Software Systems Capstone
CPSC-49200-003
Professor Michael Lewis
May 2, 2021

Group Members:
Owais Ahmed
Mohammed Osman
Asjad Ali
Deep Patel

Project Manual: Cancer Support Center

Our team member decided to choose the Cancer Center project our motivation behind this project was to learn how the cancer center works and this was the best opportunity for us to learn and develop the website. The Cancer Support Center provides comprehensive supportive and wellness programs to thousands of people annually. People who come to their Center is cared and they are here to ease the burden of the cancer journey from the beginning with diagnosis through treatment and well after someone is cancer-free. This website is an existing website that is currently running, and our task was to add new functionalities to this website such as adding a donate link, Ability to register for events, Ability to register for a classes, Ticket purchasing, and also adding a login for admins and user.

As stated above the key functionality we added to this existing website:

-Proper donate link within the website.
-Ability to register for events for users.
-Ability to register for a class for user.
-Ticket purchasing for event or class for user.
-Separate logins for admins and user.

Roles and responsibility for each member as follows:

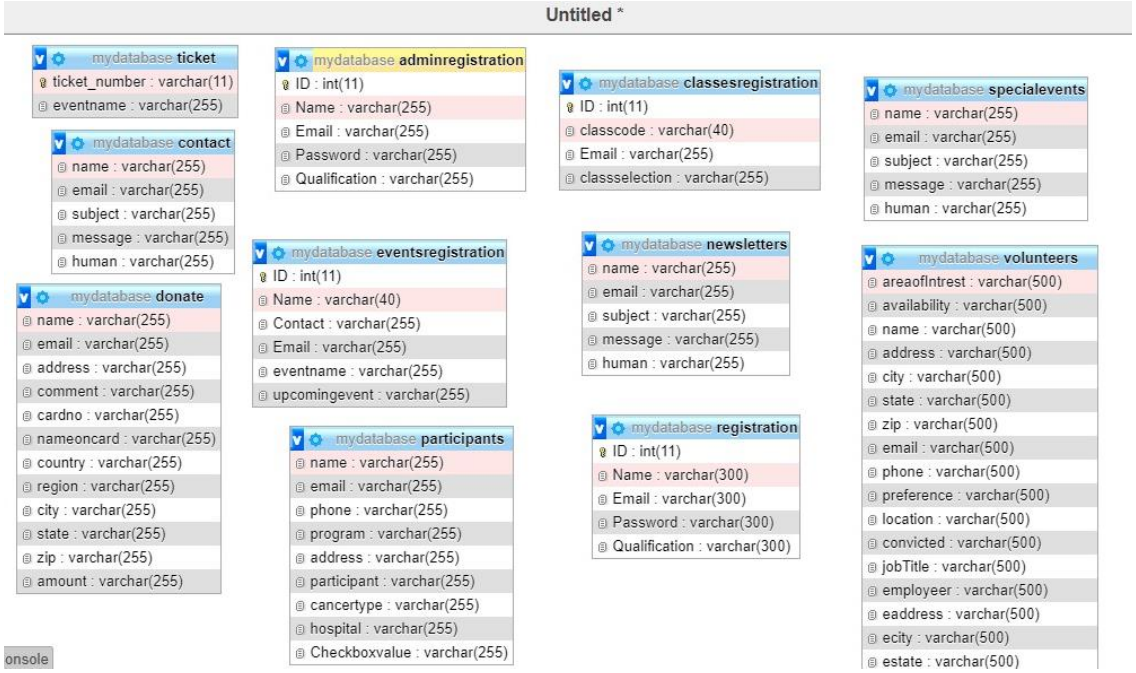
Login and Donate link implementation – Owais Ahmed
Class registration implementation – Mohammed Osman

Event registration implementation – Asjad Ali
Ticket Purchasing implementation – Deep Patel
(Please note: Even though roles were separately assigned, all work was done collaboratively.)

For software and tools we used, and we need in order for this website to function properly. As for the framework we decided not to use any since this project never had any framework and we continued with what previous semester group that used SASS for the style sheet. Moving forward to software our team decided to use Notepad++, Visual Studio, XAMPP server, and phpMyAdmin. Our team used Notepad++ and visual studio to create and edit the HTML and PHP files, and we used XAMPP server and phpMyAdmin for backend functionality which was SQL database. In order to run the website, we need to install software and follow these instructions:

1) Download the XAMPP server form browser.
2) Then open XAMPP server and run MYSQL and Apacha
3) Click on ADMIN button in front of MYSQL
4) Import database to phpMyAdmin named as mydatabase to use backend functionalities.
5) Please put the DBWORK folder in C:\xampp\htdocs. (DBWORK is the complete website project file)
6) To run you have to type the URL in browser such as http://localhost/DBWORK/index.html

Moving forward to database we decided to go with SQL database with phpMyAdmin and We implemented the database accordingly to the class diagram. We linked all the existing forms/files along with the new ones to the database. We also can see all the website data when we export the data to the database including user and admin info. Below is the Entity-relational diagram. Please note: we have used SQL database, but the relations are not specified using tables. They are being specified and maintained manually.



Untitled *

mydatabase ticket
- ticket_number : varchar(11)
- eventname : varchar(255)

mydatabase contact
- name : varchar(255)
- email : varchar(255)
- subject : varchar(255)
- message : varchar(255)
- human : varchar(255)

mydatabase donate
- name : varchar(255)
- email : varchar(255)
- address : varchar(255)
- comment : varchar(255)
- cardno : varchar(255)
- nameoncard : varchar(255)
- country : varchar(255)
- region : varchar(255)
- city : varchar(255)
- state : varchar(255)
- zip : varchar(255)
- amount : varchar(255)

mydatabase adminregistration
- ID : int(11)
- Name : varchar(255)
- Email : varchar(255)
- Password : varchar(255)
- Qualification : varchar(255)

mydatabase eventsregistration
- ID : int(11)
- Name : varchar(40)
- Contact : varchar(255)
- Email : varchar(255)
- eventname : varchar(255)
- upcomingevent : varchar(255)

mydatabase participants
- name : varchar(255)
- email : varchar(255)
- phone : varchar(255)
- program : varchar(255)
- address : varchar(255)
- participant : varchar(255)
- cancertype : varchar(255)
- hospital : varchar(255)
- Checkboxvalue : varchar(255)

mydatabase classesregistration
- ID : int(11)
- classcode : varchar(40)
- Email : varchar(255)
- classselection : varchar(255)

mydatabase newsletters
- name : varchar(255)
- email : varchar(255)
- subject : varchar(255)
- message : varchar(255)
- human : varchar(255)

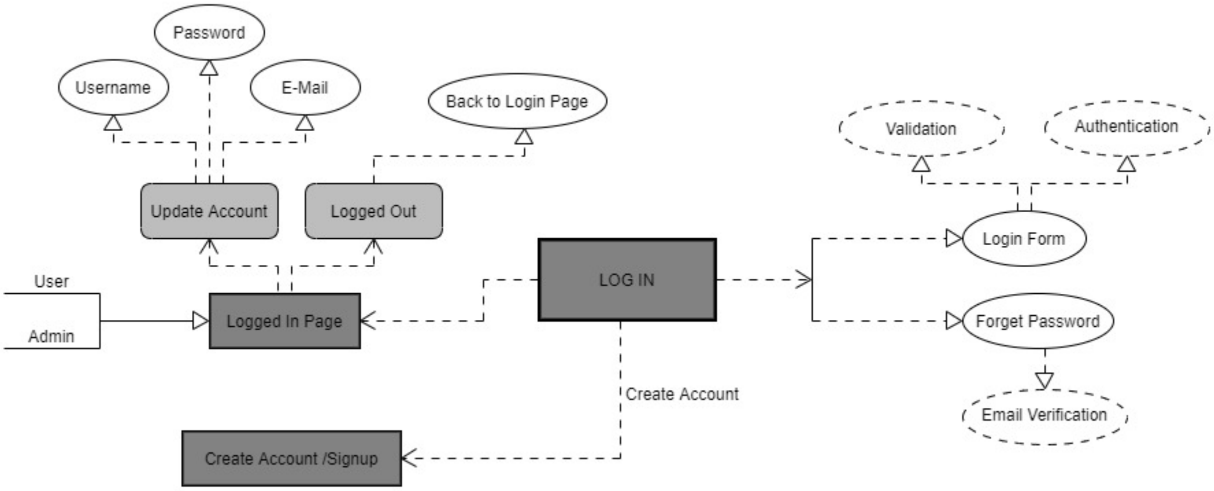
mydatabase registration
- ID : int(11)
- Name : varchar(300)
- Email : varchar(300)
- Password : varchar(300)
- Qualification : varchar(300)

mydatabase specialevents
- name : varchar(255)
- email : varchar(255)
- subject : varchar(255)
- message : varchar(255)
- human : varchar(255)

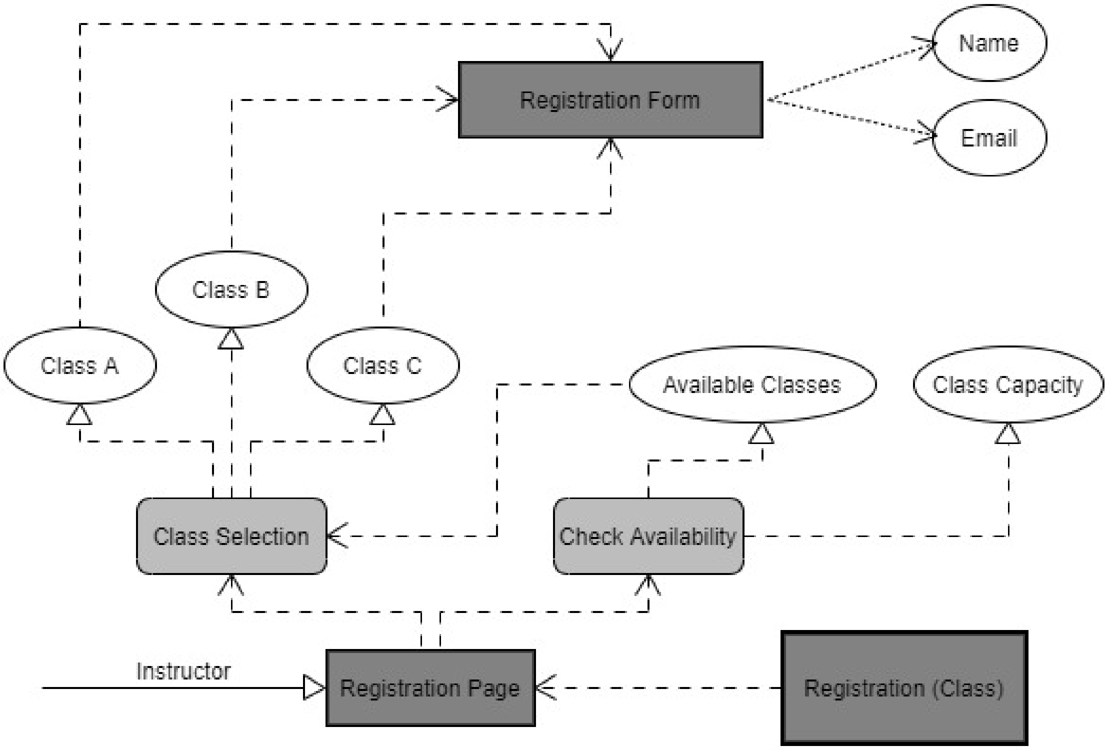
mydatabase volunteers
- areaofIntrest : varchar(500)
- availability : varchar(500)
- name : varchar(500)
- address : varchar(500)
- city : varchar(500)
- state : varchar(500)
- zip : varchar(500)
- email : varchar(500)
- phone : varchar(500)
- preference : varchar(500)
- location : varchar(500)
- convicted : varchar(500)
- jobTitle : varchar(500)
- employeer : varchar(500)
- eaddress : varchar(500)
- ecity : varchar(500)
- estate : varchar(500)

onsole

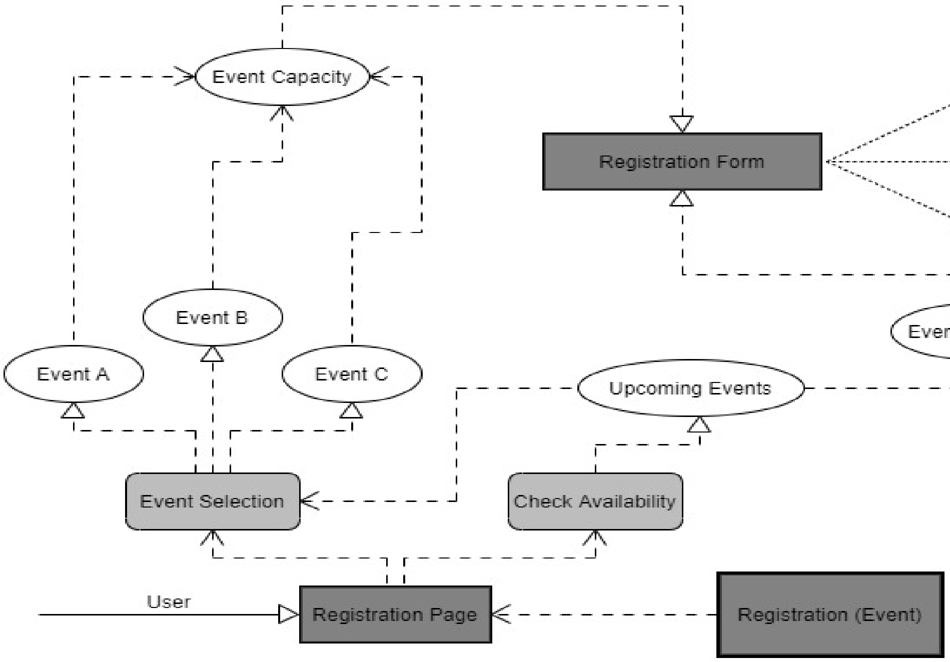
Below is the Node graph and Data Model/ Class Diagram for this project:



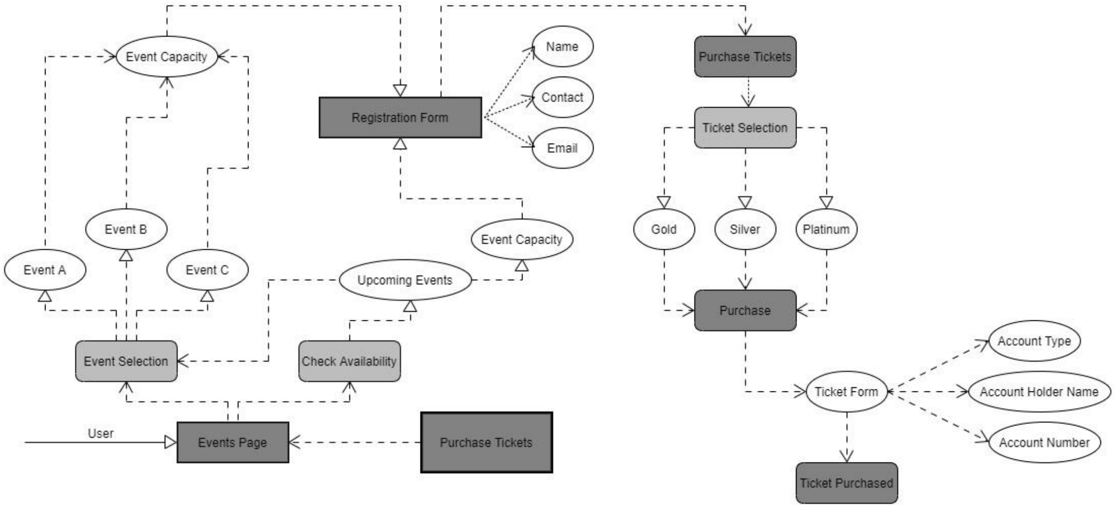
Above node is the login diagram of the website for user and for admins. In detail of the login part, we have created a two logins form, where user can create an account where it will ask their name, phone number, and email, after user successfully created an account user can after register for event, register for class, and also purchase a ticket along with that user have an option to select various types of event and different type of classes as well. We also created a login for admins, where admins can access to user account and events and classes that user registered and make changes to it if they want. User can also rest their password as well.



Above node is the class registration when a user tries to register for a class user will have more than one option to select the class this part will also check for class capacity and class availability if the class if full it will prompt the user saying "there's was an error processing" and this message can easily be change in the php file. In order for user to register for class, first user has to create account in order to access this function.

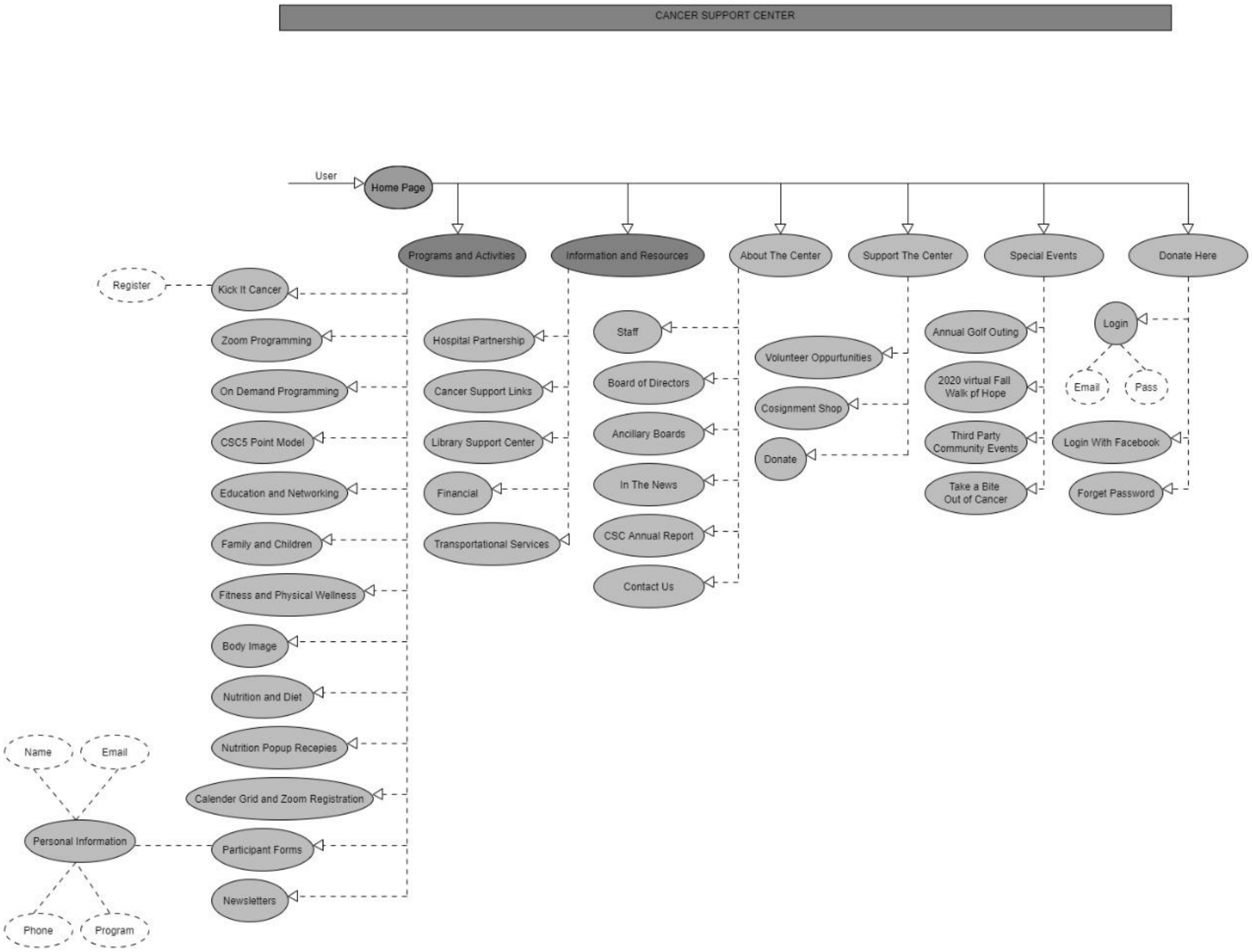


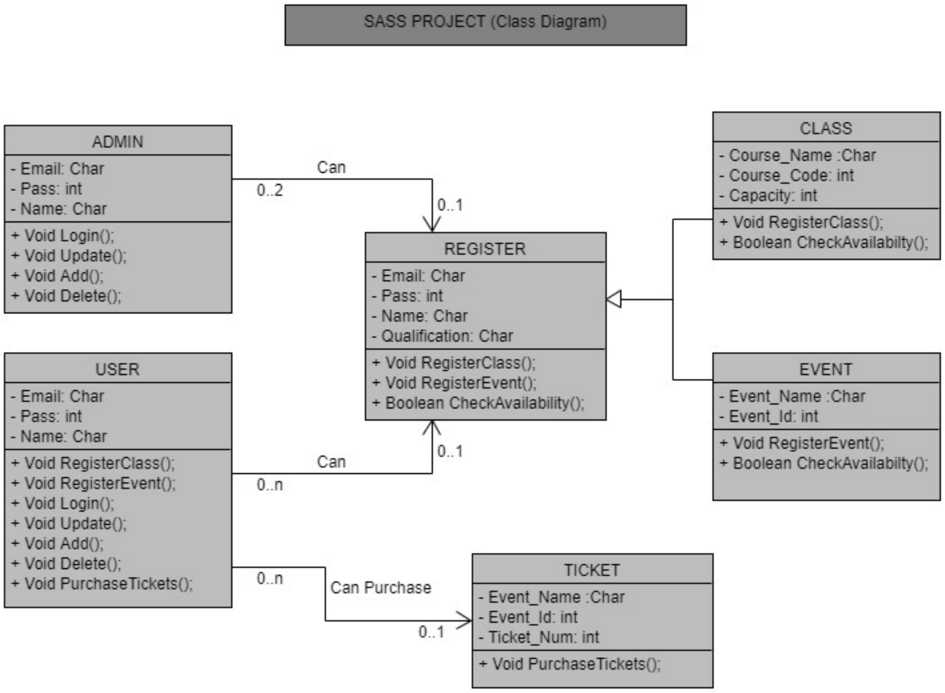
Above node is registration diagram same functionality for this when a user tries to register for the event user will have more than one option to select the class this part will also check for event capacity and if the event is full, it will prompt the error message to user.



Above node is the purchase ticket diagram where use can purchase for events. For this part there's only purchase ticket and this part will not ask users financial information, out team was planning to add but unfortunately, we could not do it due to very limited time.

Below is the class diagram, and all the functionality currently the website has.





Sperate logins for Admin and Users	New added feature
Class registration	New added feature
Event registration	New added feature
Ticket purchasing	New added feature
Donate Link	New added feature (This feature previously was running with third party website (classy.org)

Above feature was all new feature we added to existing. Donate link it was running with the third-party website (classy.org). We enhance and add the donate link within the website.

Test Plan: We fully tested the website multiple times, after couple of tests we found minor errors when admin tries to update an event for user from admin account, admin gets a fatal error. Along with that we also noticed one minor error, the user cannot access purchase ticket from his account our plan was to add this feature to user, but it got redirected to Admin, the website has the ticket purchase page, and it is working perfectly fine but only admins can access that and not user, but this can be fixed with php file, we were not able to resolve this issue due to limited time.

In conclusion our group was successfully able to implement the key functionality to the Cancer Support Center website. We can say that our group were able to accomplish all the things that we planned. Suggestion we have for future development is to enhance little more on the ticket purchasing and add the reset link feature using API. Our group added the password reset link, but we were having some difficulties with sending the “real” password reset link to user emails, but we tried our best and we hard code the password reset feature, so when a user tries to reset the password, it will be the same reset conformation number for all users.

Website Admin Login Info:

Name: Admin

Email: admin@gmail.com

Password: admin@123