of entry and returns.

Drivers and stubs represent overhead. That is, both are software that must be written but that is not delivered with the final software product. If drivers and stubs are kept simple, actual overhead is relatively low. Unfortunately, many modules cannot be adequately unit tested with “simple” overhead software. In such cases, complete testing can be postponed until the integration test step.

Unit testing is simplified when a module with high cohesion is designed. When only one function is addressed by a module, the number of test cases is reduced and errors can be more easily predicted and uncovered.

System Testing: -

Software is only one element of a larger computer-based system. Ultimately, software is incorporated with other system elements (e.g. new hardware, information), and a series of system integration and validation tests are conducted. Steps taken during software design and testing can greatly improve the probability of successful software integration in the larger system.

A classics system testing problem is “finger pointing”. This occurs when a defect is uncovered, and one system element developer blames another for the problem. Rather than including in such nonsense, the software engineer should anticipate potential interfacing problems and (1) design error handling paths that test all information coming from other elements of the system.(2) conduct a series of tests that simulate bad data or other potential errors at the software interface; (3) record the results or tests to use as “evidence” if finger pointing does occur (4) participate in the planning and design of system test to ensure that software is adequately tested.

There are many types of system tests, which are worthwhile for software-based systems, as detailed hereunder:

Recovery testing is a system test that forces the software to fail in a variety of ways that verifies that recovery is properly performed.

Security testing attempts to verify that protection mechanisms built into a system will protect it from improper penetration

Stress tests are designed to confront programs with abnormal situations.

Performance testing is designed to test the run-time performance of software within the context of an integrated system.

Integration Testing: -

A neophyte in the software world might ask a seemingly legitimate question once all modules have been unit-tested. If they all work individually, why do you doubt that they’ll work when we put tem together? The problem, of course, is putting them together – interfacing. Date can be lost across an interface; one module can have an inadvertent, adverse effect on anther, sub functions, when combined, may not produce the desired major function; individually acceptable imprecision may be magnified to unacceptable levels; global data structures can present problems. Sadly, the list goes on and on.

Integration testing is a systematic technique for construction the program structure while at the same time conduction test to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by design. There is often a tendency to attempt non-incremental integration; that is, to construct the program using a big bang approach. All modules are combined in advance. The entire program is tested as a whole. And chaos usually results! A set of errors are encountered. Correction is difficult because the isolation of causes is complicated by the vast expanse of the entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop.

Incremental integration is the antithesis of the “big bang” approach. The program is constructed and tested is small segments, where errors are easier to isolate and correct; interfaces are more likely to be tested completely, and a systematic test approach may be applied.

Integration testing can be categorized into two types, namely top-down integration or bottom-up integration. Top-down integration is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main control module. Modules subordinate to the main control module are incorporated into the structure in either a depth-first or breadth-first manner. The bottom-up integration testing as its name implies, begins construction and testing with atomic modules. Because modules are integrated for the bottom-up processing required for modules subordinate to given level is always available and the need for stubs is eliminated.

The selection of an integration strategy depends upon software characteristic and, sometime project schedule. In general, a combined approach that uses the top-down strategy for the upper levels of the program structure, coupled with a bottom-up strategy for the subordinate levels, may be the best compromise.

Scope Of Future Application

This software can be able to support internetworking with the little advancement in the coding. Then any user can upload the date to the school website and can view the data and all reports online from any part of the world. This can also be connected strongly with the internet, even if management wants, parents can view their child’s record through an attractive and graphic rich website. They can also make their child’s fee online.

This software can also be equipped with strong backup facilities to protect the important data and hence preventing any sort of problem which might occur due to lost of data.