

Project Specification: SportsLover

Team265: Wenyan Hu (wenyanh), Bin Zhu (bzhu1)

Section: A

- **Background**

Most people love all kinds of sports. But sometimes, especially when people get into work, people don't have enough available time and friends for normal or extreme sports. There are kinds of sports forums now but no real outside sports mapping e-community. SportsLover is a website that focuses on matching people who want to have a sport sometime to go out for sports together. People can be mapped to groups with specific sports plan and join them, or search for some sports places and events information to schedule a sports plan by themselves for sports fun.

- **Action List**

1. Post sports group plans.
2. Join any available sports group for sports schedule.
3. Have group talking chat room and friend communication mechanism for socialization.
4. Use pins of Map for sport places and information search.
5. Search any specific satisfied groups according to expense, date, sports classification, and so on.
6. Create sports place information and have ranking and comment mechanism about it.
7. Search any specific satisfied sports places according to expense, sports classification, popular degree, group size, rank, and so on.
8. Other basic functionality: sign up, log in & out, reset password, edit profile, edit group information, notifications, user rank and comment

- **Wireframes/ HTML Mock-ups & Actions Explanation**

Recent Sports

Nearby Sports Places

Most Popular Sports Places

Search Specific Group

Sports Items

Date Begin

Date End

Expense

Search

Map with Pins

Google Map Search

Figure1: Home Page (search page)

The home page is a main search page. It mainly has 4 parts of functions.

1. Search for general sports information SportsLover provides.

Recent Sports would go to sports group page and return all the sports group invitation posts. Nearby Sports Places would go to sports place page and return all the sports places. Most popular sports places would return the sports place sorted by visited number.

2. Search Specific Groups with detailed sports items, dates, and expense range.

If some one knows what day he/she is available and plans to join a group with specific interest and expense, he/she can use filters in search specific group panel. The “sports items” is a dropdown with classes like balls, water sports, and so on. For each class, it provides multiple checklists for users to choose any sports items they prefer. And then, they choose date range from some day to some day that they may be available. Also, expense is also a dropdown with choices of cost range, such as free, below 20\$, etc. After finish selecting filter conditions, user click search to go to sports group page.

3. Sports Pins on Map

The third part is a map in the search page with pins as group plans in some sports places. Once user enlarge the map and click some specific sports place they are interested in, they can click the pin and find sports group posts who are still available for sportsmen and sportswomen to join.

4. Google Map Help Supplement

In the bottom of the search page, SportsLover provides Google map search for sports and address information user want to know if necessary.

SportsLover	Search Sports		Sports Places	LogIn
Ball Sports	Date	Group Size	Expense	
Sky Sports	Group Title	Group Title	Group Title	
Mountain Sports	Basketball	Swimming		
Water Sports	Group Size: 5	Group Size: 2		
Motor Sports	Cost: Free	Cost: 20\$		
Ice Sports	Group Title	Group Title	Group Title	
Wild Sports				
.				
.				
.				
.				

Figure2: Groups Page (search result page)

Then if you search sports group or just click Recent Sports button, you would jump into group page and at first see all the posts of groups seeking for group of people together for some sports plan. The groups can be filtered by date, group size, and also expense. Also, groups can be sorted by date (e.g. most recently preferred), by group size (from larger to smaller size), or by expense (from cheaper to more expensive).

Moreover, when user clicks the left buttons, every sport class is a filter that would return only sports in this classification. And in the specific class, every group also has its own sports item such as basketball, swimming.

In the top navigation bar, the group page also provides search bar for searching specific sports item's group posts.

Every group post can be opened. Then you will see group as figure 6 shows, with group subject, sports item such as baseball, place such as Pittsburgh swimming pool and its detailed avenue and street address. Position existing tells user how many positions this group is still rest and if it is not full, users could send request to group owner for joining.

When a user wants to send a request, the system would remind him/her to login or signup the account. Suppose the user has had an account, the user logs in. When a user logs in, the user can click User button to go to user page. This page has the user's information with his/her group event or past history. It also includes the friends and groups list of the person.


SportsLover		Sports Places	Notification	User▲
	Most Recent Group Event/Plan			
User Name	Basic Information		Create Group	
Other Group Event		Friends		
Other Group Event		Groups		
Past Group Event		Chat Delete		
Past Group Event		Chat Delete		
Past Group Event		Chat Delete		
Past Group Event		Chat Delete		
Past Group Event		Chat Delete		
Past Group Event		Chat Delete		

Figure3: User Page

Edit Profile
Change Password
Manage your groups
Log Out

Figure4: User Dropdown

The User icon also has a dropdown with edit profile and password change, as well as log out. They are all similar to all the other websites, so I would not explain them in detail. One more feature is that group owners can modify their groups if he changes any group information such as date.

When user want to plan a new sports group for upcoming sports event, he/she can “Create Group” with information provided as figure 5. The sports item is also a dropdown. If some item is existed in the dropdown, user can directly choose it. If choose others, user should define its item content and which class it belongs. After all information set and click post, the post shows on group page and also a new pin shows on search page’s map.

Group Subject	
Sports Item	Place
Specific Address	
Average Cost	Person Number
Time Range	
Brief Event Introduction	
Post	

Figure5: Create Group and Post

Group Subject	
Sports Item	Place
Specific Address	
Average Cost	Position Existing
Time Range	
Brief Event Introduction	
Request to Join	

Figure6: View Group & Request to Join

After request approved, people in this sports plan group would be in a chat room for more information communication as figure 7 shows. If they want, they can also send friend request to group members to become friends with them. Either requesting to join the sports group or be friends with the member, the corresponding user would receive a notification as figure 8 shows.

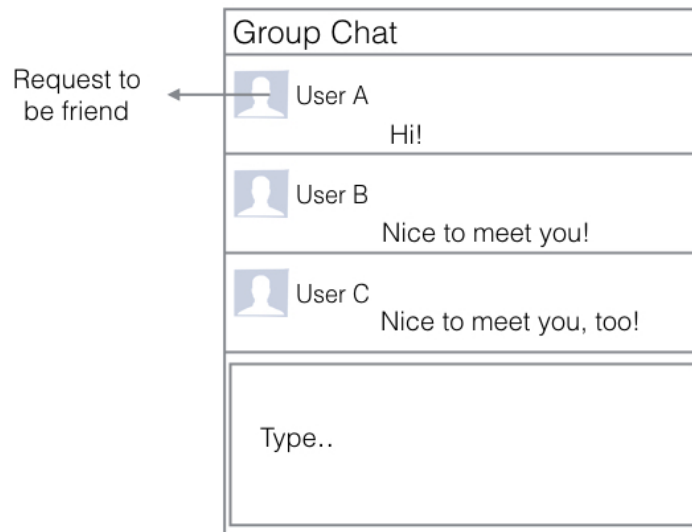


Figure7: Group Chat & Add Friends Option

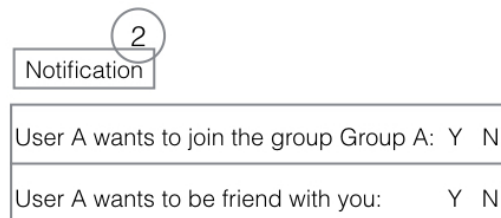


Figure8: Notifications

When you click “Sports Places” button, you are directed into sports place page. Every place is the place group owners described in their group posts. After the sports event date, member in this group are informed to rank the place, and provide photos or other comments if they want. Based on all the data provided, the sports places can be filtered by rank rate, popular degree, as well as expense. The places can also be sorted by rank, popularity, and expense, filtered by sports classes, similar to sport group page.

SportsLover	Search Places		Sports Group	Login
Ball Sports	Rank	Popularity	Expense	
Sky Sports				
Mountain Sports	Pitts Playground	Pitts Swim Pool		
Water Sports	Rank: 4.5	Rank: 3.7		
Motor Sports	Footprint: 200	Footprint: 189		
Ice Sports				
Wild Sports				
.	.	.	.	
.	.	.	.	
.	.	.	.	

Figure9: Sports Places Page

Pitts Playground	Rank: 4.5
Comment and Rank with Photo	
Comment and Rank with Photo	
Comment and Rank with Photo	
.	
.	
.	

Figure10: Sports Place Detail

As a user clicks into some specific sports place, he/she can see the detailed comments with rank rate, along with photos for this place as figure 10 shows.

● Models

User: Username and password information for login
 django.contrib.auth.models.User

Info:

```
class Info(models.Model):
    user = models.ForeignKey(User)
```

```

first_name = models.CharField(max_length=20)
last_name = models.CharField(max_length=20)
age = models.CharField(max_length=20, default="", blank=True)
bio = models.CharField(max_length=420, default="", blank=True)
image = models.ImageField(upload_to='photos', default='empty.png',
blank=True)
groups = models.ManyToManyField(Group)
friends = models.ManyToManyField(Friend)

def __unicode__(self):
    return self.user

```

SportsClass:

```

class SportsClass(models.Model):
    name = models.CharField(max_length=40)
    item = models.ManyToManyField(SportsItem)

    def __str__(self):
        return self.name
    class Meta:
        ordering = ('name',)

```

SportsItem:

```

class SportsClass(models.Model):
    item = models.CharField(max_length=40)

    def __str__(self):
        return self.item
    class Meta:
        ordering = ('item',)

```

Friend

```

class Friend(models.Model):
    name = models.CharField(max_length=40)
    user = models.ForeignKey(User)

    def __unicode__(self):
        return self.user

```

Group:

```

class Group(models.Model):
    owner = models.ForeignKey(User)
    place = models.ForeignKey(Place)
    member = models.ManyToManyField(User)

```

```
size = models.IntegerField(max_digits=5)
introduction = models.CharField(max_length=420)
cost = models.DecimalField(max_digits=5)
date_begin = models.CharField(max_length=40)
date_end = models.CharField(max_length=40)
```

```
def __unicode__(self):
    return self.owner
```

Place:

```
class Place(models.Model):
    name = models.CharField(max_length=40)
    address = models.CharField(max_length=100)
    visitor = models.ManyToManyField(User)

    def __unicode__(self):
        return self.name
```

Rank: Finally calculate average rank

```
class Rank(models.Model):
    rank = models.DecimalField(max_digits=5)
    place = models.ForeignKey(Place)

    def __unicode__(self):
        return self.rank
```

Cost: Finally calculate average cost

```
class Cost(models.Model):
    cost = models.DecimalField(max_digits=5)
    place = models.ForeignKey(Place)

    def __unicode__(self):
        return self.cost
```