

APPLICATION DESIGN DOCUMENT

National Parks Management System(NPMS)

By: Abhishek Josyula

UMD email ID:ajosyula@umd.edu

Table of Contents

1) Introduction	3
1.1 Purpose.....	3
1.2 Scope.....	3
2) Architectural Design.....	3
2.1 System Design	3
2.2 Database Design.....	4
2.3 Internal Component Design	7
2.4 Security Functionality	9
3) Testing	9

List of Figures

Figure 1: System Design.....	3
Figure 2: Data Flow Diagram	7
Figure 3: Interface Design	8

1) Introduction

This application design document is going to provide an architectural blueprint for National Parks Management System (NPMS). All the functionalities are discussed in detail here and how they are going to be met.

1.1 Purpose

The application design document is going to illustrate all the designs and standards of NPMS. This document is intended to help the developers, document personnel and the testers working on this application.

1.2 Scope

The NPMS application assists customers to view details about national parks like holidays, timings, how to reach, activities currently going on in national parks. The NPMS is designed to help customers to purchase parking passes online allowing to them to skip the long queues at the entrance and can also buy individual passes if customers are arriving by walk. The application has an Admin actor who has highest level of access to manage the whole application.

2) Architectural Design

2.1 System Design

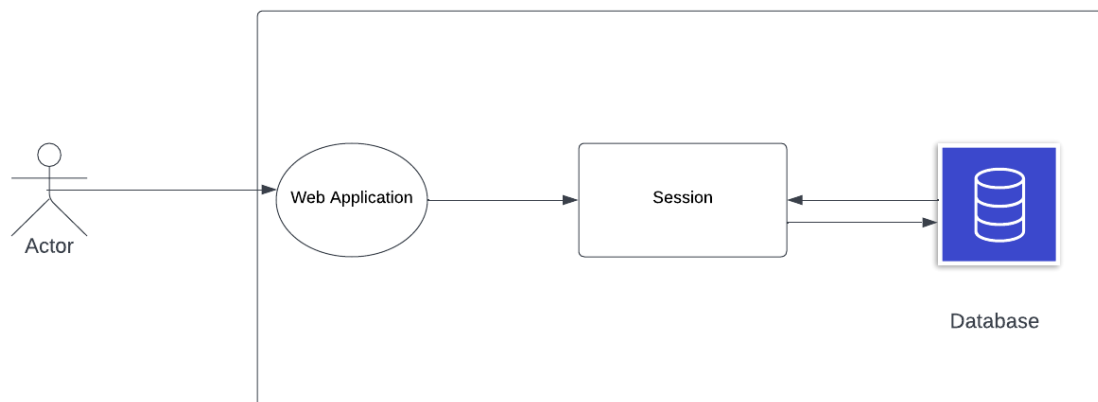


Figure 1: System Design

Customer/User is going to be interacting with the NPMS web application interface. The system is going to provide all functionalities which are accessible as roles defined in the Software Requirements Specification(SRS). Any required data is going to be fetched from database.

2.2 Database Design

Following are the tables of database:

- 1.AspNetUsers
2. Passes
3. AspNetUserRoles
4. Parks
5. Careers
6. Events
7. Carts
8. Reservations
9. OrderDetails

AspNetUsers

Field Name	Type
Id	guid
UserName	string
NormalizedUserName	string
Email	string
NormalizedEmail	string
EmailConfirmed	bool
PasswordHash	string

SecurityStamp	string
ConcurrencyStamp	string
PhoneNumber	string
PhoneNumberConfirmed	bool
TwoFactorEnabled	bool
LockoutEnd	DateTime
LockoutEnabled	bool
AccessFailedCount	Int

Careers

Field Name	Type
CareerId	Guid
CareerName	string
CareerDescription	string
CareerRecruiter	string
CareerPlace	string

AspNetUserRoles

Field Name	Type
UserId	guid
RoleId	guid

Parks

Field Name	Type
ParkName	string
ParkId	int
ParkDescription	string
ParkImageUrl	string

Events

Field Name	Type
Event Name	string
EventId	int
EventDescription	string

Passes

Field Name	Type
PassName	string
PassId	int
PassPrice	int

Reservations

Field Name	Type
Rid	guid
ReservationName	string
TypeOfEvent	string
ContactNumber	string
ReservationEmail	string
ReservationDate	DateTime
ParkName	string

Carts

Field Name	Type
OrderId	guid
Total Amount	Decimal
UserId	guid

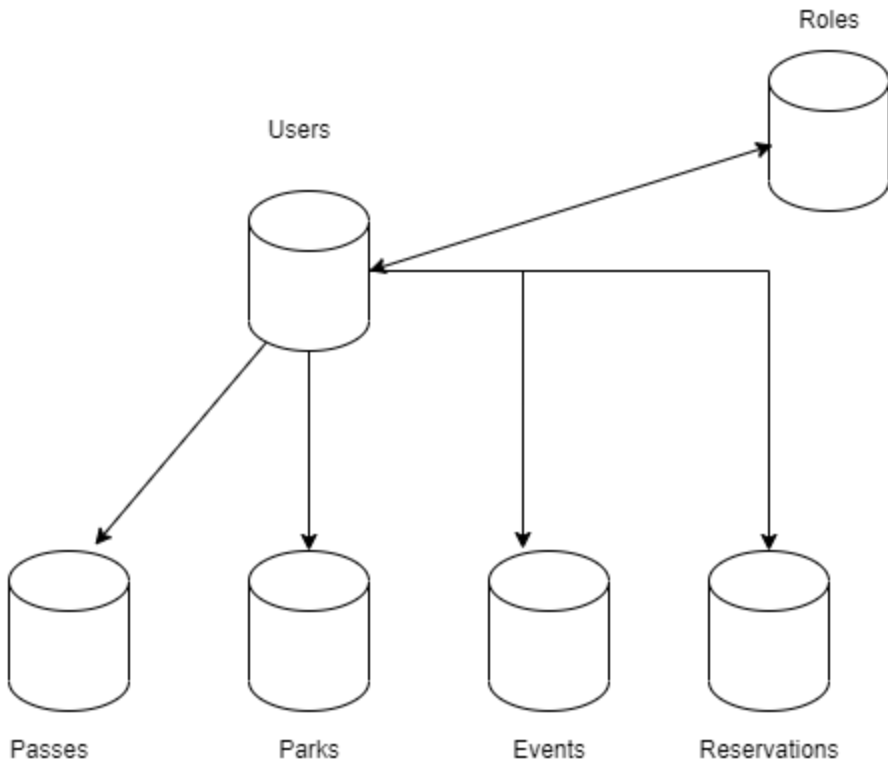


Figure 2: Data Flow Diagram

2.3 Internal Component Design

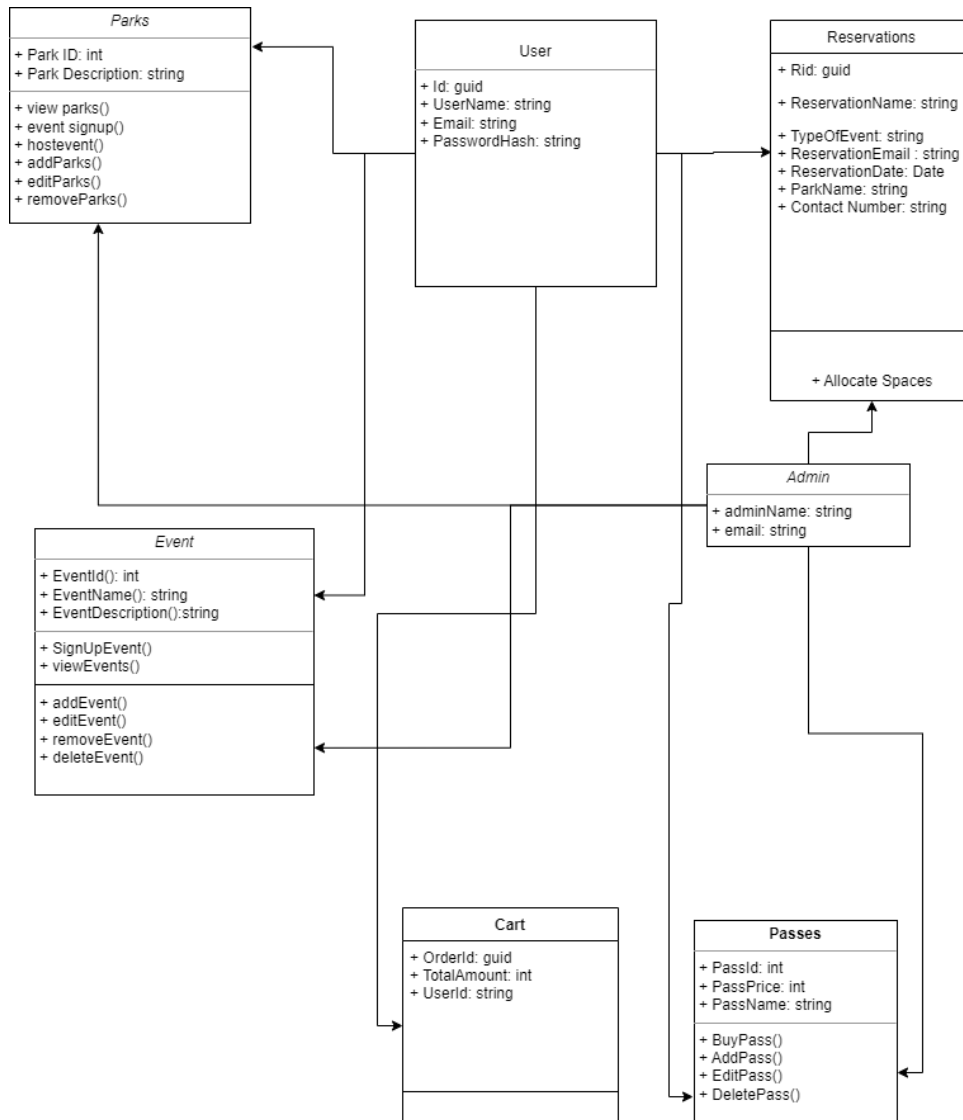


Figure 3: Interface Design

2.4 Security Functionality

- The application has several security functionalities that prevent attackers from attacking the application.
- Implement proper logging mechanism so that in the case of an incident the security team will be able to respond as quickly as possible before the situation gets out of hand.
- Ensuring the flow of traffic between the application and the internet is encrypted.
- Ensuring that there are no open sensitive ports which are accessible from outside.
- The passwords stored in the database will be in the encrypted using secured hashing algorithms format rather storing them in the plain text format
- Ensuring proper authorizations and authentication for the users of web application
- Implementing Session handling to handle sessions properly and ensuring that application creates, maintains and destroys session tokens properly over life-cycle of a user'
- Ensuring that no malicious files are uploaded to the application
- Ensuring only legitimate users are registered by sending an email to the users

3) Testing

- A valid user with valid login credentials only can login into the system
- An invalid user cannot login into the application
- A new customer can sign up with valid credentials
- A logged in customer can sign up for events by purchasing event pass
- A logged in customer can reserve a place for events
- Checking password lengths based on password policy
- Admin can add or update details of park details
- A valid user can logout successfully
- A valid user can buy parking passes
- Any customer can view details of national parks when accessed the application.

4) References

- <https://www.nps.gov/index.htm>