6-2017

CHILDREN’S SOCIAL NETWORK: KIDS CLUB

Eiman Alrashoud

Recommended Citation
Alrashoud, Eiman, "CHILDREN’S SOCIAL NETWORK: KIDS CLUB" (2017). *Electronic Theses, Projects, and Dissertations*. 451.
https://scholarworks.lib.csusb.edu/etd/451

CHILDREN'S SOCIAL NETWORK:
KIDS CLUB

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

---

by

Eiman Alrashoud

June 2017

CHILDREN'S SOCIAL NETWORK:

KIDS CLUB

A Project

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Eiman Alrashoud

June 2017

Approved by:

Dr. Ernesto Gomez, Adviser, Computer Science and Engineering

Dr. George Georgiou, Committee Member

Dr. Tong Lai Yu, Committee Member

ABSTRACT

Young children often have a profound interest that if nurtured, would develop to great social cues and skills thereby improving their social aspects of life. Parents can conveniently benefit from a swift data sharing in the collaborative scrutiny of their kid's participation, in public activities facilitated through the internet digital technology. To facilitate the involvement of shared activities among children, an interactive website is essential. The aim of my project is to develop a website that is intended to be an interactive platform for a variety of events selection. Additionally, the website will aid parents in the creation, discovery and reach for organized local events that fit their kid's interests in description and age. A variety of events will be availed at the website for scrutiny in finding friends, sharing and learning new activities. Similarly, it will be used for fun engagement. The website is implemented by using Microsoft Visual Studio 2012 Professional, C# programming language, and SQL Server Management Studio 2012 to handle the data.

ACKNOWLEDGEMENTS

My thanks and appreciations to all who helped me in this project. First and foremost, I would like to express my deep gratitude to my mother and father for their continuous prayers and encouragement. My very great thanks to my supporter and husband Abdullah for being always helping and encouraging me through my studying years to the period of working on this project. I am also grateful to my daughter Maria for supporting me spiritually and providing me the power throughout my studying and my life in general. Special thanks to my adviser, Dr. Gomez for his guidance and valuable suggestions during this project’s development. I also would thank the committee members Dr. Georgiou and Dr. Yu. Thank you.

TABLE OF CONTENTS

CHAPTER ONE: INTRODUCTION

CHAPTER TWO: REQUIREMENTS

CHAPTER THREE: SYSTEM SPECIFICATION

CHAPTER FOUR: SYSTEM DESIGN

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Introduction

Many parents are in need of suitable communities for their children to spend their leisure time and share activities with other children of similar ages. Most children make friends with their peers during their early childhood, and putting them in positions for having rich activities assists in developing social and communication skills not limited to their parents. For instance, playing is a significant method for ideal child development. It does not provide the opportunity to enjoy themselves, but also provides the availability of playmates which encourages children to communicate and practice language. The environment the children grow up in which involves sufficient interactions with their peers, paves the way for social and easily established relationships in the future. Healthy social development is an essential requirement for individuals to adjust well in their society [1].

We live in a period in which a lot of innovative technologies have been adopted. These technologies contribute to positively improving our daily lives. Technology and communication have opened up new horizons for millions of people by contributing to the social development of their societies. Social networks are an example of these technologies, and some of them are concerned with child development.

Building a network allows parents to find a suitable group for children, and provides the children with an opportunity to make new friends, and acquire experiences and skills. Due to the limited activities inside homes, this type of network offers more experiences. It prepares children for cooperative life where a child enjoys shared and organized groups. In addition, the network provides a way to create groups with those who have the same interests and facilitates by meeting them face-to-face.

Additionally, grouping networks can be used to form support groups for some specific groups of children who suffer from diseases or disorders. There is evidence that joining support groups has a positive impact on well-being. Furthermore, special needs children tend to hide their conditions from other normal peers. However, if they have contact with kids in a similar situation, they will feel more confident in socializing [2]. Moreover, unity, sharing experiences and psychological support are more likely to be present if group members have the same condition [3]. For instance, studies reveal an increase in important social skills of children with autism when they socialize with similar disability [4].

One advantage of these networks is that socialization is not restricted to particular places like schools or a certain period of the day. Additionally, it is beneficial as it becomes an easy task to discover proper groups and create events to gather children with similar ages, interests, and conditions, or who live in the same areas via the convenience of technology. The network can be accessed at any time of the day regardless of time or location. Further, it serves various households that are different in cultural, social and economic levels.

1.2 Project Objectives

The aim of my project is to develop a website that helps parents to create, discover, or join organized local events for their kids that fit their ages and interests. It is a channel to find friends, learn and share activities, spend fun time, be a member of a group and improve the concept of social life. It facilitates participation in shared activities between children in a convenient and timeless way. The website provides a wide selection of events; therefore, parents can easily pick what is best for their children.

1.3 Scope

The website is targeted at parents or individual with children. It can be used by businesses and organizations to create specific events related to children. The system should work on the following browsers: Internet Explorer, Firefox, Google Chrome, and Safari. The website can be accessed at any time in a convenience way. Furthermore, the system should be used by two types of users, and each one of them has special features:

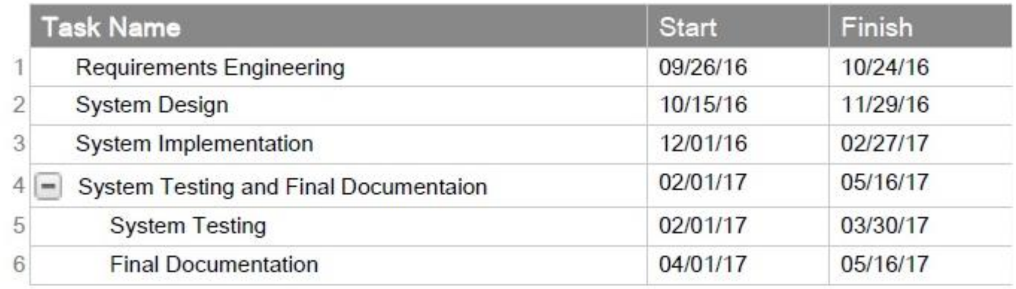
1. The member who is able to register in the system and create, view, or join organized local events for their kids that fit their ages and interests.

2. The administrator who is able to manage the website and the accounts. Further, he/she can approve, deny, or delete any event.

1.4 Project Work Plan

In this project, the work plan is divided into four tasks to achieve the project objectives. These tasks will be implemented within almost 7 months.

Table 1. Project Work Plan



	Task Name	Start	Finish
1	Requirements Engineering	09/26/16	10/24/16
2	System Design	10/15/16	11/29/16
3	System Implementation	12/01/16	02/27/17
4	System Testing and Final Documentaion	02/01/17	05/16/17
5	System Testing	02/01/17	03/30/17
6	Final Documentation	04/01/17	05/16/17

Task 1: Requirements Engineering

This task includes:

1. The requirement elicitation: it is the first step of the requirements' engineering task. It is performed by gathering requirements that represent what the website should do.
2. The Software Requirement Specification (SRS) document: creating this document, that is used in the system design task, is an important step after analyzing and validating the elicited requirements.

Task 2: System Design

This task includes:

1. The Software Design Document (SDD): creating this document helps to state all details about the system architecture, modeling, and interfaces. These details are used in the system implementation task.

Task 3: System Implementation

This task includes:

1. The process of coding: the method to integrate the system by converting the software design document into codes.
2. The unit and integration tests.

Task 4: System Testing and Final Documentation

This task includes:

1. Writing detailed test cases.
2. Checking whether the system meets all the functional and non-functional requirements.
3. Verifying all possible input to check for desired output.
4. Documenting the system requirements, design, implementation, and testing details.
5. Creating the website user manual.

CHAPTER TWO

REQUIREMENTS

2.1 Introduction

2.1.1 Purpose

The purpose of this chapter is to list and explain the requirements of the website that need to be considered to meet the users' needs. Furthermore, this chapter is used as a guide to clarify system functions and its boundary. In addition, it is used for the next phase of the project which is the system analysis and design.

2.1.2 Definitions, Acronyms, or Abbreviations

Table 2. Acronyms and Abbreviations

Term	Definition, acronyms, and abbreviations
IEEE	Institute of Electrical and Electronics Engineer
User	The user who is not a member yet
Member	The user who is registered in the system
Admin	The administrator of the system
URL	Uniform Resource Locator
UI	User Interface
GUI	Graphical User Interface

WAN	Wide Area Network
LAN	Local Area Network
CPU	Central Processing Unit
GHz	Gigahertz, 1 GHz equals 1,000,000,000 Hz
MB	Megabyte, 1 MB equals 1,000,000 byte
GB	Gigabyte, 1 GB equals1,000,000,000 byte
RAM	Random Access Memory
HTTP	Hypertext Transfer Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol

2.1.3 Format of Chapter

We follow the IEEE format for the software requirements specification [6].

2.1.4 Overview

The first section of this chapter gives a detailed description of the functional requirements. The second section includes the nonfunctional requirements. The last section covers the external interface requirements, including user, hardware, software, and communication interfaces [7].

2.2 Functional Requirements

2.2.1 The User is Able to Create an Account on the Website

The user is able to create an account and start using the website by completing the sign-up form which includes: first name, last name, user name, email, password, confirm password. After filling out the form, the system will validate the entered information, and show terms and conditions to allow the user to read them before completing the registration process. The terms and conditions include:

a. No one under 18 is allowed to create an account.
b. Members must be honest and authentic. Creating fake accounts, or impersonating another person, is illegal.
c. Members should respect other's opinions, time, and private boundaries. Additionally, the website prohibits any kind of bullying, harassment, or exploitation in order to create a safe community.
d. Any member can report inappropriate content or fake accounts. A report should be clear about the situation; consequently, the website administrator will assess the case and terminate the account if needed.
e. All interactions between members will be monitored and any behavior that is considered disrespectful, dishonest, or involves any violation will lead to account termination.
f. The website will show a legal disclaimer statement that the website is not responsible for any harm resulting from joining an event, and participating

in any event is at your own risk. Even though every member agrees to website terms and conditions, this agreement is not directly between members. Therefore, the member should be cautious and report to the police if needed when observing any suspicious behavior.

g. The private member information is protected and it will not be used for marketing or other purposes. However, it may be disclosed if required by law enforcement or government officials.

When the user agrees on these terms, an activation link is sent to his/her email in order to sign in and start using the website.

2.2.2 The Member and the Admin are Able to Sign in to the Website

They sign in to their accounts, and start using the system functions after a successful sign in by providing a correct username and password. The admin uses a different URL to sign in.

2.2.3 The Member is Able to View or Edit his/her Profile

The member can access his/her profile and view the profile details such as: first name, last name, username, email, photo, description, the number of kids, and their ages. In addition, the member is able to edit or add any of these details except the email and username.

2.2.4 The Member is Able to Create an Event

The member can create a new event or activity as an event organizer. After the website shows the conditions of creating a new event, the member provides the following information: an appropriate event topic, location (address, city, state, zip code), event description, date, time and other details. At that point, the event details are saved to be reviewed and approved by the administrator.

2.2.5 The Member is Able to View or Edit any Event that is Created Previously

The member can access any event that is created previously by him/her, and view its details such as: the event topic, description, location, date, time, and other details. In addition, the member is able to edit any of these details.

2.2.6 The Member is Able to Cancel any Event that is Created Previously

The member can cancel any event that is created previously by him/her for any reason. In this case, an email is sent to all participating guests informing them that the event has been canceled.

2.2.7 The Member is Able to Search for any Event that is Created Previously

The member can search for an event that is created previously by other members, and display a list of events based on a keyword or a zip code. After clicking on the desired event, the system displays the event page which includes: the event topic, description, location, date, time, participating guests, discussion, and other details.

2.2.8 The Member is Able to Join any Event that is Created Previously

The member can join an event that is created previously by other members. When the member clicks on the desired event, its page will be opened. The member can click on the “Join Event” button, and an email is sent to the event organizer “New member has joined your event.”

2.2.9 The Member is Able to Leave any Event that he/she Joined Previously

If the member decides not to participate in any event that he/she joined previously for any reason, he can leave the event by clicking on the "Leave Event" button. An email is sent to the event organizer "The member (name) has left your event."

2.2.10 Any Member is Able to Display the Participating Members List

Any member can display the participating members list of any event to see the other members who are coming to the event and access their profiles.

2.2.11 The Member is Able to Add a Comment

The member can add a comment and discuss with the other members about a specific event. The comments are displayed in the discussion part of the event page.

2.2.12 The Member is Able to Delete a Comment

The member can delete any comment that is already created by him/her. Consequently, the comment is removed from the discussion part of the event page.

2.2.13 The Member is Able to Like or Dislike a Comment

The member can like or dislike any comment that is already created by the other members.

2.2.14 The Admin is Able to View the Events that are Created Previously

The admin views the events that are created previously by members by displaying a list of events, and he has the ability to access any event page.

2.2.15 The Admin is Able to Delete any Event that is Created Previously

The admin can delete any event that is created previously by members if required, and an email is sent to the event creator and all participating guests informing that the event has been canceled.

2.2.16 The Admin is Able to Approve or Deny any Event

The admin should review the events that are created previously by members in order to approve and publish them to all members, and an email is sent to the event creator informing that the event has been approved or denied.

2.2.17 The Admin is Able to View all Accounts that are Registered in the System

The admin can view all accounts that are registered in the system. Additionally, the admin can search for a specific account by email or name.

2.2.18 The Admin is Able to Delete any Account that is Registered Previously

The admin is able to delete any account that is registered in the system if required.

2.3 Nonfunctional Requirements

2.3.1 Reliability

The system should provide accurate results to the user. The system will be able to handle all errors.

2.3.2 Performance

The system will work at high speed, which includes fast loading and fast response time.

2.3.3 Availability

The system will be available for members 24 hours every day. The system should be connected to the Internet in order to access the database.

2.3.4 Security

The system is accessible only to authorized users and by authorized ways.

2.3.5 Usability

The system will be implemented with a simple User Interface (UI) to guarantee easy use. The website will alert the user of required fields. Also, it will provide the user with proper clear messages, tips, and error handling messages when the user tries to enter data.

2.3.6 Portability

The system will be usable in different environments, operating systems, and Internet browsers.

2.3.7 Privacy

The private information is protected and will not be used without member permission.

2.3.8 Safety

It will not cause any harm or damage.

2.3.9 Efficiency

It requires minimal cost, time and resources.

2.3.10 Maintainability

There is an ability to update or correct any function of the system.

2.3.11 Testability

There is simplicity in creating tests that show system performance and functions.

2.4 External Interfaces

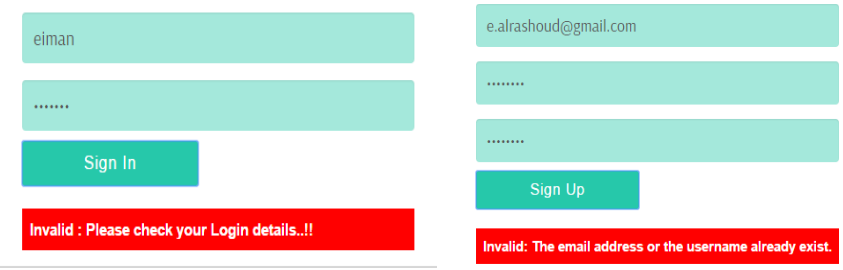
2.4.1 User Interfaces

The UI includes UI controls. Some of them support interaction with the user such as buttons and checkboxes while some of them do not such as windows and panels. The UI provides an interaction between the user and the website [5]. The Graphical User Interface (GUI) contains the website home page, member pages, and the admin pages that allow both types of users to interact with the website and show all of its functions [7].

The website home page is accessible by any user without logging in or being a member. On this page, the user is able sign in if he/she is already a member, or sign up to create a new account and be a member. Furthermore, the user is able to display "About Us", "Contact Us", "Privacy policy" pages. The admin has a different URL to sign in.

When the user signs in by entering the user name and the password, he/she will be redirected to the his/her home page. The admin has a different home page from the member home page.

To increase the usability of the website, it alerts the user of required fields. Also, it will provide the user with proper clear messages, tips, and error handling messages (See Figure 1). Furthermore, there are different messages such as when the form submission has been done successfully, or when the user attempts to change or delete any content. In this case, the website will not complete the request until the user confirms it (See Figure 2).



eiman

•••••••

Invalid : Please check your Login details..!!

e.alrashoud@gmail.com

••••••••

••••••••

Invalid: The email address or the username already exist.

Figure 1. Error Messages

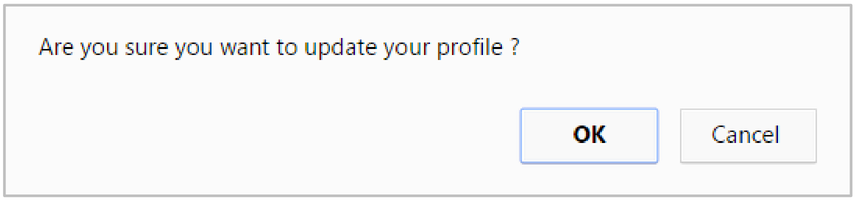

Are you sure you want to update your profile ?

Figure 2. Confirmation Message

2.4.2 Hardware Interfaces

Users of the website shall have a device running on an operating system with a mouse or a touch screen. Since the website should run over the internet, the device needs to be connected to the internet. For example, a modem, WAN or LAN, and a cable may be required [8]. The device should have the following minimum specifications: a CPU speed of 1 GHz, RAM capacity of 512 MB, hard drive capacity of 1 GB.

2.4.3 Software Interfaces

The website communicates with the database in order to get members information. Additionally, database communication is required to access events' information. It is not only for accessing but also the website may add, delete, or change the data in the system database. On the other hand, the database is only able to read from the system.

2.4.4 Communications Interfaces

The communication between the various components of the system is significant. In this project, the communication is handled by operating systems. In particular, the website uses the HTTP protocol for the internet connection and the TCP/IP protocol for the intranet communication.

CHAPTER THREE

SYSTEM SPECIFICATION

3.1 Introduction

3.1.1 Purpose

The aim in this chapter in general is to specify who uses the system, what are the functions they can use, and it describes the behavior of the system and the requirements. This chapter also provides a detailed description of the website functions and how the system performs different actions in collaboration with one or more users by using a use case diagram. Consequently, this chapter will be the first reference for the implementation task since it provides a means of understanding of the system.

3.1.2 Definitions, Acronyms or Abbreviations

Table 3. Acronyms and Abbreviations

Term	Definition, acronyms, and abbreviations
Member	The user who registers in the website, and can use the website functions.
Admin	The administrator of the website who has the authorization to manage the website.

Unregistered user	The user who is not a member yet, and can access the website homepage and sign up.

3.1.3 Overview

In this chapter, analyzing and understanding the system should be achieved. It provides a clear explanation of how the members functions will be implemented. Additionally, the administrator functions scenarios are included. This chapter provides a well-defined use case diagram along with a complete description of each use case.

3.2 Use-Case Modeling

3.2.1 Use-case diagram

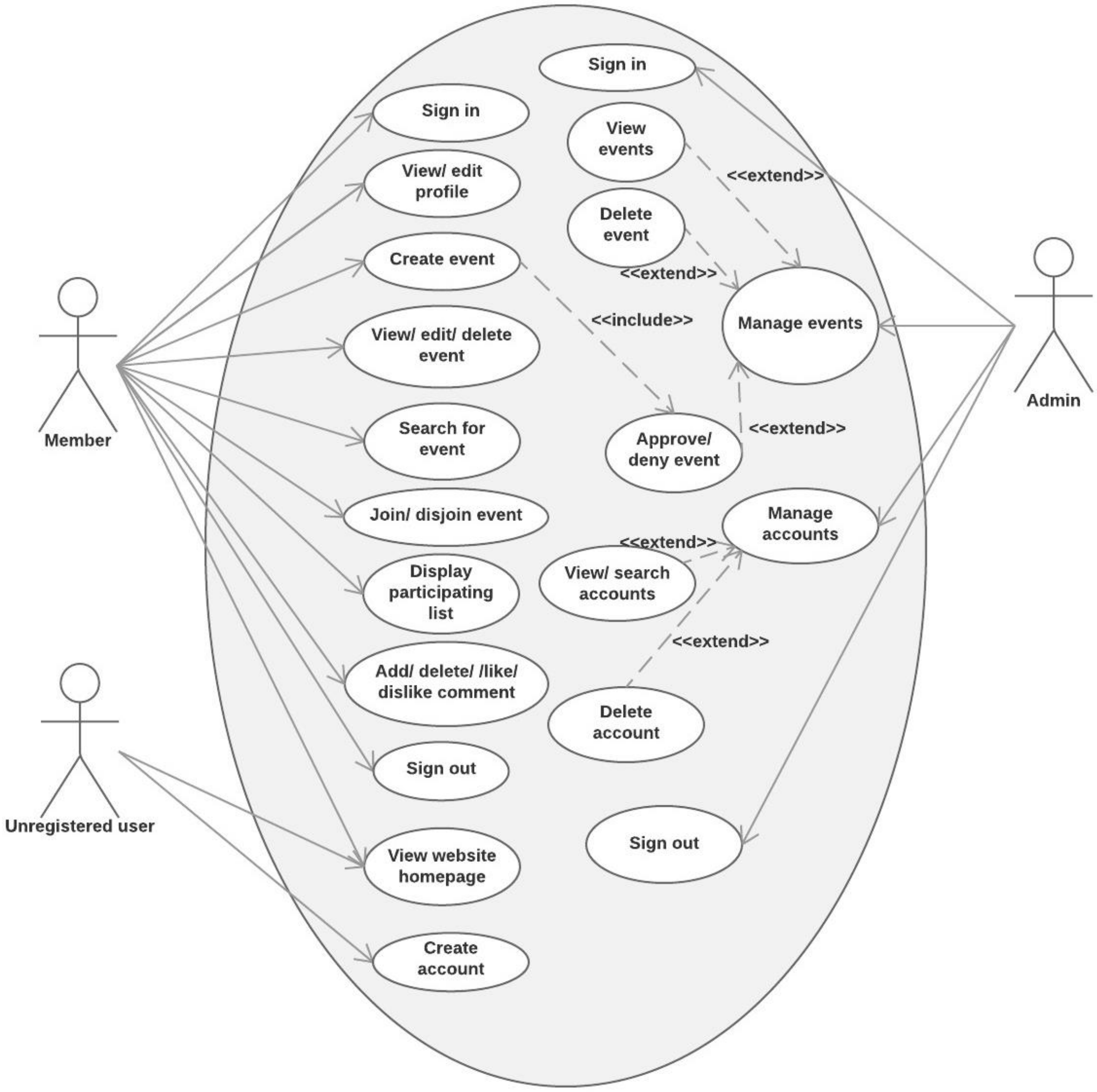


Figure 3. Use Case Diagram

3.3 Use-Case Descriptions

3.3.1 Create an Account

Actors: unregistered user.

Description: user creates an account to be a member and can use all website functions.

Preconditions: the website homepage is loaded.

Postconditions: the user is successfully registered in the system.

Normal Flow:

1. The user clicks on the “Sign up” button on the website homepage.
2. The system opens the “Sign up” page which includes the sign up form.
3. The user enters the following information such as first name, last name, user name, email, password, confirm password. The users may enter more optional information such as a description, number of kids, and their ages. Additionally, the users can upload a picture when they are already registered through their profile page.
4. The user clicks on:
    - The “Sign up” button to proceed to the next step.
    - Or the “Cancel” button, then the user is directed to the website homepage.
5. The system shows the conditions and terms page.
6. The user clicks on:
    - “I agree” button to proceed to the next step.

- Or “I disagree” button, then the user is directed to the website homepage.

7. The system validates the entered information and shows a confirmation message “You have been registered successfully. Your account will be activated after email confirmation.”

Exceptions:

4.a. In step 4 of the normal flow, if the user enters an existing username or email:

4.a.1. The system prompts the user to enter a different username or email.

4.a.2. Going back to step 3 to enter a new username or email.

4.b. In step 4 of the normal flow, if the user enters a password less than 8 characters or more than 12 characters:

4.b.1.The system prompts the user that the minimum length of the password is 8 and the maximum length of the password is 12 characters.

4.b.2. Going back to step 3 to enter a valid password.

4.c. In step 4 of the normal flow, if the user enters an invalid email format:

4.c.1. The system prompts the user to enter a valid email format.

4.c.2. Going back to step 3 to enter a valid email.

4.d. In step 4 of the normal flow, if the user enters different passwords in the password and confirmation fields.

4.d.1. The system prompts the user to enter matched passwords.

4.d.2. Going back to step 3 to enter a matched password.

4.e. In step 4 of the normal flow, if the user clicks "Sign up" and leaves any required field empty:

4.e.1. The system prompts the user “Please fill in the required fields.”

4.e.2. Going back to step 3 to fill in all the required fields.

Assumptions:

The user knows the basic skills to use a computer and has the ability to understand the English language.

3.3.2. Sign in

Actors: admin, member.

Description: The user signs in to his/her account, and starts using the system functions.

Preconditions: the user already has been registered in the system, and the website main page is loaded.

Postconditions: the user is successfully signed in.

Normal Flow:

1. The user clicks on the “Sign in” link in the homepage.
2. The website shows a sign in form.
3. The user enters the username and password, and click on the “Sign in” button.
4. The system validates the username and password, and redirects to the member home page.

Exceptions:

4.a. If the user enters invalid data:

 4.a.1. The system displays a message to enter valid data.

 4.a.2. Going back to step 3 and the user tries again to enter valid username and password. Otherwise, the user can click on the "Remember me" button to have the username and password sent to his/her email.

4.b. If the user leaves any blank field:

 4.b.1. The system displays a message to fill in the required fields.

 4.a.2. Going back to step 3, the user fills in the required fields.

Assumptions:

The user knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.3 View/ Edit Profile

Actors: member.

Description: the member can access his/her profile and view the profile details such as: name, username, password, email, photo, description, number of kids, and their ages. In addition, the member has the ability to edit any of these details except the username and the email.

Preconditions: the member is signed in to the system, and the member homepage is opened.

Postconditions: the member profile is accessed successfully, or the member profile is updated successfully.

Normal Flow:

1. The member opens the member profile page.
2. The system shows the member information such as name, username, password, email, photo, description, number of kids, and their ages.
3. The member enters the new information, and fills in the fields he/she wants to change except the username and email.
4. The member clicks on the "Update" button to save changes.
5. After clicking "Update" button, the website shows a message “Are you sure you want to update your profile?”
6. The member clicks on:
    - “Ok”, then the member proceeds to the next step.
    - "Cancel", then the member is directed to the profile page.
7. After clicking the “Ok” button, the website shows a message “Changes have been saved successfully.”
8. The updated information is shown in the member profile page.

Exceptions:

4.a. In step 4 of the normal flow, if the member clicks on “Update” and leaves any required field empty such as the first name and last name:

4.a.1. The website shows a message “Please fill in all the required fields.”

4.a.2. Going back to step 3 to fill in all the required fields.

4.b. In Step 4 of the normal flow, if the user uploads an invalid picture type or size:

4.b.1. The system prompts the user to upload a valid picture type or size.

4.b.2. Going back to step 3 to enter a valid picture.

Assumptions:

The member knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.4 Create an Event

Actors: member.

Description: the member can create a new event or activity as an event organizer.

Preconditions: the member is signed in to the system, and the member homepage is opened.

Postconditions: the new event is created successfully.

Normal Flow:

1. The member clicks on the “Create new event” link.
2. The website shows the conditions of creating new event which includes:
    - Make sure that the purpose for the event is proper, and involves sharing a goal, interest, activity, or identity.
    - Make sure that all proper rules as well as city, state and federal regulations are followed.
    - Respect other members time, opinions, and private boundaries.
    - The event should be in a public place.
    - Event topic and description must be clear and accurate to let others make the right decision about their participation.

- The event can be denied by the administrator, if a violation occurs by the member (event organizer).

3. The member clicks on the “Next” button.
4. The website shows a form of creating new event.
5. The member enters the event information:
    - Appropriate event name.
    - Location (address, city, state, zip code).
    - Event date and time.
    - Event description.
6. The member clicks on:
    - “Next” to proceed to the next step.
    - or “Back” to the conditions page.
7. The website shows questions to be answered by the organizer [9]:
    - ✓ Is there a charge admission to the event?
    - ✓ Will you be having food? If so, what kind?
    - ✓ Will the participating guests need to bring food? If so, what kind?
    - ✓ Are pets allowed in the event?
    - ✓ Is the event accessible to wheelchair members?
8. The member clicks on:
    - The “Create” button to proceed to the next step.
    - or “Back” to the event form page.

9. After clicking the “Create” button, the website shows a message “The event has been created successfully. Your request will be reviewed and we will contact you soon.”
10. The event details are saved to be reviewed by the admin.

Exceptions:

6.a. In step 6 of the normal flow, if the member clicks "Next" and leaves any required field empty:

    6.a.1. The website shows a message “Please fill in all the required fields.”
    6.a.2. Going back to step 5 to fill in all the required fields.

6.b. In step 6 of the normal flow, if the member enters an invalid zip code:

    6.b.1. The website shows a message “Please provide a valid zip code.”
    6.b.2. Going back to step 5 to enter a valid zip code.

Assumptions:

The member knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.5. View/Edit Event

Actors: member.

Description: the member can access the event that is created previously by him/her, and view the event details such as: the event topic, description, date, time, and other details. In addition, he/she had the ability to edit any of these details.

Preconditions: the member is signed in to the system, and the member homepage is opened.

Postconditions: the event details are accessed successfully, or the event information is changed successfully. The new event information is sent to all participating guests by email.

Normal Flow:

1. The user clicks on the “My events” link on the member homepage.
2. The system opens the “My events” page, and shows the list of events the member created, and the list of events the member joined.
3. The member clicks on the desired event that he/she created.
4. The event page is opened, including information such as: the event topic, description, location, date, time, participating members and other details.
5. The member enters the new information, and fills in the fields he/she wants to change.
6. The member clicks on the “Update” button to save changes.
7. After clicking the “Update” button, the website shows a message "Changes have been saved successfully.”
8. The updated information is shown on the event page, and an email is sent to all participating guests by email informing that the event information has been updated.

Exceptions:

6.a. In step 6 of the normal flow, if the member clicks on "Update" and leaves any required field empty:

6.a.1. The website shows a message “Please fill in all the required fields.”

6.a.2. Going back to step 5 to fill in all the required fields.

Assumptions:

The user knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.6. Delete Event

Actors: member.

Description: the member can cancel the event that is created previously by him/her.

Preconditions: the member is signed in to the system, and the member homepage is opened.

Postconditions: the event is canceled successfully, and an email is sent to all participating guests, informing that the event has been canceled.

Normal Flow:

1. The user clicks on the “My events” link on the member homepage.
2. The system opens the “My events” page, and shows the list of events the member created, and the list of events the member joined.
3. The member can click on the "Delete" icon of the event that he/she wants to delete.

4. The system shows a message "Are you sure you want to cancel this event?"
5. The member clicks on:
    - “Ok” to cancel the event.
    - Or “Cancel”, then the member is directed to the event page.
6. After clicking the “Yes” button, the website shows a message “Event has been canceled successfully.”
7. The event is deleted from the events’ page, and an email is sent to all participating guests informing that the event has been canceled.

Assumptions:

The user knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.7 Search for an Event

Actors: member.

Description: the member can search for an event that is created previously by other members, and display a list of events based on zip code. Each event includes all details such as: the event topic, description, location, date, time, and other details.

Preconditions: the member is signed in to the system, and the member homepage is opened.

Postconditions: the events’ list is accessed successfully, and the member is able to open any event page.

Normal Flow:

1. In the member homepage, there is an events search engine which includes text fields to enter a keyword and a zip code.
2. The member enters a keyword, a zip code, or both.
3. The member clicks on the “Search” button.
4. A list of events is shown based on the entered information.
5. The member can click on any event of the list.
6. The website opens the event page which includes event details such as: the event name, description, location, date, time, event organizer, participating members and more details.

Exceptions:

3.a. In step 3 of the normal flow, if the member clicks on “Search” and enters an invalid zip code:

3.a.1. The website shows a message “Please enter a valid zip code.”

3.a.2. Going back to step 2 to enter a valid zip code.

Assumptions:

The user knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.8 Join an Event

Actors: member.

Description: the member can join an event that is created previously by other members.

Preconditions: the member is signed in to the system, and the desired event page is opened.

Postconditions: the member joined the event and has been added to the participating members' list.

Normal Flow:

1. The member clicks on the "Join Event" button at the top of the event page.
2. The system shows a message "Are you sure you want to join this event?"
3. The member clicks on:
   - "Ok" to join the event.
   - Or "Cancel", then the member is directed to the event page.
4. After clicking the "Yes" button, the website shows a message "You have joined the event successfully."
5. The member is added to the participating members list. The event is added in the "My events" page in the section of the events the member joined, and an email is sent to the event organizer "New member has joined your event."

Assumptions:

The user knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.9 Leave an Event

Actors: member.

Description: the member can leave an event that he/she joined previously.

Preconditions: the member is signed in to the system, and the “My events” page is opened.

Postconditions: the member has left the event and has been deleted from the participating members’ list.

Normal Flow:

1. In the “My events” page, the member clicks on the event that he/she wants to leave from the section of the events the member joined.
2. The system displays the event page.
3. The member clicks on the “Leave Event” button at the top of the event page.
4. The system shows a message “Are you sure you want to leave this event?”
5. The member clicks on:
    - “Ok” to leave the event.
    - Or “Cancel”, then the member is directed to the event page.
6. After clicking the “Yes” button, the website shows a message "You have left the event successfully.”
7. The member is deleted from the participating members’ list. The event is deleted from the “My events” page in the section of the events the member joined, and an email is sent to the event organizer informing that: “The member (name) has left your event.”

Assumptions:

The user knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.10 Display Participating Members' List

Actors: member.

Description: the member can display the participating members' list of any event to see the other members who are coming to the event and access their profiles.

Preconditions: the member is signed in to the system, and the desired event page is opened.

Postconditions: the member accessed the participating members' list.

Normal Flow:

1. The member clicks on the "Members" tab at the top of the event page.
2. The system displays a list of the members who joined the event.
3. The member can click on any member to display his/her profile.
4. The system opens the member profile page, which includes: the member name, photo, description, number of kids, and their ages.

Assumptions:

The user knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.11 Add Comment

Actors: member.

Description: the member can add a comment and discuss with other members about a specific event.

Preconditions: the member is signed in to the system, and a specific event page is opened.

Postconditions: the comment is added.

Normal Flow:

1. The member clicks on the “Discussion” tab at the top of a specific event page.
2. The system shows the discussion page, which includes a text field to add a comment.
3. The member enters the comment in the text field.
4. The member clicks on “Submit” to publish the comment.
5. The system shows a message “Are you sure you want to submit this comment?"
6. The member clicks on:
    - “Ok” to confirm adding the comment.
    - Or “Cancel”, then the member is directed to the “Discussion” page.
7. After clicking the “Yes” button, the website shows a message “Your comment has been added successfully.”
8. The comment is added to the discussion part of the event page, and can be accessed by the other members.

Exceptions:

4.a. In step 4 of the normal flow, if the member clicks on "Submit" and leaves the comment field empty:

4.a.1. The website shows a message “Please fill in the required field.”

4.a.2. Going back to step 3 to fill in all the required fields.

Assumptions:

The user knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.12 Like or Dislike a Comment

Actors: member.

Description: the member can like or dislike any comment that is already created by him/her or other members.

Preconditions: the member is signed in to the system, and a specific event page is opened.

Postconditions: the “Like/dislike” icon has been changed successfully.

Normal flow:

1. The member accesses a list of the comments that have been added by the other members or by him/ her in the discussion part of a specific event page.
2. The member clicks on the “Like/dislike” icon which is displayed below each added comment.

3. The system changes the icon to represent that the member likes this comment, and it increments the number of likes which is directly beside the icon.
4. The system saves the changes and become visible by the other members.

Alternative Flow:

2. The member clicks on the “Like/dislike” icon which is displayed below each added comment.
3. The system changes the icon to represent that the member does not like this comment any more, and it decrements the number of likes which is directly beside the icon.
4. The system saves the changes and become visible by the other members.

Assumptions:

The user knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.13 Delete a Comment

Actors: member.

Description: the member can delete any comment that is already created by him/her.

Preconditions: the member is signed in to the system, and a specific event page is opened.

Postconditions: the comment has been deleted successfully

Normal Flow:

1. The member accesses a list of the comments that have been added by the other members or by him/ her in the discussion part of a specific event page.
2. The member clicks on the “Delete” icon which is displayed beside only the comments he/she added.
3. The system shows a message “Are you sure you want to delete this comment?”
4. The member clicks on:
    - "Ok" to confirm deleting the comment.
    - Or "Cancel", then the member is directed to the “Discussion” page.
5. After clicking the “Ok” button, the website shows a message “Your comment has been deleted successfully.”
6. The comment is removed from the discussion part of the event page.

Assumptions:

The user knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.14 Sign out

Actors: admin, member.

Description: the user signs out and exit the system.

Preconditions: the user is signed in to the system.

Postconditions: the user is signed out successfully.

Normal Flow:

1. The user clicks on the “Sign out” button.
2. The system signs out the user, and redirects him/her to the website homepage.

Assumptions:

The user knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.15 Manage Events/ View Events

Actors: admin.

Description: the admin views the events that are created previously by the members by displaying a list of events. Also, he can access any event page.

Preconditions: the admin is signed in to the system, and the admin homepage is opened.

Postconditions: the events’ list is accessed successfully, and the admin opens any event page.

Normal Flow:

1. The admin clicks on the “Mange events” link on the admin homepage.
2. The system opens the “Mange events” page, and a list of events is displayed.
3. The admin can click on the “View” button to view any event.
4. The website opens the event page which includes event details such as: the event topic, description, location, date, time, event organizer, participating members, and more details.

Assumptions:

The user knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.16 Manage Events/ Delete Event

Actors: admin.

Description: the admin is able to delete any event that is created previously by a member if required.

Preconditions: the admin is signed in to the system, and the admin homepage is opened.

Postconditions: the event is deleted successfully, and an email is sent to the event creator and all participating guests informing that the event has been canceled.

Normal Flow:

1. The admin clicks on the “Mange events” link on the admin homepage.
2. The system opens the “Mange events” page, and a list of events is displayed.
3. The admin clicks on the “Delete” button which appears beside the event he/she wants to delete.
4. The system shows a message “Are you sure you want to delete this event?”
5. The admin clicks on:
    - “Ok” to delete the event.

    - Or “Cancel”, then the admin is directed to the “Mange events” page.

6. After clicking on the “Ok” button, the website shows a message “Event has been deleted successfully.”
7. The event is deleted, and an email is sent to the event creator and all participating guests informing that the event has been canceled.

Assumptions:

The user knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.17 Manage Events/ Approve or Deny Event

Actors: admin.

Description: the admin should review the events that are created previously by the members in order to approve or deny them.

Preconditions: the admin is signed in to the system, and the admin homepage is opened.

Postconditions: The event is approved or denied successfully, and an email is sent to the event creator informing that the event has been approved or denied.

Normal Flow:

1. The admin clicks on the “Mange events” link on the admin homepage.
2. The system opens the “Mange events” page, and a list of events is displayed.
3. The admin accesses the “Events requests” part.
4. The admin clicks on the “View” button to review the event.

5. If the event follows the rules, then the admin clicks on the “Approve” button.
6. The system shows a message “Are you sure you want to approve this event?”
7. The admin clicks on:
   - “Ok” to approve the event.
   - Or “Cancel”, then the admin is directed to the “Mange events” page.
8. After clicking the “Yes” button, the website shows a message “Event has been approved successfully.”
9. The event is added and published to all members. An email is sent to the event creator informing that the event has been approved.

Alternative Flow:

5. If the event does not follow the rules, then the admin clicks on the “Deny” button.
6. The system shows a message “Are you sure you want to deny this event?”
7. The admin clicks on:
   - “Yes” to deny the event.
   - Or “Cancel”, then the admin is directed to the Mange events” page.
8. After clicking the “Yes” button, the website shows a message “Event has been denied successfully.”

9. The event is deleted and an email is sent to the event creator informing that the event has been denied.

Assumptions:

The user knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.18 Manage Accounts/ View/ Search Accounts

Actors: admin.

Description: the admin can view all accounts that are registered in the system, and search for a specific account by email or name.

Preconditions: the admin is signed in to the system, and the admin homepage is opened.

Postconditions: The account is viewed successfully.

Normal Flow:

1. The admin clicks on the “Mange accounts” link on the admin homepage.
2. The system opens the “Mange accounts” page. The page contains a list of accounts which includes: member name and email. In addition, there is a text field to enter a search keyword (email or name).
3. The admin is able to click on any account to open its page.
4. The website opens the account page which includes: the member name, photo, description, number of kids, and their ages.
5. The admin can click on the “Back” button to go back to the “Mange accounts” page.

Assumptions:

The user knows the basic skills to use a computer, and has the ability to understand the English language.

3.3.19 Manage Accounts/ Delete Account

Actors: admin.

Description: the admin is able to delete any account that is registered previously by a member if required.

Preconditions: the admin is signed in to the system, and the admin homepage is opened.

Postconditions: the account is deleted successfully.

Normal Flow:

1. The admin clicks on the “Mange accounts” link on the admin homepage.
6. The system opens the “Mange accounts” page. The page contains a list of accounts which includes: member name and email. In addition, there is a text field to enter a search keyword (email or name).
2. The admin clicks on the “Delete” button which appears below the account he/she wants to delete. If it is hard to find any account, the admin is able to use the search function to find it.
3. The system shows a message “Are you sure you want to delete this account?”
4. The administrator clicks on:
    - “Ok” to delete the account.

    - Or “Cancel”, then the admin is directed to the “Mange accounts” page.

5. After clicking the “Yes” button, the system deletes the account, and shows a message “Account has been deleted successfully.”

Assumptions:

The user knows the basic skills to use a computer, and has the ability to understand the English language.

CHAPTER FOUR

SYSTEM DESIGN

4.1 Introduction

4.1.1 Purpose

This chapter explains in detail the architecture of the Kids Club website. It involves a description of the system elements and the relationships between them. Although there are different views to describe system architecture and design, this chapter focuses on the logical and interaction views. This chapter is developed based on the system requirements that are discussed in the previous chapters.

4.1.2 Definitions, Acronyms or Abbreviations

Table 4. Acronyms and Abbreviations

Term	Definition, acronyms, and abbreviations
UML	Unified Modeling Language
Class diagram	A type of UML diagrams that shows the system structure.

Sequence diagram	A type of UML diagrams that shows interactions between the system elements organized based on a specific sequence.
Admin	The administrator of the website who has the authorization to manage the website.
Member	The user who registers in the website, and can use website functions.

4.1.3 Overview

In this chapter, the UML diagrams are used to illustrate how the system is composed and how its elements interact with each other. The first part of this chapter is the class diagram which is concerned with the logical view of the system. Class diagram is one type of the UML diagrams that shows the system structure. Furthermore, the classes of the system, attributes, relations, and operations are explained in this chapter. The second part includes different sequence diagrams that assist to understand the interaction between system elements, and show the exact scenarios that happen between them. These scenarios are described in a specific order [10].

4.2 Class Diagram

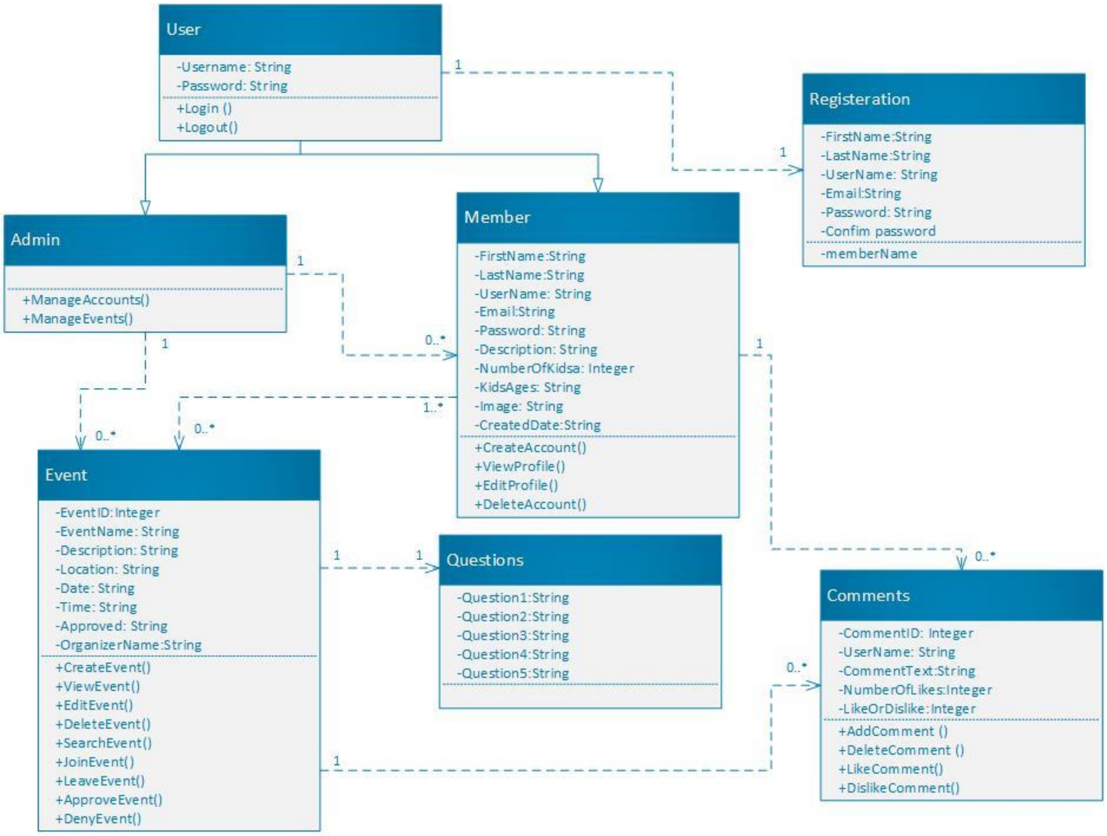


Figure 4. Class Diagram

4.3 Sequence Diagrams

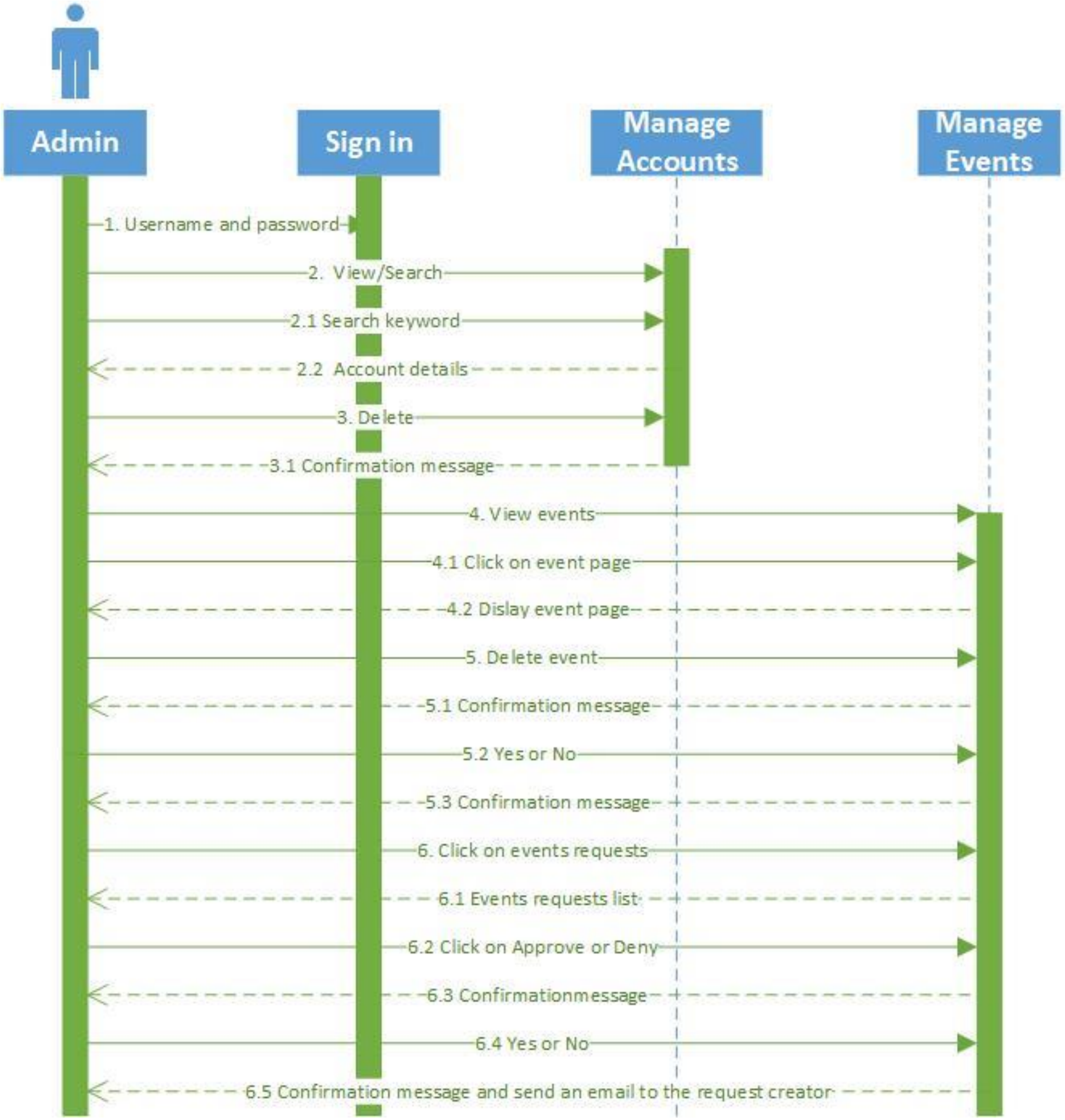


Figure 5. Admin Sequence Diagram

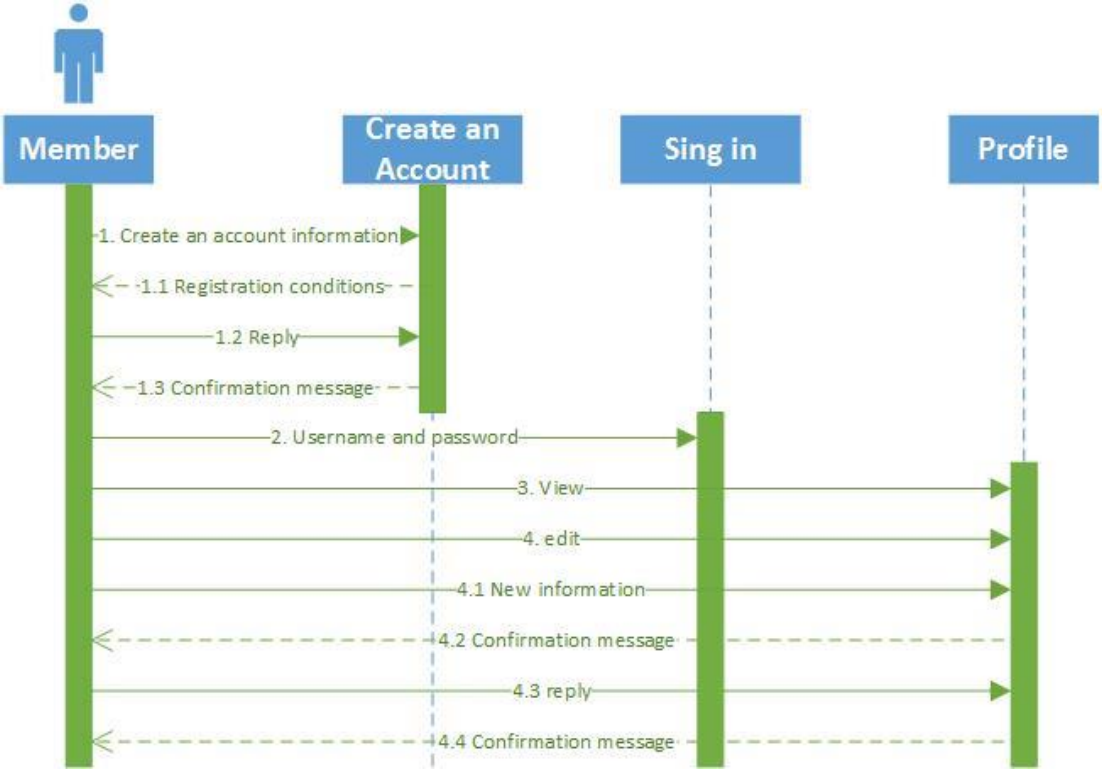


Figure 6. Member Sequence Diagram (1)

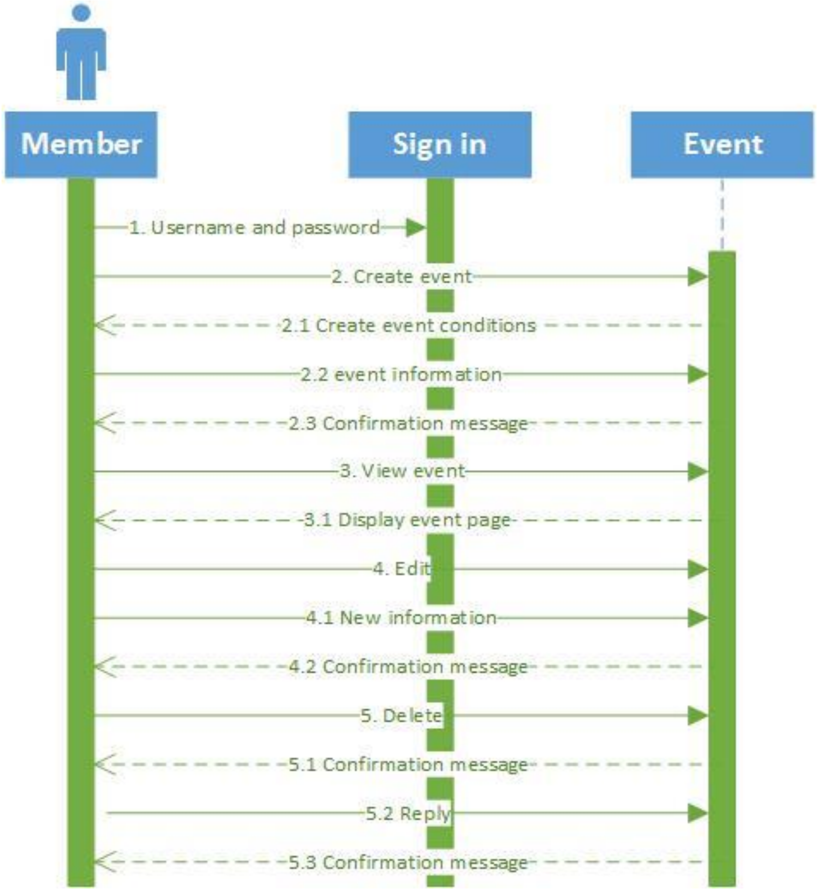


Figure 7. Member Sequence Diagram (2)

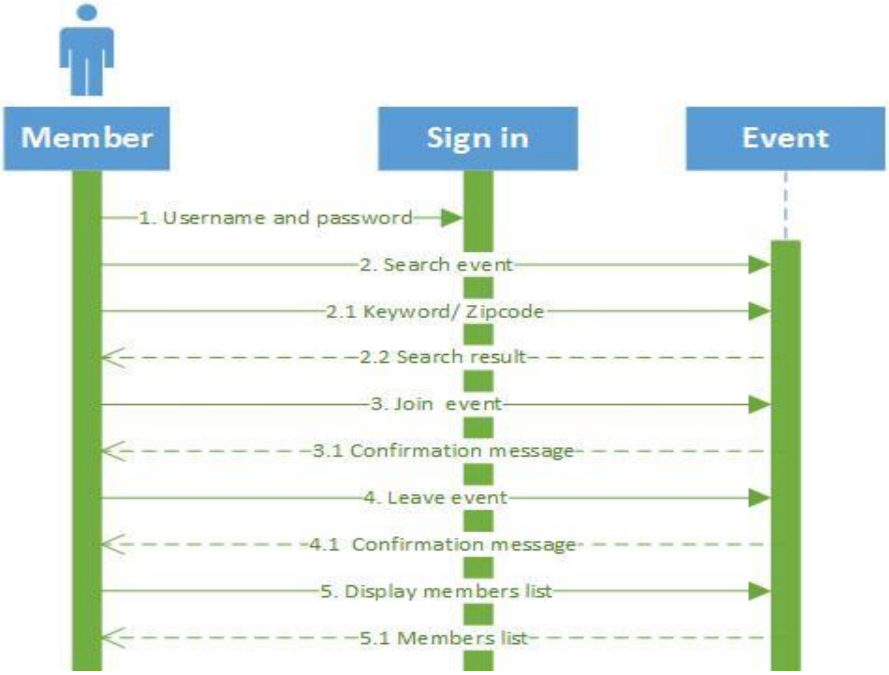


Figure 8. Member Sequence Diagram (3)

CHAPTER FIVE

SYSTEM IMPLEMENTATION

5.1 Introduction

This chapter discusses the system implementation details. It is the phase of converting the system to an executable form and can be used by the user. It includes a description of the technical environment and tools that are used to develop the website.

5.2 Development Platform Tools

1. Planning and timeline software: Gantt project.
2. Design and UML Diagrams: Dia 0.97.2, Visio Professional 2016.
3. Technology : ASP.NET, MVC, ADO.Net / Entity Framework.
4. Languages: C#, T-SQL, HTML, CSS, Angular-JS.
5. Server: Windows 2012 R2.
6. Web Server: IIS 7.5.
7. Implementation Software: Visual Studio 2012, SQL Server Management Studio 2012.
8. Development environment:
    - Operating system: Windows 10.
    - Processor: 2.3 GHZ Intel Core i5-4200U.
    - Memory: 8 GB.

5.3 Component Diagram

There are different views to describe system architecture. The development view, which is the programmer's view, focuses on the system components and describes how they are related to each other. In this case, a component diagram, which is a type of UML diagrams, is suitable to show the system components and their communications. In the following diagram, there are three main components included in the system. The first one is the user which is represented by the browser. The user can be an admin or a member, and both of them have their own functions. Generally, the user starts a communication over the Internet using HTTP in order to get a service or a content from the second component which is the web server. The web server is responsible to process incoming requests and provides users with desirable services such as HTML pages that allow the users to use website functions. On the other hand, the web server may receive content from the user such as forms or files. The third component is the SQL Server, and it is the part of the system that handles the data. It is responsible to access, retrieve, and update the data based on the queries received from other components (see Figure 9).

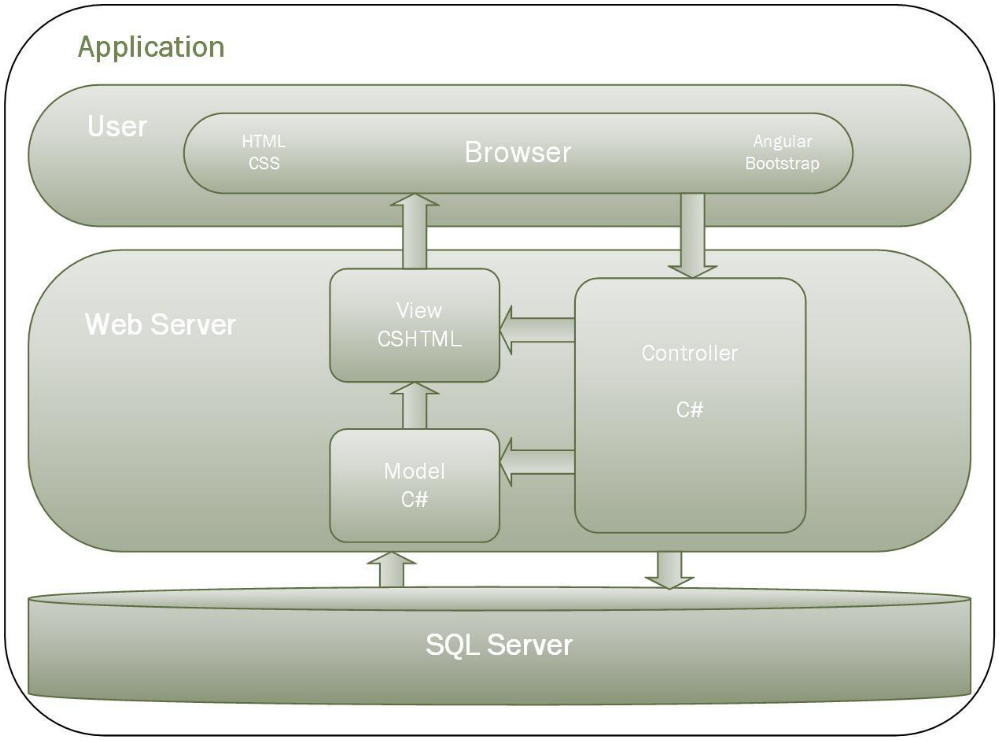


Figure 9. Component Diagram

CHAPTER SIX

SYSTEM TESTING

6.1 Introduction

6.1.1 Purpose

This chapter aims to demonstrate that the Kids Club website meets its intended requirements, and to find out any problems or weaknesses to make it ready to use. For each function, there is a test to make sure that the function does what it is supposed to do by listing different cases and showing the result of each case. Furthermore, this chapter includes defect testing that discovers any software bugs, incorrect results, or undesirable behavior. These investigations assist to fix any issue that may face the user before the website is put into use.

6.1.2 Format of Chapter

We follow the IEEE standard for software test documentation [11].

6.1.3 Overview

This chapter involves multiple steps. First and foremost, test cases are designed corresponding to all website functions. After that, the test data is prepared to run the system with this data. At this point, a comparison between the result we get and the test cases written previously is conducted to test the website functionality.

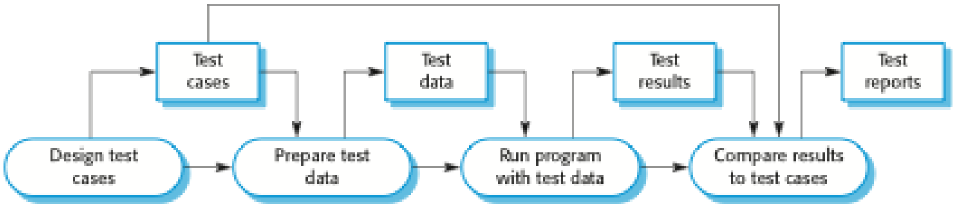


Figure 10. Testing Method

6.2 Validation Test

6.2.1 Create an Account Test Case

Test case (1): submit the form without filling one or all the required fields.

Preconditions: the “Create an account” page is opened.

Input:

- First name
- Last name
- Username
- Email
- Password
- Confirm password

Action:

1. Enter the information without filling one or all the required fields.
2. Click the “Sign up” button.

Expected output: show a warning message: “Please fill in the required fields.”

Pass/ Fail: pass.

Test case (2): submit the form with an invalid email address.

Preconditions: the “Create an account” page is opened

Input:

- First name
- Last name
- Username
- Email
- Password
- Confirm password

Action:

1. Enter the information with an invalid email address.
2. Click the “Sign up” button.

Expected output: show a warning message: “Please provide a valid email address.”

Pass/ Fail: pass.

Test case (3): submit the form with an existing username or email.

Preconditions: the “Create an account” page is opened.

Input:

- First name
- Last name
- Username

- Email
- Password
- Confirm password

Action:

1. Enter the information with an existing username or email.
2. Click the “Sign up” button.

Expected output: show a warning message: “The email address or the username already exists.”

Pass/ Fail: pass.

Test case (4): submit the form with different passwords in the password and confirmation fields.

Preconditions: the “Create an account” page is opened.

Input:

- First name
- Last name
- Username
- Email
- Password
- Confirm password

Action:

1. Enter the information with different passwords.
2. Click the “Sign up” button.

Expected output: show a warning message: “Please enter a matched password.”

Pass/ Fail: pass.

Test case (5): submit the form with a password length not between 8 and 12 character.

Preconditions: the “Create an account” page is opened.

Input:

- First name
- Last name
- Username
- Email
- Password
- Confirm password

Action:

1. Enter the information with a password length not between 8 and 12 character.
2. Click the "Sign up" button.

Expected output: show a warning message: “Please enter a password length between 8 and 12 character.”

Pass/ Fail: pass.

Test case (6): submit the form with correct information

Preconditions: the “Create an account” page is opened

Input:

- First name
- Last name
- Username
- Email
- Password
- Confirm password

Action:

1. Enter the information.
2. Click the “Sign up” button.

Expected output: show a message: “You have been registered successfully.”

Pass/ Fail: pass

6.2.2 Sign in Test Case

Test case (1): submit the form without filling one or both required fields.

Preconditions: the “Sign in” page is opened.

Input:

- Username
- Password

Action:

1. Enter the information without filling one or both required fields.
2. Click the “Sign in” button.

Expected output: show a warning message: “All fields are mandatory.”

Pass/ Fail: pass.

Test case (2): submit the form with a wrong username or password.

Preconditions: the “Sign in” page is opened.

Input:

- Username
- Password

Action:

1. Enter the information with wrong username or password.
2. Click the "Sign in" button.

Expected output: Show a warning message: “Please check your Login details.”

Pass/ Fail: pass.

Test case (3): submit the form with correct information.

Preconditions: the “Sign in” page is opened.

Input:

- Username
- Password

Action:

1. Enter the information.
2. Click the “Sign in” button.

Expected output: Member homepage is opened.

Pass/ Fail: pass.

6.2.3 View/ Edit Profile Test Case

Test case (1): view profile.

Preconditions: the member's home page is opened

Input: none.

Action: click the “Profile” link.

Expected output: profile page is opened.

Pass/ Fail: pass.

Test case (2): edit profile without filling all the required information.

Preconditions: the member's home page is opened.

Input: new information such as the first name, last name, description, the number of kids, their ages, or photo.

Action:

1. Enter new information without filling all required information.
2. Click the “Update” button.

Expected output: show a warning message: “Please fill in the required fields.”

Pass/ Fail: pass.

Test case (3): edit profile with an invalid photo.

Preconditions: the member's home page is opened

Input: new information such as the first name, last name, description, the number of kids, their ages, or photo.

Action:

1. Enter new information with an invalid photo.
2. Click the “Update” button.

Expected output: show a warning message: “Please enter a valid photo.”

Pass/ Fail: pass.

Test case (4): edit profile with valid information.

Preconditions: the member's home page is opened.

Input: new information such as the first name, last name, description, the number of kids, their ages, or photo.

Action:

1. Enter a new valid information.
2. Click the “Update” button.

Expected output: show a message: “Changes has been saved successfully.”

Pass/ Fail: pass.

6.2.4 Create Event Test Case

Test case (1): create an event without filling all the required information.

Preconditions: the “Create event” page is opened.

Input:

- Event name
- Location
- Event date
- Event time
- Event description
- Yes/ no answers

Action:

1. Enter event information without filling all the required information.

2. Click the “Create” button.

Expected output: show a waning message: “Please fill in the required fields.”

Pass/ Fail: pass.

Test case (2): create an event with all correct information.

Preconditions: create event page is opened.

Input:

- Event name
- Location
- Event date
- Event time
- Event description
- Yes/ No answers

Action:

1. Enter event information with all correct information.
2. Click the “Create” button.

Expected output: show a message: “The event has been created successfully. Your request details will be reviewed and we will contact you soon.”

Pass/ Fail: pass.

<u>6.2.5 View/ Edit Event Test Case</u>

Test case (1): view event.

Preconditions: the member's home page is opened.

Input: none.

Action:

1. Click on the “My Events” link.
2. Click on a specific event

Expected output: event page is opened.

Pass/ Fail: pass.

Test case (2): edit event without filling all the required information.

Preconditions: event page is opened.

Input: new information such as: the event name, location, event date, event time, event description, or Yes/ no answers.

Action:

1. Enter new information without filling all the required information.
2. Click the “Update” button.

Expected output: show a warning message: “Please fill in all the required fields.”

Pass/ Fail: pass.

Test case (3): edit event with all correct information.

Preconditions: event page is opened.

Input: new information such: the event name, location, event date, event time, event description, or Yes/No answers.

Action:

1. Enter new information in all required fields.
2. Click the “Update” button.

Expected output: show a message: “Changes have been saved successfully.”

Pass/ Fail: pass.

6.2.6 Delete Event Test Case

Test case (1): delete event.

Preconditions: the member's home page is opened.

Input: none.

Action:

1. Click the “My Events” link.
2. Click the “Delete” icon of a specific event.

Expected output: show a message: “Event has been deleted successfully.”

Pass/ Fail: pass.

6.2.7 Search for an Event Test Case

Test case (1): search for an event with an invalid zip code.

Preconditions: the member's home page is opened.

Input:

- Search keyword.
- Zip code.

Action:

1. Enter search input with an invalid zip code.
2. Click the “Search” icon.

Expected output: show a warning message: “Please, enter a valid zip code.”

Pass/ Fail: pass.

Test case (2): search for an event with a valid zip code.

Preconditions: the member's home page is opened.

Input:

- Search keyword.
- Zip code.

Action:

1. Enter search input.
2. Click the “Search” icon.

Expected output: a list of events based on the keyword and the zip code is displayed.

Pass/ Fail: pass.

6.2.8 Join an Event Test Case

Test case (1): join an event.

Preconditions: event page is opened.

Input: none.

Action:

1. Click on the “Join Event” link at the top of the event page.

Expected output: show a message: “You have joined the event successfully.”

Pass/ Fail: pass.

6.2.9 Leave an Event Test Case

Test case (1): leave an event.

Preconditions: event page is opened.

Input: none.

Action:

1. Click on the “Leave Event” link at the top of the event page.

Expected output: show a message: “You have left the event successfully.”

Pass/ Fail: pass.

6.2.10 Display Participating Members List Test Case

Test case (1): display participating members list.

Preconditions: event page is opened.

Input: none.

Action:

1. Click on the “Members” link at the top of the event page.

Expected output: a list of members who joined a specific event is displayed.

Pass/ Fail: pass.

6.2.11 Add a Comment Test Case

Test case (1): add a comment without filling the comment text field.

Preconditions: event page is opened.

Input: comment text.

Action:

1. Click on the “Discussion” link at the top of the event page.
2. Enter nothing in the comment text.
3. Click on “Submit.”

Expected output: show a warning message: “Please fill in the required filed.”

Pass/ Fail: pass.

Test case (2): add a comment.

Preconditions: event page is opened.

Input: comment text.

Action:

1. Click on the “Discussion” link at the top of the event page.
2. Enter a comment text.
3. Click on “Submit.”

Expected output: show a message: “Your comment has been added successfully.”

Pass/ Fail: pass.

6.2.12 Delete a Comment Test Case

Test case (1): delete a comment.

Preconditions: discussion part of event page is opened.

Input: none.

Action:

1. Click on the “Delete” icon beside a specific comment.

Expected output: show a message: “Your comment has been deleted successfully.”

Pass/ Fail: pass.

6.2.13 Like/ Dislike Comment Test Case

Test case (1): like or dislike a comment.

Preconditions: discussion part of event page is opened.

Input: none.

Action:

1. Click on “Like/dislike” icon below a specific comment.

Expected output: the icon status is changed, and the number of likes is incremented or decremented based on the action.

Pass/ Fail: pass.

6.2.14 Sign out Test Case

Test case (1): sign out.

Preconditions: any page is opened.

Input: none.

Action:

1. Click on “Sign out.”

Expected output: the user is signed out.

Pass/ Fail: pass.

6.2.15 Manage Events/ View Events Test Case

Test case (1): manage events/ view events.

Preconditions: admin home page is opened.

Input: none.

Action:

1. Click the “Manage events” link.

Expected output: a list of events is displayed.

Pass/ Fail: pass.

6.2.16 Manage Events/ Delete an Event Test Case

Test case (1): manage events/ delete an event.

Preconditions: admin home page is opened.

Input: none.

Action:

1. Click the “Manage events” link.
2. Click the “Delete” button beside a specific event.

Expected output: show a message: “Event has been deleted successfully.”

Pass/ Fail: pass.

6.2.17 Manage Events/ Approve or Deny an Event Test Case

Test case (1): manage events/ approve or deny an event.

Preconditions: admin home page is opened.

Input: none.

Action:

1. Click the “Manage events” link.
2. Go to events requests part.
3. Click the “Approve” or “Deny” button beside a specific event.

Expected output: show a message: “Event has been approved successfully.” or “Event has been denied successfully.”

Pass/ Fail: pass.

6.2.18 Manage Accounts/ View or Search for an Account Test Case

Test case (1): manage accounts/ view or search for an account.

Preconditions: admin home page is opened.

Input: none.

Action:

1. Click the “Manage accounts” link.
2. An optional keyword may be entered to search for a specific account.

Expected output: a list of accounts is displayed.

Pass/ Fail: pass.

6.2.19 Manage Accounts/ Delete an Account Test Case

Test case (1): manage accounts/ delete an account.

Preconditions: admin home page is opened.

Input: none.

Action:

1. Click the “Manage accounts” link.
2. Click the “Delete” button beside a specific account.

Expected output: show a message: “Account has been deleted successfully.”

Pass/ Fail: pass.

CHAPTER SEVEN

SYSTEM USER MANUAL

- Create an Account

In order to use the website functions, the user should create an account by filling out the registration form.

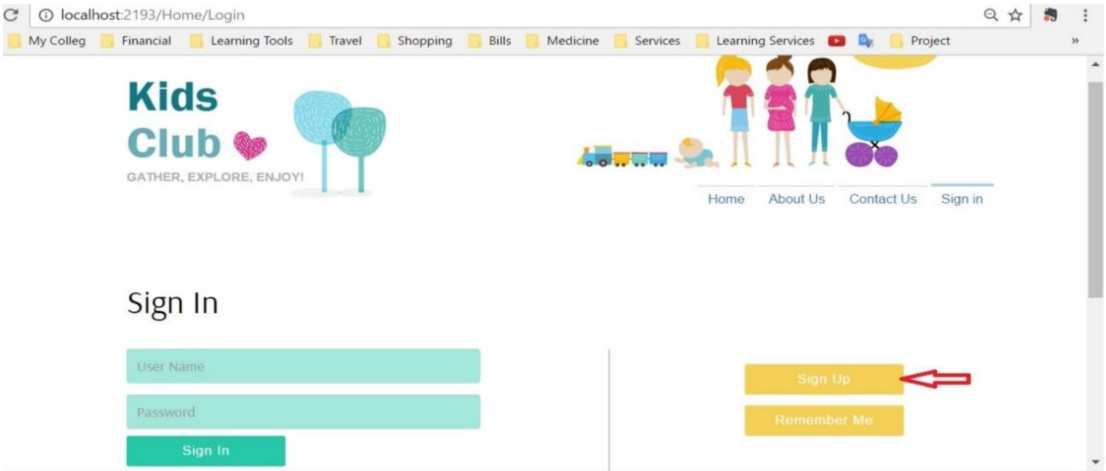


Figure 11. Create an Account (1)

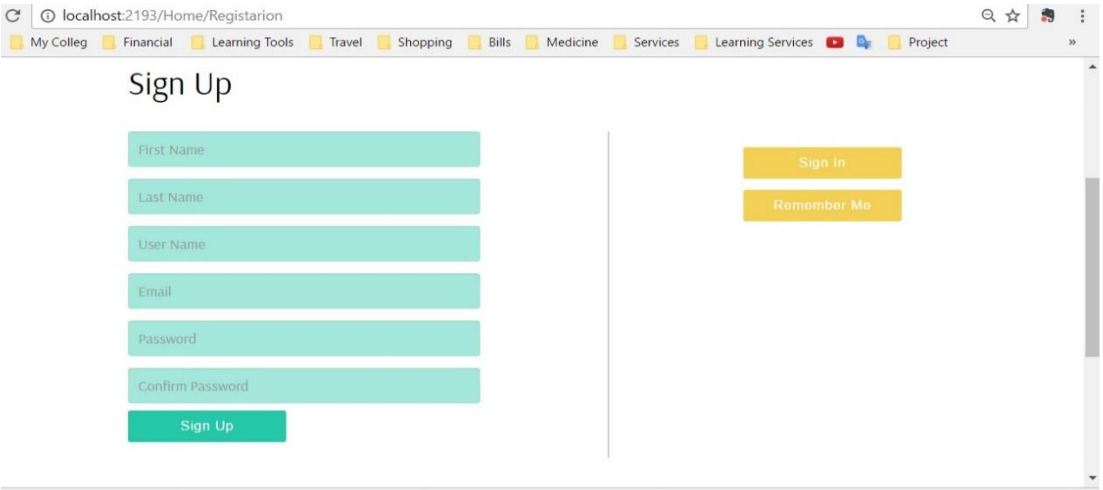


Figure 12. Create an Account (2)

- Member Log in

The member should log in to start using the website functions by entering his/her username and password.

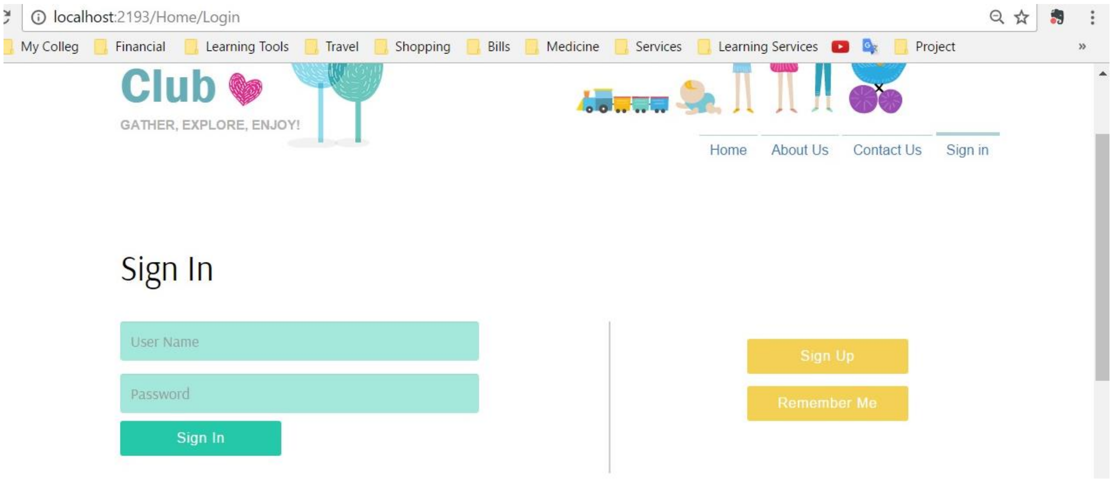


Figure 13. Member Log in

- Admin Log in

The admin should log in to start using the website functions by entering his/her username and password.

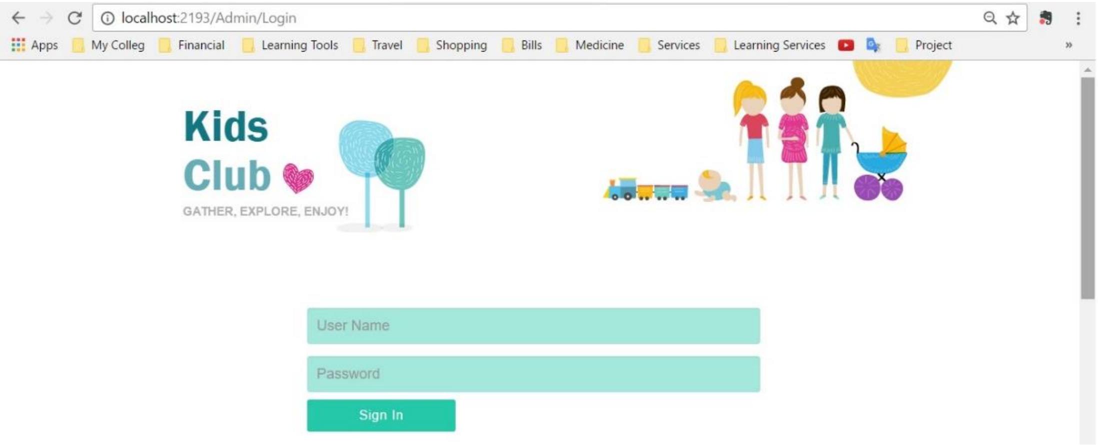
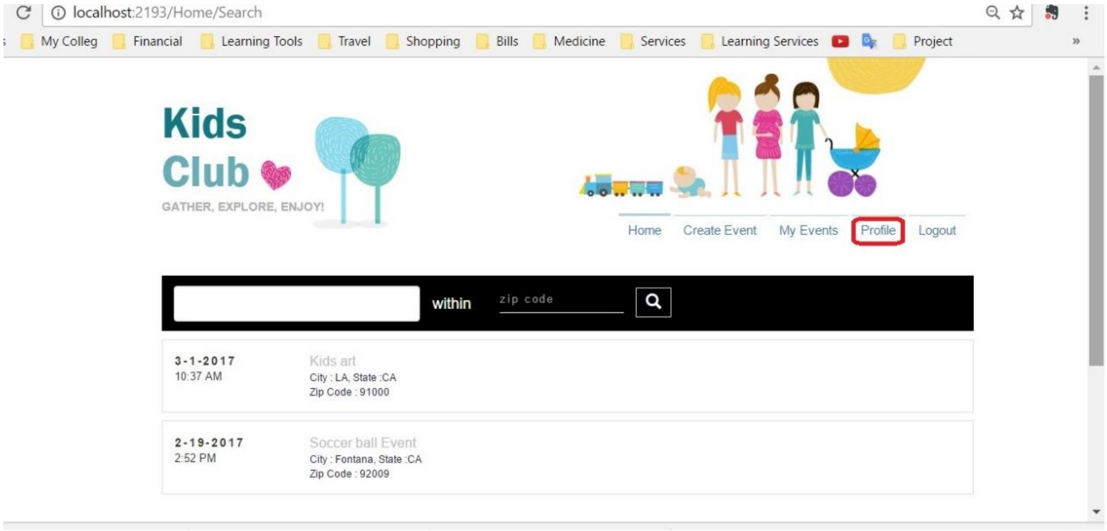


Figure 14. Admin Log in

- View/ Edit Profile

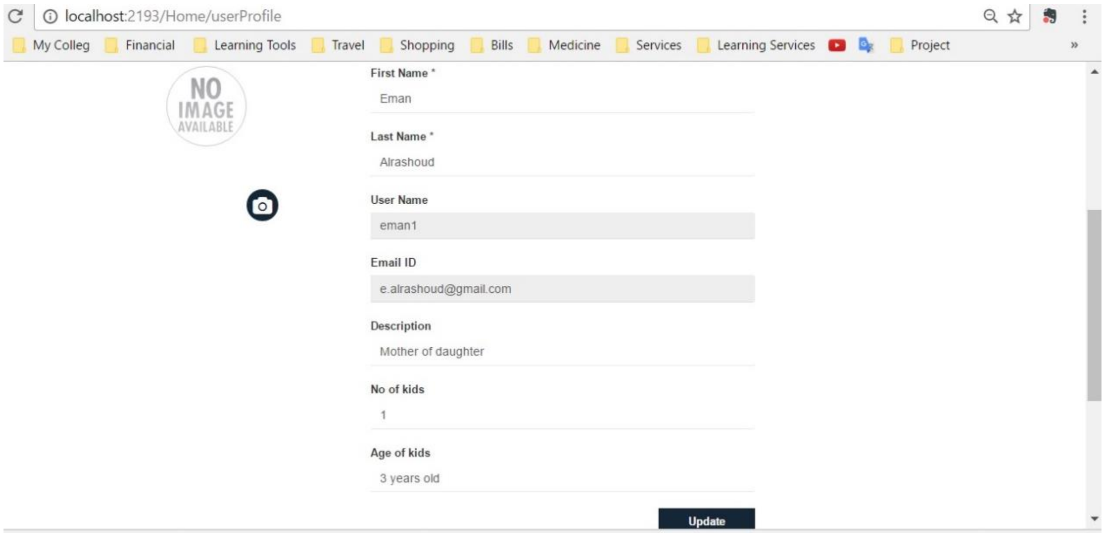
The member can view or edit his/her profile.



3-1-2017
10:37 AM
Kids art
City : LA, State :CA
Zip Code : 91000

2-19-2017
2:52 PM
Soccer ball Event
City : Fontana, State :CA
Zip Code : 92009

Figure 15. View Profile



First Name *
Eman

Last Name *
Alrashoud

User Name
eman1

Email ID
e.alrashoud@gmail.com

Description
Mother of daughter

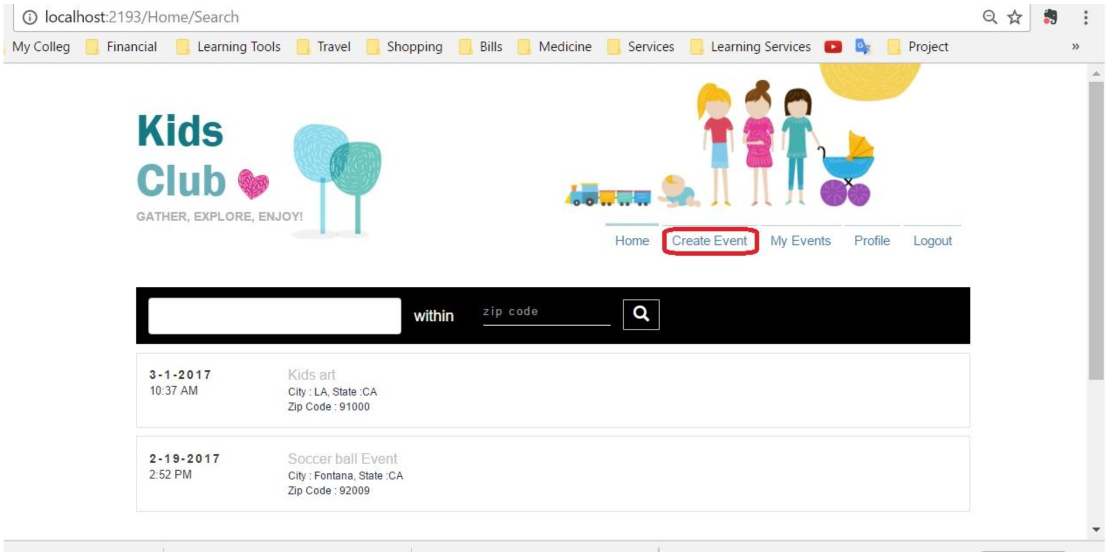
No of kids
1

Age of kids
3 years old

Figure 16. Edit Profile

- Create an Event

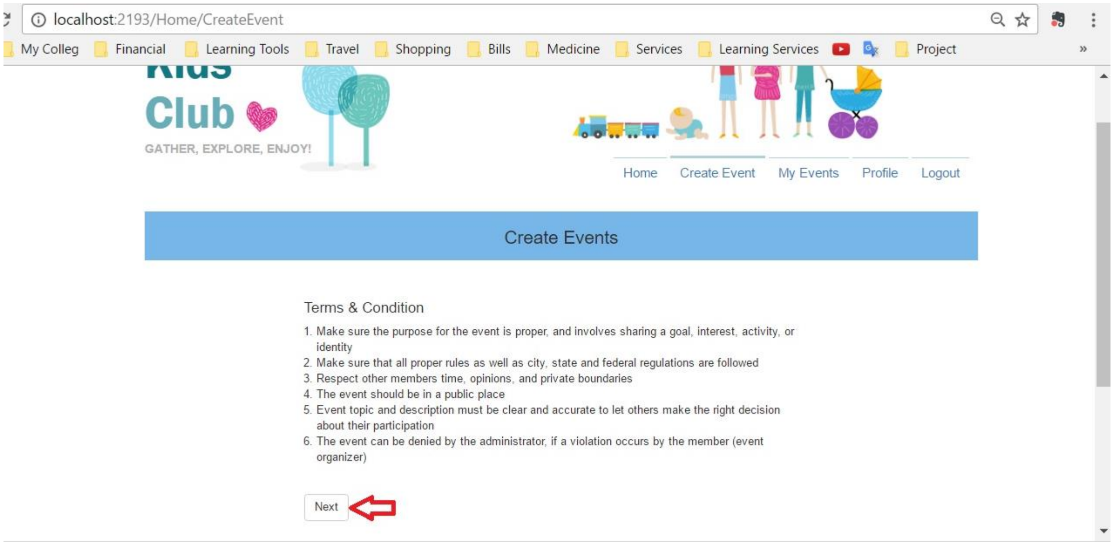
The member can create a new event as an event organizer by filling out the “Create an event” form.



3-1-2017
10:37 AM
Kids art
City : LA, State :CA
Zip Code : 91000

2-19-2017
2:52 PM
Soccer ball Event
City : Fontana, State :CA
Zip Code : 92009

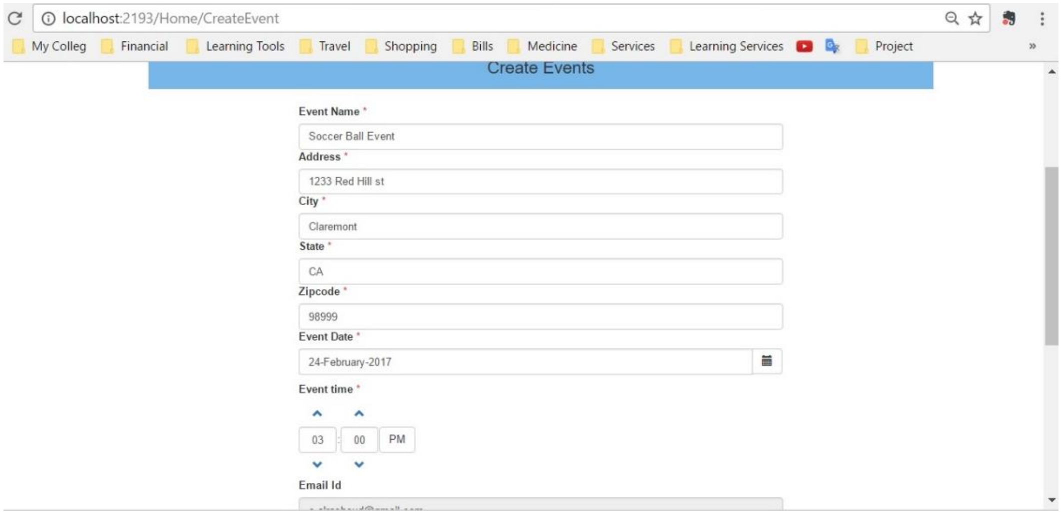
Figure 17. Create an Event (1)



Terms & Condition

1. Make sure the purpose for the event is proper, and involves sharing a goal, interest, activity, or identity
2. Make sure that all proper rules as well as city, state and federal regulations are followed
3. Respect other members time, opinions, and private boundaries
4. The event should be in a public place
5. Event topic and description must be clear and accurate to let others make the right decision about their participation
6. The event can be denied by the administrator, if a violation occurs by the member (event organizer)

Figure 18. Create an Event (2)



Create Events

Event Name *
Soccer Ball Event

Address *
1233 Red Hill st

City *
Claremont

State *
CA

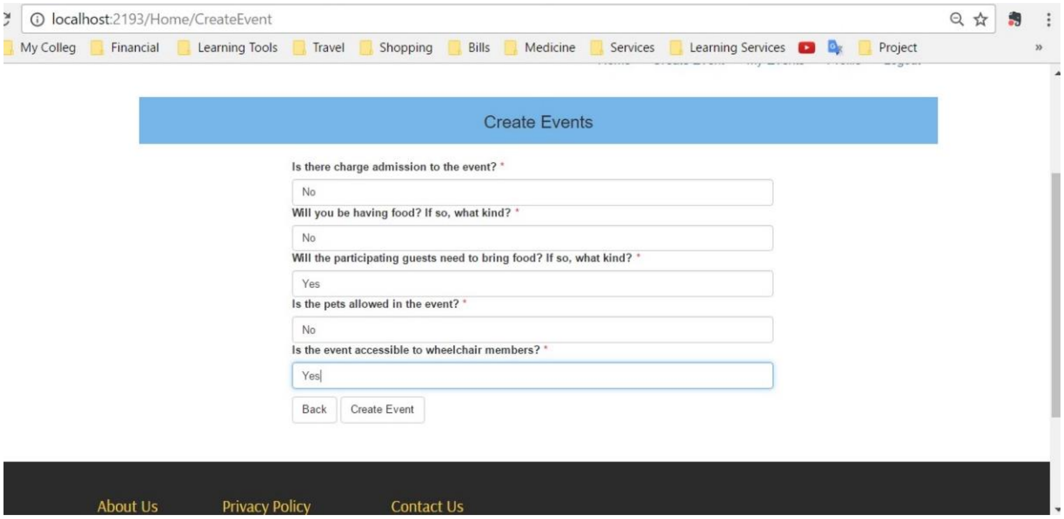
Zipcode *
98999

Event Date *
24-February-2017

Event time *
03 : 00 PM

Email Id

Figure 19. Create an Event (3)



Create Events

Is there charge admission to the event? *
No

Will you be having food? If so, what kind? *
No

Will the participating guests need to bring food? If so, what kind? *
Yes

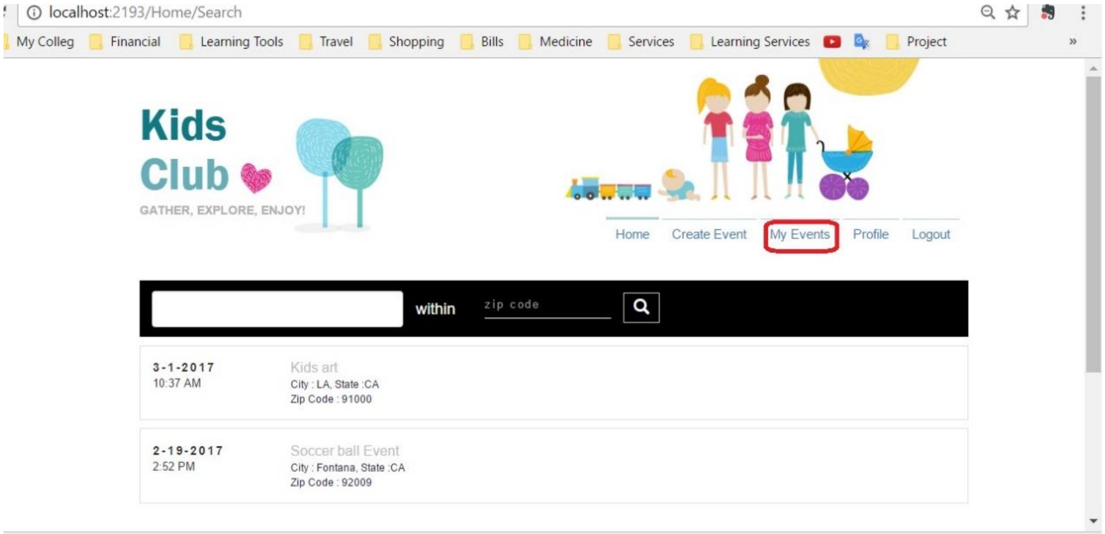
Is the pets allowed in the event? *
No

Is the event accessible to wheelchair members? *
Yes

Figure 20. Create an Event (4)

- View/ Edit Event

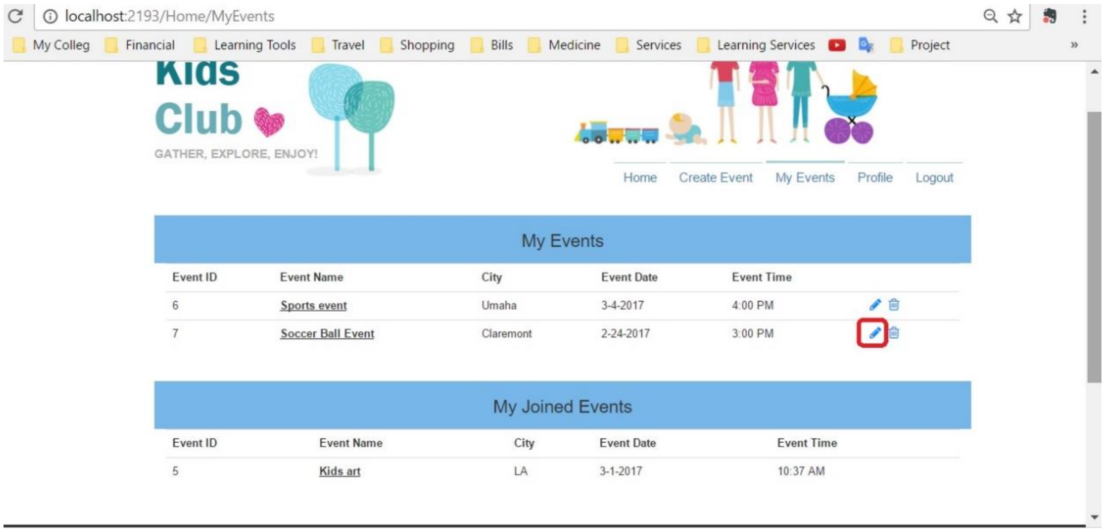
The member can view or edit the event that is created previously by him/her.



3-1-2017
10:37 AM
Kids art
City : LA, State :CA
Zip Code : 91000

2-19-2017
2:52 PM
Soccer ball Event
City : Fontana, State :CA
Zip Code : 92009

Figure 21. View an Event



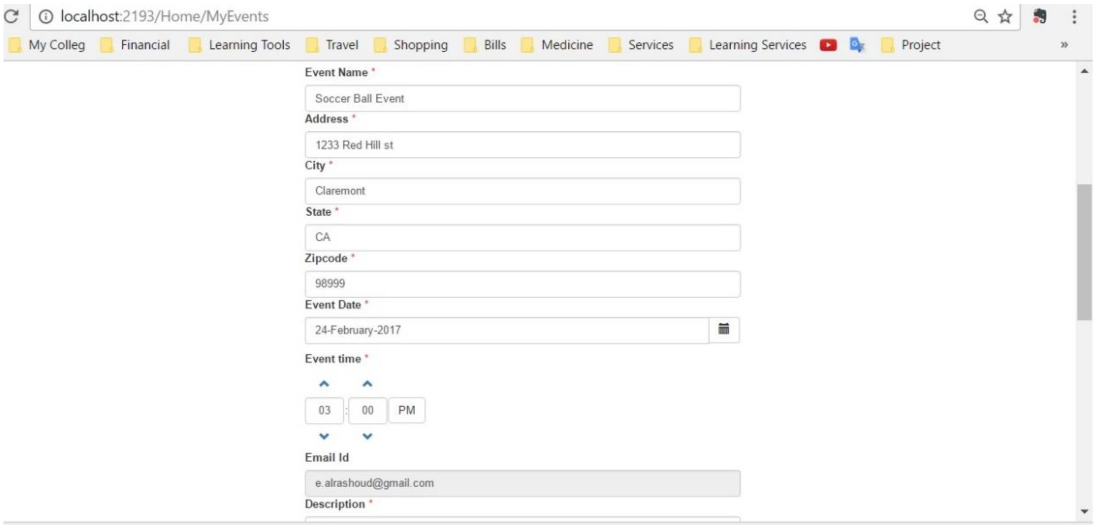
My Events

Event ID	Event Name	City	Event Date	Event Time
6	Sports event	Umaha	3-4-2017	4:00 PM
7	Soccer Ball Event	Claremont	2-24-2017	3:00 PM

My Joined Events

Event ID	Event Name	City	Event Date	Event Time
5	Kids art	LA	3-1-2017	10:37 AM

Figure 22. Edit an Event (1)

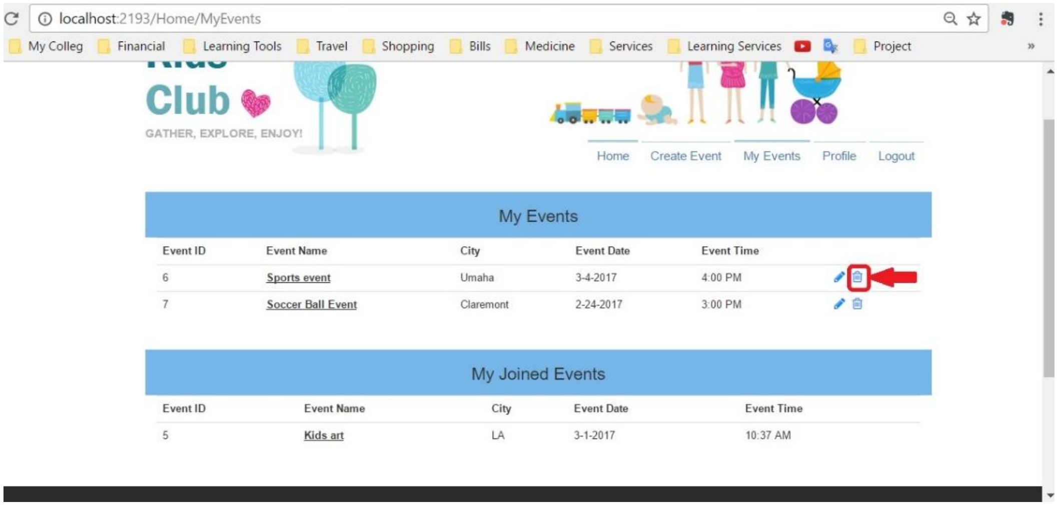


Event Name *
Soccer Ball Event
Address *
1233 Red Hill st
City *
Claremont
State *
CA
Zipcode *
98999
Event Date *
24-February-2017
Event time *
03 : 00 PM
Email Id
e.alrashoud@gmail.com
Description *

Figure 23. Edit an Event (2)

- Delete an Event

The member can cancel any event that is created previously by him/her.



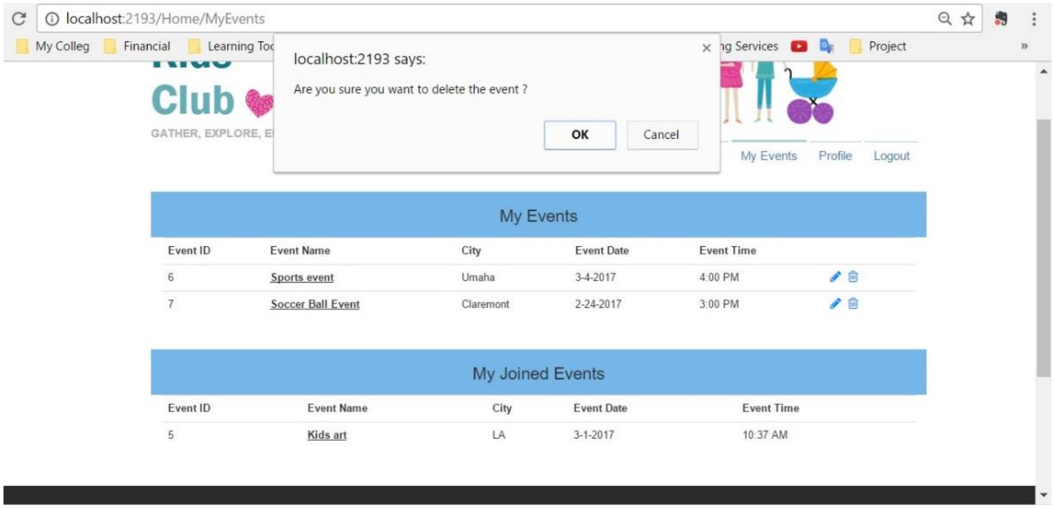
My Events

Event ID	Event Name	City	Event Date	Event Time
6	Sports event	Umaha	3-4-2017	4:00 PM
7	Soccer Ball Event	Claremont	2-24-2017	3:00 PM

My Joined Events

Event ID	Event Name	City	Event Date	Event Time
5	Kids art	LA	3-1-2017	10:37 AM

Figure 24. Delete an Event (1)



localhost:2193 says:

Are you sure you want to delete the event ?

OK Cancel

My Events

Event ID	Event Name	City	Event Date	Event Time
6	Sports event	Umaha	3-4-2017	4:00 PM
7	Soccer Ball Event	Claremont	2-24-2017	3:00 PM

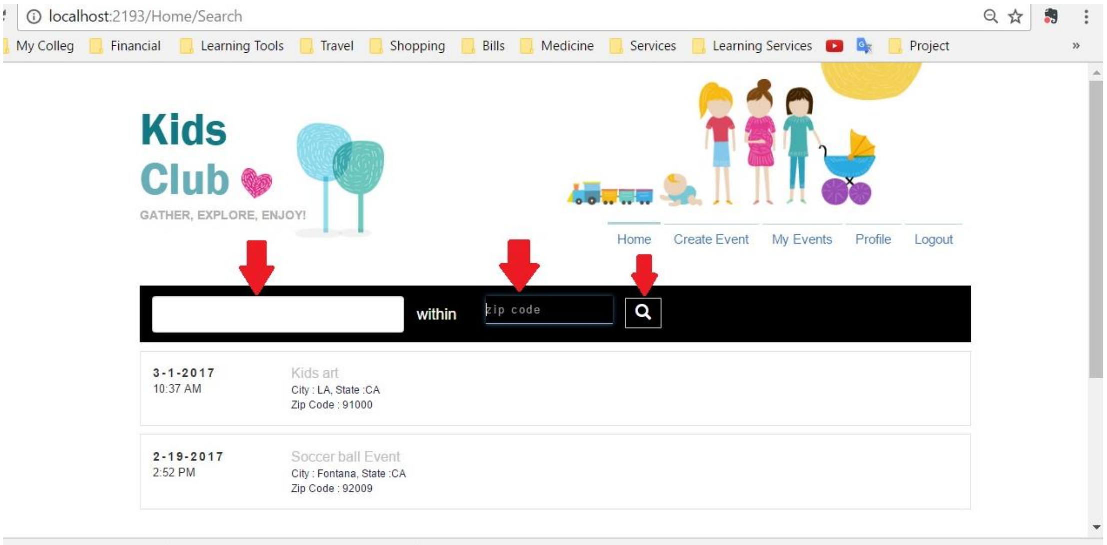
My Joined Events

Event ID	Event Name	City	Event Date	Event Time
5	Kids art	LA	3-1-2017	10:37 AM

Figure 25. Delete an Event (2)

- Search for an Event

The member can search for an event that is created previously, and display a list of events based on zip code.



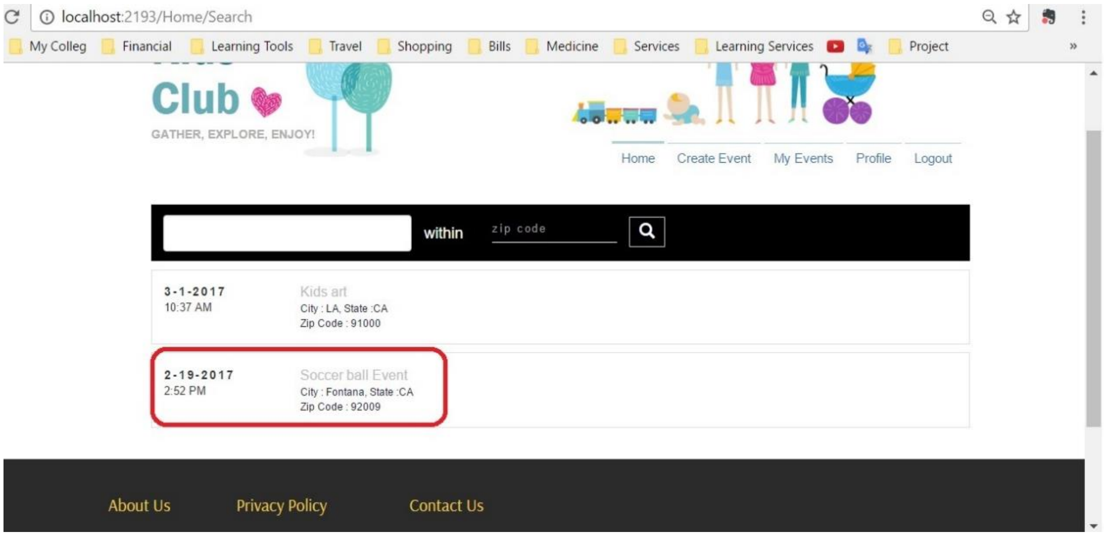
3-1-2017
10:37 AM
Kids art
City : LA, State :CA
Zip Code : 91000

2-19-2017
2:52 PM
Soccer ball Event
City : Fontana, State :CA
Zip Code : 92009

Figure 26. Search for an Even

- Join an Event

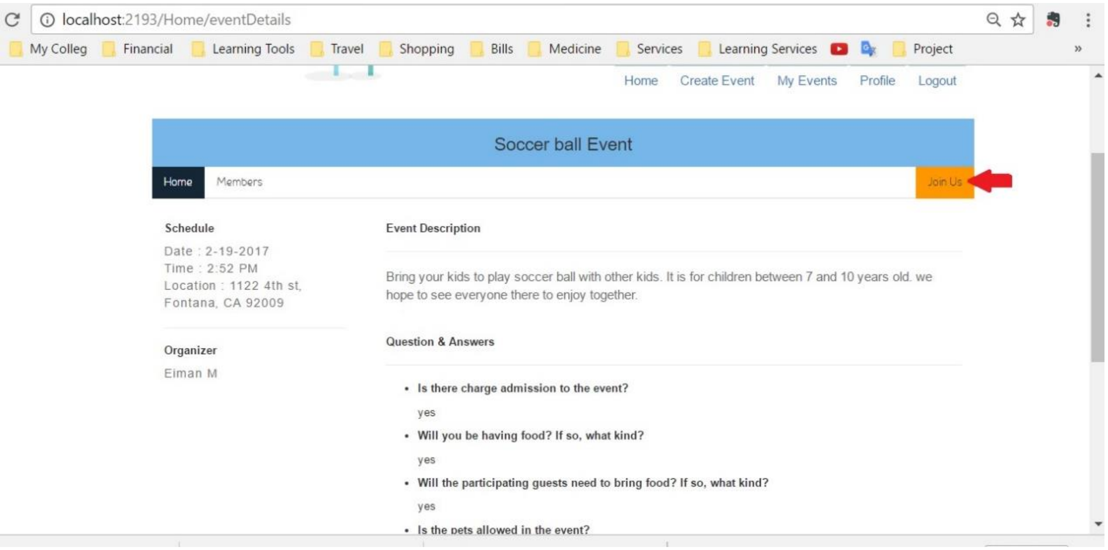
The member can join an event that is created previously by other members.



3-1-2017
10:37 AM
Kids art
City : LA, State :CA
Zip Code : 91000

2-19-2017
2:52 PM
Soccer ball Event
City : Fontana, State :CA
Zip Code : 92009

Figure 27. Join an Event (1)



Soccer ball Event

Schedule

Date : 2-19-2017
Time : 2:52 PM
Location : 1122 4th st, Fontana, CA 92009

Organizer

Eiman M

Event Description

Bring your kids to play soccer ball with other kids. It is for children between 7 and 10 years old. we hope to see everyone there to enjoy together.

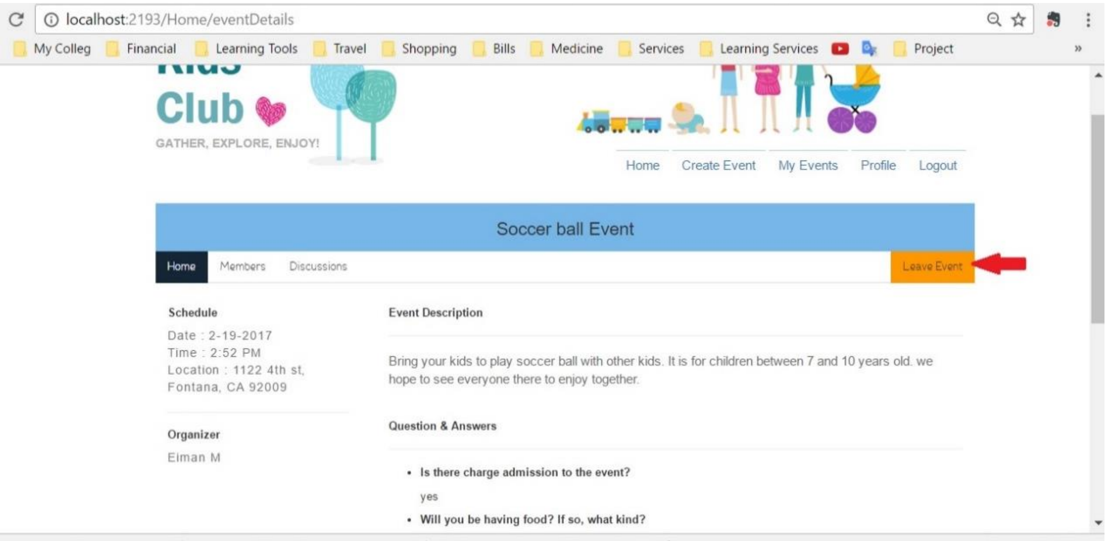
Question & Answers

- Is there charge admission to the event?
  yes
- Will you be having food? If so, what kind?
  yes
- Will the participating guests need to bring food? If so, what kind?
  yes
- Is the pets allowed in the event?

Figure 28. Join an Event (2)

- Leave an Event

The member can leave an event that he/she joined previously.



Soccer ball Event

Schedule

Date : 2-19-2017
Time : 2:52 PM
Location : 1122 4th st, Fontana, CA 92009

Organizer

Eiman M

Event Description

Bring your kids to play soccer ball with other kids. It is for children between 7 and 10 years old. we hope to see everyone there to enjoy together.

Question & Answers

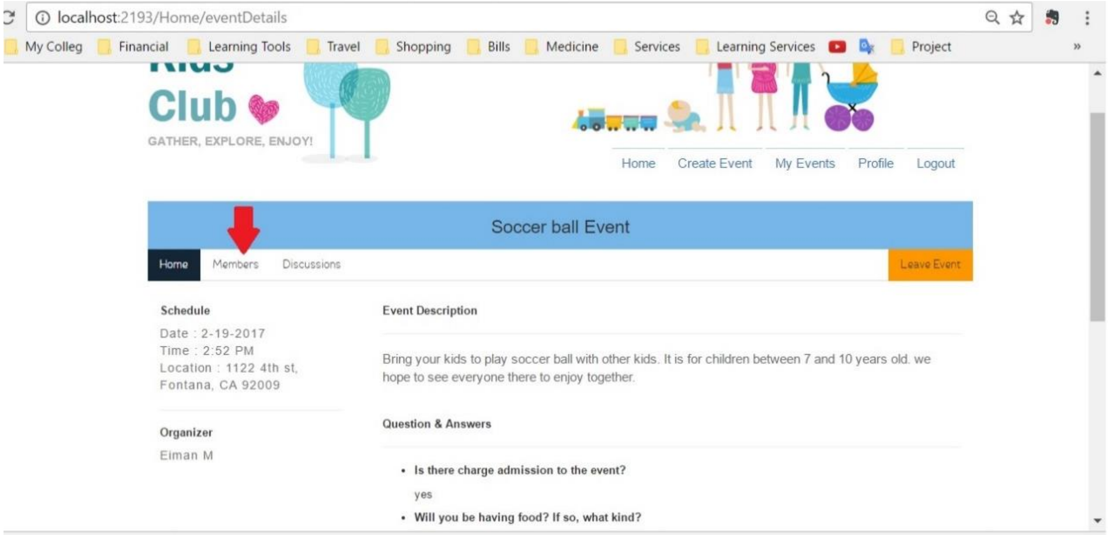
- **Is there charge admission to the event?**

  yes
- **Will you be having food? If so, what kind?**

Figure 29. Leave an Event

- Display Participating Members List

The member can display the participating members list of any event to see the other members who are coming to the event and access their profiles.



Schedule

Date : 2-19-2017
Time : 2:52 PM
Location : 1122 4th st, Fontana, CA 92009

Organizer

Eiman M

Event Description

Bring your kids to play soccer ball with other kids. It is for children between 7 and 10 years old. we hope to see everyone there to enjoy together.

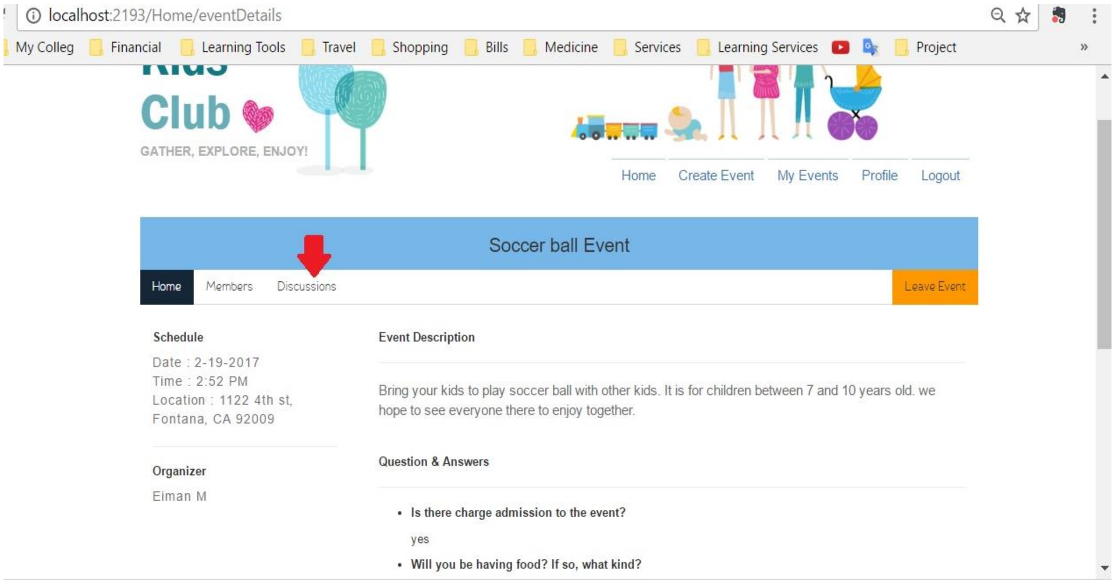
Question & Answers

- **Is there charge admission to the event?**
  yes
- **Will you be having food? If so, what kind?**

Figure 30. Display Members List

- Add a Comment

The member can add a comment and discuss with other members about a specific event.



Schedule

Date : 2-19-2017
Time : 2:52 PM
Location : 1122 4th st, Fontana, CA 92009

Organizer

Eiman M

Event Description

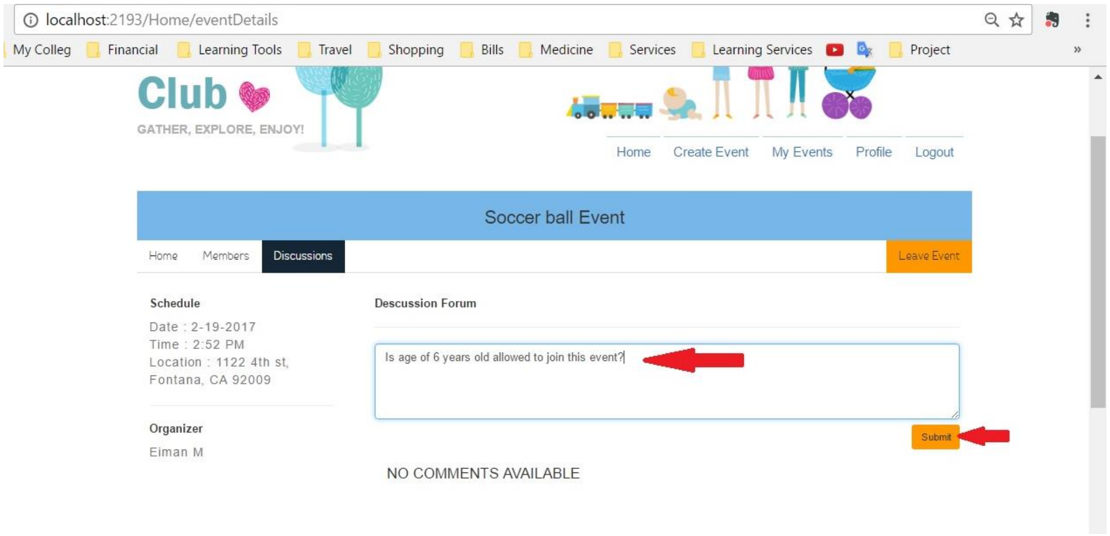
Bring your kids to play soccer ball with other kids. It is for children between 7 and 10 years old. we hope to see everyone there to enjoy together.

Question & Answers

- **Is there charge admission to the event?**

  yes
- **Will you be having food? If so, what kind?**

Figure 31. Add a Comment (1)



Schedule

Date : 2-19-2017
Time : 2:52 PM
Location : 1122 4th st, Fontana, CA 92009

Organizer

Eiman M

Descussion Forum

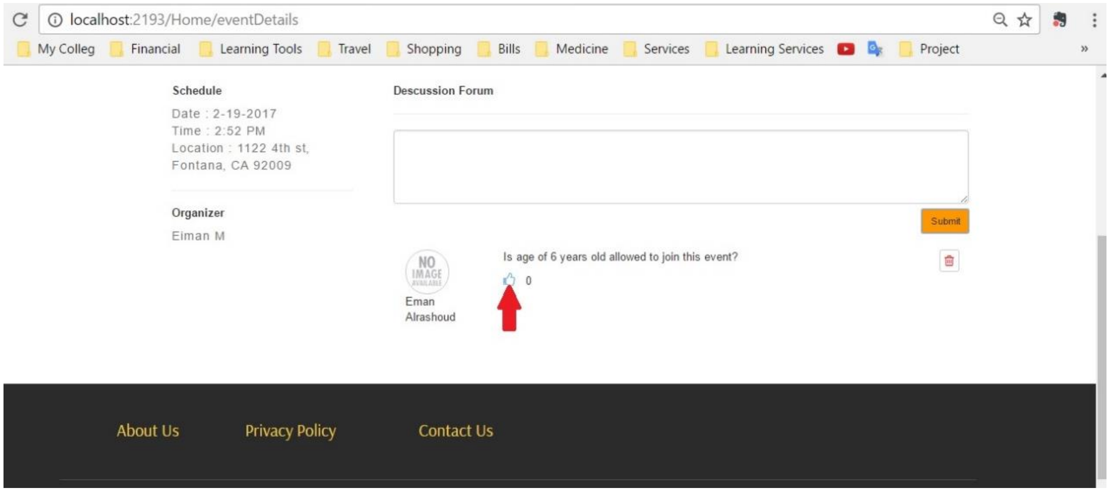
Is age of 6 years old allowed to join this event?

NO COMMENTS AVAILABLE

Figure 32. Add a Comment (2)

- Like or Dislike a Comment

The member can like or dislike any comment that is already created by him/her or other members.



Schedule

Date : 2-19-2017
Time : 2:52 PM
Location : 1122 4th st,
Fontana, CA 92009

Organizer

Eiman M

Descussion Forum

Eman
Alrashoud

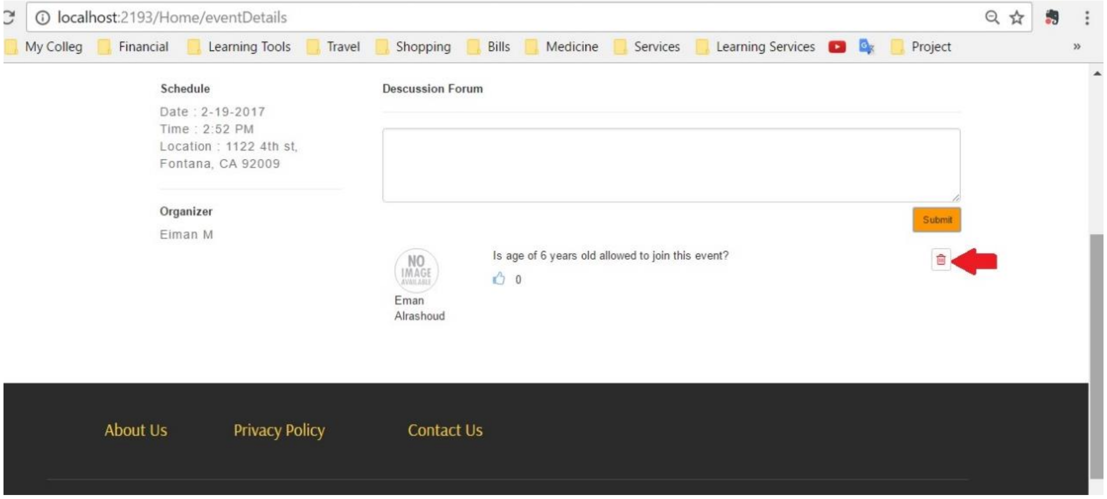
Is age of 6 years old allowed to join this event?

0

Figure 33. Like or Dislike a Comment

- Delete a Comment

The member can delete any comment that is created by him/her.



Schedule

Date : 2-19-2017
Time : 2:52 PM
Location : 1122 4th st,
Fontana, CA 92009

Organizer

Eiman M

Descussion Forum

Is age of 6 years old allowed to join this event?

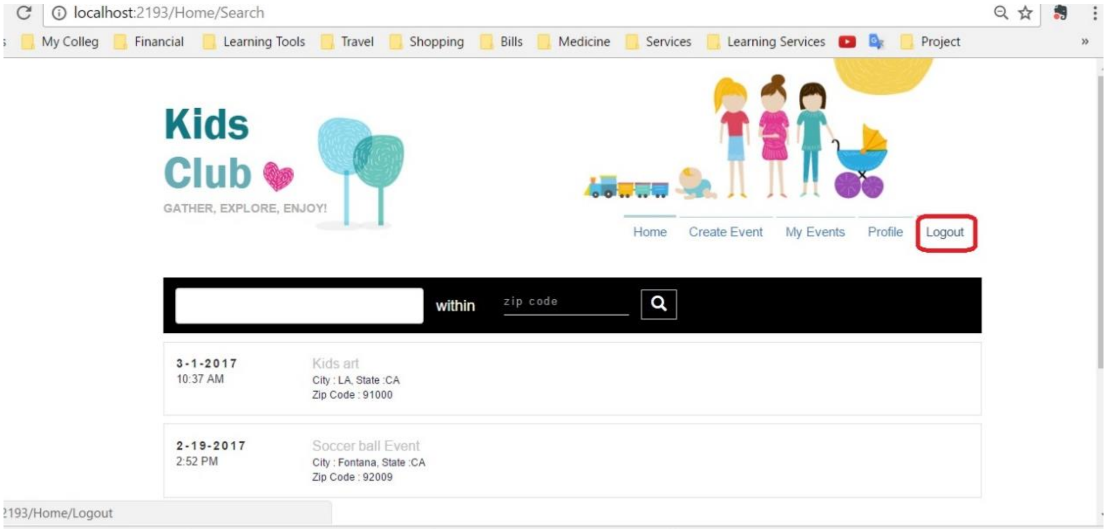
0

Eman
Alrashoud

Figure 34. Delete a Comment

- Sign out

The user signs out and exit the system.



3-1-2017
10:37 AM
Kids art
City : LA, State :CA
Zip Code : 91000

2-19-2017
2:52 PM
Soccer ball Event
City : Fontana, State :CA
Zip Code : 92009

Figure 35. Sign Out

- Manage Events/ View Events

The admin views the events that are created previously by members.

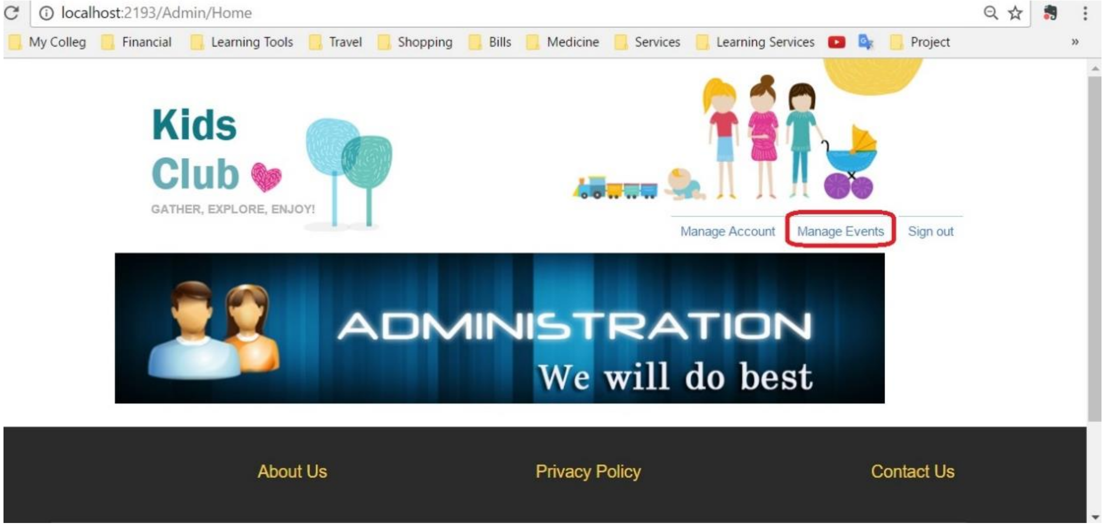
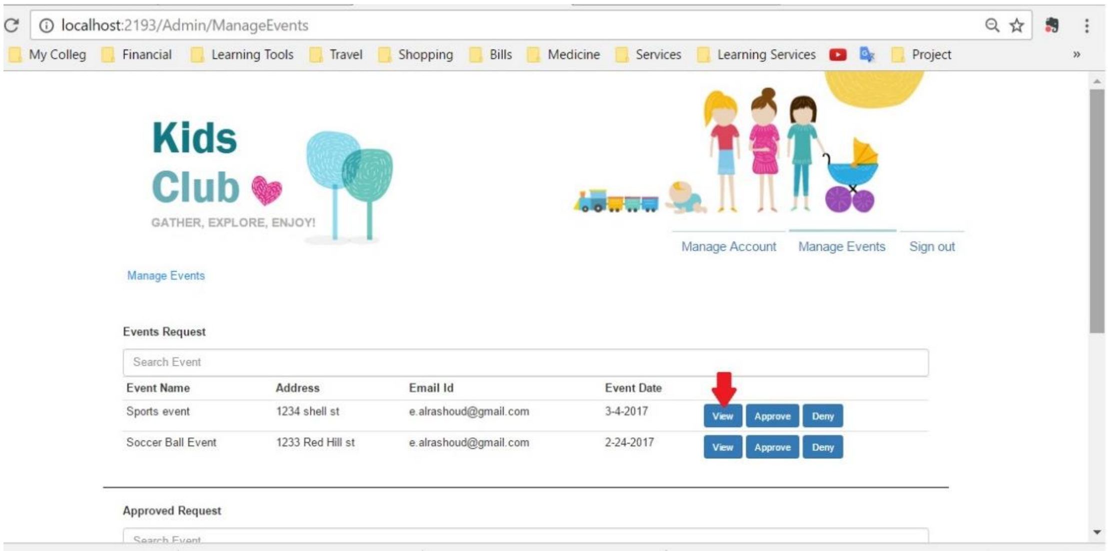


Figure 36. Manage Events/ View Event (1)



Events Request

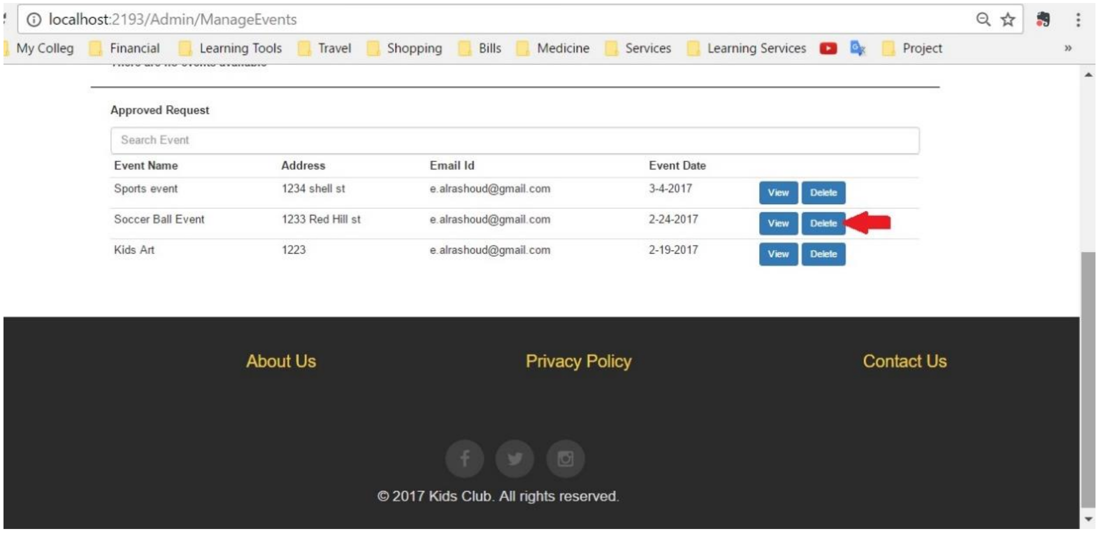
Event Name	Address	Email Id	Event Date	
Sports event	1234 shell st	e.alrashoud@gmail.com	3-4-2017	View Approve Deny
Soccer Ball Event	1233 Red Hill st	e.alrashoud@gmail.com	2-24-2017	View Approve Deny

Approved Request

Figure 37. Manage Events/ View Event (2)

- Manage Events/ Delete Event

the admin can delete any event that is created previously by members if required.



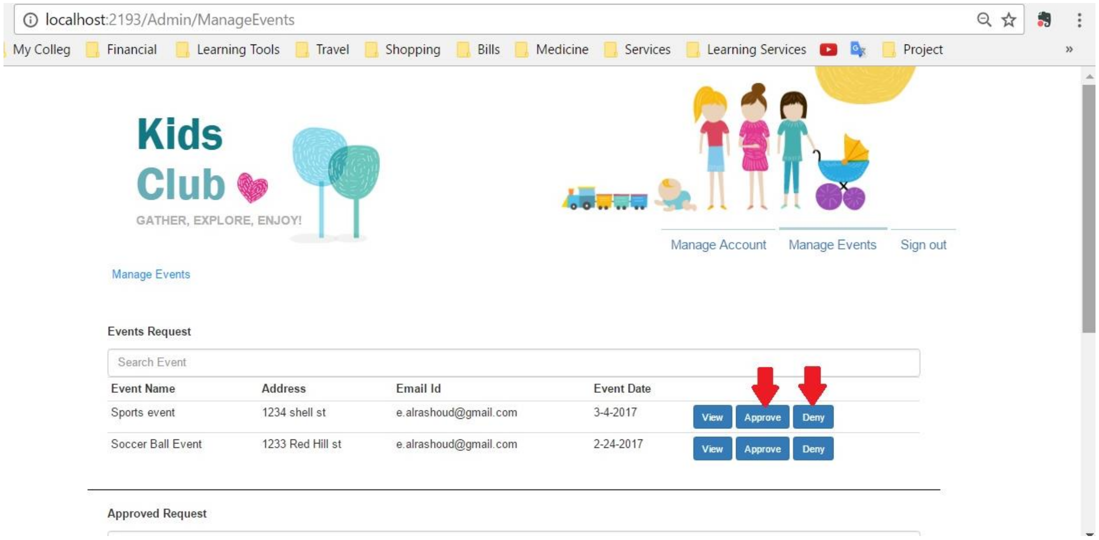
Approved Request

Event Name	Address	Email Id	Event Date	
Sports event	1234 shell st	e.alrashoud@gmail.com	3-4-2017	View Delete
Soccer Ball Event	1233 Red Hill st	e.alrashoud@gmail.com	2-24-2017	View Delete
Kids Art	1223	e.alrashoud@gmail.com	2-19-2017	View Delete

Figure 38. Manage Events/ Delete Event

- Manage Events/ Approve or Deny Event

The admin can to approve an event or deny it after he/she reviews.



Events Request

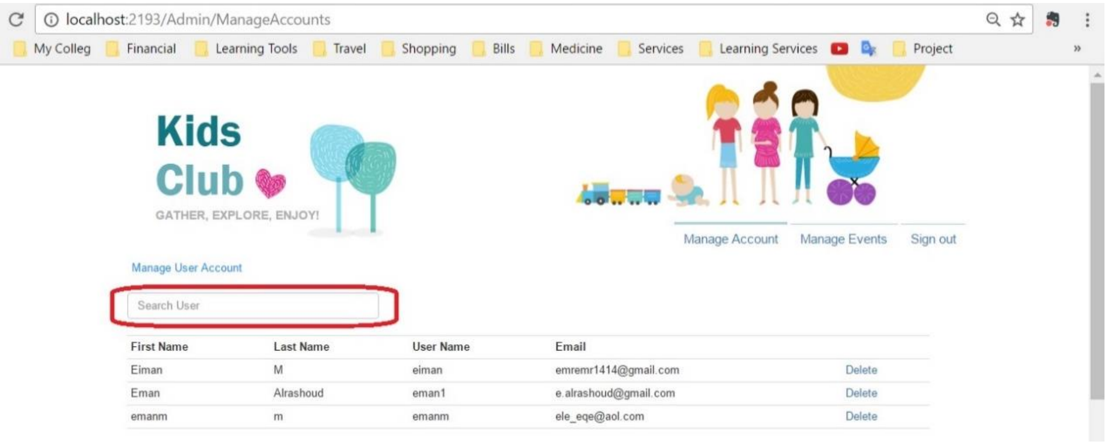
Event Name	Address	Email Id	Event Date
Sports event	1234 shell st	e.alrashoud@gmail.com	3-4-2017
Soccer Ball Event	1233 Red Hill st	e.alrashoud@gmail.com	2-24-2017

Approved Request

Figure 39. Approve or Deny an Event

- Manage accounts/ View or Search Accounts

The admin can view all accounts that are registered in the system, and search for a specific account by email or name.

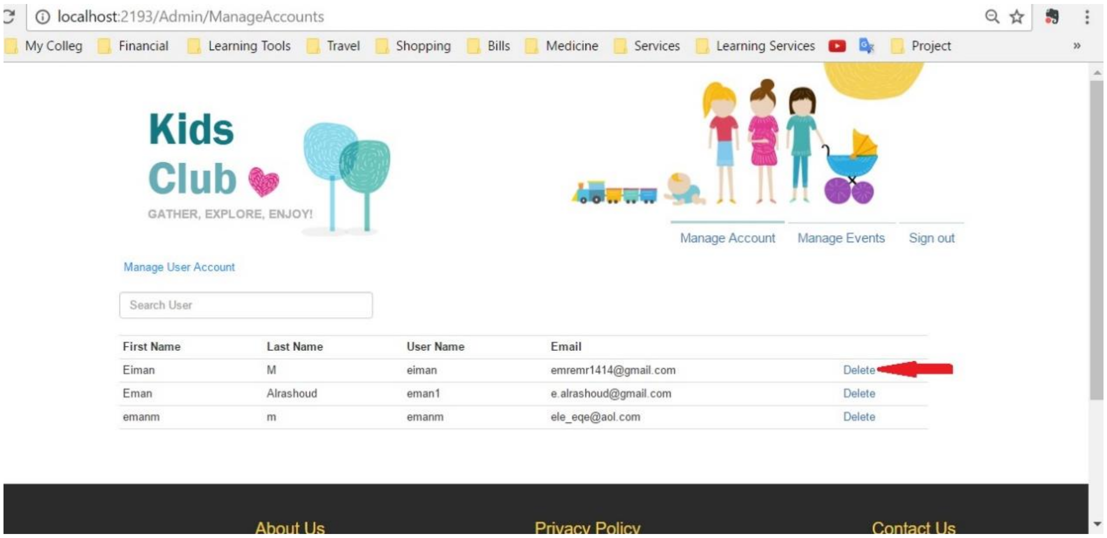


First Name	Last Name	User Name	Email	
Eiman	M	eiman	emremr1414@gmail.com	Delete
Eman	Alrashoud	eman1	e.alrashoud@gmail.com	Delete
emanm	m	emanm	ele_eqe@aol.com	Delete

Figure 40. View or Search for an Account

- Manage accounts/ delete an account

The admin can delete any account that is registered previously if required.



First Name	Last Name	User Name	Email	
Eiman	M	eiman	emremr1414@gmail.com	Delete
Eman	Alrashoud	eman1	e.alrashoud@gmail.com	Delete
emanm	m	emanm	ele_eqe@aol.com	Delete

Figure 41. Delete an Account

APPENDIX A

SYSTEM CODE

```
using KidsClub.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace KidsClub.Controllers
{
    public class Admin : Controller
    {

        public ActionResult Index()
        {
            return View();
        }

        public ActionResult Login()
        {
            return View();
        }

        public ActionResult ManageAccounts()
        {
            try
            {
                return View();
            }
            catch (Exception ex)
            {
                throw ex;
            }
        }

        public ActionResult ManageEvents()
        {
            try
            {
                return View();
            }
            catch (Exception ex)
            {
                throw ex;
            }
        }
        #endregion

        #region Manage Accounts Funtionality

        [HttpPost]

        public JsonResult GetUserAccounts();
        #endregion

        #region Delete User Account Funtionality

        [HttpPost]

        public JsonResult DeleteUserAccount(int userId);

        #endregion

        [HttpPost]

        public JsonResult GetAllEvents();

        #region Approve Funtionality

        [HttpPost]

        public JsonResult ApproveEvent(int eventId, string action);

        #endregion

        public ActionResult eventDetails()
        {
            return View();
        }

        public ActionResult userProfileView()
        {
            return View();
        }
    }
}

        #region View : Index or Home page

        public ActionResult Index()
        {
            return View();
        }
        #endregion

        #region Search Screen

        public ActionResult Search()
        {
            if (Session["User"] == null)
            {
                return RedirectToAction("Index", "Home");
            }
            return View();
```

```
}
#endregion

#region View : Login Page

public ActionResult Authenticate(int id)
{
    Registartion objUserLoginClass = new Registartion();
    string message = objUserLoginClass.AuthenticateUser(id);
    return View();
}

#region Login Page

public ActionResult Login()
{
    return View();
}
#endregion

#region View : Registarion Page

public ActionResult Registarion()
{
    return View();
}
#endregion

#region View : About Us

public ActionResult About()
{
    return View();
}
#endregion

#region View : Contact Us

public ActionResult Contact()
{
    return View();
}
#endregion

#region Logout

public ActionResult Logout()
{
    Session.Abandon();
    return RedirectToAction("Index");
}
#endregion
```

```
#region Login Funtionality

[HttpPost]

public JsonResult UserLogin(User loginData);

#endregion

#region

public JsonResult CheckUserExistance(string Username, string Email);

#endregion

#region Registration Funtionality

public JsonResult Registration(Registration registrationData);

#endregion

#region User Profile

public ActionResult userProfile()
{
    if (Session["User"] == null)
    {
        return RedirectToAction("Index", "Home");
    }
    return View();
}
public ActionResult userProfileView()
{
    if (Session["User"] == null)
    {
        return RedirectToAction("Index", "Home");
    }
    return View();
}
public ActionResult ManagePassword()
{
    return View();
}
[HttpPost]

public JsonResult updatePassword(string changePassword, int UserId);

[HttpPost]

public JsonResult RememberMe(string email);

[HttpPost]
```

```
        public ActionResult updateProfile(updateUserDetails Data)
        {
        #endregion

        #region Event Details Page

        public ActionResult eventDetails()
        {
            if (Session["User"] == null)
            {
                return RedirectToAction("Index", "Home");
            }
            return View();
        }
        #endregion

        #region Create Funtionality

        public ActionResult CreateEvent()
        {
            if (Session["User"] == null)
            {
                return RedirectToAction("Index", "Home");
            }
            return View();
        }
        #endregion

        public ActionResult MyEvents()
        {
            if (Session["User"] == null)
            {
                return RedirectToAction("Index", "Home");
            }
            return View();
        }

        [HttpPost]

        public JsonResult CreateEvent(EventsModel eventDetails, Questions
queries);

        #endregion

        [HttpPost]

        public JsonResult searchEventDetails(string searchByDetails, string
zipcode);

        [HttpPost]
        public JsonResult getEvnetDetailsByID(string Id, string UserId);

        [HttpPost]

        public JsonResult joinEvent(string EventId, string UserId, string
EventName, string FirstName, string MailID);

        [HttpPost]

        public JsonResult leaveEvent(string EventId, string UserId, string
EventName, string FirstName, string MailID);

        [HttpPost]

        public JsonResult submitDiscusion(string DiscussionData, string
EventId, string UserId);

        [HttpPost]

        public JsonResult deleteComment(string CommentID);

        [HttpPost]

        public JsonResult likeUnLike(string CommentID, string LikeOrUnLike,
string UserId);

        [HttpPost]

        public JsonResult GetAllEvents(string userId);

        [HttpPost]

        public JsonResult UpdateEvent(EventsModel eventDetails,
List<QuesNAnsEdit> QuestionsData);

        [HttpPost]

        public JsonResult DeleteEvent(string eventId);

        [HttpPost]

        public JsonResult GetEventsQuestions(string eventId)
}

        public userDetails UserLogin(LoginDTO loginData);

    public class AllUsers
    {
        string connString =
ConfigurationManager.ConnectionStrings["DBConnection"].ConnectionString.ToStri
ng();
```

```
        public List<userDetails> GetAllUsers();

        public bool DeleteUserAccount(int userId);

using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Web;

namespace KidsClub.Models
{
    public class Events
    {
        string connString =
ConfigurationManager.ConnectionStrings["DBConnection"].ConnectionString.ToStri
ng();
        Mail mail = new Mail();
        public string CreateEvent(EventsModel Data, Questions queries);

        public List<object> GetAllUserEvents(int userId);

        public List<EventsModel> GetAllEvents(int userId);

        public bool UpdateEvent(EventsModel eventDet, List<QuesNAnsEdit>
QuestionsData);

        public bool ApproveEvent(int eventId, string action);

        public bool DeleteEvent(int eventId);

        public string GetMembersMailId(int eventId,int flag);

        public List<QuesNAnsEdit> GetEventsQuestions(int eventId);

        public List<object> GetEventDetails(int ID, int UserID);

        public string JoinEvent(int EventId, int UserId);

        public string LeaveEvent(int EventId, int UserId);

    public class Mail
    {
        public string RegistrationMail(string MailID, string Subject,
StringBuilder MailBody);

        public string SendEmail(string MailID, string FirstName, string
EventName, string status);
```

```
        public string EventApprovedMail(string MailID, string Subject,
StringBuilder MailBody);

        public List<EventsModel> searchingEventData(string description, string
zipCode);

        public string ProfileUpdate(userDetails Data);

        public Registration RememberMe(string email);

        public string Registartion(RegistrationDTO Data);

        public string AuthenticateUser(int id);

        public string CheckUserExistance(string Username, string Email);

-----------------------------------USER Classes----------------------------

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace KidsClub.Models
{

    public class User
    {
        public string Username { get; set; }
        public string Password { get; set; }
    }

    public class Registration
    {
        public string First_Name { get; set; }
        public string Last_Name { get; set; }
        public string Username { get; set; }
        public string Email { get; set; }
        public string Password { get; set; }
        public string ConfPassword { get; set; }
    }

    public class Member
    {
        public int ID { get; set; }
        public string First_Name { get; set; }
        public string Last_Name { get; set; }
        public string User_Name { get; set; }
        public string Email { get; set; }
        public string Password { get; set; }
        public string Description { get; set; }
        public int No_Of_Kids { get; set; }
        public string Ages { get; set; }
        public string Image { get; set; }
        public string Created_Date { get; set; }
        public string Updated_Date { get; set; }
    }

    public class Event
    {
        public int UserId { get; set; }
        public int Id { get; set; }
        public string Address { get; set; }
        public string Location { get; set; }
        public string Description { get; set; }
        public string EventName { get; set; }
        public string EventDate { get; set; }
        public string EventTime { get; set; }
        public string City { get; set; }
        public string State { get; set; }
        public string Zipcode { get; set; }
        public string EmailId { get; set; }
        public string Approved { get; set; }
        public string OrgName { get; set; }

    }

    public class Questions
    {
        public string Question1 { get; set; }
        public string Question2 { get; set; }
        public string Question3 { get; set; }
        public string Question4 { get; set; }
        public string Question5 { get; set; }
    }

    public class comments
    {
        public int commentID { get; set; }
        public string comment { get; set; }
        public int ID { get; set; }
        public string First_Name { get; set; }
        public string Last_Name { get; set; }
        public string Image { get; set; }
        public int LikeCount { get; set; }
        public int UserLikedOrNot { get; set; }
    }
}
```

REFERENCES

[1] H. Ha, M. Hwang, T. Yeo and J. Kang, "Teachers' Efforts to Facilitate the Social Development of Multicultural Children", *Procedia - Social and Behavioral Sciences*, vol. 159, pp. 110-114, 2014.

[2] M. Stewart, N. Letourneau, J. Masuda, S. Anderson and S. McGhan, "Impacts of Online Peer Support for Children With Asthma and Allergies: "It Just Helps You Every Time You Can't Breathe Well", *Journal of Pediatric Nursing*, vol. 28, no. 5, pp. 439-452, 2013.

[3] K. Segrist, "Impact of Support Groups on Well-Being of Older Women", *J. of Gerontological Social Work*, vol. 51, no. 1, pp. 42-52, 2008.

[4] C. Harper, J. Symon and W. Frea, "Recess is Time-in: Using Peers to Improve Social Skills of Children with Autism", *J Autism Dev Disord*, vol. 38, no. 5, pp. 815-826, 2007.

[5] "Mom and Baby Center Website Template | WIX", *Wix*, 2017. [Online]. Available:http://www.wix.com/website-template/view/html/1390?originUrl=http%3A%2F%2Fwww.wix.com%2Fwebsite%2Ftemplates%2Fhtml%2Fcommunity-education%2Fcommunity%2F1&bookName=&galleryDocIndex=1&category=community-education. [Accessed: 22- Feb- 2017].

[6] IEEE Software Engineering Standards Committee, "IEEE recommended practice for software requirements specifications", *IEEE organization*, 1998.

[7] "Graphic Elements (Windows)", *Msdn.microsoft.com*, 2017. [Online]. Available:https://msdn.microsoft.com/en-us/library/windows/desktop/dn742484%28v=vs.85%29.aspx. [Accessed: 22- Feb- 2017].

[8] K. Soni, "E-Store Project", *Academia.edu*, 2017. [Online]. Available: http://www.academia.edu/11137694/E-Store_Project. [Accessed: 22- Feb- 2017].

[9] “Soar Event Planning and Registration Sheet”, *University of Nebraska*, 2017. [Online]. Available: http://involved.unl.edu/soar/docs/soar_sheets/event_planning_registration.pdf. [Accessed: 22- Feb- 2017].

[10] C. Larman, *Applying UML and patterns*, 1st ed. Upper Saddle River, N.J.: Prentice Hall Professional Technical Reference, 2005.

[11] “IEEE standard for software test documentation”, *IEEE organization*, 1998.