App Development & Modeling

BSc in Applied Computing

Produced by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
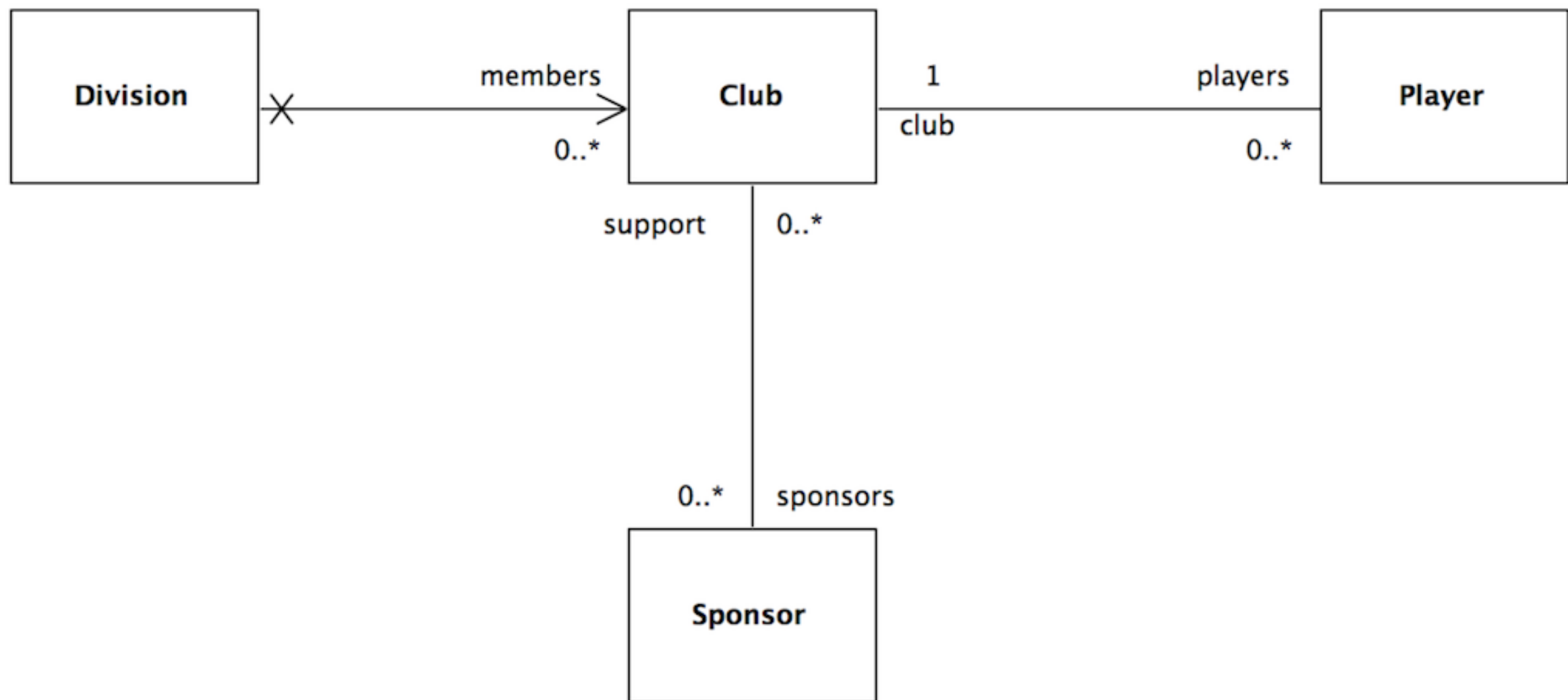Waterford Institute of Technology
http://www.wit.ie
http://elearning.wit.ie

Waterford Institute *of* Technology
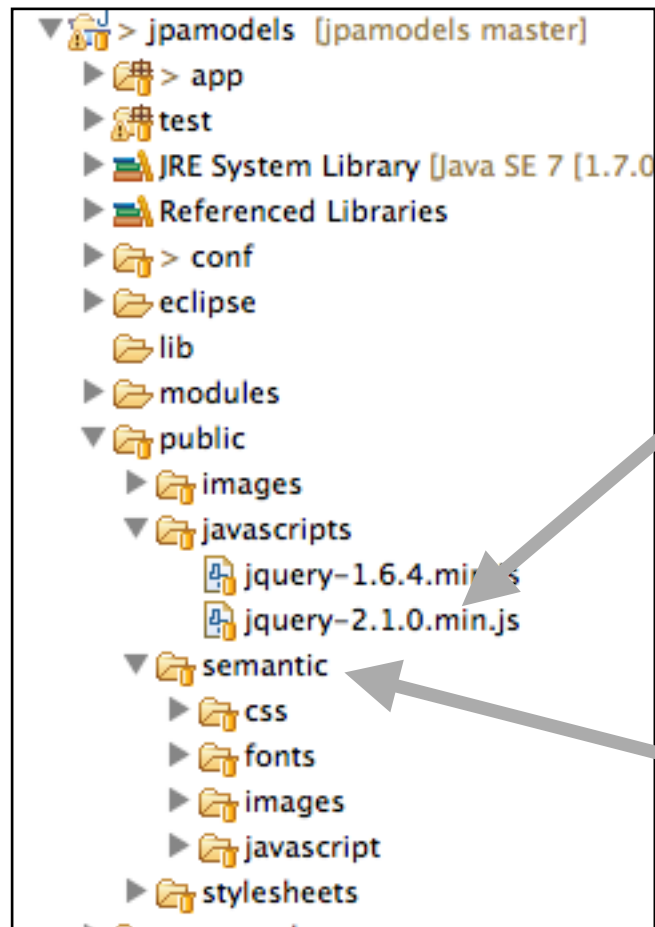INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning
support unit

JPA 2

OneToMany, ManyToOne, ManyToMany

Semantic UI & JQuery

jQuery 2.x

jQuery 2.x has the same API as jQuery 1.x, but *does not support Internet Explorer 6, 7, or 8*. All the notes in the jQuery 1.9 Upgrade Guide apply here as well. Since IE 6/7/8 are still relatively common, we recommend using the 1.x version unless you are certain no IE 6/7/8 users are visiting the site. Please read the 2.0 release notes carefully.

Download the compressed, production jQuery 2.1.0

Download the uncompressed, development jQuery 2.1.0

Download the map file for jQuery 2.1.0

jQuery 2.1.0 release notes

Content   Introduction

Semantic UI 0.15.1
UI is the vocabulary of the web.

Semantic empowers designers and developers by creating a language for sharing UI.

VIEW UI   DOWNLOAD

Delete Club

| Divisions | **Clubs** | Players | Sponsors |

Clubs

Club		
dunmore	jim mary	✖
tramore	sam john	✖
fenor	mike linda	✖

Delete Club


```html
<td>
  <a class="ui ui icon button" href="/clubs/delete/${club.id}">
   <i class="delete red icon"></i>
  </a>
</td>
```
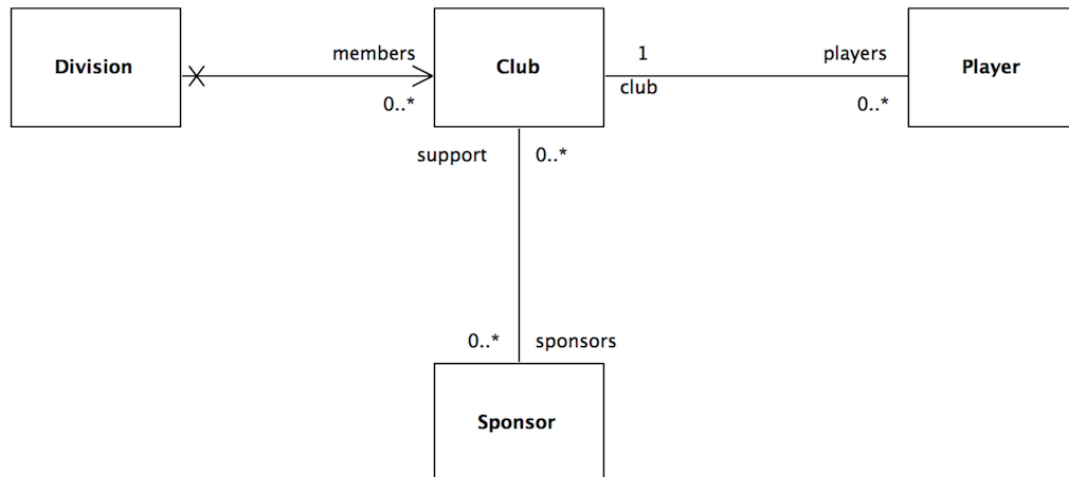
GET	/clubs/delete/{id}	Clubs.delete

```java
public static void delete(Long id)
{
  Club club = Club.findById(id);
  if (club != null)
  {
    Logger.info("Trying to delete " + club.name);
    List<Division> divisions = Division.findAll();
    for (Division division : divisions)
    {
      if (division.members.contains(club))
      {
        division.members.remove(club);
        division.save();
        Logger.info ("removing club from division");
      }
    }
    club.delete();
  }
  index();
}
```
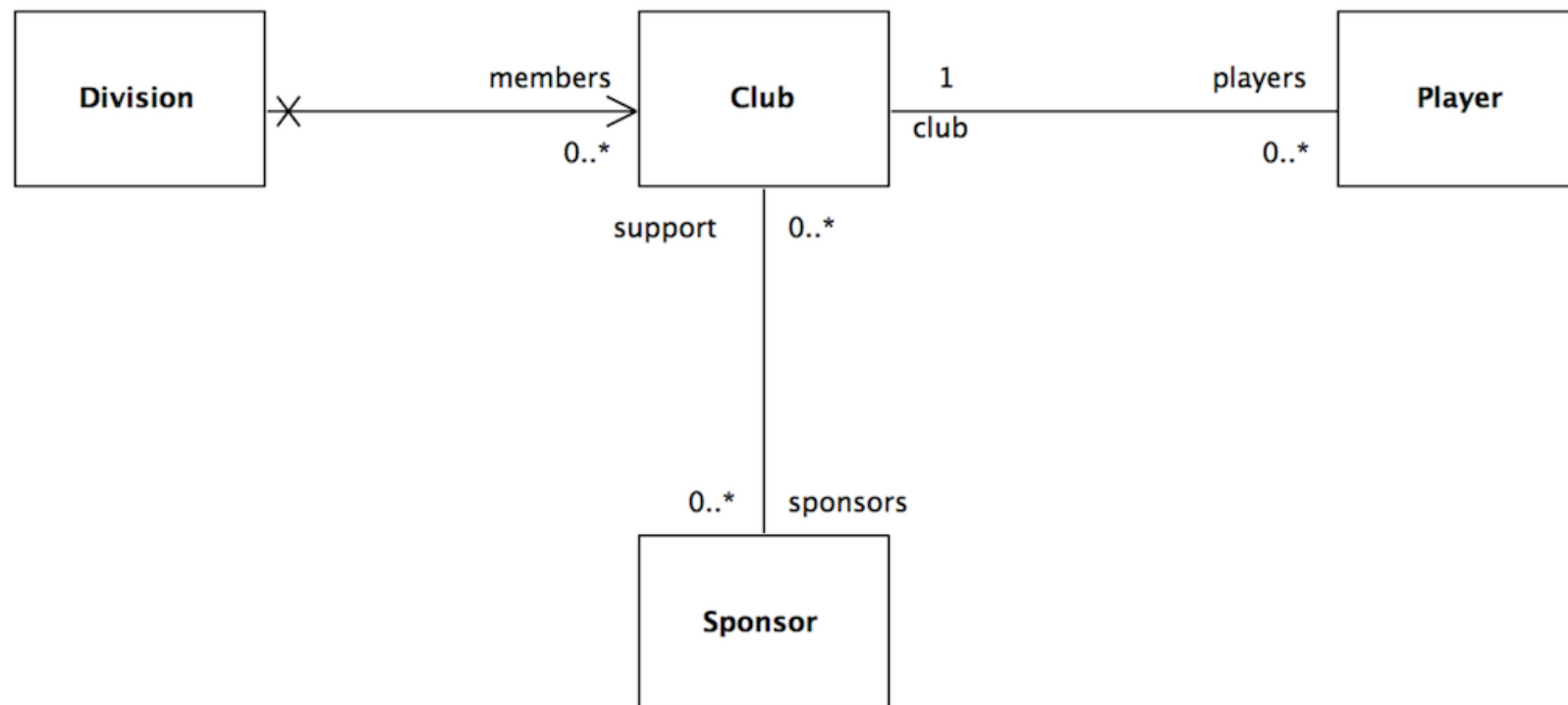
Delete Division

Divisions Clubs Players Sponsors

Divisions

Division		
senior	tramore	✖
	dunmore	
junior	fenor	✖

```
<td>
  <a class="ui ui icon button" href="/divisions/delete/${division.id}">
   <i class="delete red icon"></i>
  </a>
</td>
```

```
GET     /divisions/delete/{id}                    Divisions.delete
```

```
public static void delete(Long id)
{
  Division division = Division.findById(id);
  division.delete();
  index();
}
```

Show Sponsors Clubs

Divisions	Clubs	Players	**Sponsors**

Sponsors

Sponsor	Clubs	
pub	tramore	✖
newsagent	tramore fenor	✖

Table inside a table


```
<table class="ui table">
  <thead>
    <tr>
      <th>Sponsor</th>
      <th>Clubs</th>
      <th></th>
    </tr>
  </thead>
  <tbody>
    #{list items:sponsors, as:'sponsor'}
      <tr>
        <td>${sponsor.name}</td>
      <td>
        <table class "ui table">
          <tr>
           #{list items:sponsor.support, as:'club'}
             <td>${club.name}</td> </tr>
           #{/list}
          </tr>
        </table>
      </td>
        <td></td>
      </tr>
    #{/list}
  </tbody>
</table>
```

9

Yaml file - Froward References

- Test data in Yaml file cannot refer to objects that have not been seen in the file yet (reading from top to bottom)

- Bidirectional references can be included by including the objects twice

  - Once at top (partial)

  - Once at end (complete)

```
Sponsor(pub):
    name: pub

Sponsor(newsagent):
    name: newsagent
```

```
Club(tramore):
    name: tramore
    sponsors:
            - pub
            - newsagent

Club(fenor):
    name: fenor
    sponsors:
            - newsagent
```

```
Sponsor(newsagent):
    name: newsagent
    support:
            - tramore
            - fenor

Sponsor(pub):
    name: pub
    support:
            - tramore
```

Divisions	Clubs	Players	Sponsors

Sponsors

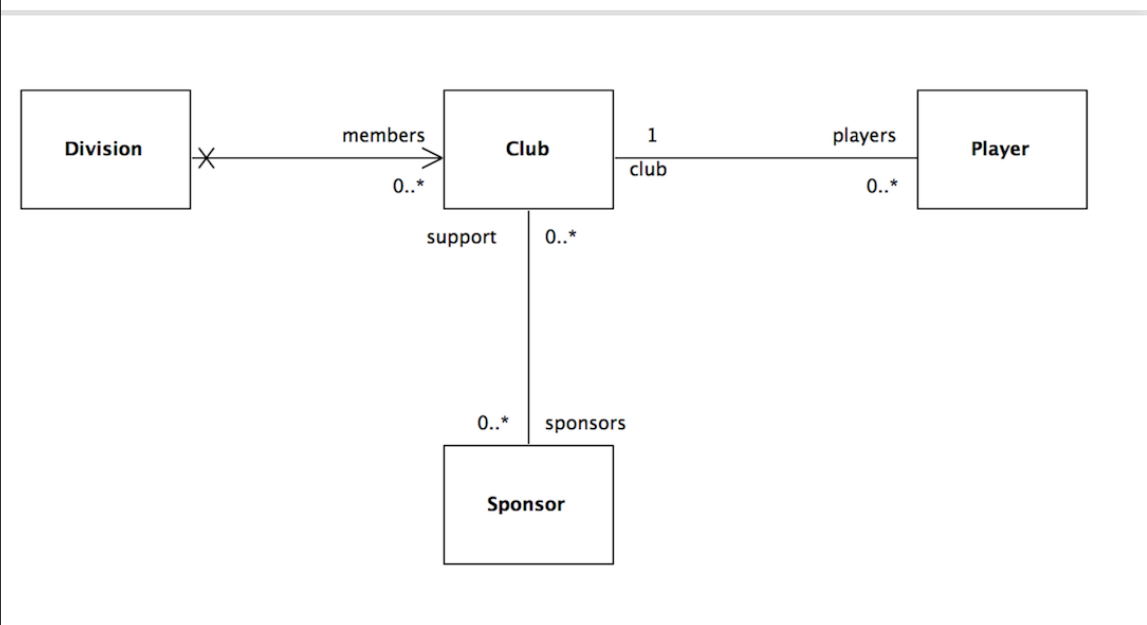| Sponsor	Clubs	
pub	tramore	✖
newsagent	tramore fenor	✖

```html
<td>
  <a class="ui ui icon button" href="/sponsors/delete/${sponsor.id}">
   <i class="delete red icon"></i>
  </a>
</td>
```

```
GET     /sponsors/delete/{id}                    Sponsors.delete
```

```java
public static void delete(Long id)
{
  Sponsor sponsor = Sponsor.findById(id);

  for (Club club  : sponsor.support)
  {
    club.sponsors.remove(sponsor);
    club.save();
  }

  sponsor.delete();
  index();
}
```

Players

Divisions	Clubs	**Players**	Sponsors

Players

Player	Club	
jim	dunmore	✖
mary	dunmore	✖
sam	tramore	✖

👤 **ADD PLAYER**

```
GET     /players/addplayer              Players.addPlayer
POST    /players/newplayer              Players.newPlayer
```

```
#{extends 'main.html' /}
#{set title:'Player Details' /}

<section class="ui raised form segment">
  <form action="/players/newplayer" method="POST">
    <div class="field">
      <label> Player Name </label>
      <input type="text" name="name">
    </div>
    <button class="ui blue submit button">Add</button>
  </form>
</section>
```

```java
public static void addPlayer()
{
  render();
}

public static void newPlayer(String name)
{
  Player player = new Player (name);
  player.save();
  index();
}
```

Player Name

james

ADD

- If the player is not in a club

  - then null type violation error here

  - Attempt to dereference null reference - there is no club member in player

```
#{list items:players, as:'player'}
  <tr>
    <td>${player.name}</td>
    <td>  ${player.club.name} </td>
    <td>
    <a class="ui ui icon button" href="/players/delete/${player.id}">
     <i class="delete red icon"></i>
    </a>
  </td>
    <td></td>
  </tr>
#{/list}
```
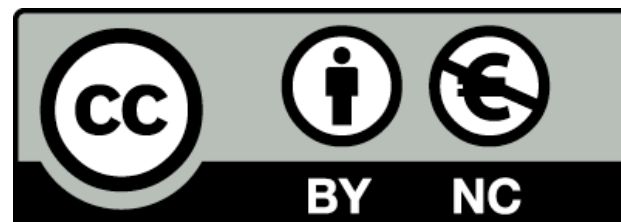
Null Safe Operator in Templates


```
#{list items:players, as:'player'}
  <tr>
    <td>${player.name}</td>
    <td>  ${player.club?.name} </td>
    <td>
    <a class="ui ui icon button" href="/players/delete/${player.id}">
     <i class="delete red icon"></i>
    </a>
    </td>
    <td></td>
  </tr>
#{/list}
```

player.club.?name

- .? is a 'null-safe' operator

- i.e. if there is a club, retrieve its name member, if not, then dont.

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning
support unit