**1. INTRODUCTION**

Our project “Magic Kingdom Synopsis” is a website in which customers can register online and book tickets. This system provides online help to the users all over the world. It uses an application that integrates with Wolfram Alfa, a computational knowledge engine to fetch information about the different theatres and events location and present them to the user in a simple and easy to understand format. It works as follows:

* The user enters the Text in the Search Box.
* Application detects if it’s a city or state, then fetches and presents information about various theatre or events. It will generate details like movies, cost, location and description.
* If the Text entered is not a city or state, a message is displayed like ‘No such place found’ This application can be developed in Java and using Wolfram Alpha API and different classification based on the event or movie. Wolfram Alpha is an answer engine that works on proprietary knowledge sources for responding to queries on various subjects in a easy to comprehend manner.
* It uses natural language to respond to Questions on various subjects like computer algebra, symbolic and numerical computation, visualization, and statistics capabilities. The most appealing aspect of the response is its presentation of curate content. Unlike most of the search engines, displaying links to web pages.
* Movies and events are the main module for the user to interact and book tickets and do the respective payment based on the cost and count of seats. A person can book only 7 tickets only from any location

The locations are classified in a search engine which is same as wolfram alpha which uses a new science of search engine to search for typed text and show the related data. The user can select a place by either typing the location name or state name.

The application analyses by processing the xml received from API Call for the input received as a search string. The possible outcomes of this analysis could be a valid location or not a valid one and displays the shows running in the place. The user gets to view the appropriate response as follows:

* The text entered is not a valid location. If so, the user gets to see an Answer such as Content of “Search Text Entered” is not listed in database as a valid place. E.g. if user had entered ‘Apple’, the answer would be displayed as ‘Apple’ is not listed in database as a valid location, you may try another name.
* In case of a valid location, the application displays the Image of location, for example in the case of ‘HYDERABAD’ it displays ‘Charminar’.
* When the user clicks it displays a list of events and movies running on the selected location giving the user to select his option and book tickets to corresponding seats.

**2. LITERATURE SURVEY**

The review of literature centers on the existing knowledge and research related to branding an organization, what a brand is, how it is developed and determined to be successful, and maintain a brand through social media.

**What Makes a Successful Wellness Brand?**

If brand equity is linked to a brand’s name and value, then a successful brand must be expressed through “a set of five dimension: brand loyalty, perceived quality/leadership, association/differentiation, awareness and market behavior Applied to successful wellness brands, according to Bernhart (2006), “Branding all health initiatives under a single title and logo is technique common to all the platinum-winning wellness programs.”

In assessing a successful brand, there is high importance in “‘customer value perception’, ‘customer satisfaction’, ‘brand loyalty’ and ‘competitive differentiation’” (Pitta and Katsanis, 1995). Research by Pitta and Katsanis suggest, “Successful companies better understand the importance "customer value-perception" plays in both the short and long run. Less successful companies appear to be more focused on the shorter term, citing "customer satisfaction" as a primary measurement tool. Ultimately, a combination of the two will be critical for successful brand management.”

**One Can Maintain a Wellness Brand through Social Media**

Teresa Bozzelli, vice president of consulting firm Sapient Government Services, says, “What differentiates electronic usage as social media is that is becomes interactive. It’s not bound by time or place. It’s an immediate dialogue and a two-way engagement.” Wellness brands can utilize this two-way connection to maintain a relationship with their customers, simultaneously building and maintaining their brand (Marshall, 2011).

Gray Box – Fans don’t just “like” us, they LOVE us too!

It’s been less than four years since Gray Box came out in 2007 and it already boasts 40% CAGR in revenues and holds over 90% market share in the online entertainment ticketing space. With a listing of over 1000 screens across 87 cities around the country, we have partnerships with all major Indian production houses as well as the Indian counterparts of some which are based abroad. Our partnerships also extend to cinema chains which include INOX, BIG Cinemas, Cine MAX and Fun to name a few. 75% of all cinema tickets sold online are via Gray Box with peak sales which have touched the mark of over a million tickets. Gray Box has already sold tickets for over 1500 plays online and we have exclusive tie-ups with theatres viz. Prithvi Theatre and the NCPA (Mumbai), Indian Habitat Centre (Delhi) and Rangshankara (Bangalore). In addition to plays, we have also been ticketing partners for concerts and sport events, too. Gray Box was the official ticketing partner for Mumbai Indians, Kings XI Punjab, and Rajasthan Royals and was the official online ticketing partner for Delhi Daredevils for the latest season of IPL in 2011.

Pre Book is our distinctive method of allowing our customers to book for tickets for their favorite’s movie well in advance so that they can be the chosen few who get to see it “first-day-first-show.”

Gray Box, as a web-based product, needed to be propagated more fully to the internet savvy customers that we had. Having a social media campaign seemed to be the best solution and we hit it head-on. “The Social Network” has helped us in bridging the gap that the customers had with Gray Box and gave them a personal experience for every problem that they faced. We also decided that our customers got the benefits of being loyal to us and organizing contests to reward them free tickets, merchandise and free previews seemed like the best thing.

We were on [Facebook](http://www.facebook.com/BookMyShowIN) as early as August 2007 but it wasn’t until Nov 2010 that we decided that our approach needed a strategic change and we started seriously focusing on our Facebook campaigns. **25Nov2016 – 15thJune2017**marks months during which we ran several campaigns to supplement our new strategy **improving visibility and reaching out to our potential consumers.**

All our customers are internet savvy as our product itself is web-based. Considering this, we found it imperative that the best way to get to our customers would be by use of something that would be web-based. Our earlier efforts to increase our customer base had primarily been advertising campaigns during key events. We felt the need to give our customers a better experience and solving their doubts, queries, and grievances in a more effective manner. What we most required was a platform which would allow all of this and also let us put our points forward and allow us to reach more people. That was the research that went behind tackling the problem at hand.

Our main aim in creating Bob was to make our Fan page friendly, informal & interactive for all our fans. Gaining maximum visibility via consumer connect was the prime idea. Instead of major spends on traditional advertising media (TV/ radio etc.) we opted for a **cost-effective way**to create our **brand image**via Bob and **directly connect**with our consumers.

We have kept our fans engaged via exclusive fresh first looks of movies& movie content, exciting contests, quizzes, movie merchandise freebies and tickets to be won. We started approaching &associating with major production houses (Hollywood &Bollywood) for content, merchandise and tickets (and later even FB Apps & cross promos for publicizing their Fan Pages). It is a cost effective win-win for both where our fans are happy and the films get great visibility and promotion. We have done the same for events, plays, concerts & sporting events as well.

We have an integrated FB application named Ticket Buddy which allows users to book tickets without leaving their social hub. Ticket Buddy allows you to book movie tickets in the samefashion as on the website. The Pre Book functionality on Ticket Buddy is, however, innovative. It allows you to “Make Your Friends Pay” by reserving seats for you and your friends but just paying for your tickets and making your friends pay for theirs by links that are posted on their wall. If the Pre Booking gets confirmed, you and your friends will all be seated together.

**Reach:**

Gray Box reach is across channels like [**Face book**](http://www.facebook.com/pages/BookMyShow/82235273865)**,** [**Twitter**](http://twitter.com/bookmyshow)**,** YouTube, and [**Gray Box Blog**](http://blog.bookmyshow.com/)**.** But our major platform that is used the best is [**Facebook**](http://www.facebook.com/pages/BookMyShow/82235273865)**.**

[**Facebook Fan Page**](http://www.facebook.com/pages/BookMyShow/82235273865) – The fan page is a more current informative platform where our posts include latest movie trailers and event updates, periodically spiced with quiz questions, movie merchandise, premier and free screening of movies and celeb Meet & Greets.

[**Blog**](http://blog.bookmyshow.com/)**:**

The blog features the latest movie reviews from BMS reviewers as well as renowned critic Rajeev Masand, a B-town Buzz section with the latest pictures from the latest Bollywood events & parties, polls, offers and discounts and miscellaneous articles about our activities, tie-ups, products, etc.

[**Twitter**](https://twitter.com/search/users?q=%40bookmyshow&category=people&source=find_on_twitter)**:**

Entertaining while interacting is our aim here. We tweet music, trivia, quizzes, polls and latest movie news. We are proud to have recently added cricket to our agenda.

**3. SYSTEM ANALYSIS**

**3.1 Scope of the Project**

Provides a way for easy access for the users to search for movies and book seats from that page. Also provides secure way of handling different set of users.

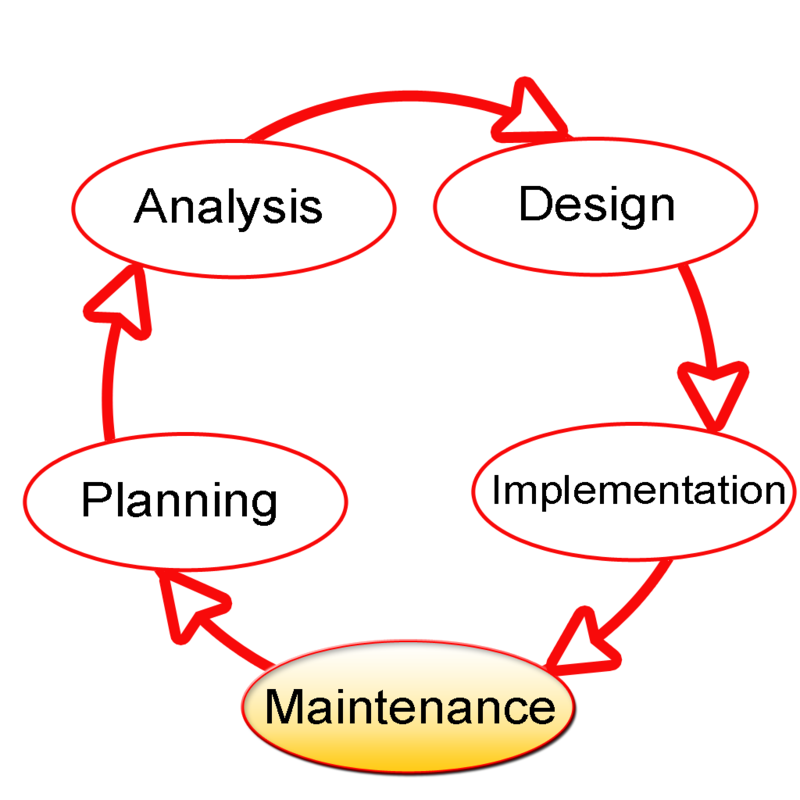
**3.1.1 Vision**

The main vision of our website is provide crew facility for an organization depending upon the requirement with help of this website crew management can know status of employees like crew status, availability, health check status, schedules, readiness for management. Based on the status of employees work will be allocated without any confusion.

**3.2 System Requirement Specification**

**3.2.1 Software Development Life Cycle (SDLC)**

The SDLC, also referred to as the application development life-cycle, is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing, and deploying an information system. The systems development life-cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.



**Figure No.3.1:** SDLC

**3.2.1.1 Spiral Model**

Spiral model wasdefined by BarryBoehminhis1988article,“AspiralModelof SoftwareDevelopmentandEnhancement.Thismodelwasnotthe firstmodeltodiscuss iterativedevelopment, but it was thefirst model to explain whythe iterationmodels.

Asoriginallyenvisioned,theiterationsweretypically6monthsto2yearslong.Eachphase startswithadesigngoalandendswithaclientreviewing theprogressthusfar.Analysis andengineeringeffortsareappliedateachphaseoftheproject,withaneyetowardtheend goal of theproject.

The steps for spiral model are generalized as follows:

* Thenewsystemrequirementsaredefinedinasmuchdetailsaspossible.This usually involvesinterviewinganumberofusersrepresentingalltheexternalor internal users and other aspects of the existingsystem.
* A preliminarydesign is createdforthe new system.
* Afirstprototypeofthenewsystemisconstructedfromthepreliminarydesign.This isusually ascaled-downsystem,andrepresentsanapproximationofthe characteristics ofthefinal product.
* A second prototypeis evolved byafourfold procedure:

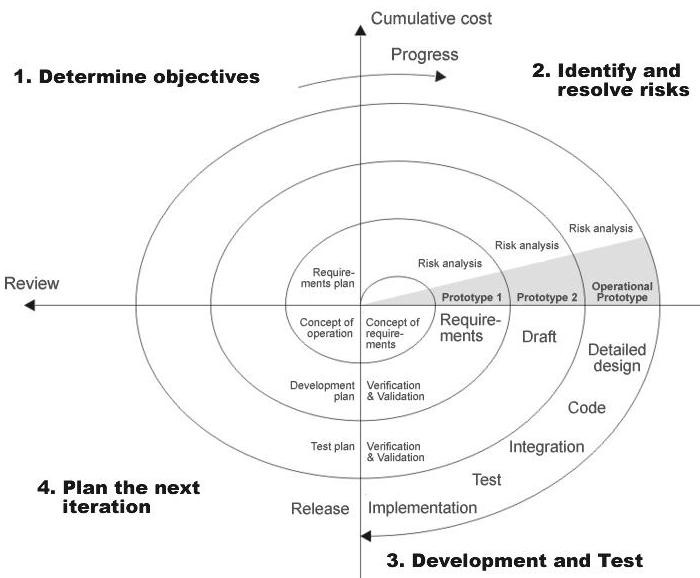
1. Evaluatingthefirst prototypein terms ofits strengths, weakness, and risks.

2. Definingthe requirements of the second prototype.

3. Planninga designing the second prototype.

4. Constructingand testingthe second prototype.

* At the customer option, the entireproject can beaborted if therisk is deemed too great. Risk factors mightinvolve development cost overruns, operating-cost
* Theexistingprototype is evaluated in thesamemanneras was theprevious prototype,andifnecessary, anotherprototypeisdevelopedfromitaccordingtothe fourfold procedureoutlined above.
* Theprecedingsteps areiterated untilthe customeris satisfied that the refined prototyperepresents thefinal product desired.
* Thefinalsystem is constructed, based on the refined prototype.
* Thefinalsystemisthoroughlyevaluatedandtested.Routinemaintenanceiscarried on a continuingbasisto prevent largescalefailures and to minimizedowntime. Thefollowingdiagram shows howaspiral modelacts like:



**Figure No.3.2:**SpiralModel

**3.2.2 Software Requirements**

Operating System : Windows7 or Linux/Solaris

User Interface : HTML, CSS

Programming Language : Java

Web Applications : JDBC, JSP

IDE/Workbench : Eclipse

Database : DB2

Server Deployment : Apache Tomcat 7.0

**3.2.3 Hardware Requirements**

Processor : Pentium IV

Hard Disk : 2GB

RAM : 1GB

**3.2.4Technologies Used**

* Java Script
* Java Server Pages (JSP)
* Servlets
* Eclipse IDE
* Cascading Style Sheets (CSS)

**3.3 Feasibility Study**

Feasibility studies aimed to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the environment the resources required to carry through, and ultimately the prospects for success.The feasibility study is used to determine if the project should get the go-ahead. If the project is to proceed, the feasibility study will produce a project plan and budget estimates for the future stages of development.

**3.3.1 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. Expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information, better employee morale.

**3.3.2 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. Early involvement reduces the chances of resistance to the system and in general and increases the likelihood of successful project.

**3.3.3 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 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.

**3.4 EXISTING SYSTEM**

* Whenever we implement new system it is developed to remove the shortcomings of an existing system.
* The firm whose production is to be monitored has to enter the data manually.
* There are chances of the firm manipulating the data after recording.

**Limitations in Existing System**

* Involves multiple people.
* Time taking process.

**3.5 PROPOSED SYSTEM**

* To book tickets at the ticket counter of respective cinema hall which is a hectic process where one should stand in long queues for hours.
* It is more time consuming process.

**Advantages Over System Existing**

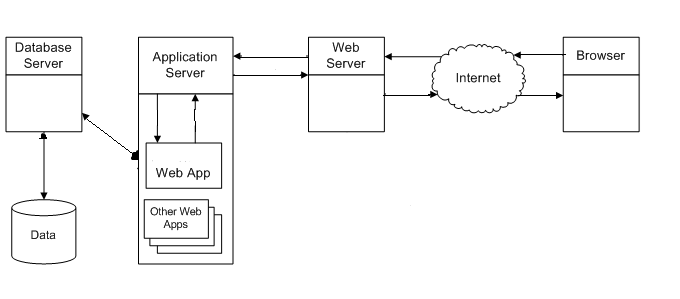
* Time saving.
* Easy to track the request sent for approval.
* Transparency.

**4. SYSTEM DESIGN**

**4.1 System Architecture**

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

A system architecture can comprise system components, the expand systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).



**Figure No.4.1:** System Architecture

**4.2 Unified Modeling Language Diagrams**

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.

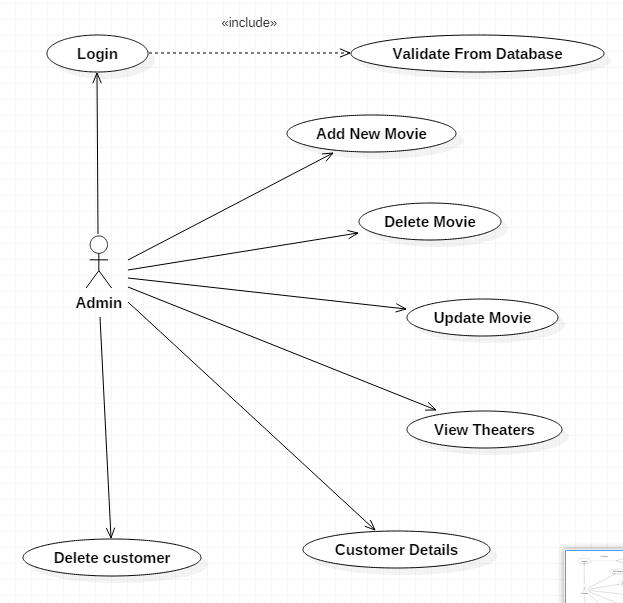
UML is specifically constructed through two different domains they are:

1. UML Analysis modeling, which focuses on the user model and structural model views of the system.
2. UML design modeling, which focuses on the behavioral modeling, implementationmodeling and environmental model views.

**4.2.1 System Use Case Diagrams**

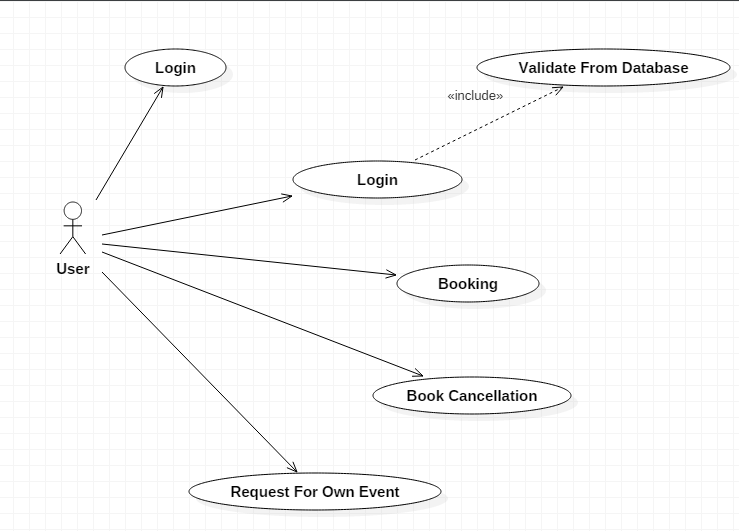
Use case Diagrams represent the functionality of the system from a user’s point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view. Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer …etc., or another system like central database.

* **Use Case Diagram for Admin**

****

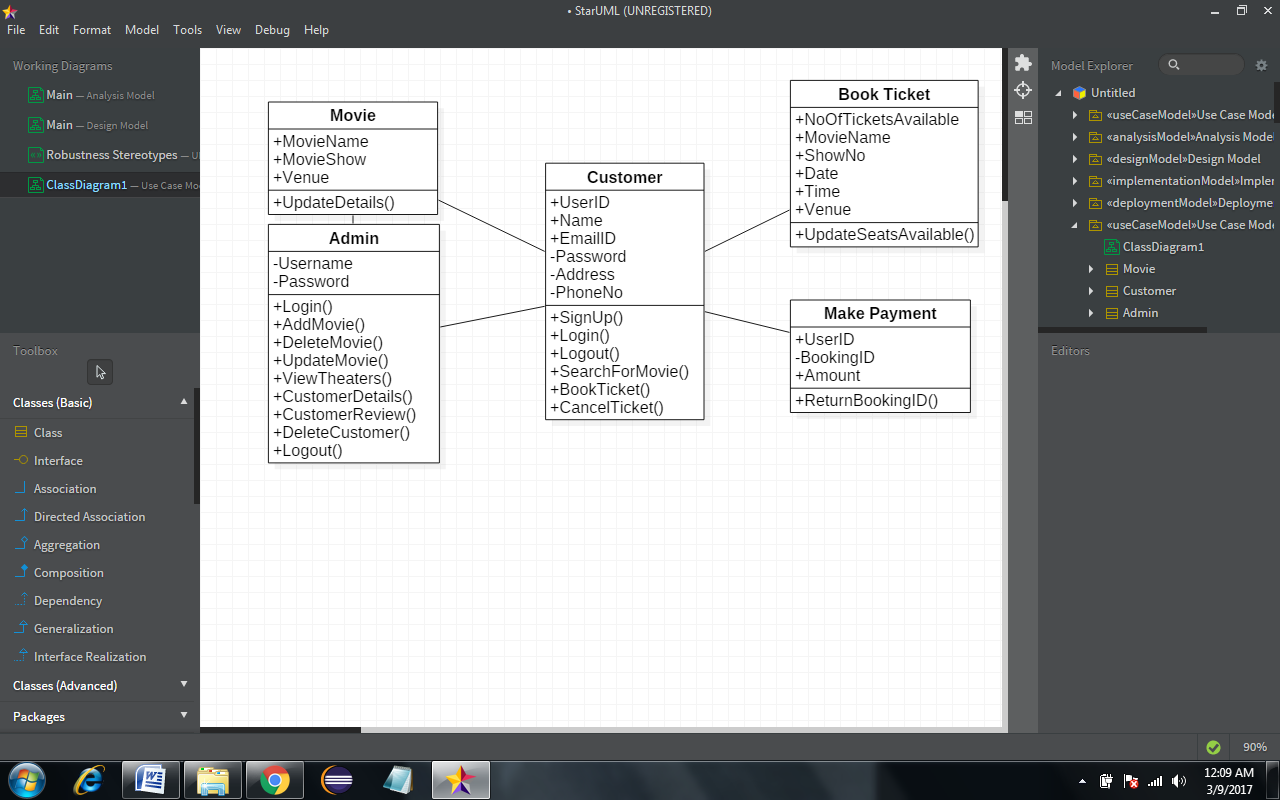
**Figure No.4.2:** Use Case Diagram for Admin

* **Use Case diagram for User:**

  
**Figure No.4.3:** Use Case Diagram for User in Magic Kingdom

The above use case diagram is an interaction between user and system. The user has privileges of viewing the details about the movies, theaters and booking tickets.

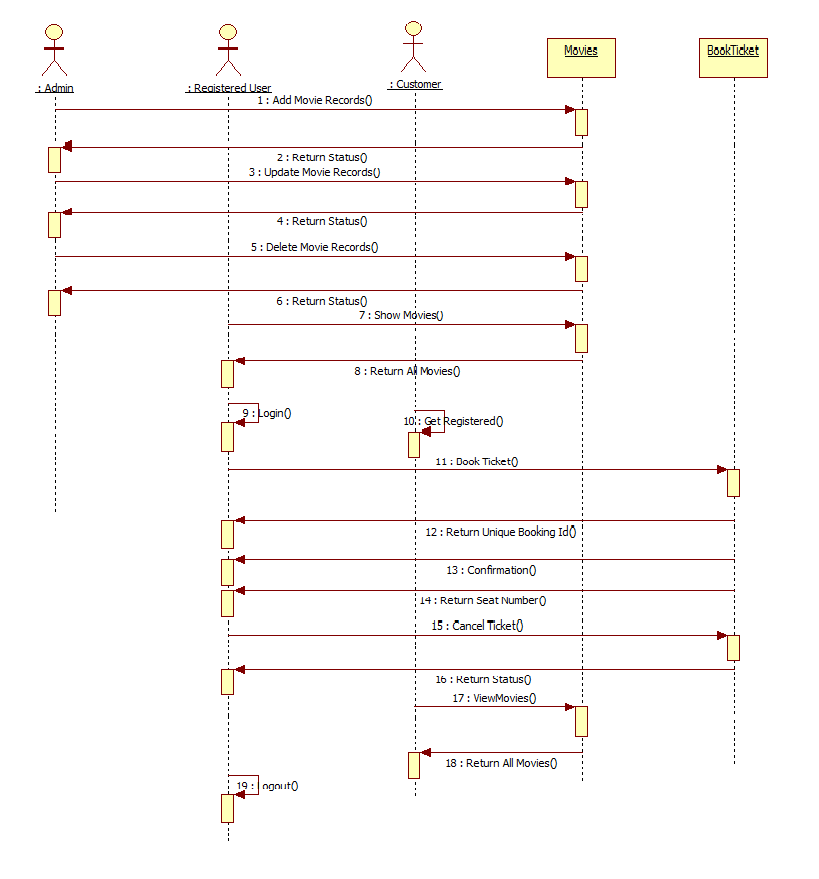
**4.2.2 Class Diagram**

A class diagram shows a set of classes, interfaces, collaborations and their relationships. Class diagrams that include active classes address the static process view of a system.

**Figure No.4.4:** Class Diagram for Magic Kingdom

**4.2.3 Sequence Diagram**

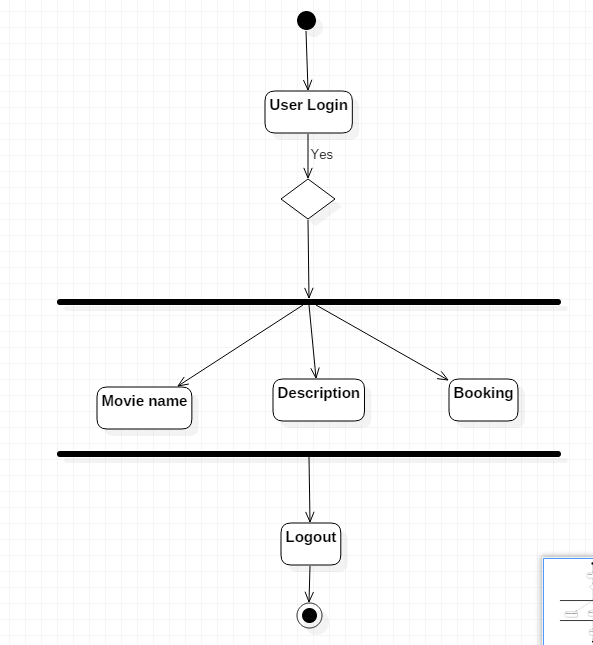
A sequence diagram is an interaction diagram that emphasizes the time-ordering of messages. Sequence diagrams and collaboration diagrams are isomorphic, meaning that you can take one and transform it into the other.

****

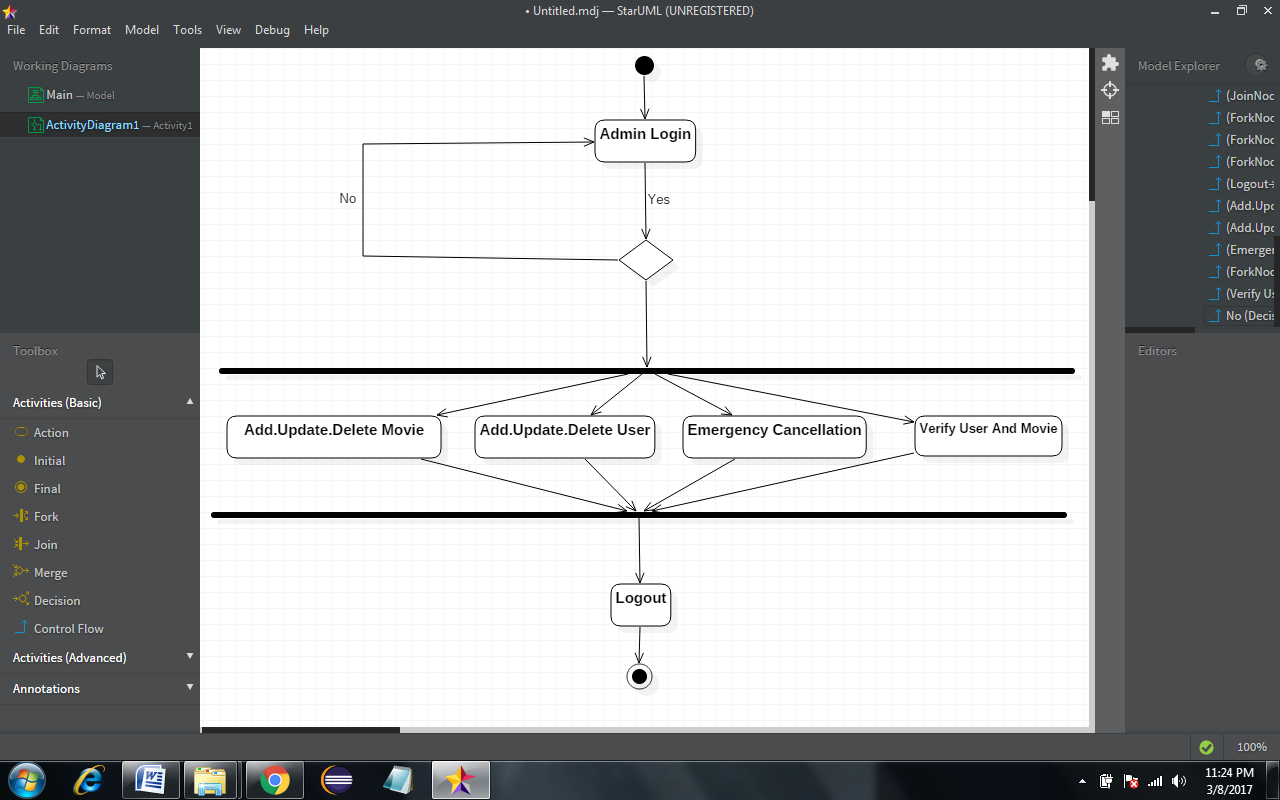
**Figure No.4.5:**Sequence Diagram for Magic Kingdom

**4.2.4 Activity Diagram**

The Unified Modeling Language has several subsets of diagrams that it can model, including structure diagrams, interaction diagrams, and behavior diagrams. Activity diagrams are a subset of the latter. Along with use case and state machine diagrams,they're used to describe business activities and software systems' functionality.Stakeholders have many issues to manage, so it's important to communicate with clarity and brevity. Activity diagrams help people on the business and development sides of an organization come together.

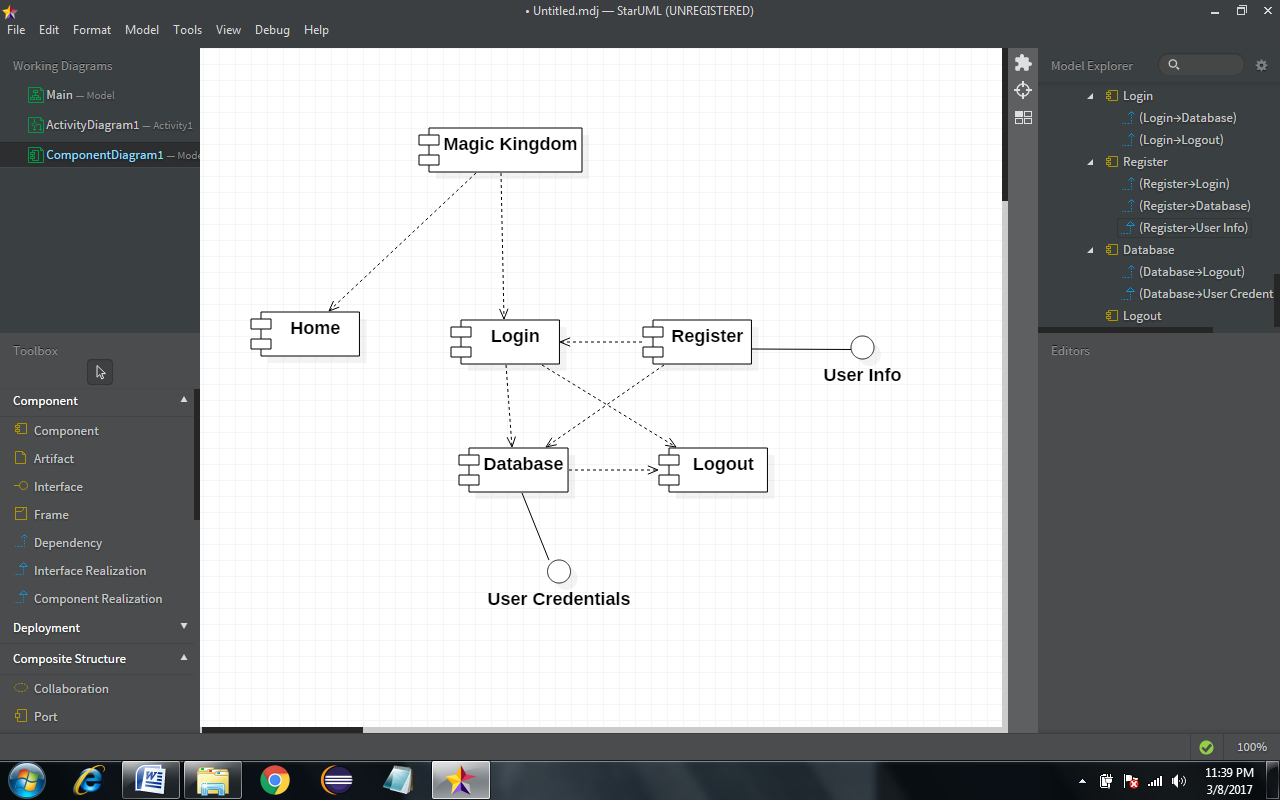


**Figure No.4.6:**Activity Diagram for User



**Figure No. 4.7:**Activity Diagram for Admin

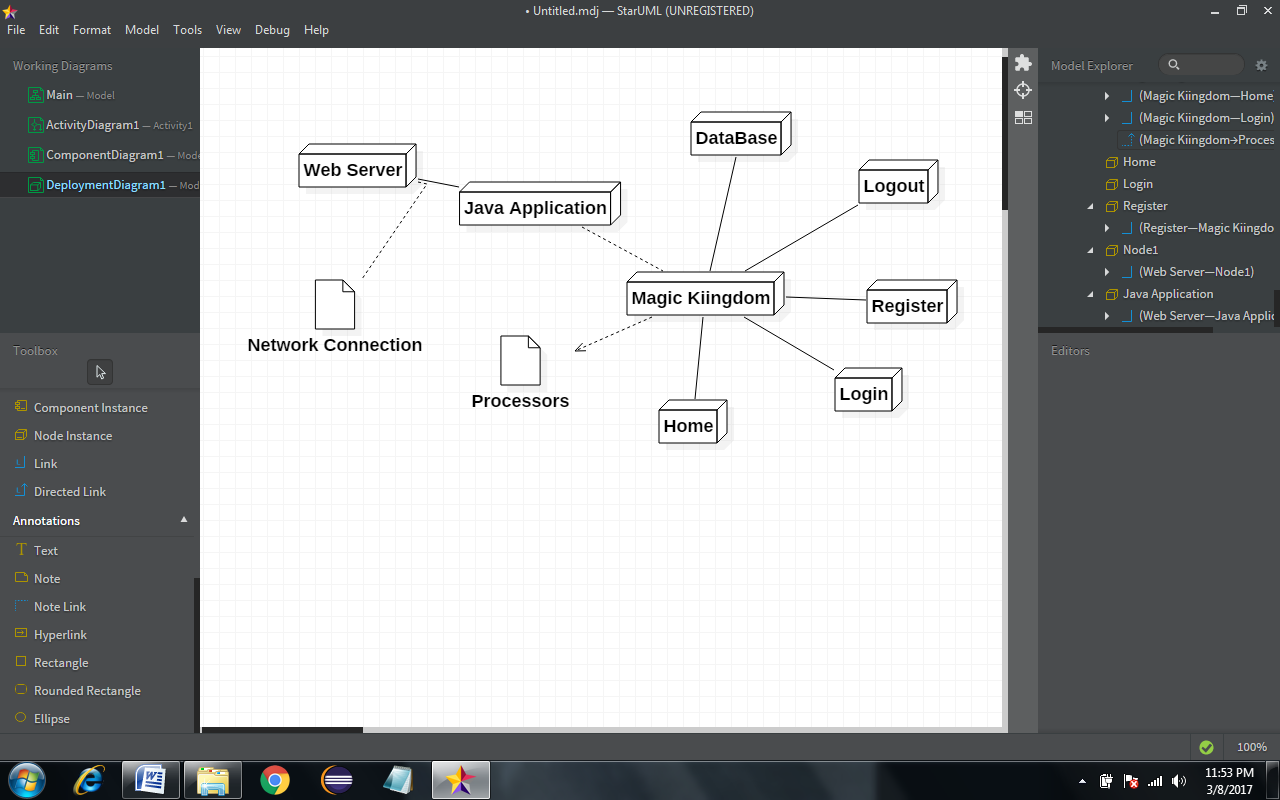
**4.2.5 Component Diagram**

A component diagram shows the organizations and dependencies among a set of components. Component diagrams address the static implementation view of a system..

**Figure No.4.8:** Component Diagram for Magic Kingdom

**4.2.6 Deployment Diagram**

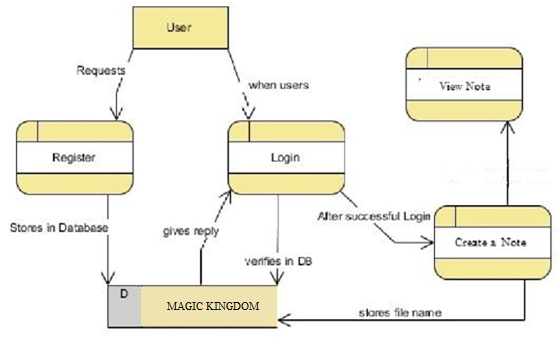
A Deployment diagram shows the configuration of run-time processing nodes and the components that live on them. Deployment diagrams address the static deployment view of architecture.



**Figure No.4.9:**Deployment Diagram for Magic Kingdom

**4.2.7 Data Flow Diagram**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

****

**Figure No.4.10:**Data Flow Diagram for Magic Kingdom

**4.3 DATA DICTIONARY:**

**Admin Login Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **KEY** | **NAME** | **DATA TYPE** | **LENGTH** | **NULLABLE** |
| PRIMARY | USERNAME | VARCHAR | 10 | NO |
|  | PASSWORD | VARCHAR | 10 | NO |

**Table No.4.1:** Admin Login

**User Registration Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **KEY** | **NAME** | **DATA TYPE** | **LENGTH** | **NULLABLE** |
|  | USERNAME | VARCHAR | 100 | NO |
|  | PASSWORD | VARCHAR | 100 | NO |
|  | FIRSTNAME | VARCHAR | 100 | NO |
|  | LASTNAME | VARCHAR | 100 | NO |
|  | GENDER | VARCHAR | 100 | NO |
|  | PHONE | BIGINT | 100 | NO |
|  | EMAIL | VARCHAR | 100 | NO |
|  | CITY | VARCHAR | 100 | NO |
|  | PIN | VARCHAR | 100 | NO |
|  | SECURITYQUESTION | VARCHAR | 100 | NO |
|  | ANSWER | VARCHAR | 100 | NO |

**Table No.4.2:** User Registration

**Movie Information Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **KEY** | **NAME** | **DATA TYPE** | **LENGTH** | **NULLABLE** |
|  | TNAME | VARCHAR | 100 | NO |
|  | MNAME | VARCHAR | 100 | NO |
|  | SCREEN | VARCHAR | 100 | NO |
|  | MOSHOW | VARCHAR | 100 | NO |
|  | MASHOW | VARCHAR | 100 | NO |
|  | FSHOW | VARCHAR | 100 | NO |
|  | SSHOW | VARCHAR | 100 | NO |
|  | ANAME | VARCHAR | 100 | NO |
|  | DNAME | VARCHAR | 100 | NO |
|  | GENRE | VARCHAR | 100 | NO |
|  | REVIEW | VARCHAR | 100 | NO |
|  | TRAILER | VARCHAR | 100 | NO |
|  | PHOTO | VARCHAR | 100 | NO |

**Table No.4.3:** Movie Information

**Booking:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **KEY** | **NAME** | **DATA TYPE** | **LENGTH** | **NULLABLE** |
|  | UNAME | VARCHAR | 100 | NO |
|  | MNAME | VARCHAR | 100 | NO |
|  | TNAME | VARCHAR | 100 | NO |
|  | DATE | VARCHAR | 100 | NO |
|  | TIME | VARCHAR | 100 | NO |
|  | VAL | VARCHAR | 100 | NO |

**Table No.4.4:**Booking

**Upcomming Movies:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **KEY** | **NAME** | **DATA TYPE** | **LENGTH** | **NULLABLE** |
|  | TNAME | VARCHAR | 100 | NO |
|  | MNAME | VARCHAR | 100 | NO |
|  | ANAME | VARCHAR | 100 | NO |
|  | DNAME | VARCHAR | 100 | NO |
|  | GENRE | VARCHAR | 100 | NO |
|  | REVIEW | VARCHAR | 100 | NO |
|  | TRAILER | VARCHAR | 100 | NO |

**Table No.4.5:**Upcoming Movies

**Customer Review:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **KEY** | **NAME** | **DATA TYPE** | **LENGTH** | **NULLABLE** |
|  | UNAME | VARCHAR | 100 | NO |
|  | MOVIE | VARCHAR | 100 | NO |
|  | REVIEW | VARCHAR | 100 | NO |
|  | RATINGS | VARCHAR | 100 | NO |

**Table No.4.6:** Customer Review

**Theater:**

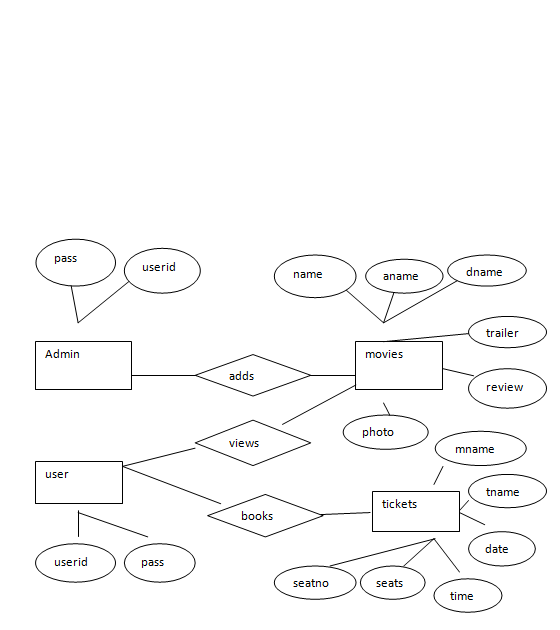
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **KEY** | **NAME** | **DATA TYPE** | **LENGTH** | **NULLABLE** |
|  | TNAME | VARCHAR | 100 | NO |
|  | AREA | VARCHAR | 100 | NO |
|  | ADDRESS | VARCHAR | 100 | NO |
|  | PHONE | VARCHAR | 100 | NO |
|  | TYPE | VARCHAR | 100 | NO |
|  | NOOFSCREENS | VARCHAR | 100 | NO |
|  | FACILITIES | VARCHAR | 100 | NO |
|  | PHOTO | LONGBLOB | 100 | NO |
|  | MAP | VARCHAR | 100 | NO |

**Table No.4.7:** Theater

**4.3.1 E-R Diagram**

An entity–relationship model is a systematic way of describing and defining a business process. The process is modeled as components (entities) that are linked with each other by relationships that express the dependencies and requirements between them, such as: one building may be divided into zero or more apartments, but one apartment can only be located in one building. Entities may have various properties (attributes) that characterize them. Diagrams created to represent these entities, attributes, and relationships graphically are called entity–relationship diagrams. ER Diagrams represent the relationship between the entities. An ER diagram is composed of the following:

* Entity is shown by rectangle.
* Attribute is shown by oval.
* Relationships with rhombus.
* Optional is shown by circle.
* Compulsory with dash.
* Primary key with underscore.



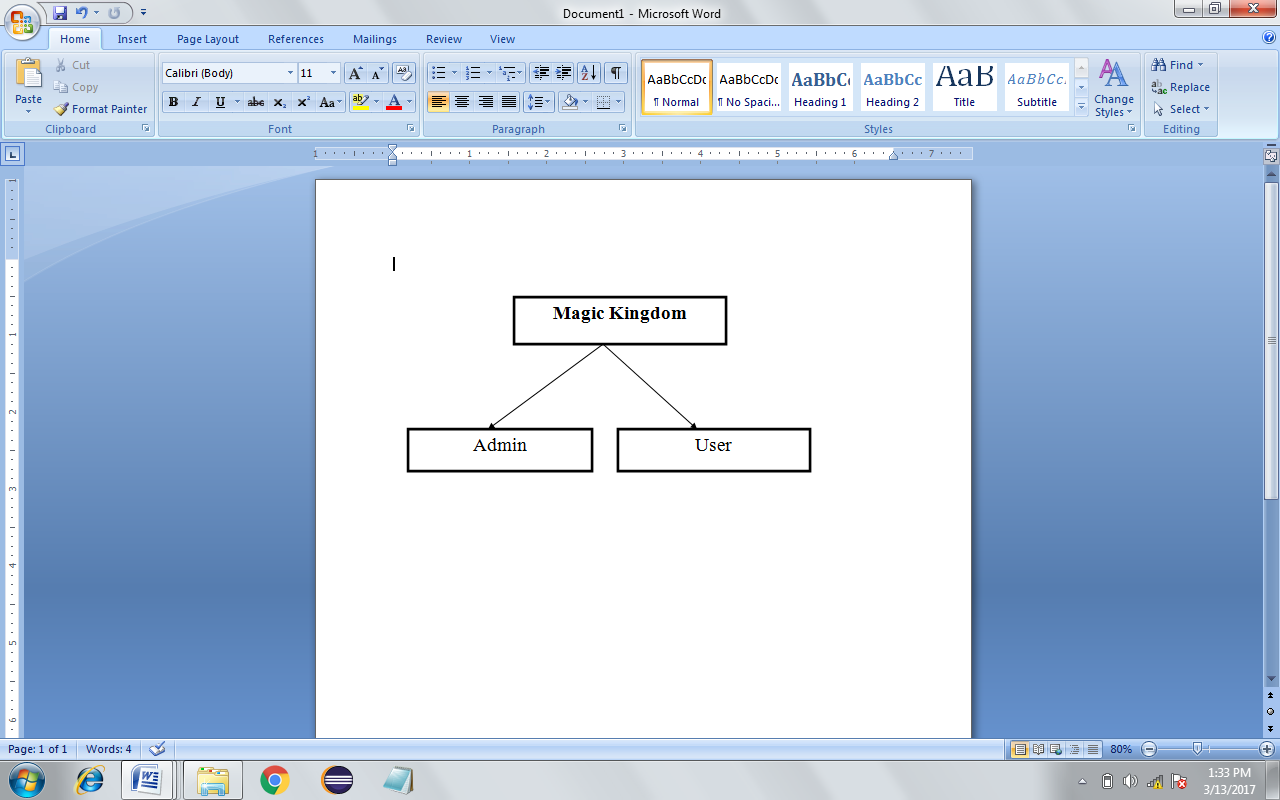
**Figure No.4.11:** E-R Diagram of Magic Kingdom Synopsis

**5. SYSTEM IMPLEMENTATION**

**5.1 Modules Description**

Our system is divided into two major parts depending upon the requirement they are.

* Admin
* User



**Figure No.5.1:** Modules

## **Administrator**

## A administrator or in specific database administrator is an IT professional responsible for the [installation](http://en.wikipedia.org/wiki/Installation_%28computer_programs%29), [configuration](http://en.wikipedia.org/wiki/Computer_configuration), [upgrading,](http://en.wikipedia.org/wiki/Upgrade)[administration](http://en.wikipedia.org/wiki/System_administrator), [monitoring](http://en.wikipedia.org/wiki/System_Monitoring), [maintenance](http://en.wikipedia.org/wiki/Software_maintenance), and [security](http://en.wikipedia.org/wiki/Computer_Security)of [databases](http://en.wikipedia.org/wiki/Databases) in an organization. The role includes the [development](http://en.wikipedia.org/wiki/Software_development)and [design](http://en.wikipedia.org/wiki/Design) of database [strategies](http://en.wikipedia.org/wiki/Strategy),and [planning](http://en.wikipedia.org/wiki/Planning) for future expansion requirements. They may also plan, [co-ordinate](http://en.wikipedia.org/wiki/Cooperation)and implement security measures to safeguard the database**.**

All the users are controlled by admin. Administrator services are as follows

* + - * View Production Manager
      * He can approve.

**Operations of User**

User is involved with some operations like viewing a route based on source/destination. The user can also view selected options utilities such as ATM, gas-station, hospitals, shopping malls etc in the route. Also, they users can save the map and places to view it later. The user can view/update profile.

* 1. **Pseudo Code**

Step 1: Displays the project home page

Step 2: Click on registration

Step 3: switch(n)

{

case 1: Customer registration

Enter the details in all the fields if(fields!=null)

{

Registered successfully

}

else

{

Enter valid details

}

break;

default :got home page; break;

}

Step 4: click on login

Step 5: switch(ch)

{

Case 1: customer login

Enter username and password if(username==null || password== null)

{

Both userid and password are mandatory

}

else if(username==userid&& password = =pwd)

{

Login successful

}

else

{

Enter valid userid and password!!

}

Break;

Case 2: admin login

Enter username and password if(username==null || password== null)

{

Both userid and password are mandatory

}

else if(username==userid&& password = =pwd)

{

Login successful

}

else

{

Enter valid user id and password!!

}

Break;

Case 3: RCPC login

Enter username and password if(username==null || password== null)

{

Both userid and password are mandatory

}

else if(username==userid&& password = =pwd)

{

Login successful

}

else

{

Enter valid user id and password!!

}

Break;

Default: got home page

Break;

Step 6: click on Upload Magic Kingdom Synopsis Report

Enter Report

If(Report==NULL)

{Cannot submit empty report}

else

{

Report submitted successfully

}

Step7: click on about

Displays portal description

Step8: click on contact us

Displays contact details

Step9: end

**6. SYSTEM TESTING**

**6.1 White Box Testing**

White Box Testing(also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing) is a [software testing method](http://softwaretestingfundamentals.com/software-testing-methods/) in which the internal structure/ design/ implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential.

This method is named so because the software program, in the eyes of the tester, is like a white/ transparent box; inside which one clearly sees.

**Example**

A tester, usually a developer as well, studies the implementation code of a certain field on a webpage, determines all legal (valid and invalid) AND illegal inputs and verifies the outputs against the expected outcomes, which is also determined by studying the implementation code.

White Box Testing is like the work of a mechanic who examines the engine to see why the car is not moving.

**Levels Applicable to**

White Box Testing method is applicable to the following levels of software testing:

* [Unit Testing](http://softwaretestingfundamentals.com/unit-testing/): For testing paths within a unit.
* [Integration Testing](http://softwaretestingfundamentals.com/integration-testing/): For testing paths between units.
* [System Testing](http://softwaretestingfundamentals.com/system-testing/): For testing paths between subsystems.

However, it is mainly applied to Unit Testing.

**White Box Testing Advantages**

* Testing can be commenced at an earlier stage. One need not wait for the GUI to be available.
* Testing is more thorough, with the possibility of covering most paths.

**White Box Testing Disadvantages**

* Since tests can be very complex, highly skilled resources are required, with thorough knowledge of programming and implementation.
* Test script maintenance can be a burden if the implementation changes too frequently.
* Since this method of testing is closely tied with the application being testing, tools to cater to every kind of implementation/platform may not be readily available.

**6.1.1 Unit Testing**

Unit testing is a [level of software testing](http://softwaretestingfundamentals.com/software-testing-levels/) where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed.

A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. In procedural programming a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.)

Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

* **Unit testing:** See component testing.
* **Component testing:**The testing of individual software components.

**METHOD**

Unit Testing is performed by using the [White Box Testing](http://softwaretestingfundamentals.com/white-box-testing/) method.

**BENEFITS**

* Unit testing increases confidence in changing/ maintaining code. If good unit tests are written and if they are run every time any code is changed, we will be able to promptly catch any defects introduced due to the change. Also, if codes are already made less interdependent to make unit testing possible, the unintended impact of changes to any code is less.
* Codes are more reusable. In order to make unit testing possible, codes need to be modular. This means that codes are easier to reuse.

**6.1.2 Functional Testing**

Functional testing is a [quality assurance](https://en.wikipedia.org/wiki/Quality_assurance) (QA) process and a type of [black-box testing](https://en.wikipedia.org/wiki/Black-box_testing) that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (unlike [white-box testing](https://en.wikipedia.org/wiki/White-box_testing)).Functional testing usually describes what the system does.

Functional testing does not imply that you are testing a function (method) of your module or class. Functional testing tests a slice of functionality of the whole system.

Functional testing differs from [system testing](https://en.wikipedia.org/wiki/System_testing) in that functional testing "[verifies](https://en.wikipedia.org/wiki/Verification_and_validation_(software)) a program by checking it against ... design document(s) or specification(s)", while system testing "[validate[s]](https://en.wikipedia.org/wiki/Verification_and_validation_(software)) a program by checking it against the published user or system requirements" (Kaner, Falk, Nguyen 1999, p. 52).

Functional testing has many types:

* [Smoke testing](https://en.wikipedia.org/wiki/Smoke_testing_(software))
* [Sanity testing](https://en.wikipedia.org/wiki/Sanity_testing)
* [Regression testing](https://en.wikipedia.org/wiki/Regression_testing)
* [Usability testing](https://en.wikipedia.org/wiki/Usability_testing)

**6.1.3 Integration Testing**

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in [software testing](https://en.wikipedia.org/wiki/Software_testing) in which individual software modules are combined and tested as a group. It occurs after [unit testing](https://en.wikipedia.org/wiki/Unit_testing) and before [validation testing](https://en.wikipedia.org/wiki/Verification_and_validation_(software)). Integration testing takes as its input [modules](https://en.wikipedia.org/wiki/Module_(programming)) that have been unit tested, groups them in larger aggregates, applies tests defined in an integration [test plan](https://en.wikipedia.org/wiki/Test_plan) to those aggregates, and delivers as its output the integrated system ready for [system testing](https://en.wikipedia.org/wiki/System_testing). The purpose of integration testing is to verify functional, performance, and reliability [requirements](https://en.wikipedia.org/wiki/Requirement) placed on major design items.

**6.1.4 System Testing**

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified [requirements](https://en.wikipedia.org/wiki/Requirements). System testing falls within the scope of [black-box testing](https://en.wikipedia.org/wiki/Black-box_testing), and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed [integration testing](https://en.wikipedia.org/wiki/Integration_testing) and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

**6.1.5 Acceptance Testing**

In [engineering](https://en.wikipedia.org/wiki/Engineering) and its various [subdisciplines](https://en.wikipedia.org/wiki/Fields_of_engineering), acceptance testing is a test conducted to determine if the requirements of a [specification](https://en.wikipedia.org/wiki/Specification) or [contract](https://en.wikipedia.org/wiki/Contract) are met. It may involve [chemical tests](https://en.wikipedia.org/wiki/Chemical_test), [physical tests](https://en.wikipedia.org/wiki/Physical_test), or [performance tests](https://en.wikipedia.org/wiki/Performance_test_(assessment)).

In [systems engineering](https://en.wikipedia.org/wiki/Systems_engineering) it may involve [black-box testing](https://en.wikipedia.org/wiki/Black-box_testing) performed on a [system](https://en.wikipedia.org/wiki/System) prior to its delivery.In [software testing](https://en.wikipedia.org/wiki/Software_testing) the [ISTQB](https://en.wikipedia.org/wiki/International_Software_Testing_Qualifications_Board) defines acceptance as: formal testing with respect to user needs, requirements, and business processes conducted to user, customers or other authorized entity to determine whether or not to accept the system. Acceptance testing is also known as user acceptance testing (UAT), end-user testing, [operational acceptance testing](https://en.wikipedia.org/wiki/Operational_acceptance_testing) (OAT) or field (acceptance) testing.

**6.2 Black box testing**

Black Box Testing, also known as Behavioral Testing, is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.Black-box testing is a method of [software testing](https://en.wikipedia.org/wiki/Software_testing) that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: [unit](https://en.wikipedia.org/wiki/Unit_test), [integration](https://en.wikipedia.org/wiki/Integration_testing), [system](https://en.wikipedia.org/wiki/System_testing) and [acceptance](https://en.wikipedia.org/wiki/Acceptance_test). It typically comprises most if not all higher level testing, but can also dominate [unit testing](https://en.wikipedia.org/wiki/Unit_testing) as well.

**6.3 Test Cases**

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.

**Types of tests that include in system testing are**:

1. Graphical user interface testing

2. Software performance testing

3. Compatibility testing

4. Exception handling

5. Usability testing … etc.

**Test Cases**

Test Cases are of two types:

1. Positive Test Case

2. Negative Test Case

**6.3.1 Guide Lines for Test Cases**

Test cases should have:

1. Test Case ID

2. Test Case Name

3. Test Case Description

4. Actual Value

5. Expected Value

**6.3.2 GUI Test Cases:**

* Total no of features that need to be checked.
* Look & Feel.
* Look for Default values, if at all any (date & Time, if at all any require).
* Look for spell check.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case No** | **Description** | **Expected value** | **Actual value** | **Result** |
| 1 | Check for all the features in the screen | The screen must contain all features | It contains all features | True |
| 2 | Check for the alignment of the objects as per the validations | The alignment should be in proper way | The alignments are in the proper way | True |

**Table No.6.1:** Guidelines for test cases

**6.3.3 Positive Test Cases:**

* The positive flow of the functionality must be considered
* Valid inputs must be used for testing
* Must have the positive perception to verify whether the requirements are justified.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TestCase No** | **Description** | **Expected value** | **Actual value** | **Result** |
| 1 | Check for the date  Time Auto Display | The date and time of the system must be displayed | The date and time is displayed | True |
| 2 | Enter the valid login id and password | It should accept | Valid | True |

**Table No.6.2:** Positive test cases

**6.3.4 Negative Test Cases:**

* Must have negative perception.
* Invalid inputs must be used for test.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case No** | **Description** | **Expected value** | **Actual value** | **Result** |
| 1 | Try to modify the information in date and time | Modification should not be allowed | Not valid | False |
| 2 | Enter invalid data in to the user details form, click on save | It should not accept invalid data, save should not allow | Valid | True |

**Table No.6.3:** Negative test cases

**6.3.5 Login Page Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Name** | **Test Case Description** | **Test Steps** | | **Result** |
| **Expected** | **Actual** |
| 1.Login | Leave the user id empty and click Submit button | Error message | An error message “User id is empty” must be displayed | True |
| Enter User id and password and click submit button | If they are correct, redirect to home page or display error message. | Login success full and redirect to home page or an error message Invalid User id or Password” must be displayed |  |

**Table No.6.4**: Login page test case

**6.3.6 Registration Validation for Empty Field Page Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Name** | **Test Case Description** | **Test Steps** | | **Result** |
| **Expected** | **Actual** |
| Registration  (Empty Fields Validation) | To verify that Full name of user on Registration is entered or left empty | Error message if field left empty | An error message Full Name is empty | True |
| To verify that Mobile number is entered or left empty | Error Message if field left empty | An error message Mobile number is empty | True |
| To verify that E-mail id on Registration page is entered or left empty | Error Message if field left empty | An error message Email id is empty | True |
| To verify that Squestion on Registration page is selected or not | Error Message if field left empty | An error Squestion is not selected | True |
| To verify that answer on registration page is not entered or left empty | Error Message if field left empty | An error message answer is empty | True |

**Table No.6.5:** Registration validation for empty field test case

**6.3.7 Registration Page Validation for Valid Data Test Case:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Registration  (Validation  Of Data) | **Test Case Description** | **Test Steps** | | | **Result** |
| **Expected** | **Actual** | |
| To verify that Mobile number is valid or not | Error Message if mobile number entered is less than 10 digits an error message | | Phone number should have only 10 digits | True |
| To verify that E-mail id on Registration page is valid or not | Error Message if proper email format is not maintained ex:abc@bcd.com | | An error message Email id is invalid | True |

**Table No.6.6**: Registration page validation for valid data test case

**7. DOCUMENTATION AND MAINTAINANCE**

**7.1 Specifying and Reporting**

Specifying and Reporting is a document created by system analyst after the requirements are collected from various stakeholders.

Specifying and Reporting defines how the intended software will interact with hardware, external interfaces, speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc.

The requirements received from client are written in natural language. It is the responsibility of system analyst to document the requirements in technical language so that they can be comprehended and useful by the software development team.It should be robust. That is, it should be tolerant towards incompleteness and complexity.

**7.2 Objectives and Requirements**

Include the detailed user requirements satisfied by this particular software unit. It may a matrix that traces the system requirements from the objectives through the system design document and test plan for the specific software units. Other life cycle documentation may be referenced as appropriate.The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client’s point of view.

**7.3 Corrective, Adaptive and Perfective Maintenance**

**Corrective Maintenance**

In the event of system failure due to an error, actions are taken to restore operations of the software system. Due to the management pressure, sometime maintenance personnel use the emergence fix instead of following the proper maintenance process that results in an increased complexity of software and unforeseen ripple effects (i.e. a change in one part of a program may affect other sections of the software). This is often due to lack of time to carry out a thorough impact analysis before effecting the change.

**Adaptive Maintenance**

Changes in the environment are inevitable. So whenever the environment changes, the software are to be modified to adapt it according to the environment. So adaptive maintenance means changing the software to some new environment such as a different hardware platform or use with a different operating system. The software functionality does not radically change. This type of maintenance includes any work initiated as a consequence of moving the software to a different hardware or a software platform-compiler, operating system or a new processor.

**Perfective Maintenance**

Perfective maintenance activities involve making enhancements to software products i.e. implementing new functional or non-functional system requirements, improving user displays and modes of interaction, upgrading external documents and internal documentation, or upgrading the performance characteristics of the system. These requirements are generated by software customers as their organization or business changes

**7.4 Optimizations of Functionalities**

Requirements, which are related to Optimizations of functionalities aspect of software fall into this category. They define functions and functionality within and from the software system.

* Search option given to user to search from various invoices.
* User should be able to mail any report to management.
* Users can be divided into groups and groups can be given separate rights.
* Should comply business rules and administrative functions.

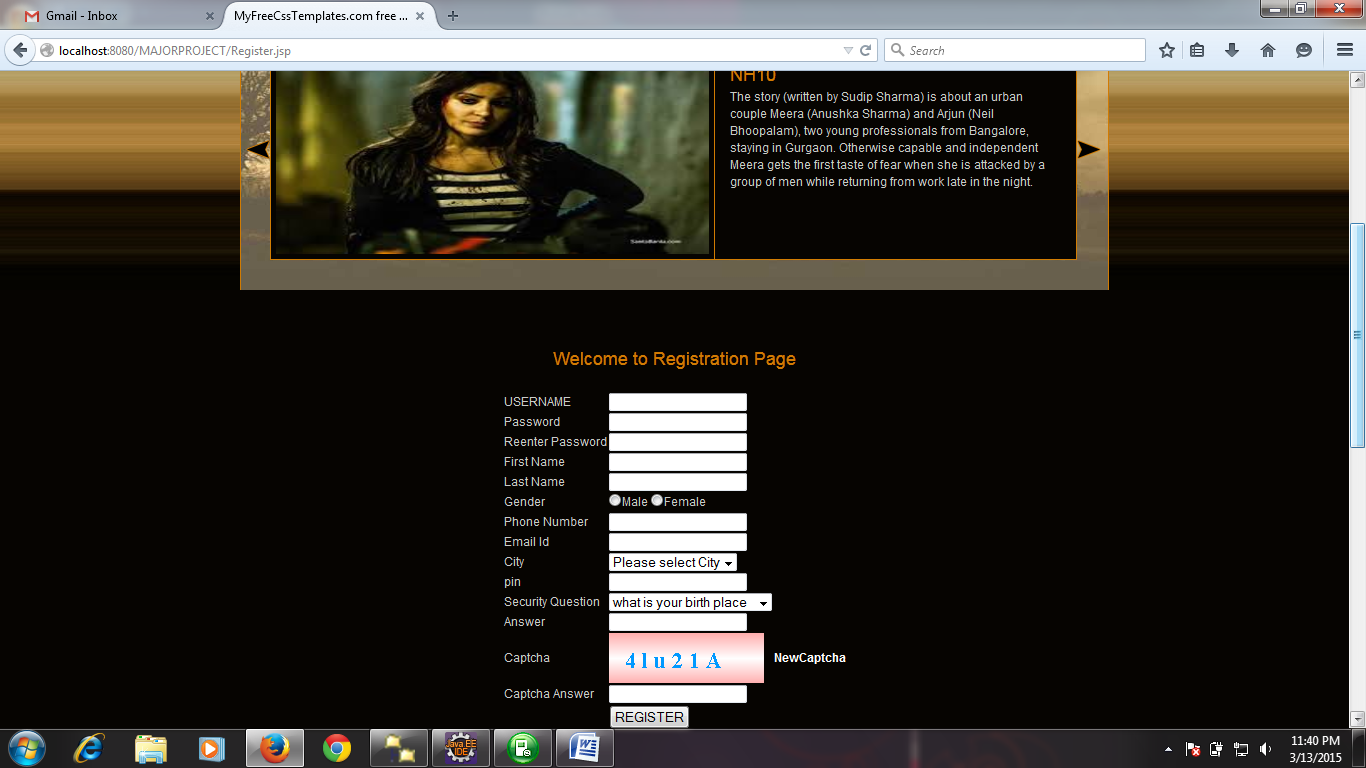
**8.OUTPUTS**

**Step 1:** Home page of Magic Kingdom online movie booking system and user can redirect to various pages from home page.

****

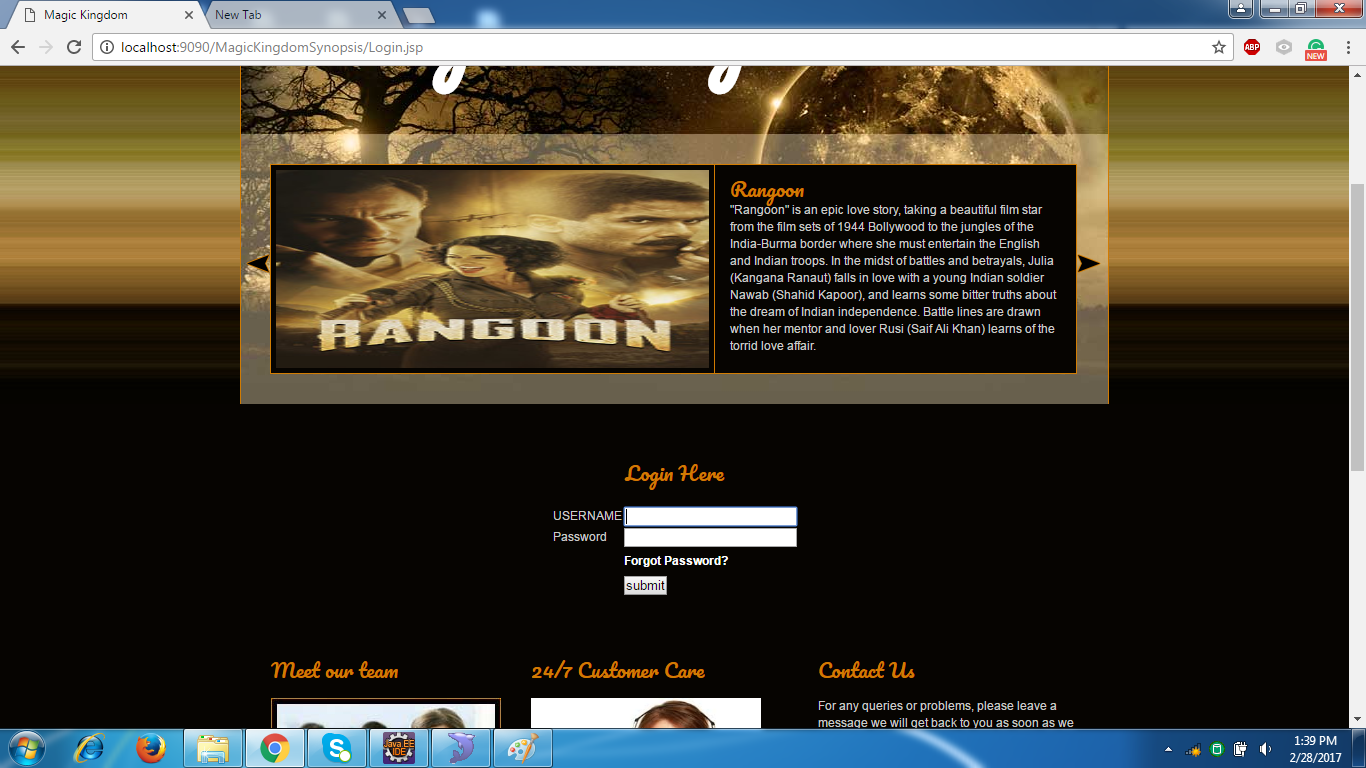
**Figure No.8.1:** Home page

**Step 2:** The user can register here and after successful registration user can book the tickets



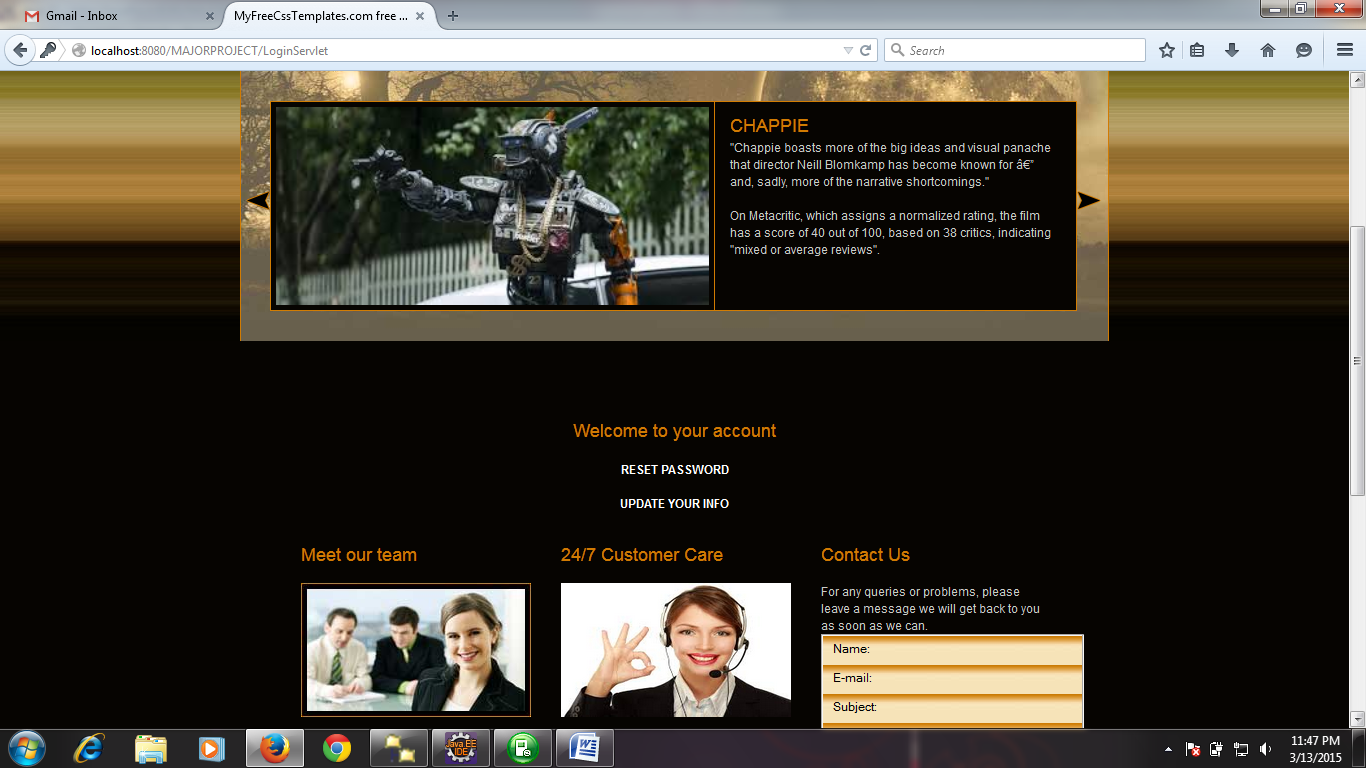
**Figure No.8.2:**Registration Page for User

**Step 3:** The registered users can login here

****

**Figure No.8.3:**Login Page for Users

**Step 4:** When the users log in to their accounts, they are navigated to this page

****

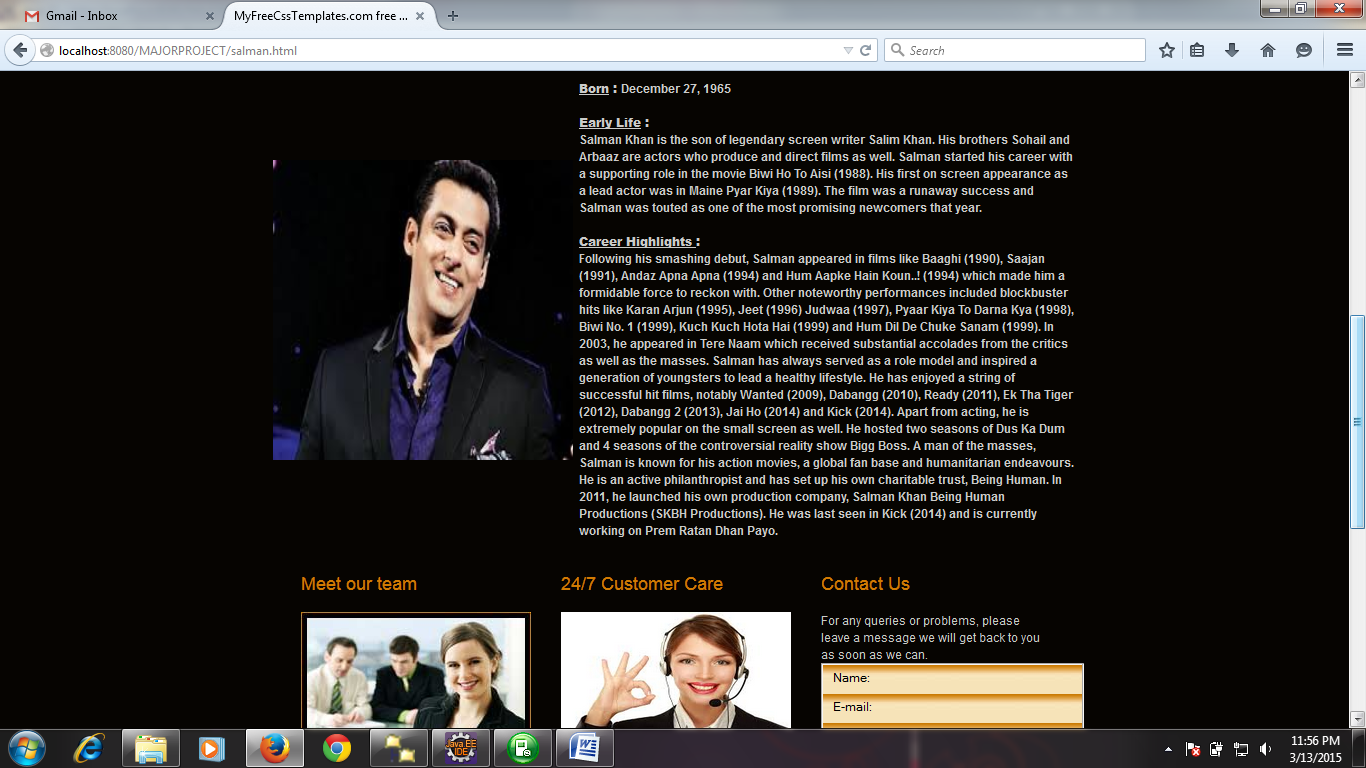
**Figure No.8.4:**User Account Page

**Step 5:** The user can book tickets for his favorite movies here**.**

****

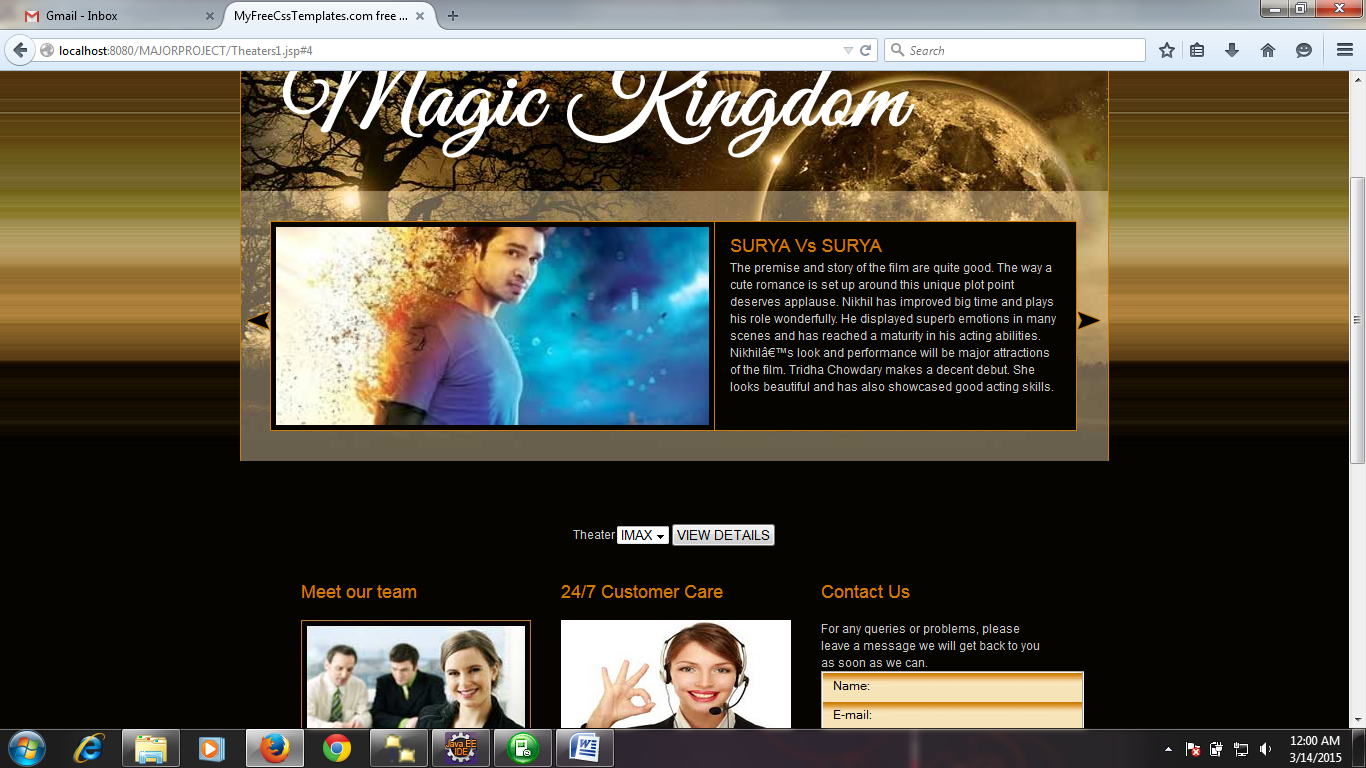
**Figure No.8.5:** Ticket Booking Page

**Step 6:** The users can view about their favorite celebrities from the celebrity’s page.

****

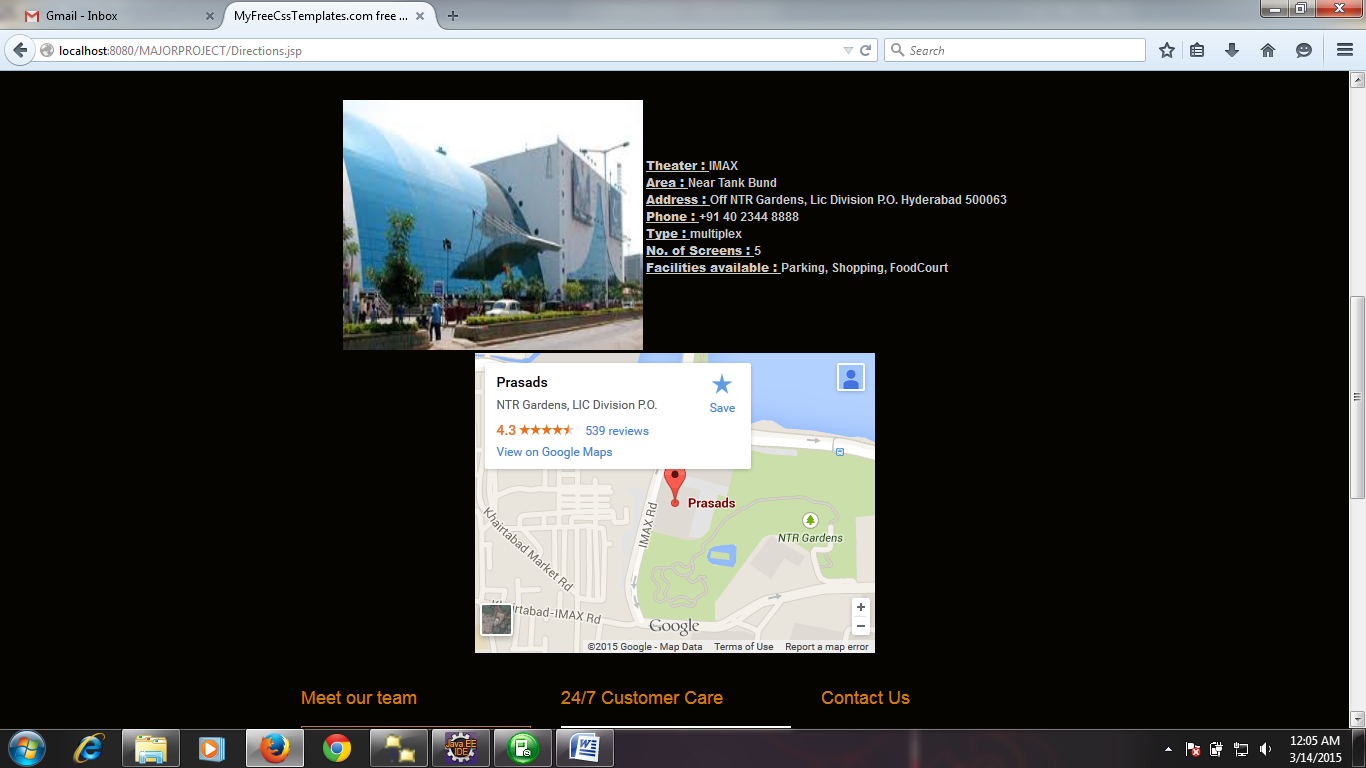
**Figure No.8.6:**Celebrities Information Page

**Step 7:** The users can view details about the various theaters here**.**

****

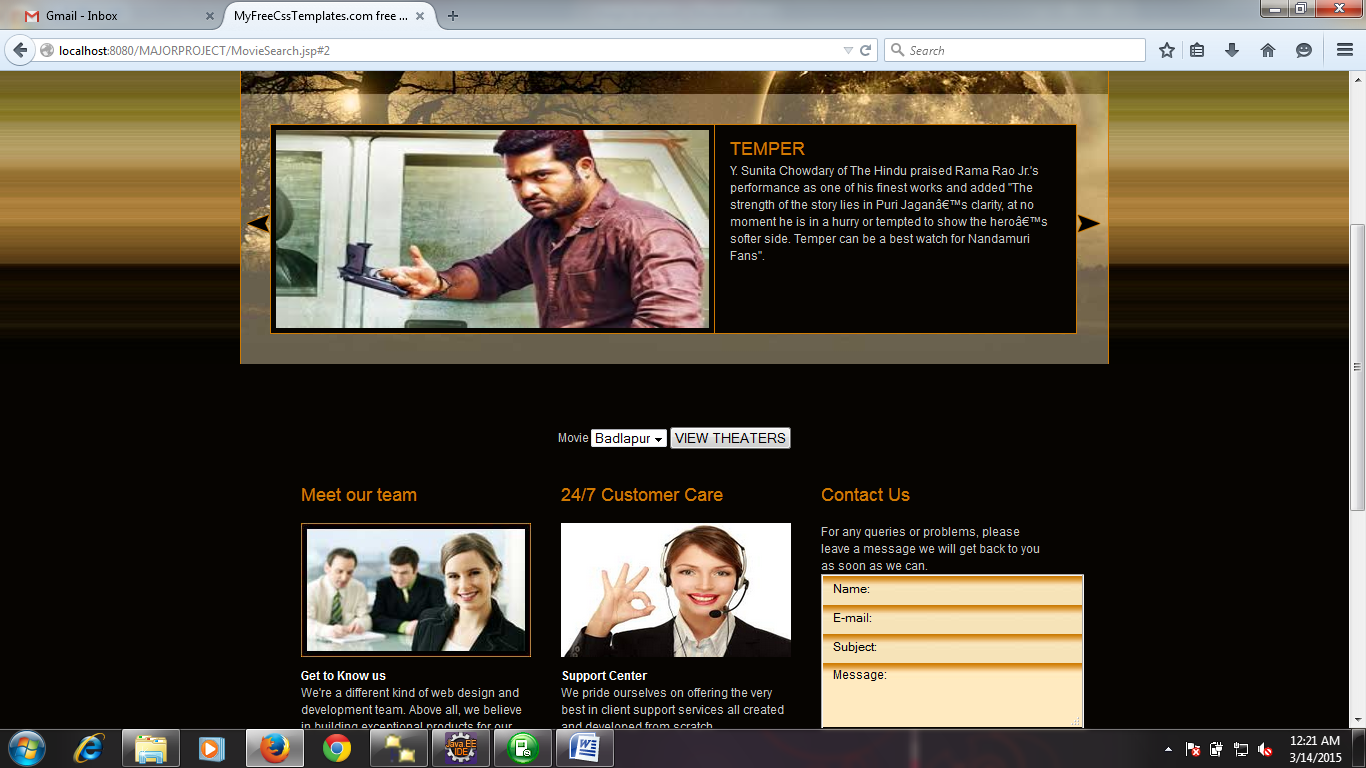
**Figure No.8.7:**Theater selection for details Page

**Step 8:** The users can view the movie details including the map to show directions.

****

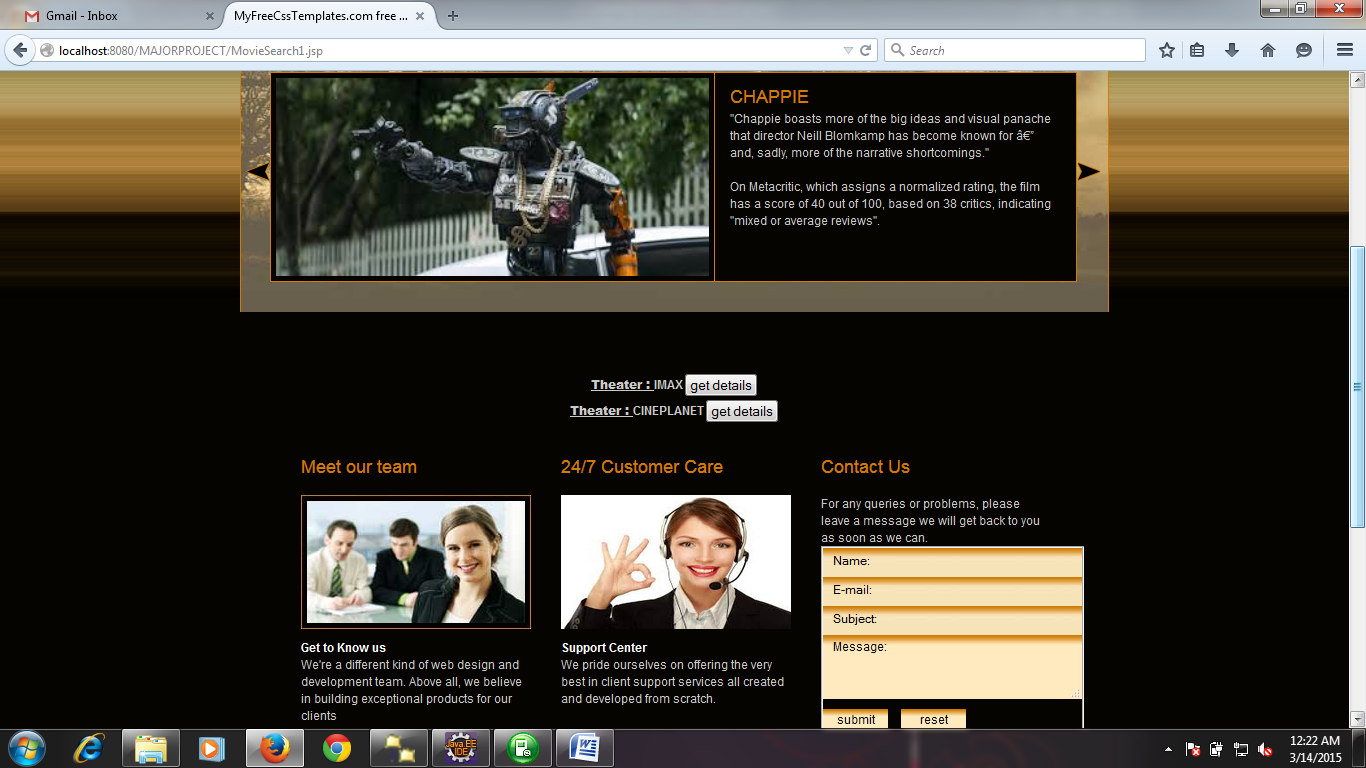
**Figure No.8.8:** Theater Details Page

**Step 9:** The users can view the theater details for the movies selected by them.

****

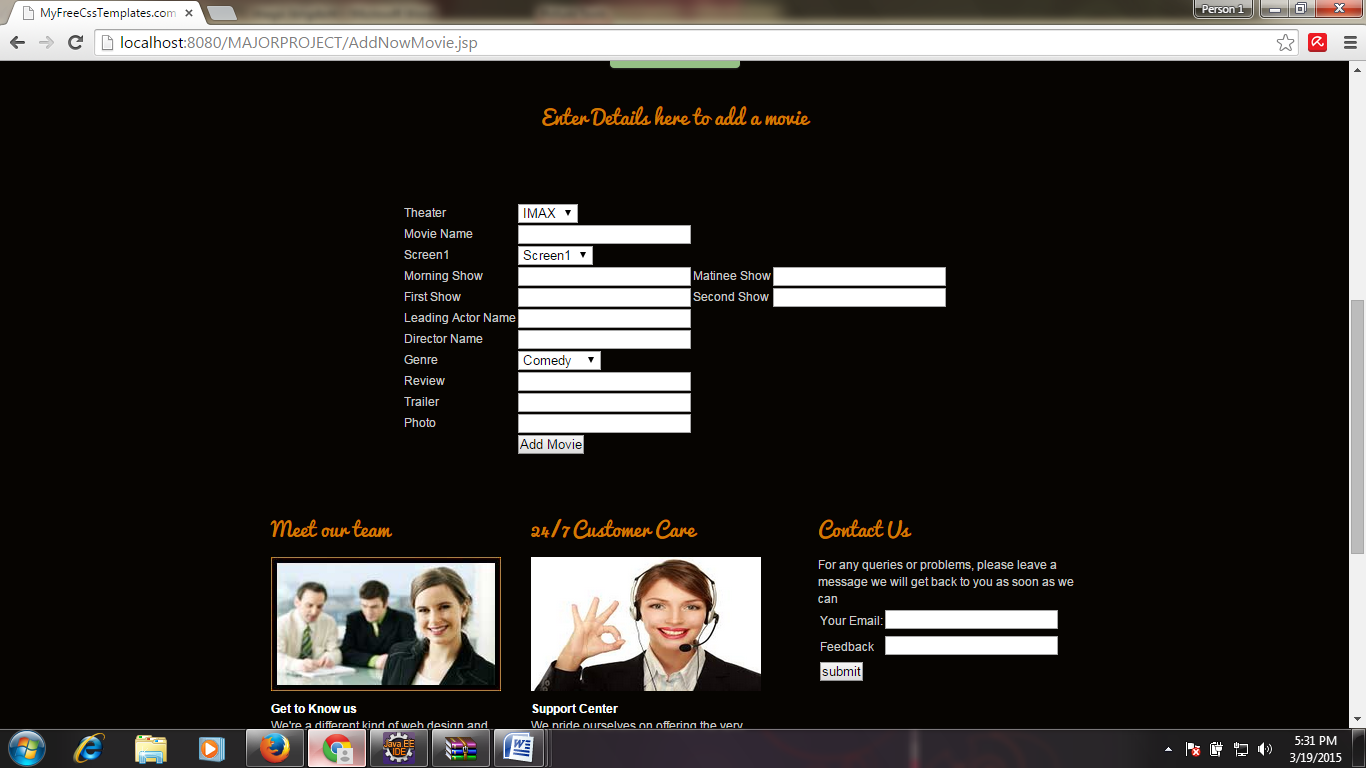
**Figure No.8.9:**Theaters for Movies Page

**Step 10:** The users can view the list of theaters for the movies they select**.**

****

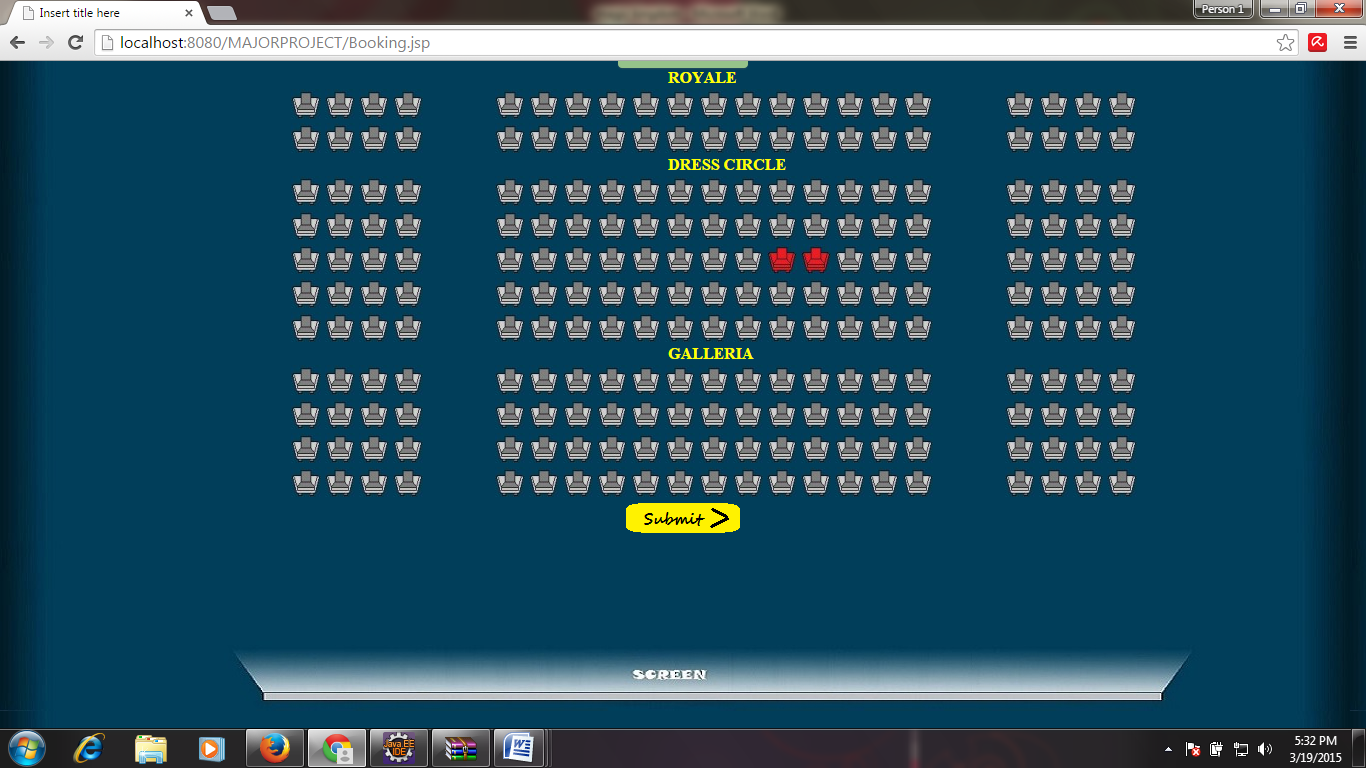
**Figure No.8.10:** Theaters list for movies Page

**Step 11:** The admin can add new movies into the database here.

****

**Figure No.8.11:** Add new movies Page

**Step 12:** The user can select seats for their selected movies here**.**

****

**Figure No.8.12:**Seat Selection Page

**9. CONCLUSION AND FUTURE ENHANCEMENT**

In conclusion, results indicate that an artisanal Gouda cheese producer located in the central coast of California could survive in the current economy. Minimal debt can be accrued through strategic funding, and a low cost price paired with a premium market price. Greater profit margins could be obtained by marketing the product in Santa Barbara and production occurring in a nearby town, such as Los Alamos. Furthermore, assuming no financial aid other than loans and capital funds, the initial start-up costs and expenses for an entire year could be paid off and the business could start making a profit in one year if the cheese market price was 11.99 per pound.

**9.1 Limitations of the system**

* The Administrator should always be active.
* The approver might always not check the mail to accept or decline the request.

**9.2 Future Enhancements**

We see the most important next step as enhanced testing of the “NSF Fraud Protection Model” model against known incidents and to build a larger data set of mapped products. We are in discussion with a major international retailer to undertake this work. It is to be hoped that given the scale of the task the industry will work collaboratively within its own ranks and with regulators to develop a better 65 understanding of fraud risk and the appropriate controls and interventions to minimize impact and the potential for harm. The original model itself stood up satisfactorily to various tests and comparisons with known incidents and the stakeholders own perceptions of high and low risk categories/products although it was necessary to amend the original choice of horizontal axis as research indicated the likelihood of detection was more important to the fraudster than ease of perpetrating the fraud. We are confident that we have developed a useful framework and the basis for further enhancement that will with support and investment eventually lead to a dynamic tool collecting key data in near real time that will help identify

**BIBLIOGRAPHY**

**Books**

**[1]** ELMASRI, R. AND NAVATHE, S.R. Elmasri and S. Navathe, Fundamentals of database systems, 1st ed. 1272 Seiten: [s.n.], 2017.

**[2]**J. Goodwill, Developing Java servlets, 1st ed. Indianapolis, Ind.: SAMS, 2001.

**[3]**D. Hougland and A. Tavistock, Core JSP, 1st ed. Upper Saddle River, N.J.: Prentice Hall, 2001.

**[4]**"Javatpoint - A Solution of all Technology", www.javatpoint.com, 2017. [Online]. Available: http://www.javatpoint.com/. [Accessed: 25- Feb- 2017].

**[5]**Y. Liang, Introduction to Java programming and data structures, 1st edition.

**Websites**

[1] [http://www.w3schools.com/](http://www.w3schools.com/%20)

[2] [http://www.cafevida.net/](http://www.cafevida.net/%20)

[3] [http://stackoverflow.com/](http://stackoverflow.com/%20)

[4] <http://en.wikipedia.org/wiki/Main_Page>