**Report**

**Of**

**DJMS**

**By Manik Vaid**

**INDEX**

|  |  |  |
| --- | --- | --- |
| S NO. | CONTENT | PAGE NO |
| 1 | **Abstract** | **3** |
| 2 | **Introduction** | **4-5** |
| 3 | **System Analysis** | **6-7** |
| 4 | **Feasibility Study** | **8-10** |
| 5 | **System Requirement Specification** | **11-18** |
|  | **5.1 SDLC** | **15-16** |
|  | **5.2 Model** | **17-18** |
| 6 | **System Design** | **19-23** |
|  | **6.1 UML Diagrams** | **21** |
|  | **6.2 Data Flow Diagrams** | **22** |
|  | **6.3 Data Dictionary** | **23** |
| 7 | **Screen Shots** | **24-27** |
|  |  |  |
|  |  |  |

**ABSTRACT**

**Objective**

The DJMS System is the web application used by the Partners of the business by login into the application with a valid username and password. Then Event Bookings can be noted with proper details filled into the Agreement Form. Has features like Event Booking Agreement Form , Registration for the Partner Users Etc.

**Description**

* Registration screen.
* Login Screen.
* Contact information.
* Agreement Form
* Agreement Listing

**Some of the features are**

* Interface to Check all the Event Booking Agreements.
* Interface to Partners User’s

**INTRODUCTION**

DJMS is a web application built in Asp.net as the front end and SQL server as backend for the application.

**Vision**The purpose of “DJMS” is to provide a platform to the Partner user’s of the different organizations to be able to add Agreement details for the Event Bookings Like Marriage Functions, DJ Setups, Parties, Kitties Etc.

**Scope**

The system is used for Event Bookings and Different organizations to make them able to prepare a proper Agreement based upon the information provided by the customer at the time of booking.

**System Analysis**

**Purpose of the System**

The purpose of “DJMS” is to serve as a platform to the different organizations for creating and maintaining Agreement Records against the Event Booking that take up by an organization or event management company.

**How DJMS helps the Existing System**

* Different Events has different setup and arrangements to be taken care it gives proper detail form to add all in the record.
* Agreement Listing Helps to check the Latest and upcoming events that are in the list.
* By customer registration we able multiple no of company employee to add agreement as per customer details for the different events.
* It will help in reporting the correct data in the system against the Events.

**Feasibility Study**

**Economic Feasibility**

Economic feasibility attempts 2 weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system.

A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation.

**Operational Feasibility**

Proposed project is beneficial only if it can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project:

Is there sufficient support for the project from management from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.

Are the current business methods acceptable to the user? If they are not, Users may welcome a change that will bring about a more operational and useful systems.

Have the user been involved in the planning and development of the project?

Early involvement reduces the chances of resistance to the system and in general and increases the likelihood of successful project. Since the proposed system was to help reduce the hardships encountered. In the existing manual system, the new system was considered to be operational feasible.

**Technical Feasibility**

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed design of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis.

Understand the different technologies involved in the proposed system before commencing the project we have to be very clear about what are the technologies that are to be required for the development of the new system. Find out whether the organization currently possesses the required technologies. Is the required technology available with the organization?

**System Requirement Specification**

**Study of the System**

In the flexibility of uses the interface has been developed a graphics concepts in mind, associated through a browser Interface

**No of Modules**

The system after careful analysis has been identified to be presented with the following modules:

The Modules involved are

* User Login
* User Registration
* Event booking as Agreement
* Agreement Listing

**Description for Modules**

* User Login
  1. It add the username
  2. It add the password
  3. It has the Login Button
  4. It has the Link to go to Registration as (Click here to register)
* User Registration

1. It add the username
2. It add the password
3. It add the Mobile
4. It add the Email
5. It has Register Button

* Agreement Form

In this module user will add all the Vital information against the Event booking for the customer such as customer name , booking date, customer address, Venue of the Event , financial info as total amount , advance and balance etc.

* Security and Authentication
  1. User registration
  2. Login as user
  3. No blank record validation
* Agreement Listing

In this module all the agreements showing as in Grid view for the user that is logged in to be able to see the records.

**Software Requirement**

Operating System : (Windows 8/Above) or Linux

User Interface : Asp.net, CSS

Programming Language : Asp.net

Web Applications : Google chrome or any other browser

IDE/Workbench : Visual Studio

Database : SQL server

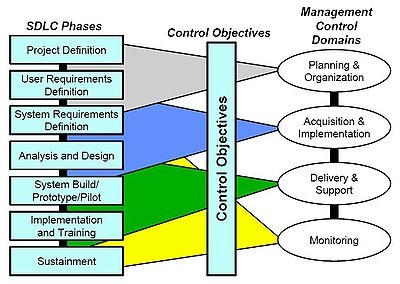
**Hardware Requirements**

Processor : Pentium IV

Hard Disk : 100GB

RAM : 2GB or more

**System Development Life Cycle**



A software development process, also known as a software development life-cycle (SDLC), is a structure imposed on the development of a software product.

Similar terms include software life cycle and software process. It is often considered a subset of systems development life cycle. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process.

Some people consider a life-cycle model a more general term and a software development process a more specific term. For example, there are many specific software development processes that 'fit' the spiral life-cycle model.

Information system development involves various activities performed together. The stages involved during System Life Cycle are:-

* Recognition of need.
* Feasibility study.
* Analysis.
* Design.
* Coding & Testing.
* Implementation.
* Post implementation and maintenance.

THE INCREMENTAL MODEL

Incremental model is an evolution of waterfall model. The product is designed, implemented, integrated and tested as a series of incremental builds. It is a popular model software evolution used many commercial software companies and system vendor. Incremental software development model may be applicable to projects where:

Software Requirements are well defined, but realization may be delayed.  
The basic software functionality is required early.

The incremental model is a method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements. This model combines the elements of the waterfall model with the iterative philosophy of prototyping.

The incremental modelcombines elements of the linear sequential model (applied repetitively) with the iterative philosophy of prototyping. The incremental model applies linear sequences in a staggered fashion as calendar time progresses.

Each linear sequence produces a deliverable “increment” of the software. For example, word-processing software developed using the incremental paradigm might deliver basic file management, editing, and document production functions in the first increment; more sophisticated editing and document production capabilities in the second increment; spelling and grammar checking in the third increment; and advanced page layout capability in the fourth increment.

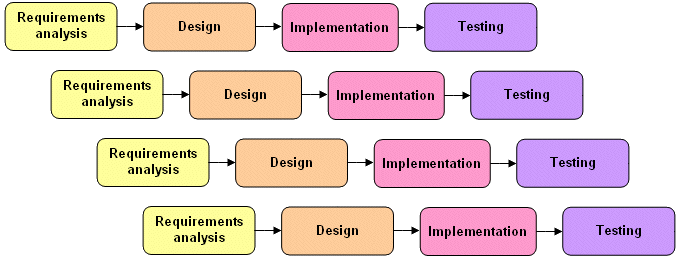
It should be noted that the process flow for any increment can incorporate the prototyping paradigm. When an incremental model is used, the first increment is often a core product*.* That is, basic requirements are addressed, but many supplementary features (some known, others unknown) remain undelivered. The core product is used by the customer (or undergoes detailed review). As a result of use and/or evaluation, a plan is developed for the next increment. The plan addresses the modification of the core product to better meet the needs of the customer and the delivery of additional features and functionality.

This process is repeated following the delivery of each increment, until the complete product is produced. The incremental process model, like prototyping and other evolutionary approaches, is iterative in nature. But unlike prototyping, the incremental model focuses on the delivery of an operational product with each increment.

Early Increments are stripped down versions of the final product, but they do provide capability that serves the user and also provide a platform for evaluation by the user.

Incremental development is particularly useful when staffing is unavailable for a complete implementation by the business deadline that has been established for the project. Early increments can be implemented with fewer people. If the core products well received, then additional staff (if required) can be added to implement the next increment.

The incremental model diagram



**ADVANTAGES OF INCREMENTAL MODEL**

## It generates working software quickly and early during the software life cycle.

## Flexibility is more and less costly.

## Testing and debugging becomes easier during a smaller iteration.

## Risk can be managed more easily because they can be identified easily during iteration.

## Early increments can be implemented with fewer people.

## The versions are provided after each iteration of the incremental model.

## Incremental model includes use of the software by user to for changes.

## It is flexible and easy to manage more manageable process and better software making and better software structure.

## Testing is easy.

## DISADVANTAGES OF INCREMENTAL MODEL

## Each phase of an iteration is rigid and do not overlap each other.

Problems may arise pertaining to system architecture because not all requirements are gathered up front for the entire software life cycle.

**System Design**

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software. The importance can be stated with a single word “Quality”. Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer’s view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

**Authentication**

Authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be. In private and public computer networks (including the Internet), authentication is commonly done through the use of [logon](http://searchSecurity.techtarget.com/sDefinition/0,,sid14_gci212499,00.html) passwords.

**Functional Description**

* + Login to the system through the first page of the application.
  + Able to register the user
  + Fill up the Agreement form detail against the Event booking and Listing screen for the same as well.
  + Help from the system.

**UML Diagrams (Unified Modeling Language**)

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

User Model View

* This view represents the system from the user’s perspective.
* The analysis representation describes a usage scenario from the end-users perspective.

Structural model view

* In this model the data and functionality are arrived from inside the system.
* This model view models the static structures.

Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View

In this the structural and behavioral as parts of the system are represented as they are to be built.

Environmental Model View

In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are:

* UML Analysis modeling, this focuses on the user model and structural model views of the system.
* UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

##### DATA FLOW DIAGRAMS

A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also known as a data flow graph or a bubble chart.

DFDs are the model of the proposed system. The Basic Notation used to create a DFD’s are as follows:

**Dataflow:** Data move in a specific direction from an origin to a destination.

**Process:** People, procedures, or devices that use or produce (Transform) Data. The physical component is not identified.

**Source:** External sources or destination of data entities.

**Data Store:** Here data are stored or referenced by a process in the System.

**DATA DICTONARY**

After carefully understanding the requirements of the client the entire data storage requirements are divided into tables. The below tables are normalized to avoid any anomalies during the course of data entry.

Table 1:- Registration

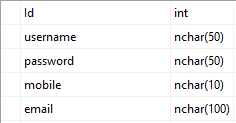
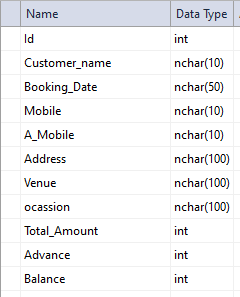
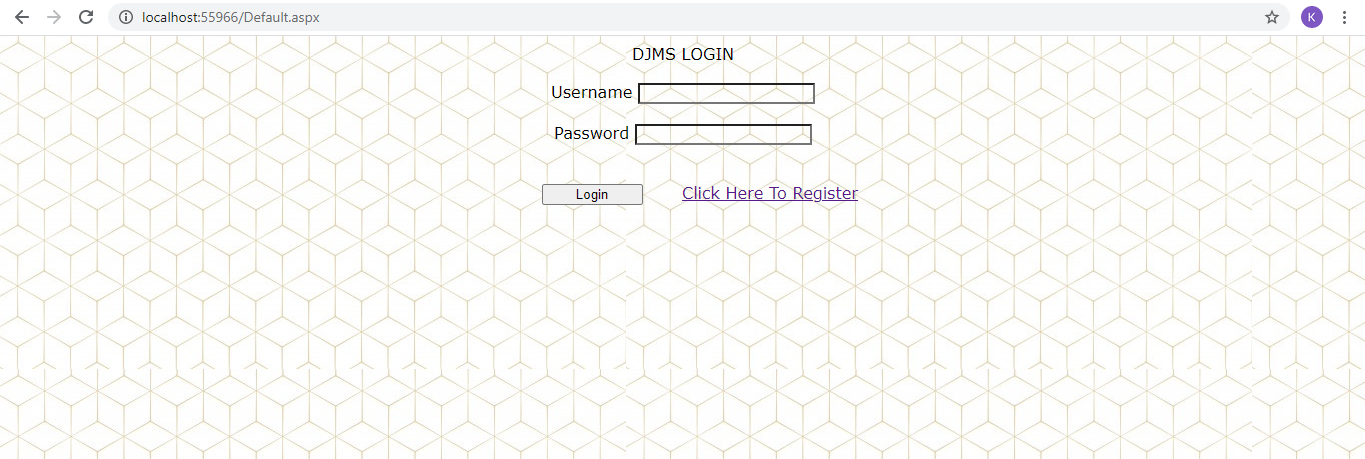


Table 2:- Agreement

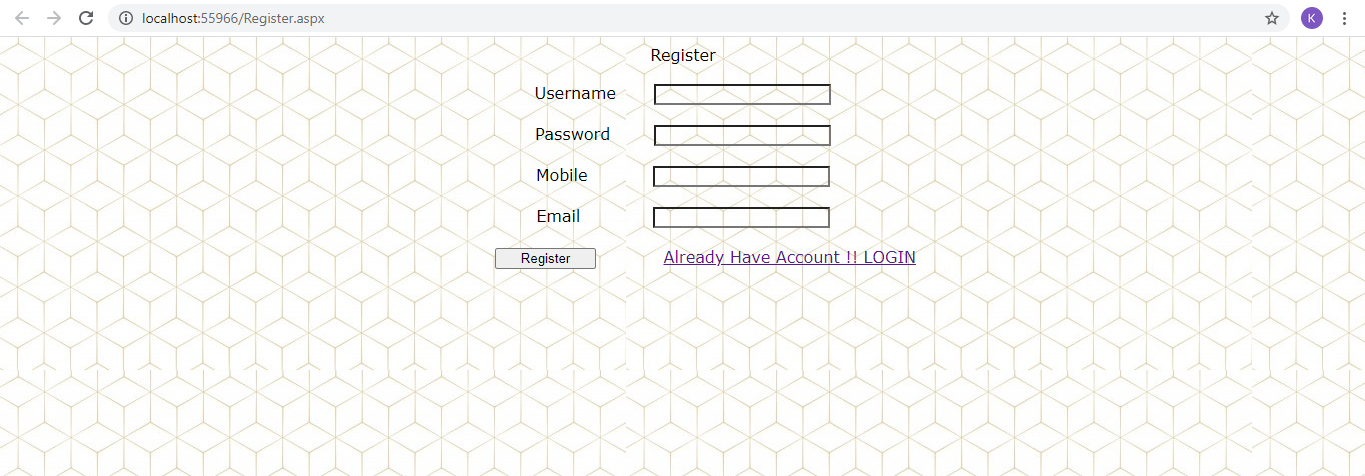


**Screen Shots**

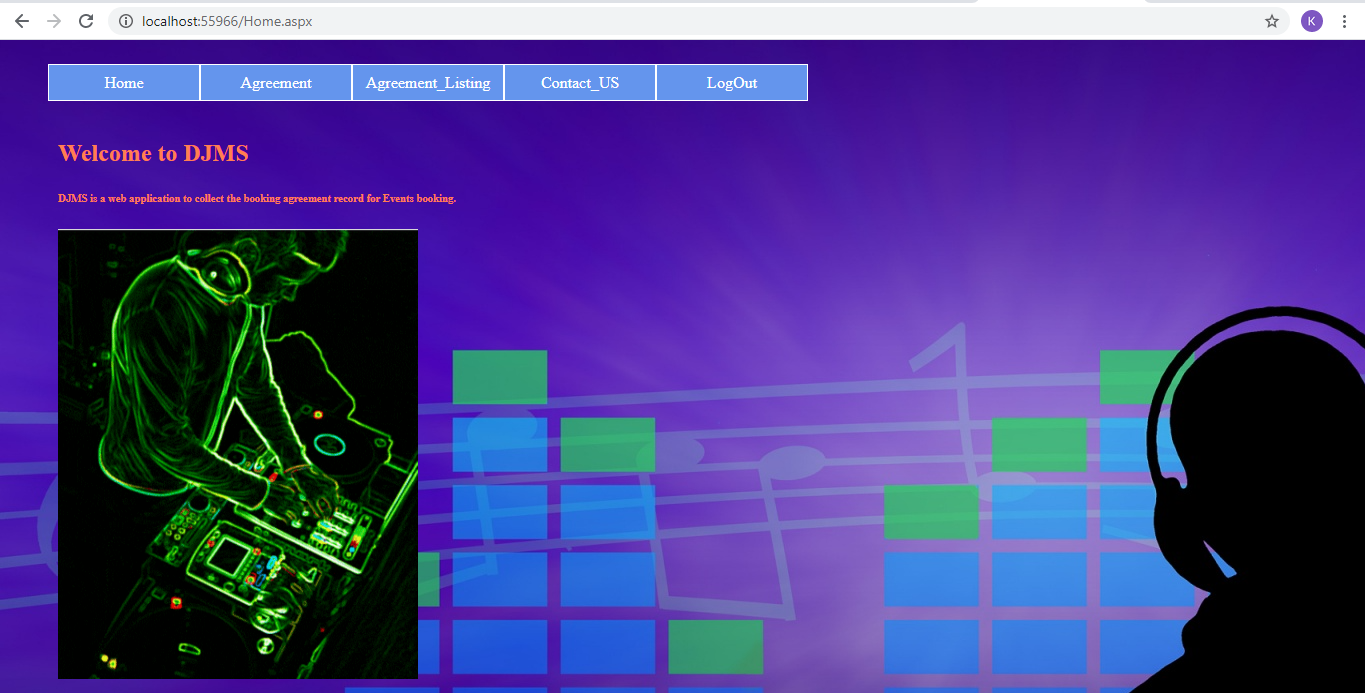
Login Screen:-



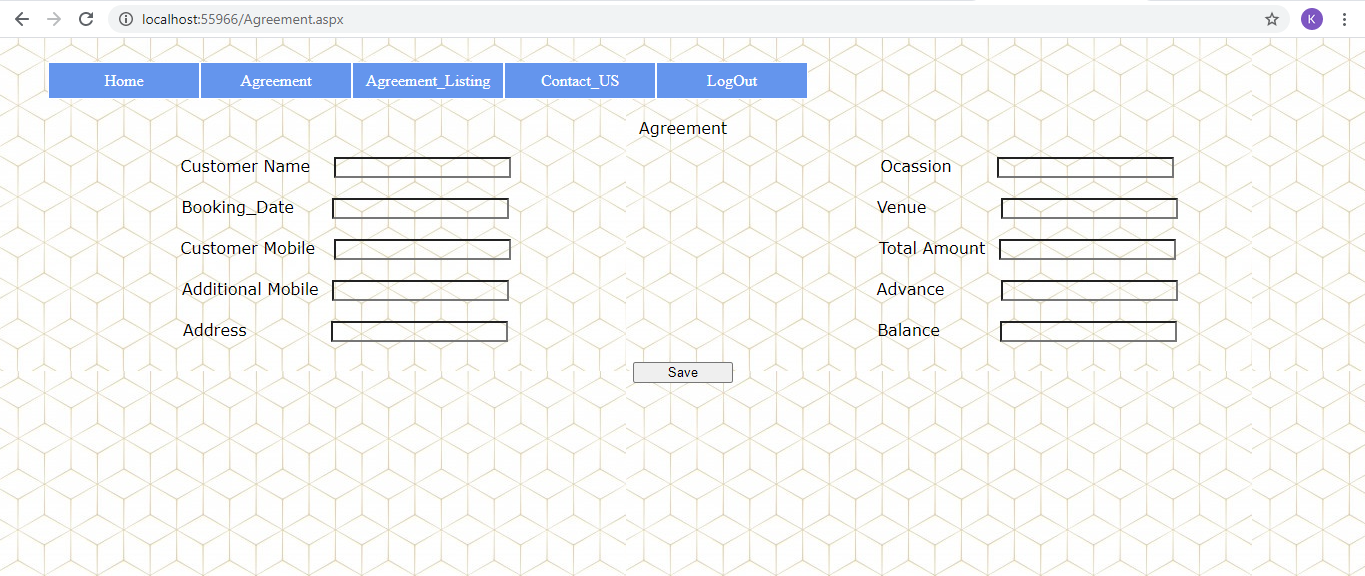
Registration Screen:-



Home Screen :-



Agreement Form Screen:-



Agreement Listing Screen :-



Contact Us Screen :- 