E-TICKET BOOKING

A Main Project submitted
in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE AND ENGINEERING

Submitted by

1. P.DURGA PRASAD REDDY (15PA1A05D7)

Under the esteemed guidance of
T.VAMSI KRISHNA
Assistant Professor

VISHNU
UNIVERSAL LEARNING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VISHNU INSTITUTE OF TECHNOLOGY

(Approved by AICTE, Accredited by NBA & NAAC and permanently affiliated to JNTU Kakinada)
BHIMAVARAM – 534 202
2018 – 2019

VISHNU INSTITUTE OF TECHNOLOGY

(Approved by AICTE, Accredited by NBA & NAAC and permanently affiliated to JNTU Kakinada)

BHIMAVARAM-534202

2018-2019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "EVENT MANAGEMENT SYSTEM", is being submitted by ***P.DURGA PRASAD REDDY*** bearing the **REGD.NO: 15PA1A05D7** submitted in fulfilment for the award of the degree of **"BACHELOR OF TECHNOLOGY"** in **"COMPUTER SCIENCE AND ENGINEERING"** is a record of bonafide work carried out by them under my guidance and supervision during the academic year 2018-2019 and it has been found worthy of acceptance according to the requirements of university.

Internal Guide | **Head of the Department**

T.Vamsi Krishna | Dr. Sumit Gupta

External Examiner

ACKNOWLEDGEMENT

It is nature and inevitable that the thoughts and ideas of other people tend to drift in to the subconscious due to various human parameters, where one feels acknowledge the help and guidance derived from others. We acknowledge each of those who have contributed for the fulfilment of this project.

We take the opportunity to express our sincere gratitude to **Dr.D.Suryanaryana,** director and principal, VIT**,** Bhimavaram whose guidance from time to time helped us to complete this project successfully.

We are very much thankful to **Dr Sumit Gupta**, Head of the Department, Department of Computer Science and Engineering for his continuous and unrelenting support and guidance. We thank and acknowledge our gratitude to her for his valuable guidance and support expended to us right from the conception of the idea to the completion of this project.

We are very much thankful to **T.Vamsi Krishna**, Assistant Professor, our internal guide whose guidance from time to time helped us to complete this project successfully.

P.DURGA PRASAD REDDY (15PA1A05D7)

Project Associates

ABSTRACT

This project is aimed at developing an online ticket reservation system for a Cinema Hall. The Ticket Reservation System is an Internet based application that can be accessed throughout the Net and can be accessed by anyone who has a net connection. This application will automate the reservation of tickets.This online ticket reservation system provides a website for a cinema hall where any user of internet can access it. This cinema hall is a multiplex with a screen and screen has different types of classes/seats. Only 50% of seats can be reserved online. User is required to login to the system for booking the tickets.The website provides complete information regarding currently running movies on all the screens with details of show timings, available seats and fare charges of different classes. Seats can be reserved for different classes as well for same show and screen also. Ticket reservations are done using credit card and can be cancelled if needed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF DIAGRAMS

INTRODUCTION

1. INTRODUCTION

1.1.1 Introduction

Cinema-going is one of the most popular out-of-home cultural activities, affecting a serious of social, economic and cultural phenomena in modern societies. Cinemas are considered to be an integral part of cities and they contribute to the definition of a local geography and identity. They also contribute to the preservation of the collective memory, since they constitute a significant social and cultural practice linked to a specific place, which acts as a common reference or landmark for many individuals. Through this project we present a comprehensive solution for ticket booking in multiplexes. Theater management system, an online ticket selling software that is easy to understand, easy to use and offers the simplicity of fast point-and-click service to the customers. This powerful software program is specifically designed for theater owners, to sell tickets online. This intuitive visual interface makes day-to-day aspects of selling, exchanging, refunding, and reporting fast and easy for both the user and administrators. Theater management controls all back-end functionalities like, movie details, ticket rate and show time, customer information and sales history saved in a database, etc. Theater admin manages the report details like counter wise report, daily, weekly, monthly report and movie report etc.

1.2 LIMITATIONS

Our project has the following limitations:- 1. In this system the customer will not select seat number. The authorized people in this process will be the employee who works at the cinema. 2. In this system the customer can book the tickets only for the present day. Advanced booking for the upcoming movies (2 or 3 months before the movie is released) is not included but can be included later. 3. Also, a printing system will not be in this project. In future, a ticket printing system can be integrated. 4. The customers who have booked the tickets must reach the multiplex 30 minutes prior to the show time or else ticket will get cancelled

SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2.1 Existing System

Manual ticketing system have some disadvantages

- One should go to movie theatre to get a E-ticket
- User have to wait in queue to get a movie ticket
- User did not get ticket in desired row.

2.2 Proposed System

People are becoming too busy to go to movie theatre to book the movie Tickets. The main idea of this work is to provide ease and comfort to customers while booking E- ticket and it also resolves the problems that the customers has to wait in Queue.

To build this project, many languages are used to complete this project like Python, JavaScript,Jquery and also used frameworks like Django etc. It takes lot of time to complete the project and also it is hard to identify the errors. Many of the developers have to work on the code and sometimes, there would be a necessity to optimize the code.

2.3 Feasibility Report

Once the problem is clearly understood, the next step is to conduct the feasibility study. The objective is to determine whether or not the proposed system is feasible. At the present moment, the application is entirely functional, save the few minor bugs which are bound to present themselves during more extensive testing. A user is currently able to register and log in to the site and book a ticket

Much of what is left to do focuses not on improving functionality, but rather on improving user experience by creating richer user interfaces for the user to interact with and modifying the application’s icons and colour schemes to make them more pleasing to look at and use.

This application is technically feasible as well as economically feasible. It is observed that the proposed system is very user friendly and since the system is built with enough help, even persons with knowledge can utilize the system very easy. This project will be able to view from anywhere having a proper internet connection to the system.

2.4 Software Requirements

Tool	:	Visual Studio Code
Operating System	:	WINDOWS/Linux
User Interface	:	HTML, CSS, JavaScript,Python
Framework	:	Django

2.5 Hardware Requirements

Processor	:	ANY
RAM	:	2.00 GB

SYSTEM DESIGN

3. SYSTEM DESIGN

3.1 Introduction

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. System design could be seen as the application of system theory to product development.

Architectural Design

The architectural design of a system emphasizes the design of the system architecture that describes the structure, behaviour and more views of that system and analysis.

Logical Design

The logical design of a system pertains to an abstract representation of the data flows,inputs and outputs of the system. This is often conducted via modelling, using an over-abstract(and sometimes graphical) model of the actual system. In the context of systems, designs are included. Logical design includes entity-relationship diagrams (ER diagrams).

Physical Design

The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified/ authenticated, how it is processed, and how it is displayed. In physical design, the following requirements about the system are decided:

1. Input requirements
2. Output requirements
3. Storage requirements
4. Processing requirements

5. System control and backup or recovery

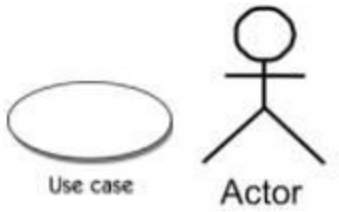
3.2 UML Diagrams

UML stands for **U**nified **M**odelling**L**anguage is a general-purpose, standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. It is defined as a simple modelling mechanism to model all possible practical systems in today's complex environment.

3.2.1 Use case Diagram

The purpose of use case diagram is to capture the dynamic aspect of a system. These are used to gather the requirements of a system including internal and external influences. Use case is represented as an eclipse with a name inside it.

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors.



3.2.2 Sequence Diagram

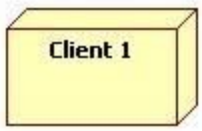
Sequence diagram emphasizes on time sequence of messages. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

- To capture the dynamic behaviour of a system.
- To describe the message flow in the system.
- To describe the structural organization of the objects.
- To describe the interaction among objects

3.2.3 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. It consists of nodes and their relationships.

- Visualize the hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe the runtime processing nodes.



3.2.4 Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. This flow can be sequential, branched, or concurrent.

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

3.2.5 Collaboration Diagram

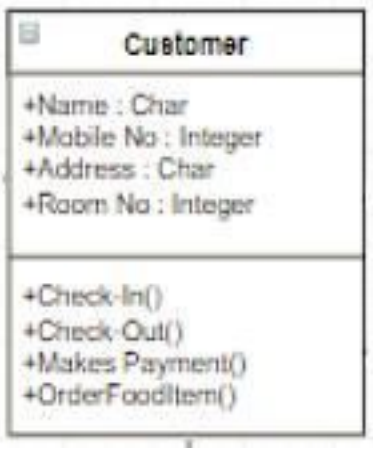
A collaboration diagram, also called a communication diagram or interaction diagram. Collaboration diagram emphasizes on the structural organization of the objects that send and receive messages.

- To capture the dynamic behaviour of a system.
- To describe the message flow in the system.
- To describe the structural organization of the objects.
- To describe the interaction among objects.

3.2.6 Class Diagram

The purpose of class diagram is to model the static view of an application. These are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.



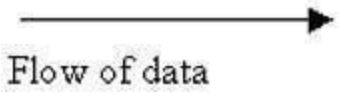
3.3 Data Flow Diagrams

A graphical tool used to describe and analyse 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.

3.3.1 Basic Notation

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.

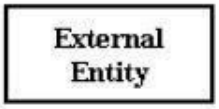


Flow of data

Process: People, procedures, or devices that use or produce data. The physicalcomponent is not identified.

Process

Source: External sources or destination of data, which may be People, programs,Organizations or other entities.



3.4 ER Model (Entity–Relationship)

The E-R model can be used to describe the data involved in a real world enterprise in terms of objects and their relationships. An ER diagram is a means of visualizing how the information a system produces is related.

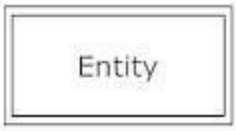
There are five main components of an ERD:

- Entities
- Actions
- Attributes
- Connecting lines

Entities

Entities, which are represented by rectangles. An entity is an object or concept about which you want to store information.

A **weak entity** is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.



Actions

Actions, which are represented by diamond shapes, show how two entities share information in the database.

Attributes

Attributes, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity.

A **multivalued attribute** can have more than one value. For example, an employee entity can have multiple skill values.

A **derived attribute** is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.

Connecting lines

Connecting lines, solid lines that connect attributes to show the relationships of entities in the diagram.

Relationships in E-R Diagrams:

There are three types of relationship in ER diagrams,

- One-to-one
- One-to-many
- Many-to-Many

2. One-to-one (1:1)

A one-to-one relationship between two entities indicates that each occurrence of one entity in the relationship is associated with a single occurrence in the related entity. There is a one-to-one mapping between the two, such that knowing the value of one entity gives the value of second.

2. One-to-many (1:M)

A one-to-many or a Many-to-many relationship between two entities indicates that a single occurrence of one entity is associated with one or more occurrence in the related entity.

3. Many-to-many (M:M)

A Many-to-many relationship between two entities indicates that either entity participating in the relation may occur one or several times.

Uses:

1. These models can be used in database design.
2. It provides useful concepts that allow us to move from an informal description to precise description.
3. This model was developed to facilitate database design by allowing the specification of overall logical structure of a database.

4. It is extremely useful in mapping the meanings and interactions of real world enterprises onto a conceptual schema.

5. These models can be used for the conceptual design of database applications.

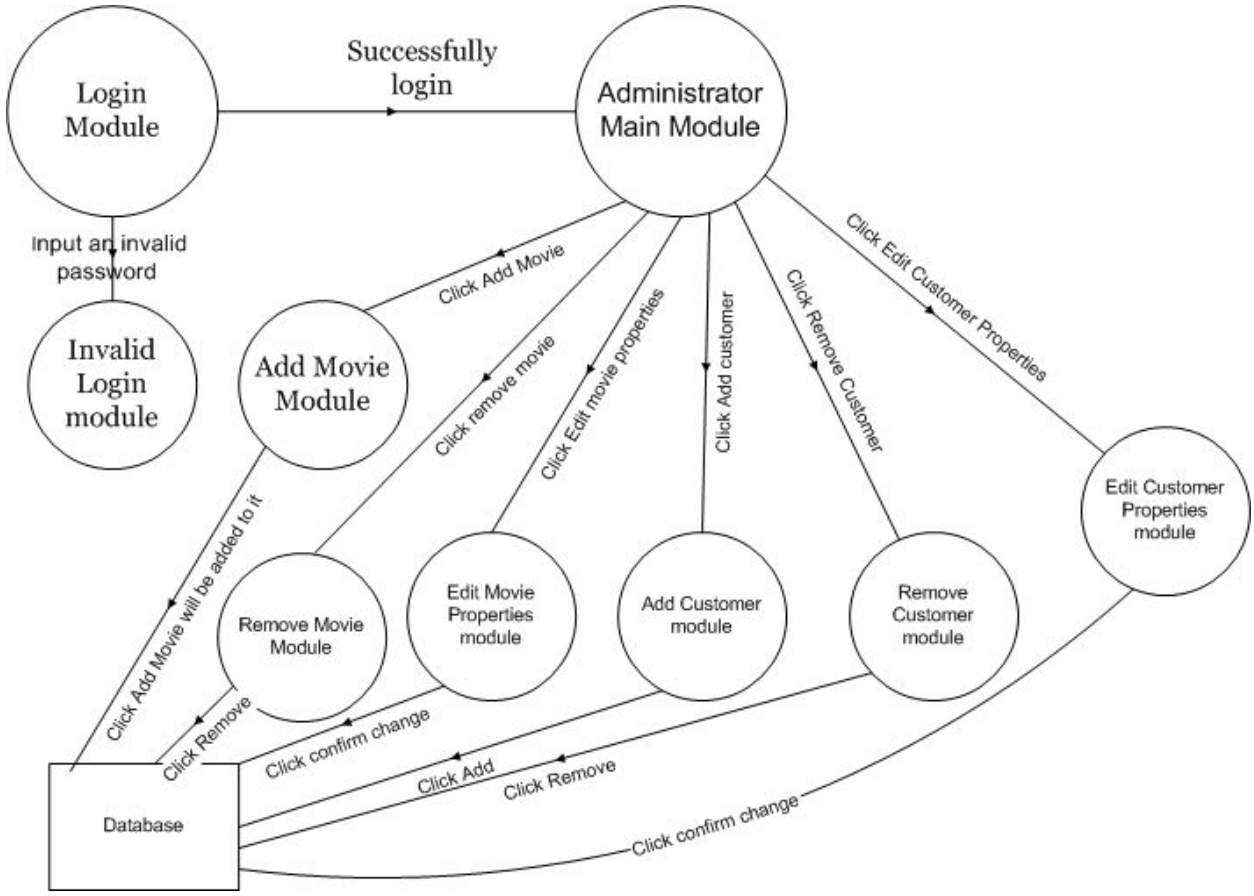


Fig 1.1 Data Flow Diagram

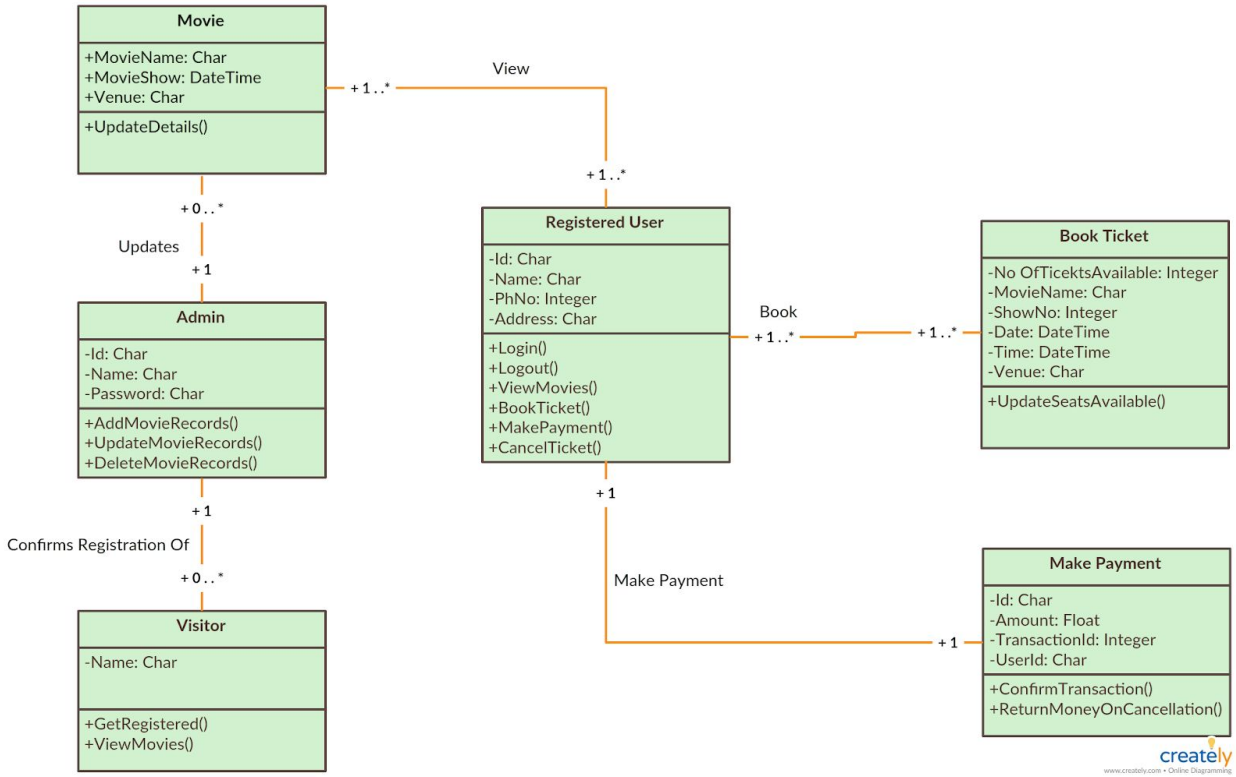


Fig 1.2 Class Diagram

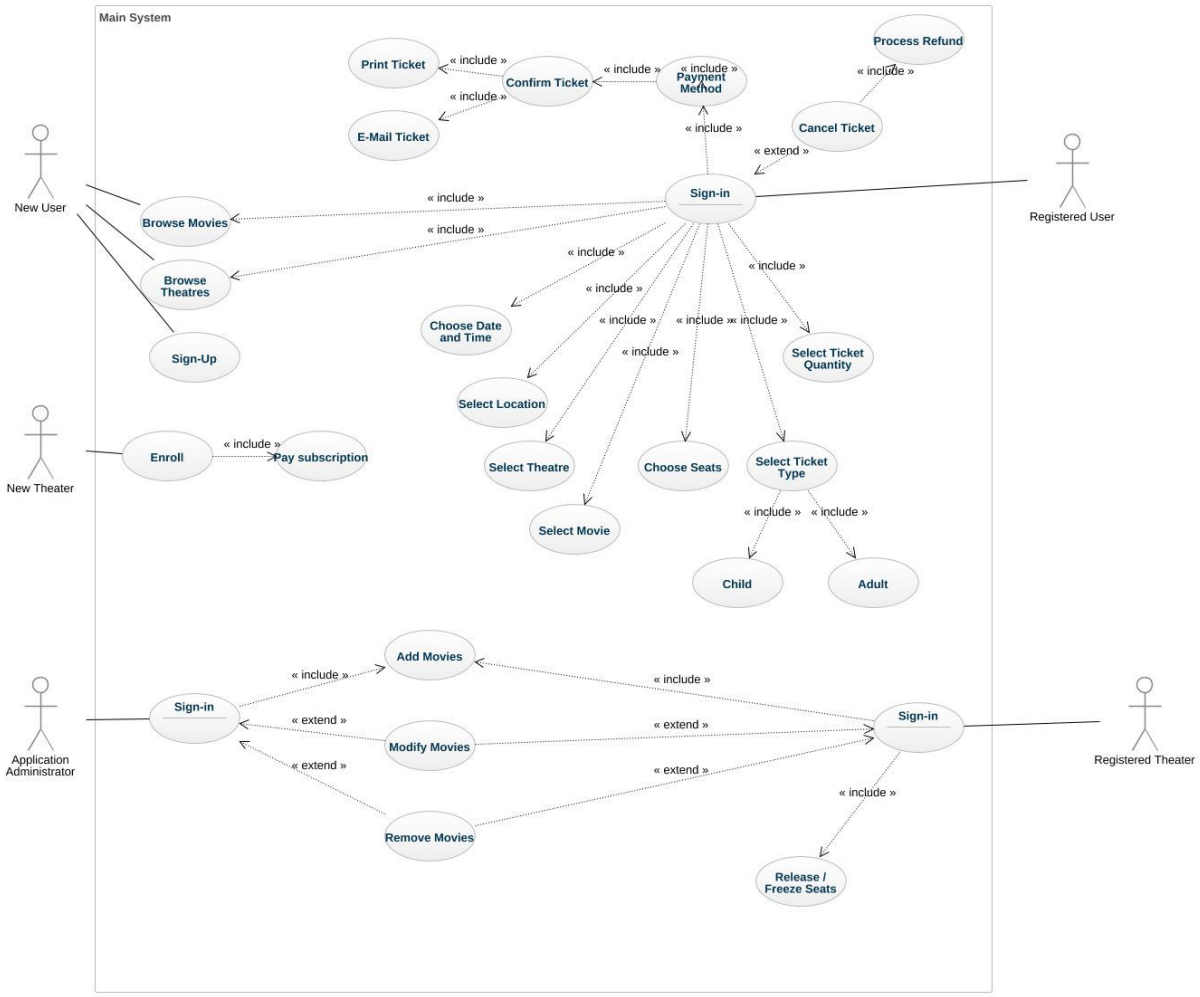


Fig 1.3 UseCase Diagram

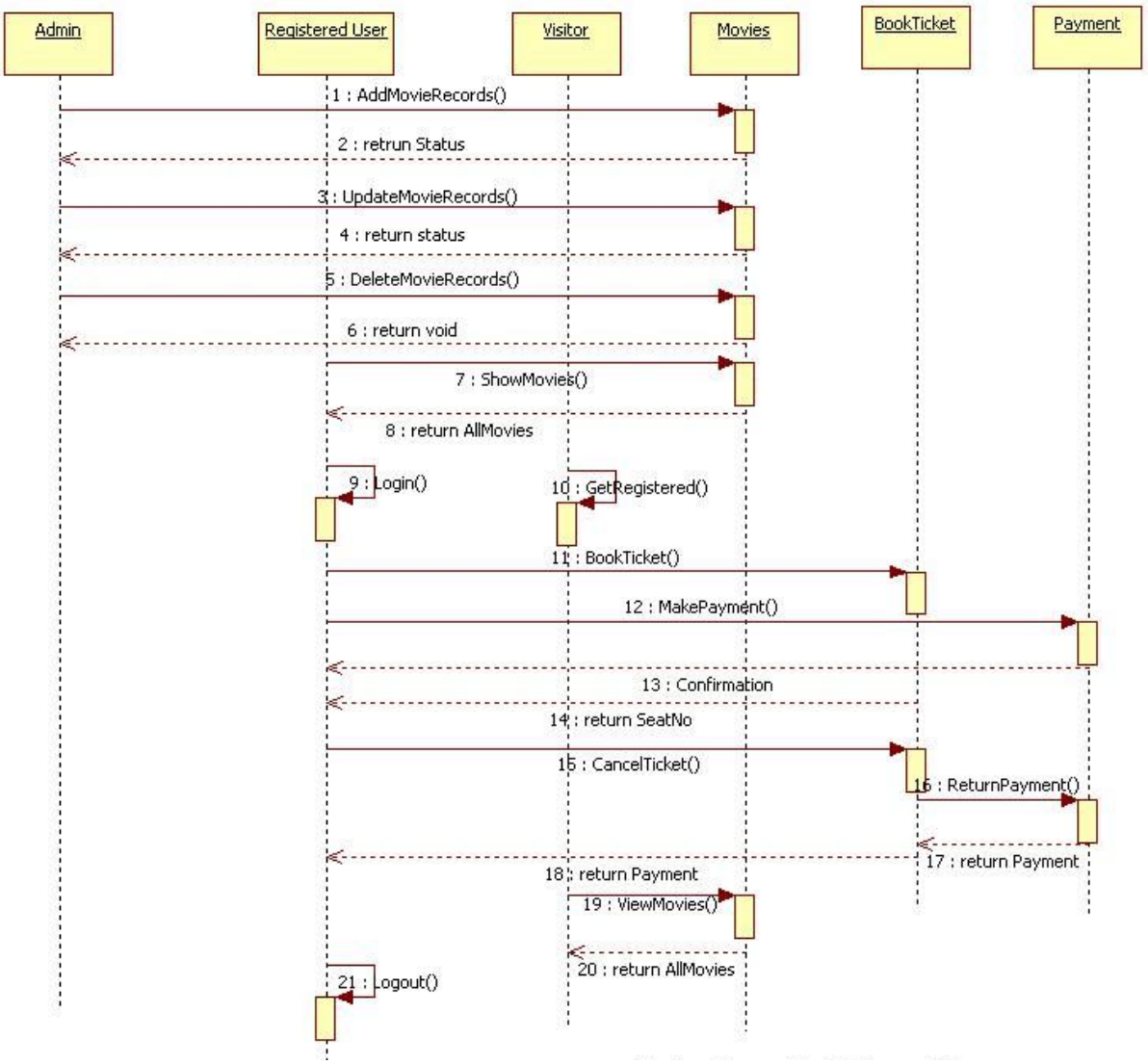


Fig 1.4 Sequence Diagram

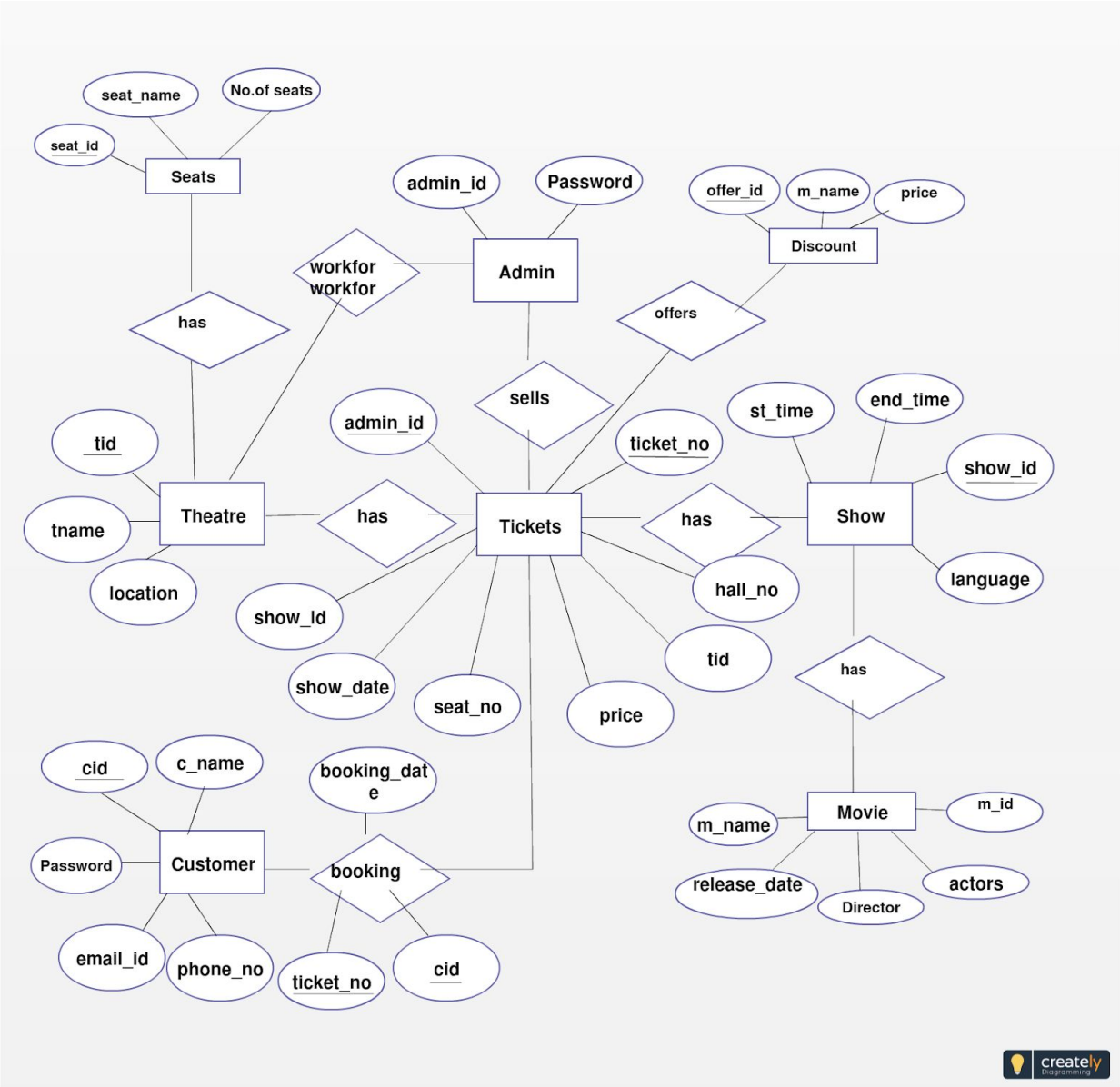


Fig 1.5 ER Diagram

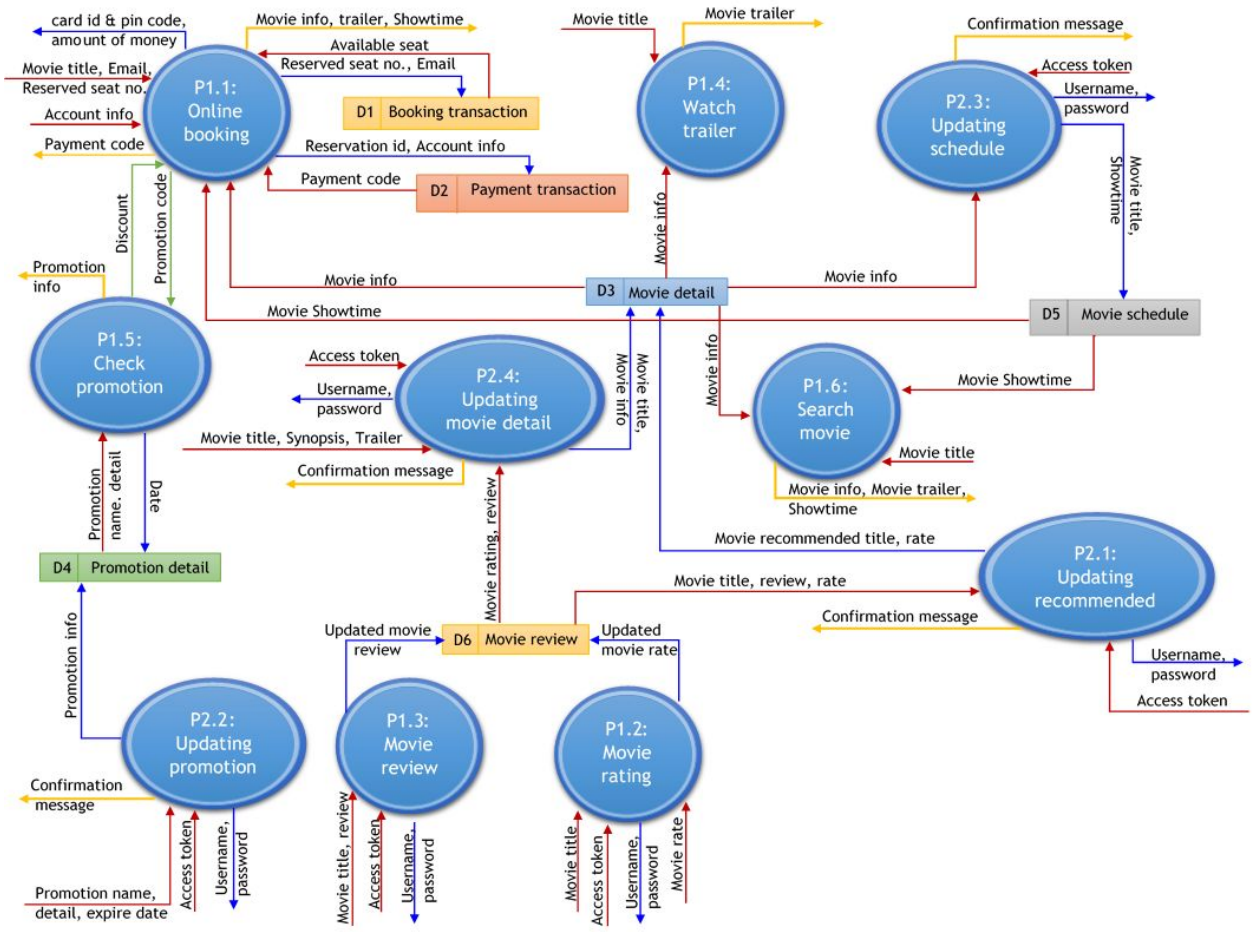


Fig 1.6 Data Flow Diagram

MODULE DESCRIPTIONS

4. MODULE DESCRIPTION

4.1 MODULE 1 - OWNER PROFILE

The owner has full access to the system. The owner can graphically view all the details, and he has the authority to change the Cost of tickets, ticket Availability and much more. The owner is provided with an id and password.

He can put various kinds of notifications on the website. He can also check the details of seats left and seats reserved. He can also monitor the transactions made throughout the day and the month and an algorithm will check the progress in the booking of tickets.

He also can give various kinds of offers to the viewers. He can also keep some seats hidden from the users for their offline reservation. He can add new movies and their show timings. Also, he can delete some video from the website.om the website.

4.2 MODULE 2 - MANAGER PROFILE

In Manager Profile, who so ever is the Manager he must log-in and then he can book the tickets of the film for the viewers, and this will serve the viewers in offline mode.

When he will log-in into the system, his attendance will be taken, and it will be easy for the owner to monitor them. Here they will also get a notification of booked room so that they can beforehand maintain the place.

4.3 MODULE 3 - TICKET BOOKING

The viewer can quickly search for the desired movie and the number of seats from the various options available.This all will be so user-friendly that the audience will not find any trouble in booking the movie ticket.

Once the available movies are searched, and viewer finds the movie of his choice then this module helps the audience to book the film. If everything is done properly, he will be asked to

pay partial or complete money. And then he will be forwarded to Payment Gateway where he can quickly pay Online.

4.4 MODULE 4- MOVIE MODULE

This module will help the viewer to see the details of the movies available for the show. He can even search for any specific film. The number of seats available will be displayed to the viewer along with their timings and the cost of different types of tickets.

Then he can reserve the required tickets. The information of upcoming movies will also be displayed to the viewers.

4.5 MODULE 5- PAYMENT DETAILS

After filling the details like the name of the movie, category of cards, the timings of the show and the number of seats to be reserved, the viewers need to pay the money using different options available like cash/net banking/ATM card/credit/debit card.

Payment portal would use a payment gateway to remove the fees. Once payment will clear receipt of payment will be generated automatically and the message will be sent to the viewer on his mobile.

Once payment will clear receipt of payment will be created automatically and the message will be directed to the viewer on his mobile.

IMPLEMENTATION

5.IMPLEMENTATION

5.1 TECHNOLOGIES USED

- HTML
- CSS
- JAVASCRIPT
- JQUERY
- PYTHON
- DJANGO(framework)
- BOOTSTRAP
- MYSQL(database)
- APACHE(server)

5.2 SAMPLE CODE

models.py

```
from __future__ import unicode_literals

from django.db import models

# Create your models here.
class Tickets(models.Model):
    id = models.IntegerField(primary_key=True)
    user_name = models.TextField(max_length=100, null=True)
    seat_count = models.TextField(max_length=100, null=True)
    seat_numbers = models.TextField(max_length=100, null=True)
    total_amount = models.TextField(max_length=100, null=True)
    date_time = models.DateTimeField(auto_now=True)

    def __str__(self):
       return
"{},{},{},{},{},{}".format(self.id,self.user_name,self.seat_count,self.sea
t_numbers,self.total_amount,self.date_time)
class AddPrice(models.Model):
```

```
    price = models.IntegerField(primary_key=True)
    def __str__(self):
       return "{}".format(self.price)
```

urls.py

```
from django.conf.urls import url, include
from django.contrib.auth import views
from mysite.core import views as core_views
from django.contrib import admin


urlpatterns = [
   url(r'^admin/', admin.site.urls),
   url(r'^$', core_views.home, name='home'),
   url(r'^login/$', views.LoginView.as_view(template_name='login.html'),
name='login'),
   url(r'^logout/$', views.LogoutView.as_view(next_page='login'),
name='logout'),
   url(r'^signup/$', core_views.signup, name='signup'),
   url(r'^booking/$', core_views.booking, name='booking'),
   url(r'^bookings/$', core_views.bookings, name='bookings'),
   url(r'^confirm/$', core_views.confirm, name='confirm'),
   url(r'^ticket/$', core_views.ticket, name='ticket'),
   url(r'^auth/', include('social_django.urls', namespace='social')),

   url('^', include('django.contrib.auth.urls')),
]
```

views.py

```
from django.contrib.auth.decorators import login_required
from django.contrib.auth import login, authenticate
from django.contrib.auth.forms import UserCreationForm
from django.shortcuts import render, redirect
```

```
from django.shortcuts import render
from .models import Tickets, AddPrice

@login_required
def homee(request):
   return render(request, 'base.html')
def bookings(request):
   booking_view =request.POST.get('booking_view')
   tickets = Tickets.objects.filter(user_name=booking_view)
   context = {'tickets': tickets}
   return render(request, 'history.html',context=context)
def ticket(request):
   tickets = Tickets.objects.all().order_by('-id')[:1]
   context = {'tickets': tickets}
   return render(request, 'ticket.html',context=context)
def home(request):
   price = AddPrice.objects.all()[:1]
   context = {'price': price}
   return render(request, 'home.html', context=context)
def booking(request):
   return render(request, 'booking.html')
def confirm(request):
   print("hello form is submitted")
  # user_name = request.POST['user_name']
   user_name=request.POST.get('user_name')
   seat_count = request.POST.get('seat_count')
   seat_numbers = request.POST.get('seat_numbers')
   total_amount = request.POST.get('total_amount')

   tickets =
Tickets(user_name=user_name,seat_count=seat_count,seat_numbers=seat_number
s,total_amount=total_amount)

   tickets.save()
   tickets =
Tickets.objects.filter(user_name=user_name).order_by('-id')[:1]
   context = {'tickets': tickets}
```

```
    return render(request, 'dis.html',context=context)



def signup(request):
    if request.method == 'POST':
        form = UserCreationForm(request.POST)
        if form.is_valid():
            form.save()
            username = form.cleaned_data.get('username')
            raw_password = form.cleaned_data.get('password1')
            user = authenticate(username=username, password=raw_password)
            login(request, user)
            return redirect('home')
    else:
        form = UserCreationForm()
    return render(request, 'signup.html', {'form': form})
```

index.html

```
<head>

        <link rel="icon" href="https://i.ibb.co/JqdGK2K/2.png">

    <title>Vishnu Miniplex</title>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.cs
s">

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></s
cript>
```

```
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
</script>

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.cs
s">

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></s
cript>

    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
</script>

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></
script>

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min
.css">

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.bund
le.min.js">

    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.6.3/css/all.css">


</head>


<body>

    <header>
```

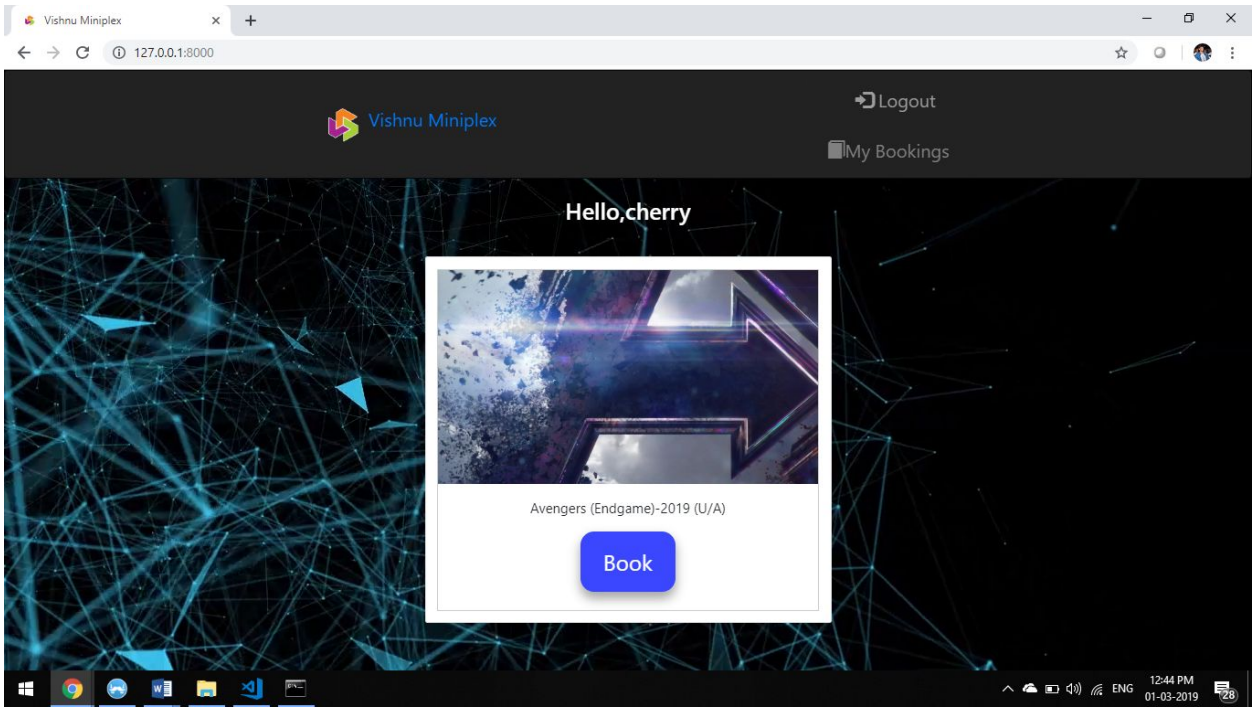
```
        <nav class="navbar navbar-inverse">
            <div class="container-fluid">
                <div class="navbar-header">
                    <img  src="https://i.ibb.co/JqdGK2K/2.png"
                 width="50px" height="50px">
                    <a class="title" style="text-align:center;" href="{%
url 'home' %}">Vishnu Miniplex</a>
                </div>
                <ul class="nav navbar-nav">
                    {% if user.is_authenticated %}
                    <ul class="nav navbar-nav navbar-right">

                        <li><a style="float: right;" href="{% url 'logout'
%}"><span class="glyphicon glyphicon-log-in"></span> Logout</a> My
Bookings</a></li>
                        <form id="bookingsform" method="post" action="{%
url 'bookings' %}">
                                {% csrf_token %}
                                <input type="hidden" id="custId"
name="booking_view" value="{{user.username}}"><br>

                                <li><a class="titlee" style="float:
right;" href="#"
```

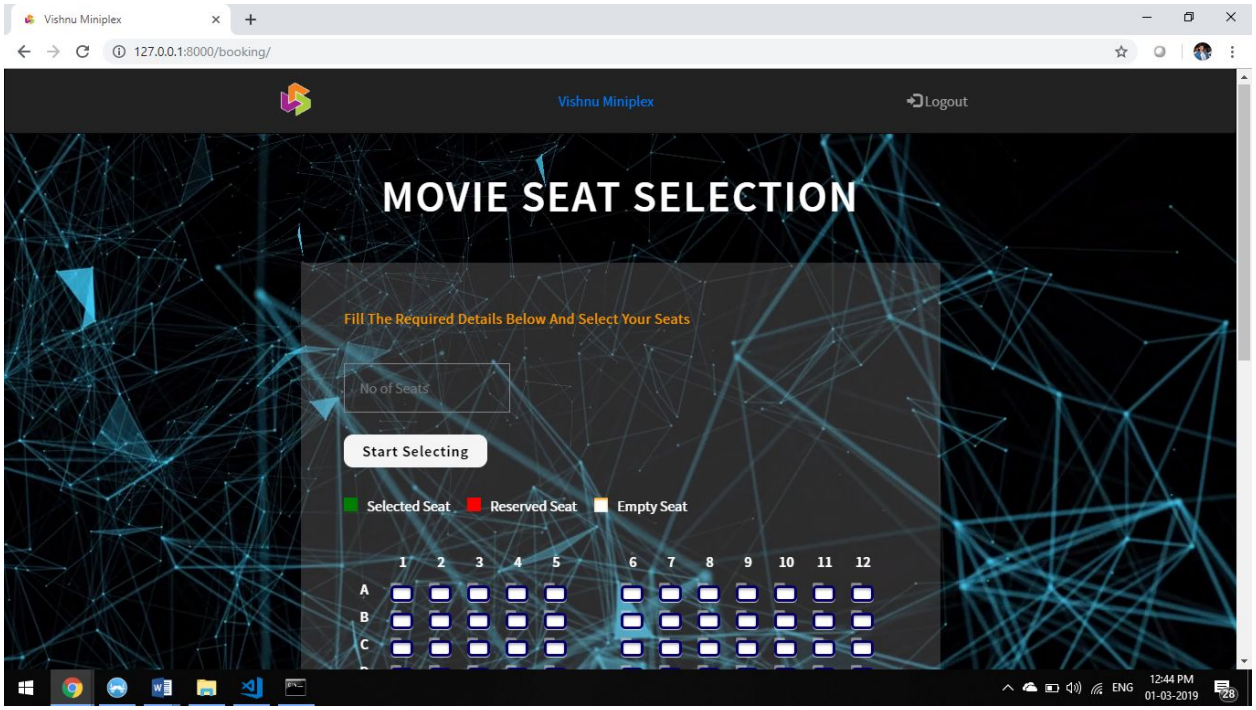
```
onclick="document.getElementById('bookingsform').submit();"><span
class="glyphicon glyphicon-book"></span>My Bookings</a></li>
                                </form>



                            {% else %}
                            <li><a style="float: right;" href="{% url 'login'
%}"><span class="glyphicon glyphicon-log-in"></span> Login</a>
                                <a style="float: right;" href="{% url 'signup'
%}"><span class="glyphicon glyphicon-user"></span> Sign Up</a>
                                </li>
                        </ul>
                        {% endif %}
                </div>
            </nav>


    </header>
    <video autoplay loop id="video-background" muted plays-inline>
        <source src="css/surfers.mp4" type="video/mp4">
       </video>


    {% block content %}
    {% if user.is_authenticated %}


    <h1>Hello,{{ user.username }} </h1>
```

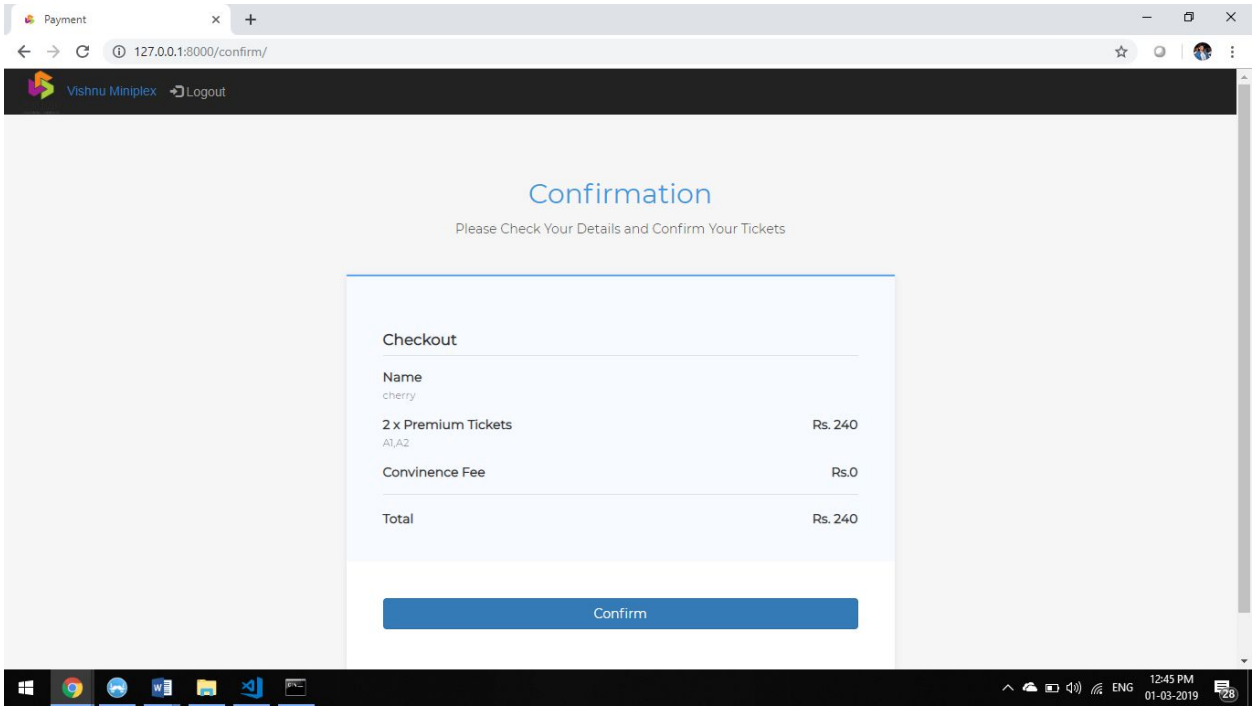
```
    <img src="{{ user.socialaccount_set.all.0.get_avatar_url }}" />


    <div class="container">

        <div class="row">

            <div class="col-sm-9 col-md-7 col-lg-5 mx-auto">

                <div class="card card-signin my-5">

                    <div class="card-body">

                        <!-- Movie Div -->

                        <div class="gallery">

                            <a target="#" href="#">


                                <img
src="http://hdqwalls.com/download/avengers-end-game-poster-o0-5120x2880.jp
g" alt="Cinque Terre" width="800" height="400">

                            </a>

                            <div class="desc">Avengers (Endgame)-2019
(U/A)



                            </div>

                            <center><a href="{% url 'booking' %}"
class="button button2" value="Book"> Book</a></center>

                        </div>

                        <h2></h2>
```

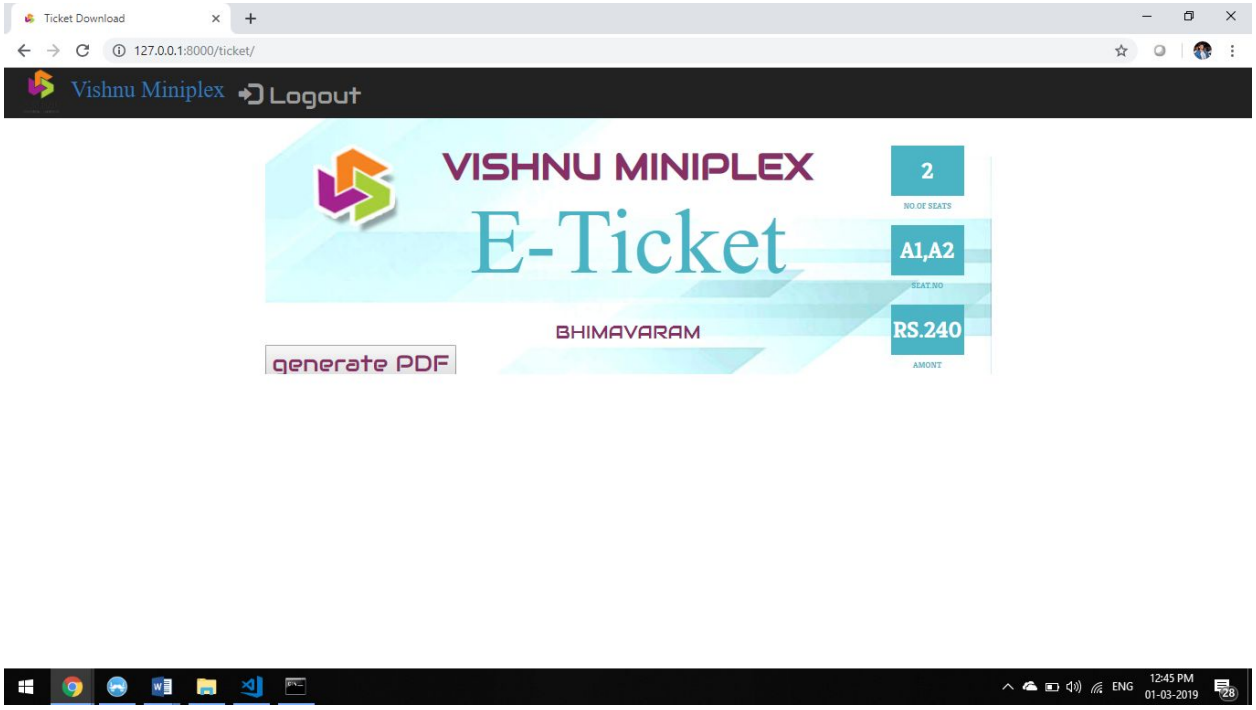
```
                </div>
            </div>
        </div>
    </div>
</div>


{% else %}


{% endif %}
{% endblock content %}
```

5.3 SCREENSHOTS OF WEB PAGES






Payment
127.0.0.1:8000/confirm/
Vishnu Miniplex
Logout
Confirmation
Please Check Your Details and Confirm Your Tickets
Checkout
Name
cherry
2 x Premium Tickets
A1,A2
Rs. 240
Convinence Fee
Rs.0
Total
Rs. 240
Confirm
ENG
12:45 PM
01-03-2019


Ticket Download
127.0.0.1:8000/ticket/
Vishnu Miniplex
Logout
VISHNU MINIPLEX
E-Ticket
BHIMAVARAM
generate PDF
2
NO.OF SEATS
A1,A2
SEAT.NO
RS.240
AMONT
ENG
12:45 PM
01-03-2019

TESTING

6. SYSTEM TESTING

Software testing is a process of executing a program or application with the intent of finding the software bugs. It can also be stated as the process of validating and verifying that a software program or application or product: Meets the business and technical requirements that guided its design and development.

Testing is required for an effective performance of software application or product. It's important to ensure that the application should not result into any failures because it can be very expensive in the future or in the later stages of the development. It's required to stay in the business.

6.1 Software Testing Levels

6.1.1 Unit Testing

A level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.

6.1.2 Integration Testing

A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. System testing a level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

6.1.3 Acceptance Testing

A level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

6.2 Black Box Testing

Black Box Testing also known as behavioural testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester. .

6.3 White Box Testing

White Box Testing also known as glass box testing is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality i.e. black-box testing. It is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester.

CONCLUSION

7.CONCLUSION

Nowadays, traditional reservation ways of cinema ticketing is dying. It's new age where technology dominates human life. With the software and technological devices, exceptions are reduced and even terminated. Also, people prefer easy, quick and safe way for every part of his life. This project is designed to meet the requirements of a cinema ticket booking system. It has been developed in PHP and the database has been built in My SQL server keeping in mind the specifications of the system. In our project: with this cinema ticketing system; cinema companies can satisfy comfortable facilities to their customers. The relationship between cinema manager, employee, and customer satisfy a good communication to complete ticketing process. With this platform we developed, we are hoping to reduce time wasting, avoid misunderstandings, provide easy data flow, customer pleasure, and less hard work. We believe that we have accomplished our goals and satisfied with the code we developed

BIBLIOGRAPHY

8.BIBLIOGRAPHY

Web References

1. https://stackoverflow.com
2. https://www.w3schools.com
3. https://djangodocumentation.com