**Abstract**

Realizing the difficulties and needs of the campus, our team tried to make a project named EaseInfy where the key idea was to make our daily lives easier. We analyzed the problem of long ques for ordering the food and booking movie tickets on weekends. There came the idea of digitalizing the entire process by making a platform where we can get all the details of food courts and make the payment using OTP.

The technologies used are ASP.net, HTML, CSS, SQL queries, C# and MVC. The provided platform is Visual Studio for performing and testing and Microsoft SQL Server Management Studio 2017 for the database.

**CHAPTER 1: Introduction**

**1.1 Overview**

EaseInfy is a web platform that is used to show all the details of availability food courts, food vendors and the menu available, the other features include the updation of menu by vendors, addition/deletion of food courts or food vendors. Also the second part comprises of the streaming movies for the weekend, OTP generation for the verification by user and view and selection of the seats available according to the previous booking.

Also the required database is made by the team itself.

**1.2 Problem Statement**

There were numerous problems faced by the people like waiting in long queues for the payment and selection of food, Vendors did not have a proper platform to update menu.

Therefore, the entire idea of developing a Web Application was to get one platform to perform the above mentioned tasks.

The aim was to:

For Food Courts:

1. View all the food courts and its availability.
2. View of the vendors and its menu.
3. Generation of OTP for ordering the food items.
4. For vendors, alteration in menu.

For streaming movies in Multiplex:

1. Prior information about the movies streaming next weekend.
2. Checking the availability of seats for a particular time of the show.
3. Generation of OTP for the verification of user.
4. Booking of tickets.

**1.3 Objective**

* To create a website that has a user friendly Interface.
* To create a website which enables the user to view the list of food courts, food items and vendors to update the food items list.
* To create a website which enables the user to view the movies streaming for the weekend and book the tickets.

**1.4 Methodology**

The following methodologies were used for the implementation of our project.

* Creation of Database using SQL server.
* Creation of Web Application using MODEL VIEW CONTROLLER structure.
* Creation of the DATA ACCESS LAYER to connect the MVC project to database.
* Creation of ENTITY DATA MODEL DESIGNER.
* Using Model classes for the basic structure of Views.
* Use of Controllers and Views to perform the required functionality.
* Use of HTML, CSS, BOOTSTRAP and JAVA SCRIPT to create an interface that is user friendly.
* Performed Unit Testing, Integration Testing using white box approach.

**CHAPTER 2: Literature Reading**

**2.1 Asp.Net.MVC**

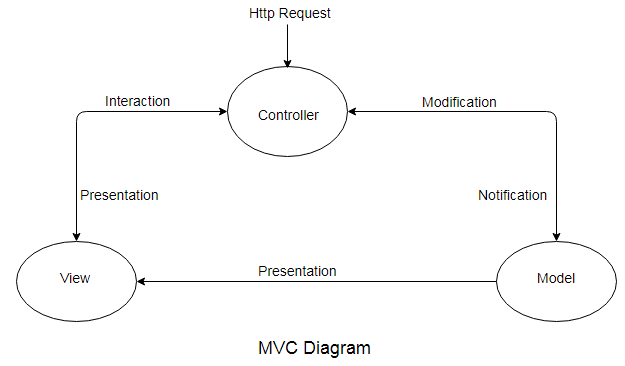
.NET Core is a new Microsoft platform, which is modular, cross-platform, open source and has a better support for cloud

MVC comprises of Models, Controllers and Views. Asp.Net.MVC is a source for creating a web application.

MVC provides loose coupling between different aspects.

Description of the Components of MVC:

1. **Model**: It is used to pass data to Views so that it can be executed further by the concerned. It handles the data related to business.
2. **Controllers**: The functionalities in controller lead to views. It responds to user input and informs the other two parts to make appropriate changes (to display to end user -View part, manage data -Model part)
3. **Views**:  The part of the software that handles what is to be displayed to user (output) and takes data from the Model part.



**Advantages of MVC**

Most of the software applications these days (except a few like embedded systems, time-sensitive apps) contain different parts/components.

It is always better to separate these components so that they can be managed easily.

It reduces the complexity as the applications are divided in the model, view and controllers.

It also gives the user full control over the application.

**Features of the ASP.NET MVC FRAMEWORK**

ASP.NET MVC views (user interface pages) are created using**razor-syntax**, that helps to create dynamic server pages by embedding C# code in html files.

These files are commonly identified with .cshtml extension.

The run time responsible to evaluate these .cshtml pages is known as Razor view engine.

**2.2 HTML**

HTML is a markup language that defines the structure of web pages and determines how content is displayed online.

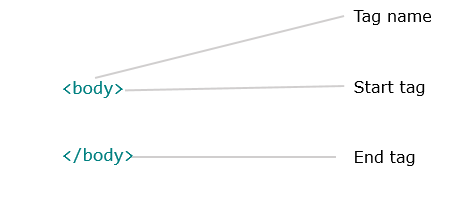
* Hypertext is the text displayed on a computer which refers to other text or image that the user can access
* Markup language is a set of tags which helps the web browser in presenting text in a document
* HTML is case-insensitive.

**Working**

HTML elements are made up of 2 things:

1) Tags - HTML instructions

2) Content - On which the HTML instructions should be applied



**Limitations faced while working with HTML**

It provides a simple and basic layout. Therefore, we use Cascading Stylesheet to add features to the layout of web pages.

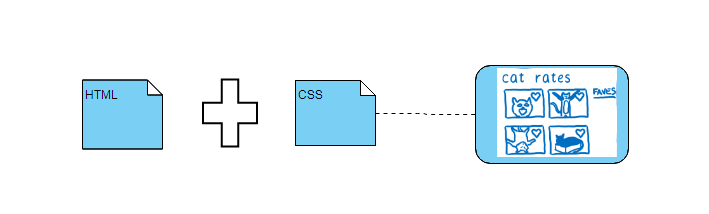
**2.3 Cascading Style Sheet**

CSS (or Cascading Style Sheets) is a language/technology that is used for **defining the presentation of an HTML page**.  
  
It was introduced primarily to separate the page content from the presentation. This provides greater flexibility and control over the styling. It helps in separating the **style** information and the **HTML** content.

**Working**

There are three ways of applying CSS to a web page

1. External CSS - a separate file is created that contains CSS instructions with a file extension of .css. And in the HTML page, this CSS file can be imported to provide styling to the layout.
2. Internal CSS - CSS instructions are placed at the top of the HTML web page document in the head tag of the web page.
3. Inline CSS - These kind of styles can be placed at any place where required in the tag you wish to decorate.



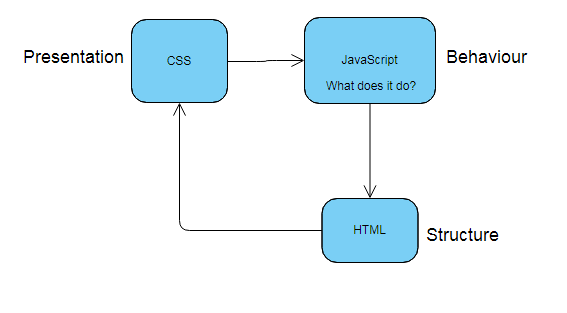
**2.4 Java Script**

The main usage of JavaScript is to make the website dynamic such as selecting from drop down or side navigation buttons. Html and CSS are used to make web site pages in which CSS helps to beautify the page but what about dependent functionalities i.e., open this page when click that such as showing error pages, this all done with the help of JavaScript

It is written with in:

<script>

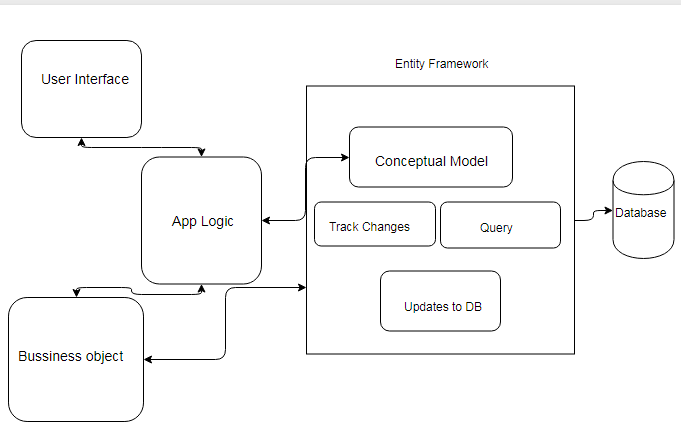
</script>

You can add functions inside the JavaScript.

**2.4 Entity Framework**

Entity Framework eliminates the need for writing a lot of code for implementing the data access logic by executing queries using LINQ syntax.

It is an open-source ORM (Object Relational mapper) framework for .NET applications supported by Microsoft that makes it easy for developers to work with data which is kept in the database without actually using the database.



**CHAPTER 3: PROJECT DEVELOPMENT**

**3.1 Database Implementation**

The below mentioned are the database Tables and their attributes that are used for the project.

1. Employee: Attributes contained by this table are

* User Name
* User Password
* Phone Number
* Date of Birth
* Address

1. Vendor: Table contains attributes

* Vendor Id
* Vendor Password
* Vendor Name

1. FCoupon: Table contains attributes

* Coupon Id
* Item Id
* User Name
* Vendor Id
* Price

1. Food Coupon: Table contains attributes

* Coupon Id
* Item Id
* User Name
* Vendor Id
* Price

1. Food Court: Table contains attributes

* Food Id
* Fc Name

1. Food Item: Table contains attributes

* Item Id
* Name
* Price
* Vendor Id

1. MCoupon: Table contains attributes

* Coupon Id
* Mid
* User Id
* Price

1. Movie Coupon: Table contains attributes

* Coupon Id
* Mid
* User Id
* Price

1. Movie Vendor: Table contains attributes

* Movie Vendor Id
* MPassword

1. Multiplex: Table contains attributes

* Mid
* Movie Name
* Screen No.
* Total Seats
* Price
* At Time1
* At Time2
* Evening Seats
* Morning Seats

1. Rating: Table contains attributes

* Rating Id
* User Name
* Rating Val
* Message
  1. **Data Access Layer**

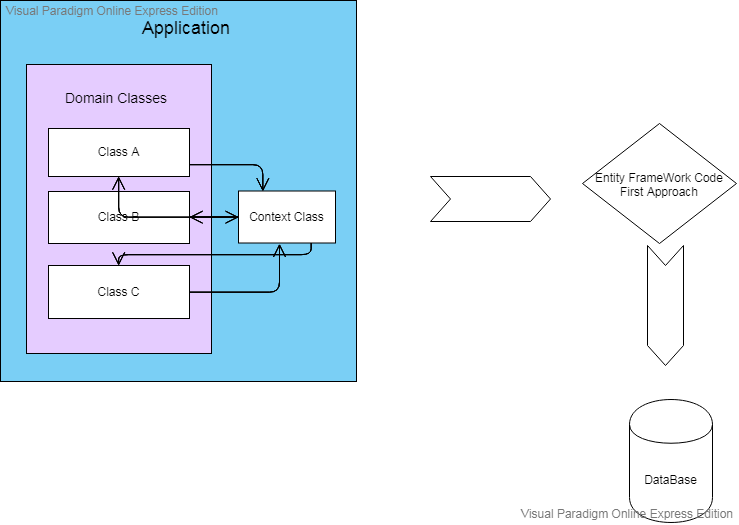
This layer is an intermediate between the MVC Layer and database.

Its main purpose is to help in performing the CRUD operations on database.

CRUD –Create, Read, Update, Delete.

The EaseInfy Repository contains the following functionalities:

* **Get All Food Courts**: It extracts the name all the food courts from database with respect to their availability.
* **View Rating**: It extracts top/most recent five ratings from the database.
* **Get All Vendors**: It extracts the name all the vendors associated.
* **Get All Food Items**: It is specifically for vendors and extracts the name all the food items that the vendor has uploaded.
* **Get Food items**: It is specifically for the employees and extracts the food items that the vendors have mentioned.
* **Get Employee**: It extracts all the details of the registered employees.
* **Get Food List by Name**: The food items ordered by the user.
* **Get all Movies**: Movies that the Vendor has mentioned for the weekend.
* **View Customers**: Extract the details of the customers that have registered themselves.
* **Add Food Item**: The recently added food items by the vendor are extracted.
* **Add FCoupon**: Coupons that can be added by the vendor including itemid, price and vendor id.
* **Bill Amount**: It extracts the bill amount according to the price of the food item mentioned by the vendor.
* **Get Bill**: Extracts the Bill details.
* **Get FCoupons**: It shows the itemid, price and the vendor from which item is purchased.
* **Clear Bill**: The database clears all the details of bill after its generation.
* **Get OTP**: It extracts a random number for every entry.
* **Find OTP**: It finds the OTP in the data set for the input given by the user.
* **Verify OTP**: Checks if the generated OTP is same as that of Random OTP generated.
* **Delete Food Item**: List of food items that are removed from the menu.
* **Validate Employee**: At the time of login database checks the data with the existing data set.
* **Validate Vendor**: At the time of login database checks the data with the existing data set.
* **Validate Movie Vendor**: At the time of login database checks the data with the existing data set.
* **Update Employee**: Any updation of new employee or updation of details of existing employee.
* **Update Food Item**: The food items added/deleted by vendor is extracted.
* **Update Multiplex**: The movie name, seat number, screen number, timings can be updated.
* **Get User**: Users who availed the services from the vendors.
* **Register User**: The basic details for the registration of Users.
* **Save Rating**: Stores the rating done by the customers.
  1. **Design Architecture/System/Sub-System Context Diagram**

****

* 1. **Controllers**

1. **Home Controller**: This controller is responsible to perform credential Validations and login and logout sessions, also the details of the web application regarding what it offers and its contact details also are a part of Home Controller.

The Functionalities included are:

* Login Employee
* Login Vendor
* Login Movie
* About
* Contact
* Register
* Check Employee
* Check Credentials
* Check Vendors
* Check Movie Vendors

1. **Employee Controller**: It controls all the functionalities that the employee as a user would need like selection, purchase, OTP generation and Bill generation.

It consists of the following functionalities:

* + - * + View Rating
        + Rating
        + Save Rating
        + View Food Court
        + View Vendor
        + View Food Items
        + View Movies
        + Select Food
        + Purchase Food
        + Select Movie
        + View Bill
        + Generate OTP
        + Generate Bill
        + Generate Bill Movie
        + Verify OTP
        + Payment Done
        + Print Bill
        + Update Profile
        + Same Updated Profile
        + Logout

1. **Movie Controller**: It controls the updation of movies by the vendor and view of the movies its timings, screen number, seats availability and selection.

The Functionalities included are:

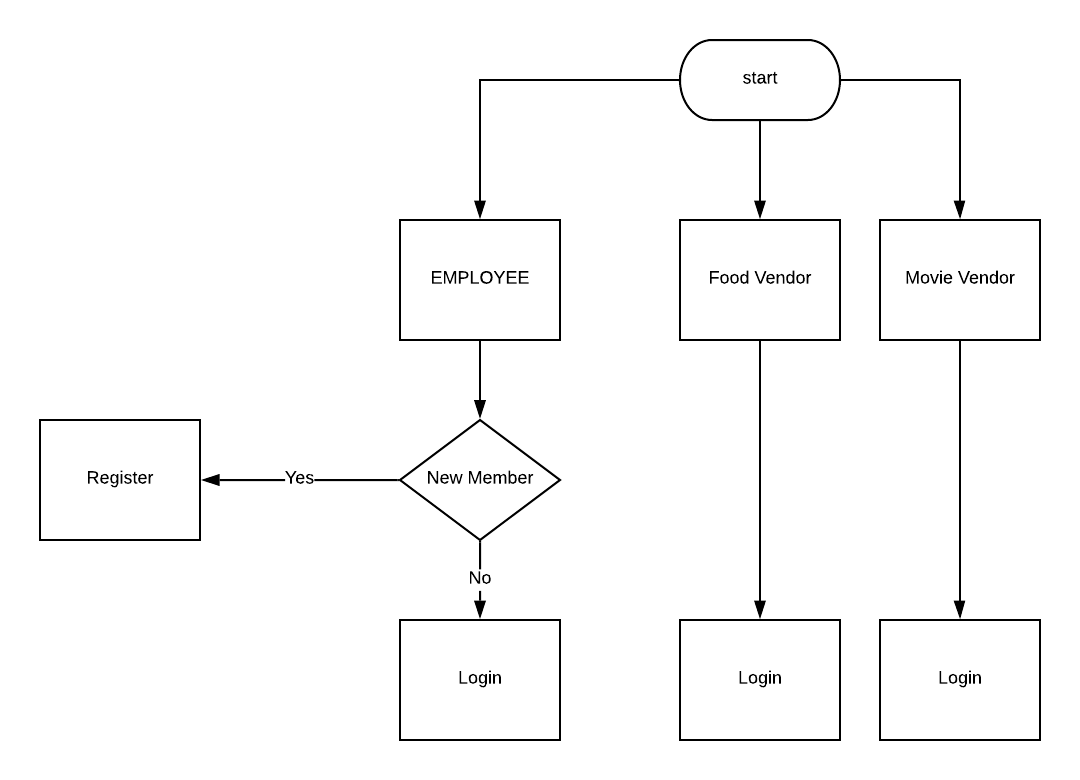
* View Movie
* Update Movie

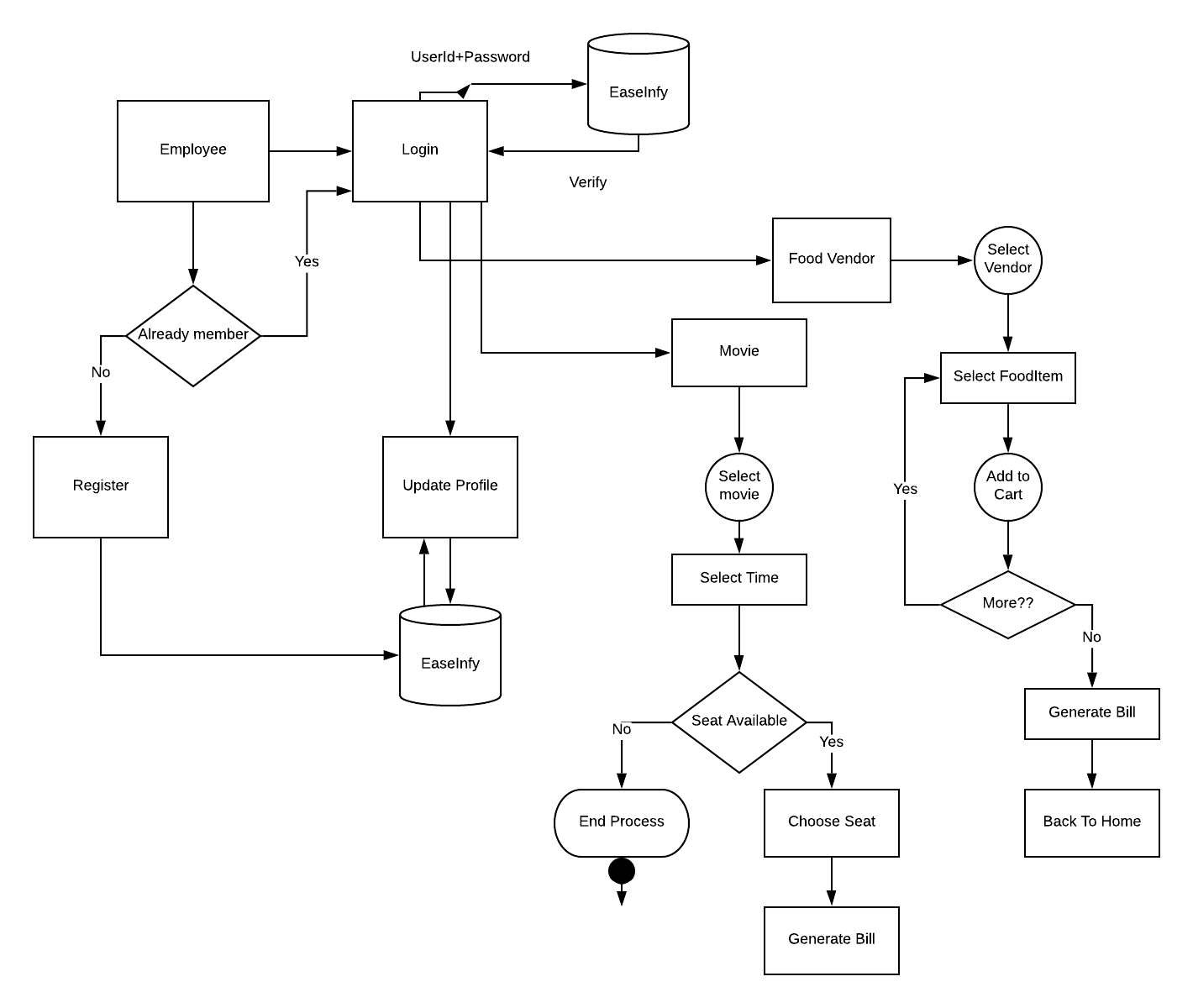
1. **Vendor Controller**: This controller’s functionality is to view who all are its customers, add/delete food items and save them.

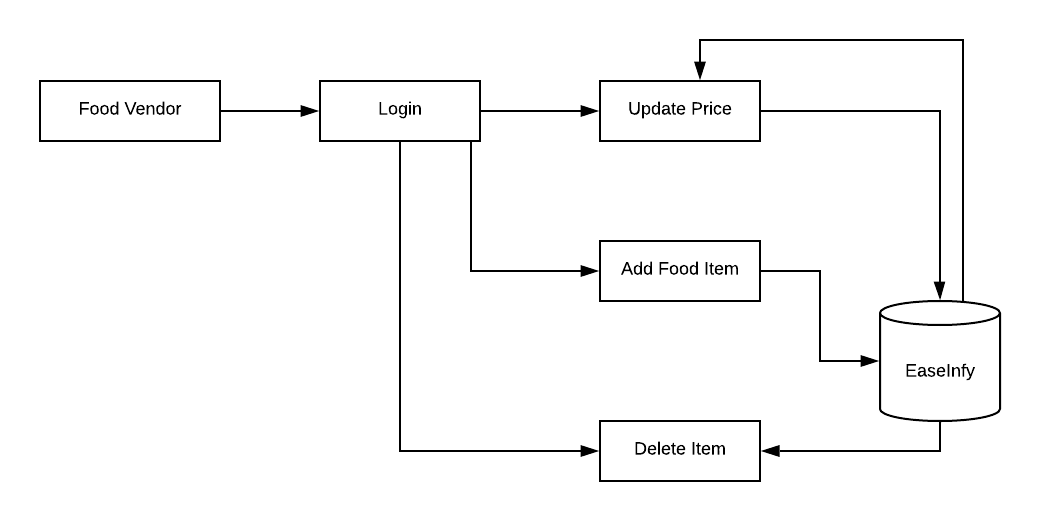
The Functionalities included are:

* + - * + View Customer
        + View Food Items
        + Delete Items
        + Add Food Items
        + Save Added Items
        + Save Update Price
        + Save Deletion
  1. **Views, Functionalities and Control Flow**

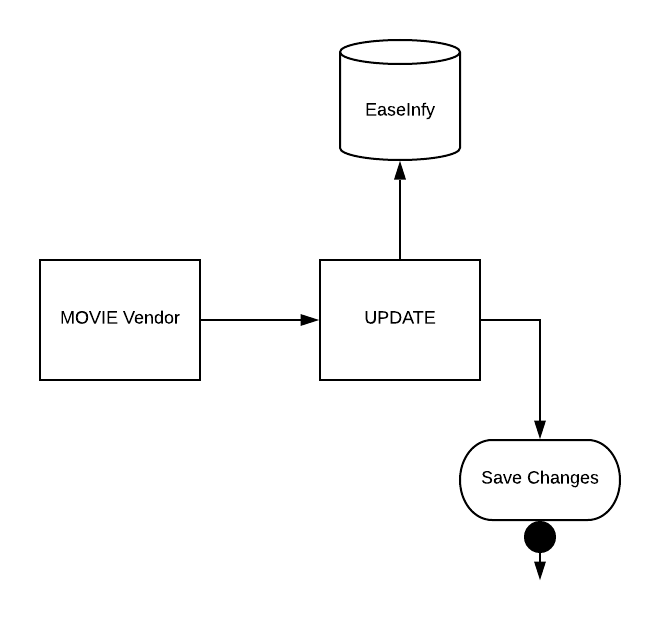
1. **This Picture shows who can use it: Employee, Food Vendor, Movie Vendor**



1. **Functionalities of Employee**
2. **Functionalities of Food Vendor**



1. **Functionalities of Movie Vendor**



1. **VIEWS**
2. Employee:

* View Rating
* View Food Court
* View Vendor
* View Food Items
* View Movies
* Select Food
* Select Movie
* View Bill
* Generate OTP
* Generate Bill
* Generate Bill Movie
* Update Profile

1. Home:

* About
* Contact
* Index
* Login Employee
* Login Movie
* Login Vendor
* Privacy
* Register
* Success

-

1. Movie:

* Update Movie
* View Movie

1. Vendor:

* View Imports
* View Vendors

1. Shared Layouts:

Many views are used as common layout in other views. The logo, header, footer are displayed all over the webpages. It is used to maintain a consistent look and structure. Therefore to avoid repetition the shared layout file is maintained.

Below mentioned are the layouts used in the project:

* Layout
* Layout Error
* Layout Movie
* Layout Vendor
* Employee Layout
* Error

* 1. **Models**
* Employee
* Vendor
* FCoupon
* Food Coupon
* Food Court
* Food Item
* MCoupon
* Movie Coupon
* Movie Vendor
* Multiplex
* Rating

**Chapter 4: System Development**

**4.1 Functional Requirements**

The employees of the organization who will use this application for View of Food Items, Food Courts, Movies for the weekend, Seat Selection will be the user of this application.

A user can see the basic functionalities of the Web Application without logging in. But to make changes the User be it Employee or Vendor has to register and then login.

After login to the application

**For User:**

* View the list of all the food courts.
* View all the food items with its price.
* Select the Food item to order.
* Place order of food items.
* View the list of movies
* Select seats
* Generate OTP for verification
* Generate Bill
* Rate Services
* View previous ratings

**For Guest User:**

* View the services provided by the web application
* Register as an employee

**For Food Vendors:**

* Add food item to list
* Delete food items from list
* Change price of a particular food item
* View of its customers
* View food items

**For Movie Vendors:**

* Update Movie for weekend
* View customers

**4.2 User Interface and Hardware/ Software and Platform**

**User Interface:**

* Should be easy to understand and user friendly.
* Should adapt to different screen sizes.
* Any user can view the web application but only registered users can avail the services.

**Software:**

* Windows 7 Desktop

**Platform:**

* Visual Studio 2017
* Microsoft SQL Server Management Studio 2017

**Chapter 5: Performance Analysis**

**5.1 Test Environment**

* Hardware: Window 7 Desktop
* Software: Visual Studio 2017, Microsoft SQL Server Management Studio 2017

1. Login

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Condition to be tested** | **Expected Result** | **Test Status** |
| 1 | Successful Login | Admin Page | Pass |
| 2 | Validation of credentials | Alert Box | Pass |
| 3 | Validation of credentials | Proper User Friendly Message | Pass |

2)Register Page

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Condition to be tested** | **Expected Result** | **Test Status** |
| 1 | Successful Registration | Login Page | Pass |
| 2 | Validation of credentials | Proper User Friendly Message | Pass |
| 3 | Validation of Data Annotations | Proper User Friendly Message | Pass |
| 4 | Unsuccessful Registration | Error Page | Pass |

3) Movie Vendor Page

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Condition to be tested** | **Expected Result** | **Test Status** |
| 1 | Successful Login | Ticket booking Page | Pass |
| 2 | Unsuccessful Login | Error Page | Pass |
| 3 | Validation of Credentials | Display of proper message | Pass |
| 4 | Validation of Data Annotations | Display of proper message | Pass |

4. Food Court Vendor Page

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Condition to be tested** | **Expected Result** | **Test Status** |
| 1 | Successful Login | Food Menu Page | Pass |
| 2 | Validation of Credentials | Display proper message | Pass |
| 3 | Unsuccessful Login | Error Page | Pass |
| 4 | Validation of Data Annotations | Display proper message | Pass |

5. OTP Generation

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Condition to be tested** | **Expected Result** | **Test Status** |
| 1 | OTP is generated successfully | Payment Page | Pass |
| 2 | OTP not generated successfully | Error Page | Pass |
| 3 | OTP matched correctly | Payment Page | Pass |
| 4 | Incorrect match of OTP | Error Page | Pass |

6. Menu Display Page

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Condition to be tested** | **Expected Result** | **Test Status** |
| 1 | Selection of any food item | Bill Generation Page | Pass |
| 2 | Deselection of any unwanted food item | Same page stays | Pass |
| 3 | Final submission of selected food items | Bill Generation Page | Pass |

7. Movie Display Page

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Condition to be tested** | **Expected Result** | **Test Status** |
| 1 | Selection of movie currently being showed | Seat selection Page | Pass |
| 2 | Unselecting or Changing of already selected movie | Same page stays | Pass |

8. Seat Generation Page

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Condition to be tested** | **Expected Result** | **Test Status** |
| 1 | Selecting a seat in the multiplex | Bill generation page | Pass |
| 2 | Unselecting or Changing already selected seats | Same page stays | Pass |