

Governors State University
OPUS Open Portal to University Scholarship

All Capstone Projects

Student Capstone Projects

Fall 2016

GSU Event Portal

Kranthi Kumar Naik Banoth
Governors State University

Hanuman Prasad Dharmapuri
Governors State University

Sravya Mokrala
Governors State University

Nagamani Veepuri
Governors State University

Follow this and additional works at: <http://opus.govst.edu/capstones>

 Part of the [Computer Sciences Commons](#)

Recommended Citation

Banoth, Kranthi Kumar Naik; Dharmapuri, Hanuman Prasad; Mokrala, Sravya; and Veepuri, Nagamani, "GSU Event Portal" (2016).
All Capstone Projects. 266.
<http://opus.govst.edu/capstones/266>

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to
http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the [Governors State Computer Science Department](#)

This Project Summary is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.

ABSTRACT

This project deals with making a website, which gives data about the different sorts of events. The enrolled clients in this website can make or alter events and they can audit the quantity of enlistments that had happened for their occasion. There will be an administrator for the website to monitor all the activities, the administrator will have rights to check, alter or erase events and check client accounts. Along these lines, this webpage for the most part network clients to discover the insights about the events that happen around them and in the event that they are keen on going to any occasion, they can just enroll for such occasion, by paying the particular sum, which was expressed by occasion coordinator.

In the event if there are any adjustments in the occasion all the enrolled clients get advised through emails. In the event that an enrolled client of this site, register for any occasion then an confirmation page will be produced, and the client can utilize such page as a proof of enrollment, to go to the occasion. This Website monitors every one of the events which are enrolled by all the approved clients.

Software Requirements:

Frontend Technologies: ASP .net, XML, Java Script

Backend Database: Microsoft SQL Server

Table of Content

1	<i>Project Description</i>	1
1.1	Competitive Information	1
1.2	Relationship to Other Applications/Projects	1
1.3	Assumptions and Dependencies	1
1.4	Future Enhancements.....	2
1.5	Definitions and Acronyms	2
2	<i>Project Technical Description</i>	2
2.1	Application Architecture	2
2.2	Application Information flows.....	3
2.3	Interactions with other Projects (if Any)	4
2.4	Interactions with other Applications	4
2.5	Capabilities	4
2.6	Risk Assessment and Management.....	5
3	<i>Project Requirements</i>	5
3.1	Identification of Requirements	5
3.2	Operations, Administration, Maintenance and Provisioning (OAM&P).....	7
3.3	Security and Fraud Prevention.....	7
3.4	Release and Transition Plan.....	7
4	<i>Project Design Description</i>	7
5	<i>Internal/external Interface Impacts and Specification</i>	8
6	<i>Design Units Impacts</i>	14
6.1	Functional Area A/Design Unit A	14
6.1.1	<i>Functional Overview</i>	14
6.1.2	<i>Impacts</i>	14
6.1.3	<i>Requirements</i>	14
6.2	Functional Area B/Design Unit B.....	15
6.2.1	<i>Functional Overview</i>	15
6.2.2	<i>Impacts</i>	18
6.2.3	<i>Requirements</i>	19
7	<i>Open Issues</i>	19
8	<i>Acknowledgements</i>	19
9	<i>References</i>	19

1 Project Description

The project we developed is GSU Event Portal. The system gives solutions for most of the problems that we have identified in the event reservation currently. All most all the tasks were handled manually by the management and the employees work at the events. The system that we are going to introduce will address those problems accordingly the ability to search about specific information or detail before and after doing anevent reservation in an easy way with customizing event details and checking the validity of details are some purpose to build system which makes requirements possible with easy and fast way. GSU Event Portal is built to find suitable solution for reservation and customization of events, concerts, payments, administration, and services.

This system deals with the database as an end back which is based on SQL server and its interface based on Visual Studio C#. The interface aims to make reservation and using other tools easy to every one without needing to learn how to use. The system that is going to be handed over to the client will address most of the problems in the event portal currently. The tasks that now are carried out manually will be able to do with the system in more easy way. The data that are now kept in large physical files will be stored in the centralized database of the system. That will reduce the damages that can be happened to the data unexpectedly. The calculations that are done manually will able to be done through the system to get more accurate results.

Presently the information regarding the employees is handled manually. The introducing system is able to handle the problems in a more convenient and accurate manner. The details of the daily income, reservations can be inserted to the system easily and retrieve them whenever the user requires. Records and details of the events are also handled through the proposed system with giving the solutions for the problems that arise with the current manual procedure of the event portal.

The features of the introducing system will call upon the problems that we have encountered from the current system that is prevailing in the event portal now to make the tasks done at the event location comfortably and much more efficiently.

1.1 Competitive Information

There are events portals web applications, our team is continuously putting effort to add features and to make the application is top amongst the competitors.

1.2 Relationship to Other Applications/Projects

This project is not related to any of the other Applications/Projects.

1.3 Assumptions and Dependencies

Initially, user has to login into the system software by using Event PortalAccount Username and Password. System software authenticates the user by verifying the details in the Events portal database.

1.4 Future Enhancements

Planning to keep application update by adding future enhancements like filtering results by location coordinates. Future work continues.

1.5 Definitions and Acronyms

Acronym items should be included here. For each special term supply a definition here.

2 Project Technical Description

GSU Event Portal with web-based architecture can provide the feature such as:

- Maintains the information related to events in database tables
- The grid view fetches the events details and displays the information
- Customer feedback in the website for the things to be improved
- Payment process with faster response and also securely.
- Good communication between users and company

Considering the access levels, we designed the application in two main modules. They are

- User Module
- Admin Module

User Module:

In this module user can view the information of the website and he/she can also register for event registration in advance with fully advance payment. User has to login into the system for the registration. User can view only the front end of the website. This contains sub modules like information module, login module, contact module, registration module and reservation module.

Admin Module:

In this module admin controls the whole project. This contains sub modules such as Admin login module, employee module, customer module and event registration module. Administrator has an authority to handle the front end and also the back end process of the system.

After gathering and reviewing the project requirements, we have come up with the architecture styles to be used, class diagrams and also the design pattern that we should follow. The architecture pattern and the design patterns are listed below.

2.1 Application Architecture

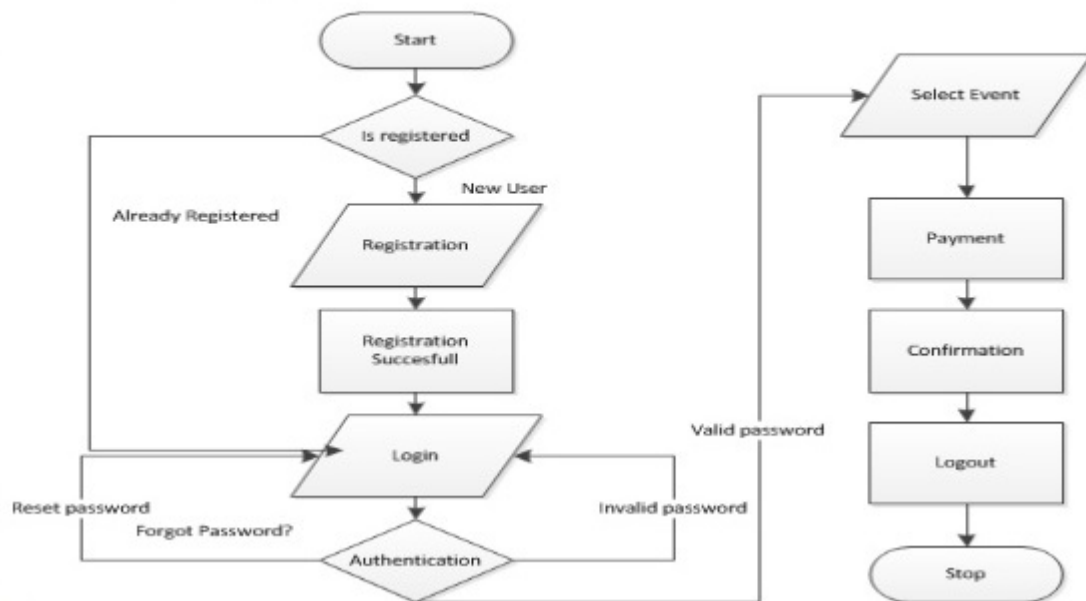
This architecture is purely designed by our team it is not a reference from any diagram from internet. We just collected the pictures from web and arranged them according to our priority. Coming to the technical architecture we used MVC as the development architecture. MVC architecture would be the best choice for the Event Portal application. It is more apt for the enterprise applications.

It designs the software in an organized manner by dividing the application into three different modules; Model, View and Controller. We can design simple and clean interfaces using this architectural approach. In the design perspective it is the best option than the other available architecture styles. The architecture is depicted in the below given figure.

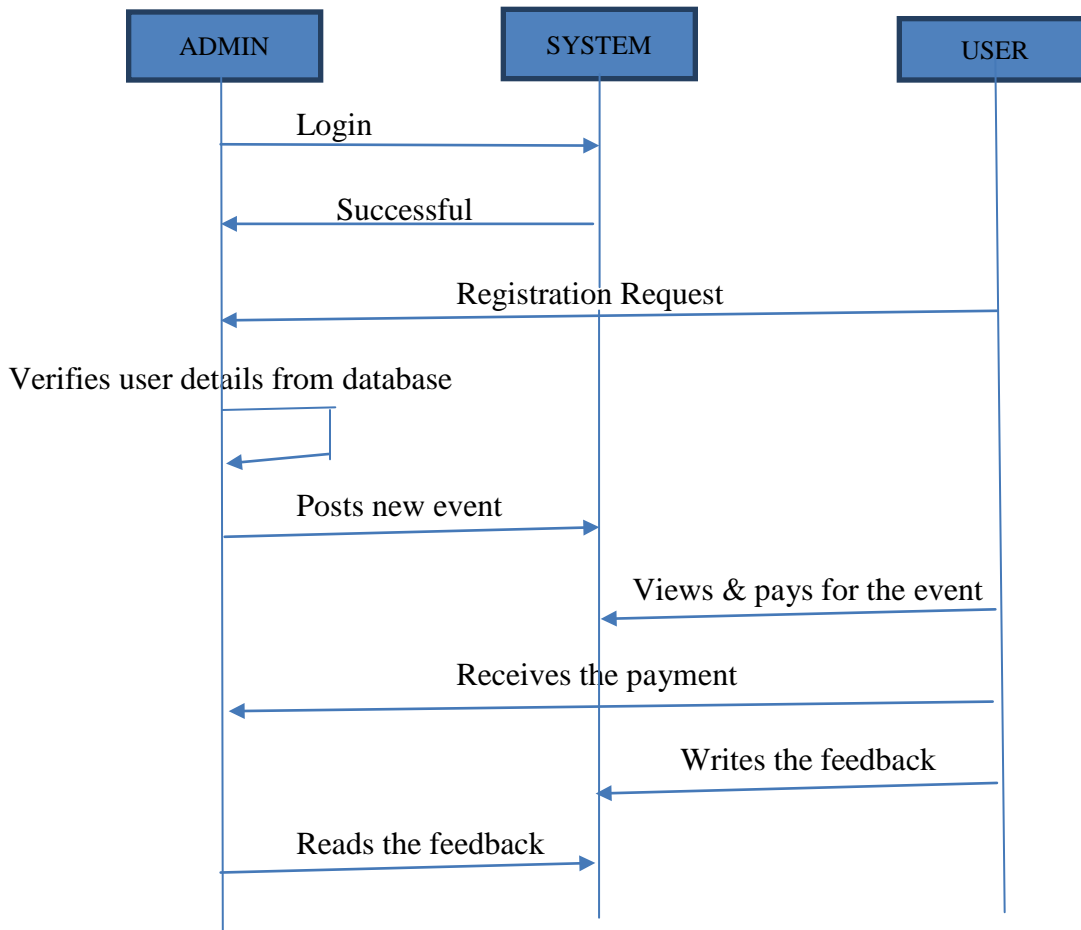
2.2 Application Information flows

User has to create the account before use the Events portal application, if the user wants to post any events. If user wants to buy tickets, then no need of account creation. Once the user creates the account, then the user can see the list of events, and of course the user can add the event. User can filter the list of events by applying filters to the categories.

Flowchart



Admin work flow and the event flow diagrams



2.3 Interactions with other Projects (if Any)

There are no interactions with the other projects.

2.4 Interactions with other Applications

There are no interactions with other applications at present, but planning to integrate Google API for maps purpose.

2.5 Capabilities

This application provides capabilities such as retrieving events, deleting events, updating events and adding the events. It also provides the capabilities to store payment details and also reservation history.

2.6 Risk Assessment and Management

User card details are being transferred to this application. There are several security providers available to secure the payment details. We are planning to provide 128-bit Secure Socket Layer security to secure the payment details.

3 Project Requirements

3.1 Identification of Requirements

<GSU-EventSYS-1.0-User-Add-000001>

The project must allow new users to be added by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-User-Update-Profile-000002>

The project must allow users to update the profile details by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-User-Update-Password-000003>

The project must allow users to update the password by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-User-Event-List-000004>

The project must allow users to view the event list by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-User-Event-Add-000005>

The project must allow users to add the event by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-User-Event-Update-000006>

The project must allow users to update the event details by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-User-Event-Delete-000007>

The project must allow users to delete the event details by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-User-Event-Details-000008>

The project must allow users to view the selected event details by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-User-Ticket-Reserve-000009>

The project must allow users to reserve the ticket by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-User-Ticket-Reserve-View-000010>

The project must allow users to view the ticket reservation by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-User-Payment-Add-000011>

The project must allow users to add the payment details by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-User-Payment-View-000012>

The project must allow users to view the payment details by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-User-Feedback-Add-000013>

The project must allow users to give the feedback for the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-Admin-Event-List-000014>

The project must allow administrator to view the event list by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-Admin-Event-Add-000015>

The project must allow administrator to add the event by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-Admin-Event-Update-000016>

The project must allow administrator to update the event details by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-Admin-Event-Delete-000017>

The project must allow administrator to delete the event details by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-Admin-Event-Details-000018>

The project must allow administrator to view the selected event details by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-Admin-Payment-View-000019>

The project must allow administrator to view the payment details added by the users by the application.

Implementation: Mandatory

<GSU-EventSYS-1.0-Admin-Feedback-View-000020>

The project must allow the administrator to view the feedback details given by the users for the application.

Implementation: Mandatory

3.2 Operations, Administration, Maintenance and Provisioning (OAM&P)

User data backup will be provided. So, if any issue happened then the user data can be rolled back. Technical team keeps monitor the application quality and performance for the maintenance purpose. Because of this highest level of performance can be achieved.

3.3 Security and Fraud Prevention

The data of the user should be in private so that it cannot be accessed by other users. The user has to be authenticated by the system software using the OAuth method. Because of SSL fraud payments can be detected.

3.4 Release and Transition Plan

The Integrated Development Environment has the option to deploy the application. This application can be placed in a server where everybody can send the request to the server to get the response from the application. If any bug fixes or the new features are added, then the older version can be replaced with the new deployed version just like as the normal file in the file explorer.

4 Project Design Description

Project was designed using an object oriented approach. While developing the system we had to come across several diagrams in order to achieve the quality of the software product. After creating the ER diagram it was mapped to a relational schema and normalized. Finally, the database was created. Class diagram was designed with class stereotypes which included with relationships. Class diagram helped in programming the solution.

Formal system is well defined by the forms, reports, policy manuals and organization charts. Above mentioned tactic helped a lot in providing a clear view of the Event portal management system. Majorly we found the users of the system after interviews. It promotes the user roles of the system. Intention of building the system was clearly understood after this phase of software life cycle. Further analysis revealed where and when the system will be used. After gathering and analyzing the client requirement it was expressed by a use case and the rest of the development process was built on it.

We followed the most of all the design patterns such as component, factory, and provider patterns. Component pattern is only given for because of time constraints on us. The component design pattern clearly depicts the necessity to design the software using that design pattern only.

For an instance, we illustrated the class diagram and the necessity of the design pattern below:

Class Diagram:

As part of the development process, we come up with the classes. To better understand the relationship between the classes, we have gone through the class relationships. We have come up with the following dependencies, and associations. The below figure clearly depicts the relationships between the classes we used in our application

Design Pattern:

Container pattern is useful for the design problems for the reservation module in the GSU Event Portal.

Name of the Pattern: Container Pattern

Intent of the pattern: To create objects to hold other objects and manage the objects.

Problem: The reservation has to hold the information about the event, user and payment.

Solution: Instantiate the reservation class with the instances for Event, User, and Payment classes.

Participants and Collaborators: Reservation, User, Event, and Payment.

Consequences: Users can't see the reservation details containing of event and payment information.

Implementation: Using the container pattern we create the instance for the Reservation class which contains the instances for the User, Event, and Payment classes. So, the class can store the information about the User, Event, and Payment information.

5 Internal/external Interface Impacts and Specification

Interfaces are the major part of the GSU Event Portal. This application contains two modules based on the access level specification to provide the facility to control for the application under a specific person known as Administrator. The user module and Admin module both are contains the interfaces according to the functionality. The user module contains the interfaces to create the account, to log in to the account using the credentials. User can also create the events, update the events. There is an option to add payment to reserve for the specific event that the user would like to attend. In each and every web form user has enforced to enter the accurate data. As part of this process, ASP.NET default validations are being used. Some of the web form designs are provided below.

Login Page:

This purpose of the screen is to validate the user and admin by seeking the respective credentials. The below given figure depicts the interface for login module.

User Login

User Name:

Password:

Admin Login

User ID:

Password:

User Signup

User ID:

User Name:

Password:

Add Event Page:

This page is designed to provide the option to add an event by the user. This interface seeks the input from user such as event name, location, date, and also event description, etc. The given below figure depicts an instance of the event list.

Create Event

Event Title:

Location:

Starts:

Ends:

Price (USD):

Available Quantity:

Description:

Home Page:

This page is designed to give the overview of the events to the user. Along with event info, each event has the option to edit, delete and reserve links which does respective actions accordingly. The given below figure depicts an instance of the event list.

Welcome, hanuman!

Location: Time:

Title	Location	Starts	Ends	Price	Quantity	Note	Action
Sample Event	University	2/25/2016 12:00:00 AM	2/25/2016 12:00:00 AM	3.00	100	test	Book Edit Delete
Cricket	University Ground	6/29/2016 3:00:00 PM	6/29/2016 5:00:00 PM	5.00	150	Good to go event and chill out	Book Edit Delete
Baseball	Baseball Stadium	6/30/2016 2:00:00 PM	6/30/2016 5:00:00 PM	10.00	50	Test	Book Edit Delete
Football	Baseball Ground	6/30/2016 9:00:00 AM	6/30/2016 1:00:00 PM	20.00	20	SAmple	Book Edit Delete

Event Reservation Page:

This is interface to enter the information to reserve for the event. The figure given below depicts the event reservation process.

Event Details

Event Title: Sample Event

Location: University

Starts: 2/25/2016 12:00:00 AM

Ends: 2/25/2016 12:00:00 AM

Price (USD): 3

Total Quantity: 100

Quantity Required:

Payment Page:

The purpose of this screen is to provide the interface for the user to enter the payment details. The below figure depicts the design for the payment process.

Enter Payment Details

Total amount \$ 3

Card Number

Name on Card

CVV

Expiration Date

Pay

FeedbackPage:

The purpose of this screen is to provide the interface for the user to provide the feedback for the GSU Event Portal and also for the web application. The below figure depicts the design for the Feedback page.

Feedback

Feedback

Rating

1 2 3 4 5

Submit

ProfilePage:

The purpose of this page is to provide the interface for the user to update the profile details and also to change the password. The below figure depicts the design for the Profile page.

Update User Details	Change Password
User Name: <input type="text" value="hanuman"/>	Current Password: <input type="text"/>
Email: <input type="text" value="1@2.com"/>	New Password: <input type="text"/>
Phone: <input type="text"/>	Confirm Password: <input type="text"/>
<input type="button" value="Update"/>	<input type="button" value="Update"/>

Validations:

This application has a feature to validate the data from the user input. The validations are implemented using the ASP.NET default validation controls. The below figure depicts the validations.

Enter Payment Details

Totla amount \$ 3

Card Number Please enter card number

Name on Card Please enter name on card

CVV Please enter CVV

Expiration Date 01 ▾ 2016 ▾

Admin Payments Page:

The purpose of this page is for the administrator to provide the facility to view the payments done by the users at the time of event reservation. The below figure depicts the interface for the payments screen in the admin module.



Welcome Admin!

Users Payment Summary

Amount	Card Name	Card Number	CVV	Expiry Month	Expiry Year
15.00	s	4111111111	123	12	2014
6.00	s	4111	123	01	2015
3.00	1	4111	133	01	2016
3.00	1	1	111	08	2016

Admin Feedback Page:

The purpose of this page is to provide the administrator to view the feedback given by the GSU Event Portal web application users. The below figure depicts the interface for the Feedback page in the Admin module.

Welcome Admin!

Customers Feedback Summary

Email	Rating	Comments
1@2.com	3	test
Average	3	

6 Design Units Impacts

In this module user can view the information of the website and he/she can also register for reserve for the event in advance with fully advance payment. User has to login into the system for the registration. User can view only the front end of the website. This contains sub modules like information module, login module, reservation module, and registration module.

6.1 Functional Area A/Design UnitA

6.1.1 Functional Overview

User module contains the functionality to set up an account with the Event portal web application, posting new event to the portal and also to reserve the ticket for any event that the user would like to watch.

6.1.2 Impacts

This is main module of the GSU Event Portal. This module contains all the information that user can register, login, manage the event information. In this module, user can register for the event, pay for the event. The user can also update the information.

6.1.3 Requirements

User Requirements:

The user must be able to create an event, and also to reserve for the event in the system software. User should be able to read the user card details to reserve the ticket.

Access/Security requirements:

The data of the user should be in private so that it cannot be accessed by other users. The user has to be authenticated by the system software using the OAuth method.

Interface requirements:

The user interface provides a way to create event, view events, reserve for event and to enter the payment details. The user should be able to edit the existing events and update event details easily.

Scalability requirements:

The system software must be able to handle many user requests. Bulk amount of data should be easily stored, retrieved and processed by the system software.

Integration requirements:

Both the event management services and payment service must be integrated and deployed in the system software.

Quality and Performance requirements:

Data storage & retrieval should be as quick as possible. The maintenance of events and payment details should have maintained with highest level of security.

6.2 Functional Area B/Design Unit B

6.2.1 Functional Overview

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. The database schema is designed using Structured Query Language (SQL) provided by Microsoft. The following figures represent the schema for the different tables in the GSU Event Portal Database.

User Table:

This table holds the information about the user such as user id, user name, email, password, and phone, etc. The below figure depicts the table design for the user table.

Name	Data Type	Allow Nulls	Default
user_id	nvarchar(50)	<input type="checkbox"/>	
user_name	nvarchar(50)	<input type="checkbox"/>	
user_password	nvarchar(50)	<input type="checkbox"/>	
user_email	nvarchar(75)	<input type="checkbox"/>	
user_phone	nvarchar(15)	<input checked="" type="checkbox"/>	

Keys (1)
 <unnamed> (Primary Key, Clustered: user_id)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Admin User Table:

This table holds the information about the admin user such as user id, password, and etc. The below figure depicts the table design for the admin user table.

Name	Data Type	Allow Nulls	Default
user_id	nvarchar(50)	<input type="checkbox"/>	
password	nvarchar(50)	<input type="checkbox"/>	

Keys (1)
 <unnamed> (Primary Key, Clustered: user_id)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Events Table:

This table holds the information about the events such as event name, location, date, and description, etc. The below figure depicts the table design for the events table.

Name	Data Type	Allow Nulls	Default
Id	int	<input type="checkbox"/>	
title	varchar(200)	<input type="checkbox"/>	
location	varchar(500)	<input checked="" type="checkbox"/>	
starttime	datetime	<input type="checkbox"/>	
endtime	datetime	<input type="checkbox"/>	
description	varchar(MAX)	<input checked="" type="checkbox"/>	
price	decimal(6,2)	<input type="checkbox"/>	
quantityAvailable	int	<input type="checkbox"/>	

Keys (1)
 <unnamed> (Primary Key, Clustered: Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Reservation Table:

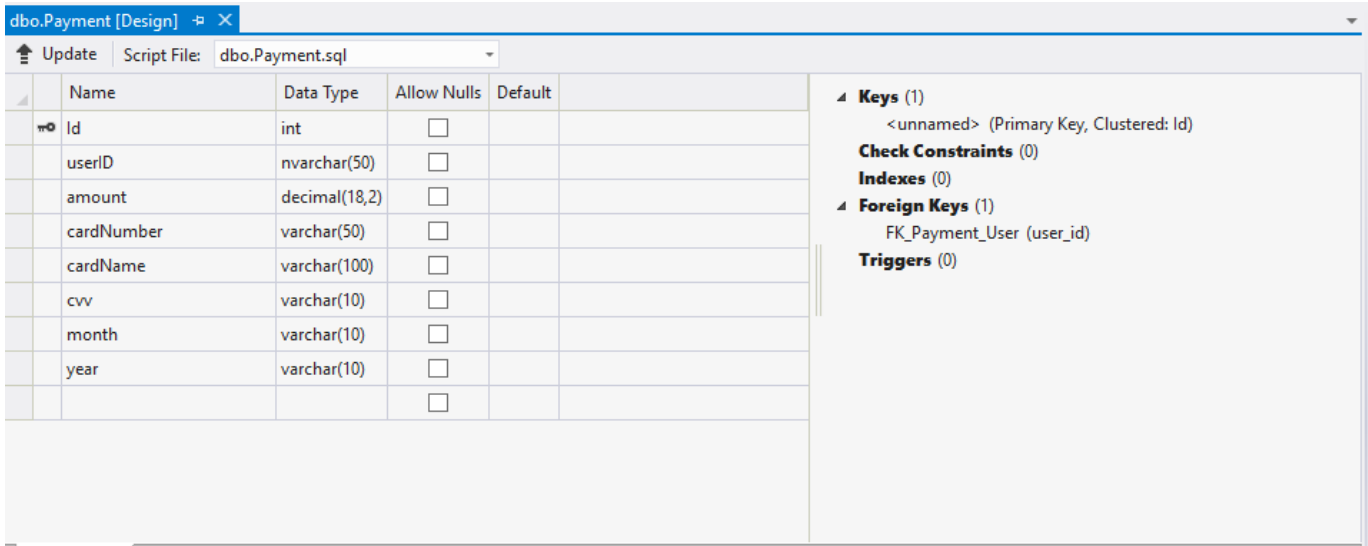
This table holds the information about the reservation details such as event id, quantity required, user id, etc. The below figure depicts the table design for the reservation table.

Name	Data Type	Allow Nulls	Default
Id	int	<input type="checkbox"/>	
userID	nvarchar(50)	<input type="checkbox"/>	
eventID	int	<input type="checkbox"/>	
quantityRequired	int	<input type="checkbox"/>	

Keys (1)
 <unnamed> (Primary Key, Clustered: Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (2)
 FK_Reservation_User (user_id)
 FK_Reservation_Event (Id)
Triggers (0)

Payment Table:

This table holds the information about the payment details such as payment amount, card holder name, card number, security code, expiration date, etc. The below figure depicts the table design for the payment table.

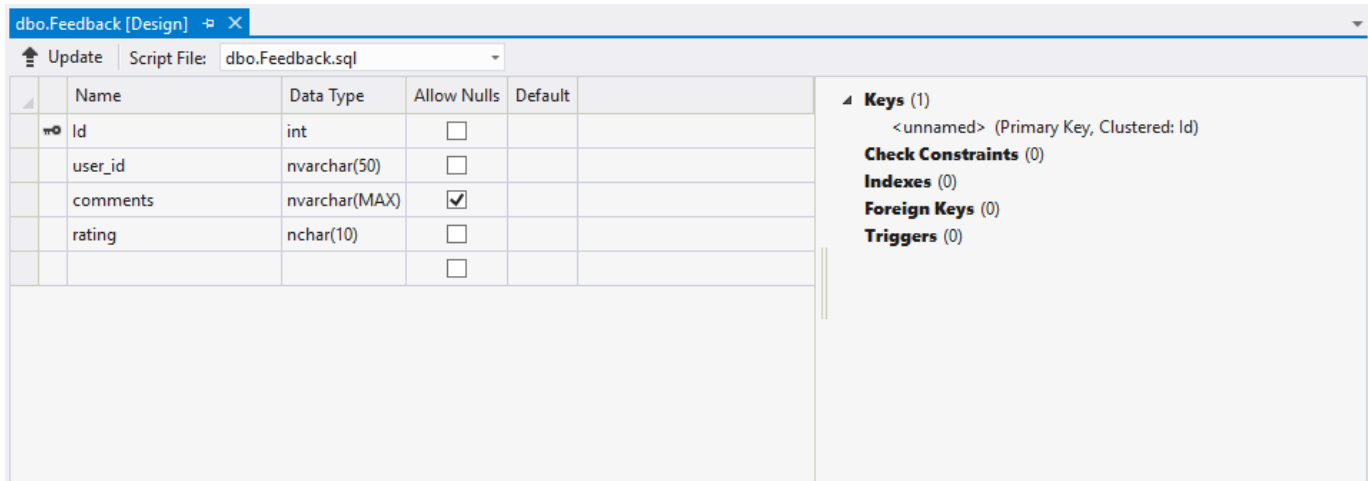


Name	Data Type	Allow Nulls	Default
Id	int	<input type="checkbox"/>	
userID	nvarchar(50)	<input type="checkbox"/>	
amount	decimal(18,2)	<input type="checkbox"/>	
cardNumber	varchar(50)	<input type="checkbox"/>	
cardName	varchar(100)	<input type="checkbox"/>	
cvv	varchar(10)	<input type="checkbox"/>	
month	varchar(10)	<input type="checkbox"/>	
year	varchar(10)	<input type="checkbox"/>	

Keys (1)
 <unnamed> (Primary Key, Clustered: Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (1)
 FK_Payment_User (user_id)
Triggers (0)

Feedback Table:

This table holds the information about the feedback details such as rating, user id, description, etc. The below figure depicts the table design for the feedback table.



Name	Data Type	Allow Nulls	Default
Id	int	<input type="checkbox"/>	
user_id	nvarchar(50)	<input type="checkbox"/>	
comments	nvarchar(MAX)	<input checked="" type="checkbox"/>	
rating	nchar(10)	<input type="checkbox"/>	

Keys (1)
 <unnamed> (Primary Key, Clustered: Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

6.2.2 Impacts

This is main module of the GSU Event Portal to store the information about the users, administrators, and the events, etc. This module contains all the information that user can register, login, manage the event information. In this module, user can store for the event, pay for the event. The user can also update the information.

6.2.3 Requirements

The application must be able to store an event, and also to reserve for the event in the system software. User should be able to read the user card details to reserve the ticket. The data of the user should be in private so that it cannot be accessed by other users. The system software must be able to handle many user requests. Bulk amount of data should be easily stored, retrieved and processed by the system software. Both the event management services and payment service must be integrated and deployed in the system software. Data storage & retrieval should be as quick as possible. The maintenance of events and payment details should have maintained with highest level of security.

7 Open Issues

There are no current open issues at this time.

8 Acknowledgements

We have taken effort in this project report. However, it wouldn't have possibly done without the kind support from our professor and help of many individuals and organizations. We would like to extend our sincere thanks to all of them who assisted us in our project completion.

9 References

E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patters: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.