**System Design Document**

Project: Playhouse Theatre  
By: Team Java

1 INTRODUCTION

1.1 PURPOSE  
The main purpose is to replace a current paper-based booking system with a new box office application which will help customers to see all the require information about the shows, book it online. It will help Playhouse Theatre to easily manage all the advertising and update any information and control all the database and extract any required information for any marketing and financial purposes.

1.2 DESCRIPTION

The new online system will allow customers to see all the information regarding the shows and book it online. Customers will also be able to see any promotions and future shows. It will help theatre staff to manage all the bookings, advertising and any promotions and update any content or any related information for the shows online without any hassle including any refunds or seats change or availability of the seats. It will save a lot of time and will help staff to concentrate on other tasks. All this information will be saved in a database and can easily extract any information according to the specific needs like for financial reports, weekly or monthly sales etc.

1.3 SCOPE  
The focus will be providing all the information for the shows online and promote online bookings of current and future shows.

User stories:

1. **Theatre Customer:** As a theatre customer I should be able to see all the information regarding shows and book tickets online in advance to save my time.
2. **Staff:** As a staff, I should be able to see all the required information regarding shows and manage booking information (create, update, cancel) including selling of tickets over a counter to give better customer service.
3. **Marketing Manager:** As a marketing manager, I should be able to update or publish any information related to the shows online or on other social media to attract more customers and grow the business to another level.
4. **Theatre Manager:** As a theatre manager, I should be able to set the ticket prices and set any promotional prices or discounts or mange any refunds to compete with the other competitors in the market.

1.4 TARGET USERS  
The users of the application for this release will be:

1. **Theatre Customer**: They will explore shows and other information online and book online tickets.
2. **Box Office Clerk/ Staff:** This system will help staff to see all the information and update it (refund, seat change or customer service) for better performance and will increase productivity in the team.
3. **Admin User**: They will be able to excess any data from the database which will be used to maintain all proper records for the business
4. **Theatre manager:** Theatre manager will be able to update prices, shows and set any promotional or advertising material.

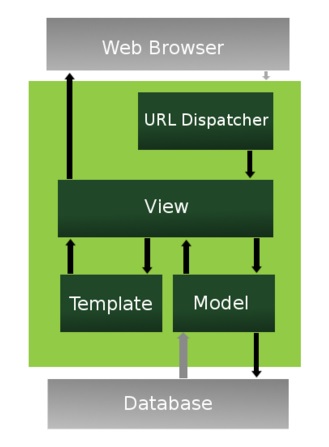
2 SOLUTION DESIGN

2.1 ARCHITECTURE OVERVIEW  
The solution will be built using a Content Management System (CMS) which provides the engine for the website and allows a simple and fast way to develop a first working solution. CMS is a software that helps users to create, manage, and modify content on a website, without writing code. It allows us to edit, modify, organize, delete, publish the website content with the help of predefined templates, themes and plugins.

The diagram below gives an overview of a CMS architecture. The CMS provides developers with templates for:

* The **models:** models are entities in database, including users, tickets, bookings performance to be stored in the database.
* The **views:** views represent the user interface or web pages that displayed on the browser.
* The **controller:** controller is like acombination of framework (the light green box in the picture shown below) and the URL dispatcher (dispatch a unique identifier used to locate a resource on the Internet)

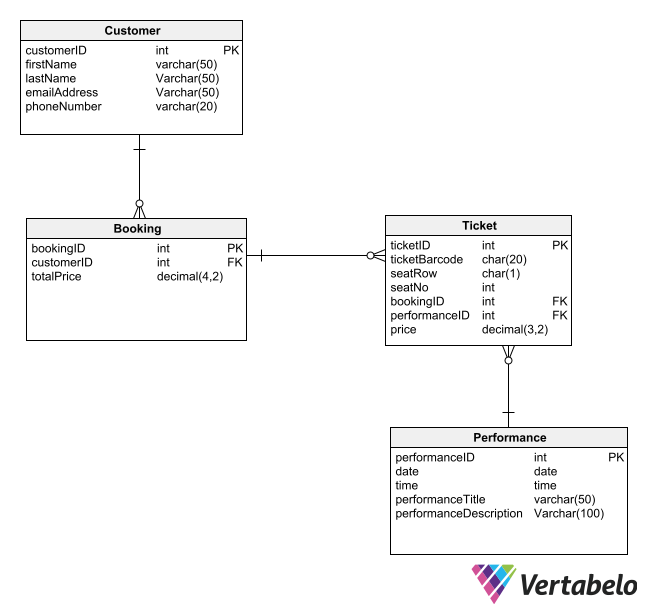
Developers will define the **Views** (based on templates) and the **Models** (based on the entities)**,** and the rest will be provided by CMS.

Warter (2009)

The sections below will document the Model and Views that are required for this application.

2.2 MODEL DESIGN: DATA MODEL.   
The following database model represents a high-level ERD (Entity-Relationship Diagram) that highlights all the data required to be kept for the system we are developing including their attributes, and their relationships. This will be created upon the first release of the website and may be updated upon further version releases:

*Customer should only be able to make 1 booking per customer, need to change relationship cardinality*

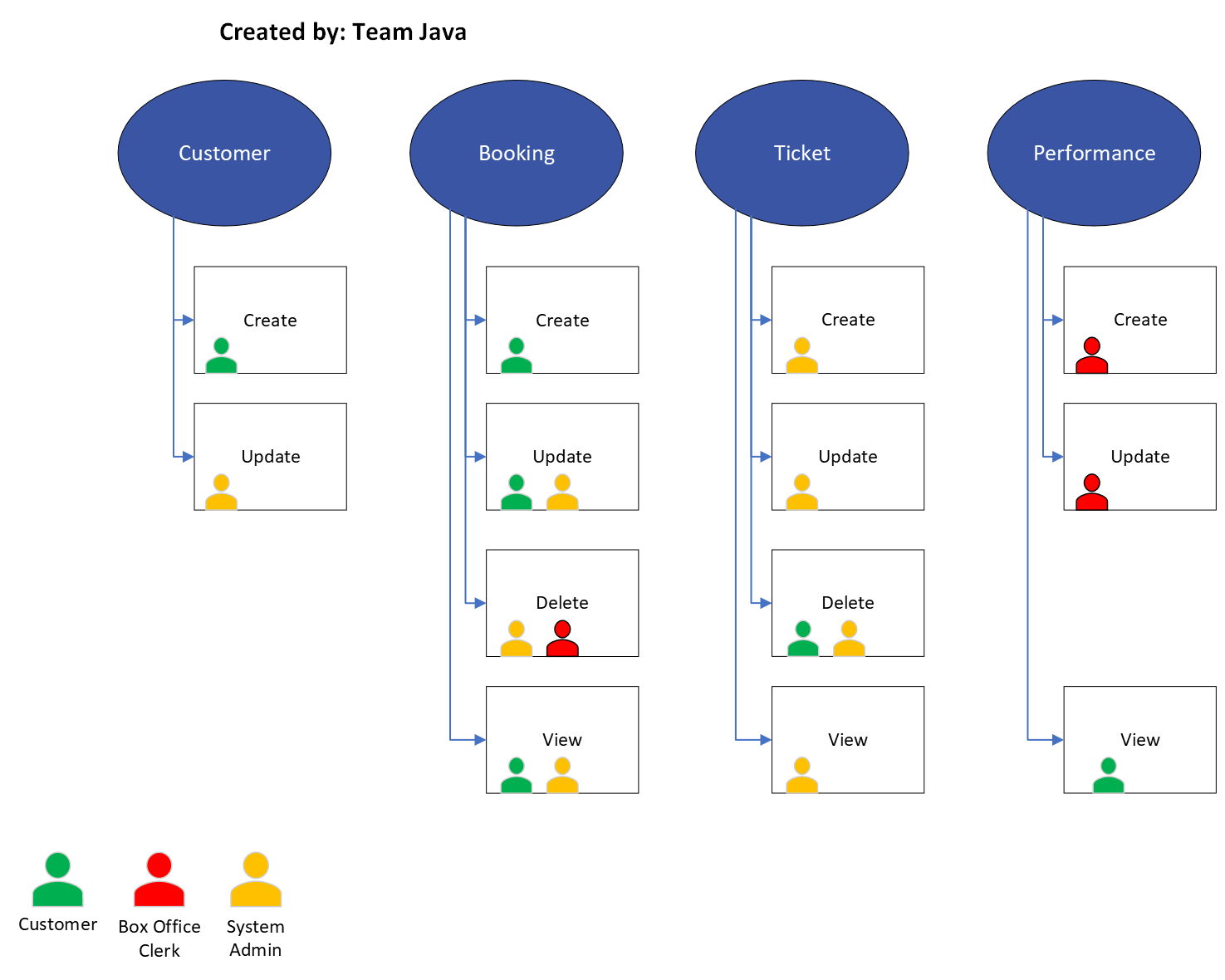


Notes:

* We may continue to use the clients Pentium desktop and explore options of using their device as the database storage location. This is in the case that client does not feel the need to spend more on upgrading their databases.)
* This is ERD applies to the first release, it is kept simplified and holds an idea of what we can expect for the systems database to look like.
* For each performance, there is one ticket per customer as this is to be expected in a user story.

Box

2.3 FUNCTIONAL DECOMPOSITION (please update complete 2.3 section -- Dallas)  
Select the key entities in the data model. From the user stories identify the types of functions needed on each entity, and which users will be performing them. This will provide you with a basis for identifying the Views that you will need.



2.4 VIEW DESIGN: UI  
*List of user's interfaces needed to perform each user story.*

Site map:

1. Home Page (museum information and shows)
2. Shows (complete list of shows)
3. Book Now
4. Contact
5. About Us(link)
6. Sign-Up/ Sign-In

2.4.1 Theatre Customer

As a theatre customer I should be able to see all the information regarding shows and book tickets online in advance to save my time.

1. View: View shows information
2. View: Creating booking
3. View: Updating booking
4. View: Adding multiple Tickets
5. View: Reviewing Tickets (before submission)
6. Online Payment (safe and secure payment)

2.4.2 Box Office Clerk/Admin (Mary)

As a staff, I should be able to see all the required information regarding shows and manage booking information (create, update, cancel) including selling of tickets over a counter to give better customer service.

Views required:

1. Creating Ticket
2. Updating Ticket
3. Deleting Ticket
4. Ticket refund
5. Seat Changes
6. Booking View/Update

2.4.3 Marketing Manager/ Theatre Manager (same person)

For the release 1 version, theatre manager is going to handle all marketing as well. The views required are:

As a marketing manager, I should be able to update or publish any information related to the shows online or on other social media to attract more customers and grow the business to another level.

As a theatre manager, I should be able to set the ticket prices and set any promotional prices or discounts or mange any refunds to compete with the other competitors in the market.

1. Performance Create
2. Performance Update
3. Performance Delete

2.5 DESIGN CONSIDERATIONS

2.5.1 User authentication

2.5.1.1 As an admin user I should be able to authenticate all staff users and customers.

The user authentication will be provided by the Content Management System. The CMS will be configured customer and staff user logons, with the following user groups:

User IDs and passwords will be encrypted.

1. Admin: They will need to access any data from the database to maintain all proper records to make sure the business running in good order.
2. Staff: Staffs can check all the information and update them (refund, seat change or customer service) in the system so that the team can be more productive.
3. Customers: The system provides customers the information about shows and other events and enable customers to book tickets online.

2.5.2 Security

Security requirements to be included:

2.5.2.1 Prevent the system from most common security threats and attacks

Django is under consideration for the CMS because it provides the following security features:

1. Cross site scripting (XSS) protection
2. Cross site request forgery (CSRF) protection
3. SQL injection protection
4. Clickjacking protection
5. SSL/HTTPS
6. Host header validation
7. Session security

2.5.2.2 Protect customer information

Data transferred over the internet will be protected via SSL/HTTPS. Any password will be encrypted in the database and all access to the database must have authentication and is limited using permissions

2.5.2.3 Use a secure payment process

A reputable secure payment package compatible with the CMS will be used.

Verifone: <https://www.2checkout.com/>

2.5.2.4 Provide non-forgeable electronic tickets

Tickets will be produced in PDF format and will have a unique barcode or QR code that can be scanned on paper or directly from a mobile device.

2.5.3 Usability

2.5.3.1 Application must be easy to understand by all users and easily navigates users to book the tickets.

2.5.4 Portability

2.5.4.1 For Customers, the priority is the online booking system. Detailed user interface designs for the core user story will cover:

1. Desktop
2. Tablets
3. Mobile phone

UI mock-ups will include the desktop interface for the Theatre Customer user story.

2.5.4.2 For staff users, the priority is desktop or tablet.

Detailed user interface designs for staff users will include wireframes for:

1. Desktop
2. Tablets
3. Mobile

**Service Provider:** Digital Ocean (https://www.digitalocean.com/), monthly charges (Managed Databases $15/month; App Platform $0/month), for 1 year subscription special prices -None

Glossary:

1. **CMS models:** models are entities in database, including users, tickets, bookings performance to be stored in the database.
2. **CMS** **views:** views represent the user interface or web pages that displayed on the browser.
3. **CMS** **controller:** acombination of framework (the light green box in the picture shown below) and the URL dispatcher (dispatch a unique identifier used to locate a resource on the Internet)

References

1. Digital Ocean. (retrieved 2022, April). https://www.digitalocean.com/
2. Verifone.(retrieved 2022, April). <https://www.2checkout.com/>
3. Vertablo. (retrieved 2022, April). https://vertabelo.com/
4. Warter, F. (2009). *The Django Web Application Framework*. Retrieved from https://www.slideshare.net/fishwarter/the-django-web- application-framework-2-1221388