TM470 Project Report

A Fencing Competition Results Web Service

Submitted in partial fulfillment of
the requirements for the award of the degree of

Bachelor of Science
in
Computing and IT

Submitted by

Matthew Anthony Carus B3951972

Under the guidance of
Prof. Peter Smith

Department of Computing and IT
The Open University
Milton Keynes, United Kingdom

In collaboration with

BRITISH FENCING

British Fencing
London, United Kingdom

Abstract

This project will create a web service capable of storing and serving up the results of fencing competitions.

Contents

List of Figures

List of Tables

Listings

1 Problem Definition

The sport of Fencing has a fairly small but loyal following in the UK. As such, the support systems around the sport are not as developed as they are in other sports. The National Governing Body of Fencing in the UK, British Fencing, host the results of fencing competitions on their website. The results are represented as static HTML files and there is no way to present the data other than as a list of the results of each competition (i.e. you can't view all of the results of a particular fencer).

2 Project Goals

The goal of this project is to produce a working system that meets requirements that will later be identified. The system will be designed and built using the knowledge that I have built up during the course of my Open University studies. As such it is likely to be designed using UML and implemented using Java. The specific elements that I will deliver will be as follows:

- Output from the modelling phase of the project. This will include a domain model, class models etc.
- A database schema, designed to best practices, that will be used to store all of the relevant information derived from the modelling exercise.
- A working database implementing the database schema, populated with real data.
- An API to the database such that data can be added to the database and data already in the database can be retrieved. The API should use standard data formats for accepting data and for presenting it. This API should be designed in such a way that it can be extended in the future to allow for other operations (for example updating data already in the database, searching for data etc.)
- A web interface to allow viewing of the data held within the database.

Some activities will specifically be excluded from the scope of the project. This is primarily due to time constraints but also to properly complete the activities would require more knowledge than I have gained during the course of my studies. If during the course of my research and development work it becomes evident that this can be added in a trivial manner then they may later be re-introduced into the scope of the project. These are:

- API Security. It is anticipated that I will not implement access control on the API. A live deployment of the API will either require that authentication be added or an application firewall be utilised to only permit certain operations.
- Web interface to advanced operations. It is accepted that the web interface will only allow viewing of the data exposed by the API. In order to add data to the database the API will be used directly, rather than via a web interface.

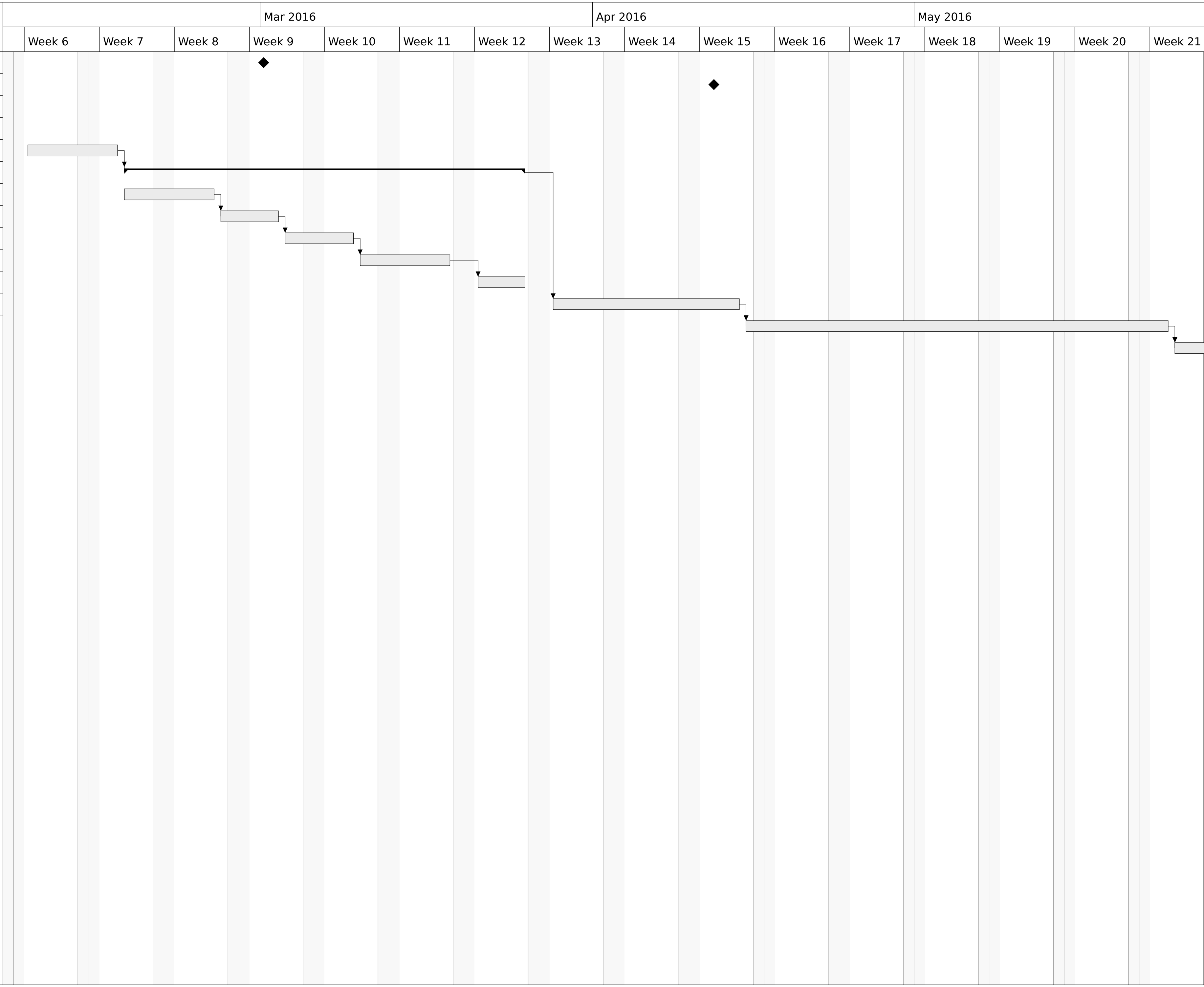
Aside from the domain research needed for the modelling phase and associated requirements generation, the following additional research tasks have been identified:

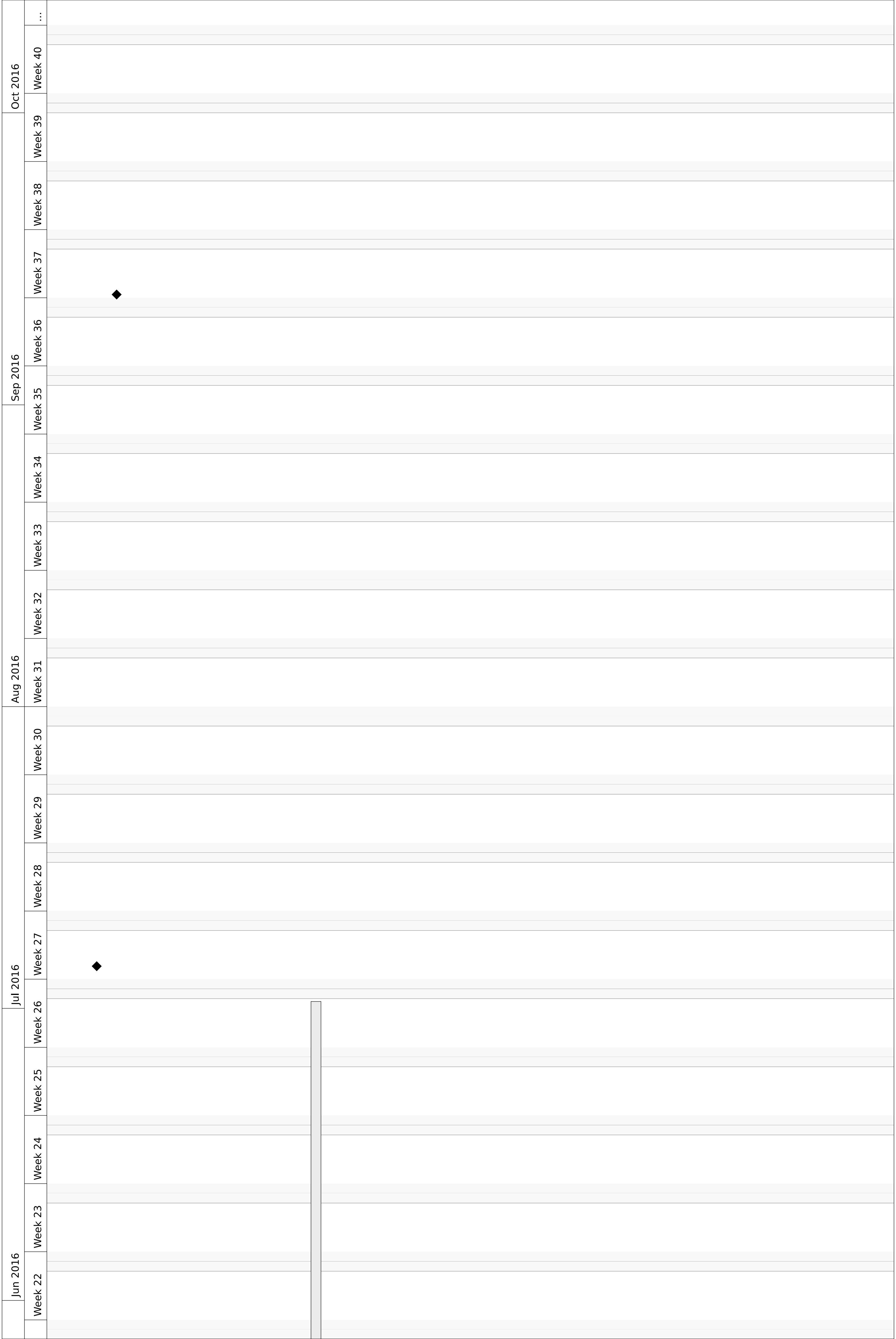
1. Research into the data formats that will be used to populate the database via the API, and the desired output formats from the API.

The output of this research is presented in section **??** beginning on page **??**

3 Project Plan

Name	Work
TMA01 Deadline	
TMA02 Deadline	
TMA03 Deadline	
EMA Deadline	
Project Planning	7d
Modelling	**28d**
Get ??? from BF	7d
Generate project glossary	4d
Generate domain model	5d
Validate domain model with BF	7d
Create class diagrams	5d
Database Schema	14d
API	28d
Web Interface	28d




Jun 2016
Jul 2016
Aug 2016
Sep 2016
Oct 2016
Week 22
Week 23
Week 24
Week 25
Week 26
Week 27
Week 28
Week 29
Week 30
Week 31
Week 32
Week 33
Week 34
Week 35
Week 36
Week 37
Week 38
Week 39
Week 40
...

4 Research

4.1 Data Formats

From speaking to Katie Rhodes (British Fencing) and to other competition organisers, it has become clear that one of two pieces of software are used to organise fencing competitions. These are called *Engarde* and *Fencing Time*. Both programs have the ability to output XML files detailing the results of the competition (Engarde Escrime, undated) and (Fencing Time, undated). XML samples from each program are supplied in Appendix A on page 56. Clearly, the two programs use different formats. For the output format from the API, it would be possible to re-use one (or both) of these two formats, or a different format could be used. If a different format is chosen then it could either be a format that I create specifically for this system, or it could be a format published elsewhere. Further internet-based research has indicated that there are standards in existence for representing sports data. One such standard is known as SportsML, the current version of which (as of Feb 2016) is SportsML-G2. This standard is defined by the *International Press Telecommunications Council (IPTC)*. I contacted the IPTC via their developer forums to ask if fencing data could be represented in SportsML format and the response was that it should be able to be represented but no-one was aware of anyone currently using SportsML for fencing data. Two interesting responses were received from the initial posting, one from Steve Potts at the *BBC* and another from Jean Fèvre of *L'Agence France-Presse (AFP)*.
Steve Potts, in an email to me stated that the BBC did use SportsML as their favoured format for receiving data, he also stated:

> The BBC are interested in the concept of local generated sports stats (compare with user generated, which it is not), and also in publishing stats for lesser-participated sports. The effort we expend in obtaining and publishing stats is naturally weighted towards the more popular sports, so we are investigating opening channels (with a lower barrier to entry) of sports stats ingest from outside our primary suppliers. Fencing isn't included in our supplier contracts, so creating a vocabulary for it removes one barrier for us to publish its stats. Our ideal scenario is for us to ingest a routine automated SportsML feed from British Fencing for fixtures, results and standings to appear across the BBC Website, Mobile App, Red Button TV, Connected TV. (Potts, 2016)

From this I take that the BBC would be interested in integrating with the system I am to build in order to receive data automatically from British Fencing. This is something the British Fencing are quite excited about! Conversely, Jean Fèvre, again in an email, states:

> I am in SportML group since long time but I dont like SportML too much because its very complex and it must be adapted to each sport. I work with IOC on Olympic Games since very long time (1988) and we (IOC, ASOIF,

> many international federations) have defined ODF XML format. For Rio games we will receive ODF 2. (Fèvre, 2016)

After studying at the two formats, I would dispute Fèvre's assertion that SportsML is very complex - I understood it more quickly than the ODF format. It is clear that there are two competing formats and it makes sense for me to decide to support primarily one format, with support for the second being made possible if it is able to be added. One outstanding issue I had to face before making a decision on my primary data format was that of whether or not SportsML would actually support fencing data or not. As suggested by members of the SportML development group, I decided to jump in at the deep end and actually produce a valid SportsML file with fencing data in it. This file is available in Appendix B on page 69 I made a copy of this file available to the SportsML group for their comments, which were supportive. There remained a small issue of embedding the weapon used for a particular competition in the format (something that is not taken care of in the core SportsML standard) but I'm confident that a solution can be found for that as the standard is easily extensible. Although it might seem counter-intuitive to reject ODF (especially as fencing in an Olympic sport), I find the SportsML standard easier to use and ease-of-use is important as people who wish to integrate with the platform will not necessarily be full-time professional developers. Additionally, British Fencing is not and never will be a primary supplier of Olympic fencing results so support for ODF is not a requirement.

5 Models

5.1 Grammatical Parse

After consultation with British Fencing, I was directed to a useful resource which describes how fencing competitions operate (British Fencing, 2010). Not all of this document is relevant but the sections that are are reproduced below, with **nouns** in bold and verbs underlined. Only the nouns and verbs deemed to be directly relevant to the process of competing in a competition have been included.

> Check In: All **competitions** start by **fencers** visiting the **check in desk** to confirm that they are present. Dont miss this bit out - your **entry** will be scratched. When checking in, **fencers** are required to show their **British Fencing card**. See the box (right) for details. This carries **insurance**. Without it, you may not fence. Fencing usually starts about 30 - 60 minutes after check in closes. Pools: After **check in**, **competitors** are divided into **pools** - groups of 5 - 7 **fencers** who all fence each other up to 5 **hits**. (4 **hits** for some under 9 **competitions**). Time is limited to 2 or 3 minutes. Sometimes there may be two rounds of **pools**, particularly in **age group competitions**. Direct Elimination: The **results** of the **pools** are used to seed the **knockout phase** of the **competition**. In some **competitions**, up to 30% of the **fencers** who did worst are eliminated, but in most cases all **fencers** go through to the **direct elimination (DE)** stage. The **DE** rewards **fencers** who do well in the **pool** stages, and keeps the strong **fencers** apart until near the end of the **competition**. In a **competition** with 64 **entrants**, the first round of DEs would see 1st place fence 64th, 2nd place fence 63rd and so on. If the number of **entrants** is not a power of 2, (ie 8, 16, 32, 64 etc) then those **fencers** who did best in the **pools** will get a **bye** through the first **DE** round. After several **DE** rounds, there will only be two **fencers** left - the **finalists**. **Direct elimination** fights are up to 15 **hits** (adults) 10 **hits** (under 13s) or 8 **hits** (under 9s). **DE fights** are normally 3 x 3 minutes (sometimes less for **younger fencers**) with a 60 second break between **periods**.

5.2 Project Glossary

From the grammatical parse in section 5.1 a project glossary can be compiled. As a part of this, I removed all duplicate words, plurals and synonyms, choosing the most appropriate word from any sets of synonyms as the noun or verb to represent that entity/action. Note that synonyms in this sense are not necessarily true synonyms but are equivalent terms in the context of fencing and fencing competitions. Objects that have been identified as being a synonym of another object, but not a complete synonym have had *(sub-type)* appended to them in the synonym list. The description of the words comes either from the text above, from basic internet searching, or from my own knowledge of the sport.

Table 5.1: Project Glossary

Nouns	
Competition	An over-arching event at which one or more events take place *synonyms: age group competition (sub-type)*
Fencer	An individual (human being) who competes in a fencing competition *synonyms: competitor, entrant, finalists (sub-type), younger fencers (sub-type)*
Check-in desk	The location that fencers present themselves to in order to confirm that they will take part in a competition
Entry	The link between a fencer and a competition
British Fencing Card	A form of identity card indicating that a fencer has membership of British Fencing, and by extension is insured to compete *synonyms: insurance*
Hit	The act of one fencer striking another and scoring a point
Result	A listing of fencers in order of their measured performance in a particular round
Direct Elimination	The phase of a competition where defeating another fencer results in their elimination from the competition and your promotion to the next round. *synonyms: DE, knockout phase*
Bye	The act of progressing through a round of the Direct Elimination stage of the competition without having to face another fencer. This is used in the event that the number of fencers is not a power of 2.
Bout	An individual fencing match between two fencers *synonyms: fight*
Period	A time-based subdivision of a bout
Poule	A small grouping of fencers who all fence against one another in a series of bouts. *synonyms: pool*
Event	A tournament in which fencers of the same gender, age group and weapon fence
Verbs	
Check In	The act of a fencer confirming that they have arrived at the venue of the competition and they intend to compete in it. *synonyms: to confirm (attendance at the competition)*
Be Scratched	To be removed from the competition without competing
Fence	The act of one fencer engaging in a fencing bout with another fencer *synonyms: fight*
Seed	To use the results of one or more round(s) of competition to determine the structure of the next round.
Eliminated	To be removed from the competition as a result of loosing a bout in a Direct Elimination round, or (in some cases) not being ranked high enough after a round of Poules
Go Through	To progress from one round to the next, either by being ranked high enough after a poules round, or by winning a Direct Elimination bout

Get a Bye	Be in receipt of free passage from one round of Direct Elimination to the next, without having to face another fencer. This is used in the case that the number of competitors in a round of Direct Elimination is not a power of 2

5.3 Business Processes

I have decided that for the purposes of this project, modelling the business processes associated with the process of a fencer taking part in a competition is not appropriate. The reason for this is that the software system will not seek to implement these processes, but rather be an archive or the results of these processes. Other software (such as Engarde and Fencing Time, mentioned in section 4 on page 6) already implement these processes. The business processes that I will model will be the process by which data gets added to the system and viewed by users of the system.

5.4 Use Case Diagram

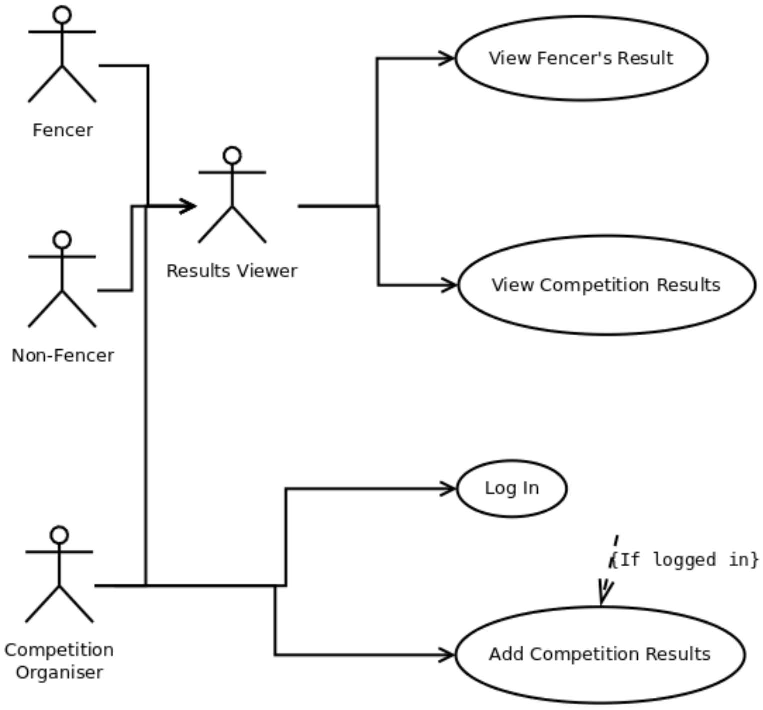


Figure 5.1: Use Case Diagram

5.5 Class Diagram

As mentioned above, the project glossary includes terms relating to the actual competition process. Some of these terms are relevant to a results hosting service and some are not.

Terms like *Check-in desk* and *British Fencing Card* are only relavent during the actual competition itself, therefore I have excluded them from the class diagram in 5.2. The

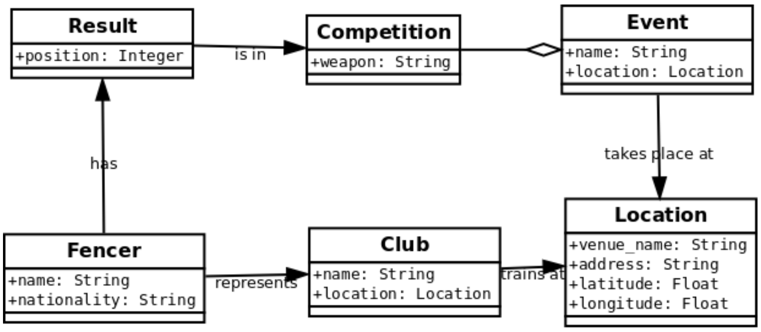


Figure 5.2: Class Diagram

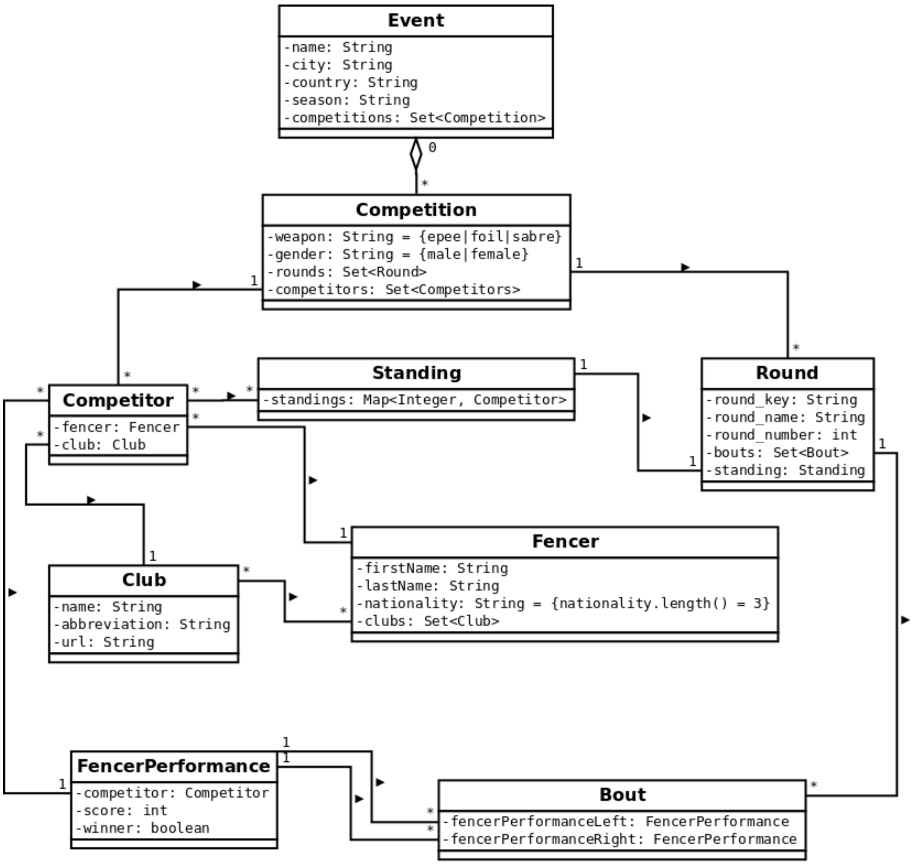


Figure 5.3: Full Class Diagram

class diagrams in 5.5 on page 13[1] and 5.6 on page 14[2] reflects what was actually coded in the *net.fencingarchive* and *sportsml* packages and were generated automatically (but re-arranged manually).

5.6 Database Design

[1] Also available at http://github.com/mattcarus/something/here.png

[2] Also available at http://github.com/mattcarus/something/here.png

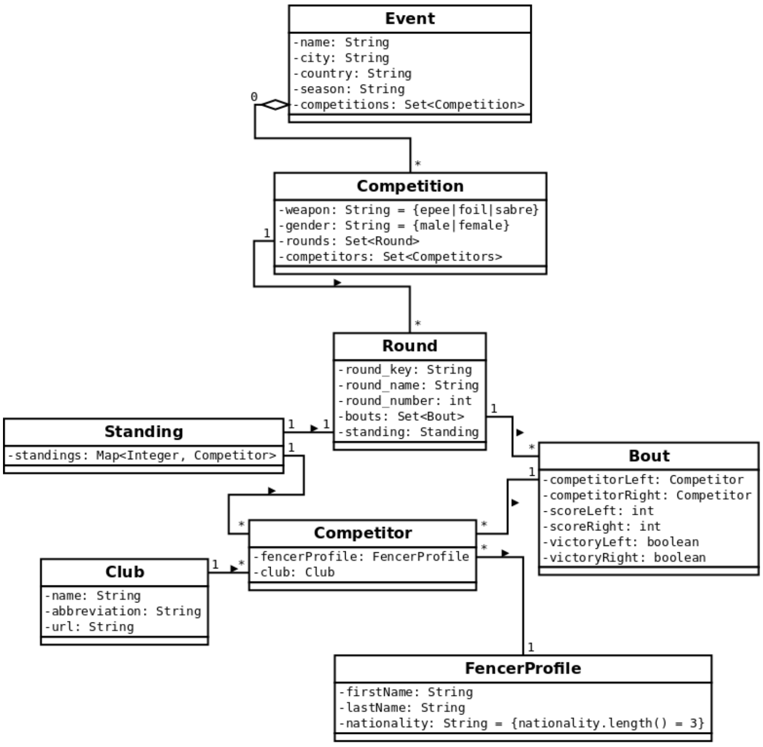


Figure 5.4: Reduced Class Diagram

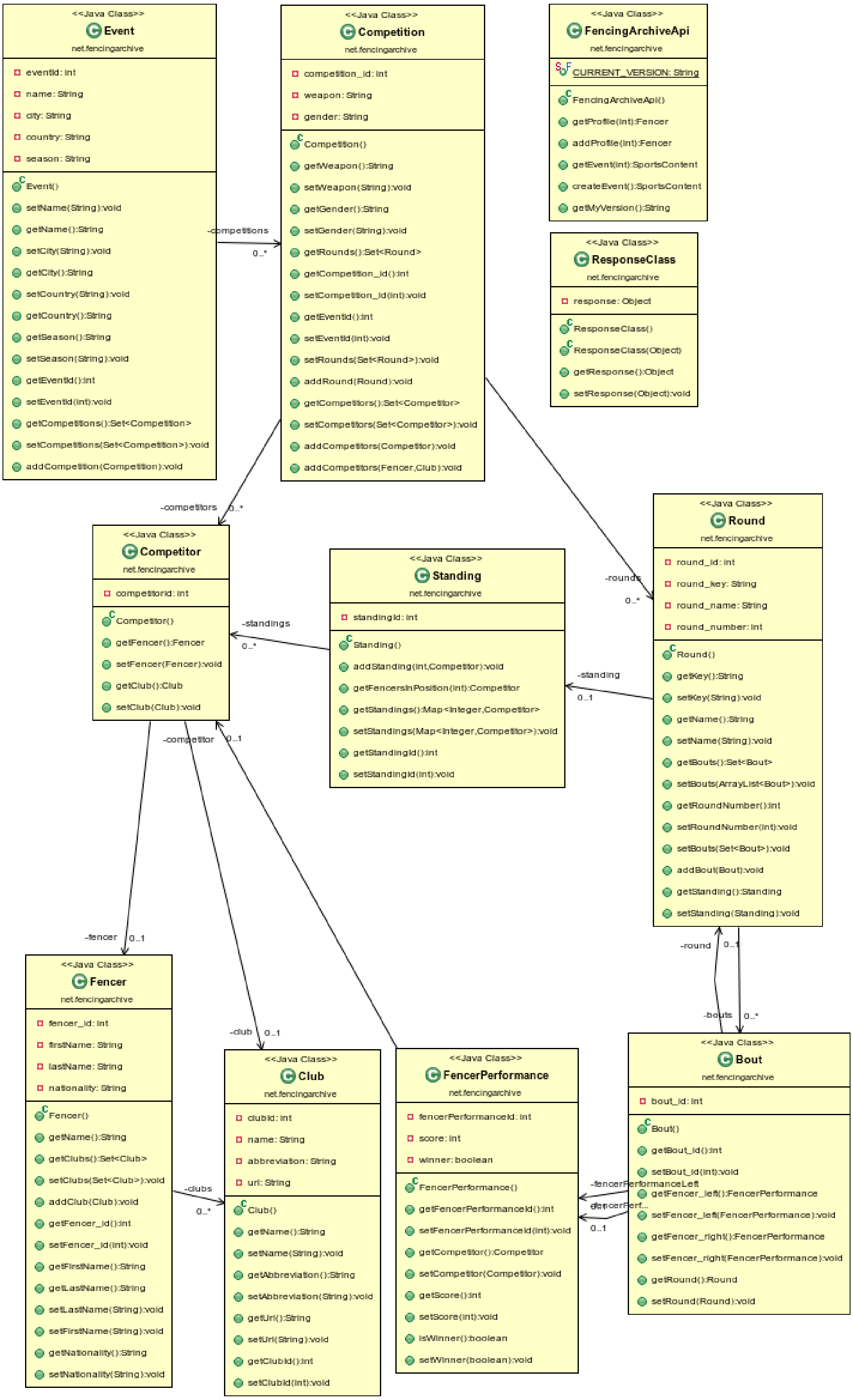


Figure 5.5: net.fencingarchive Class Diagram - Generated from source code

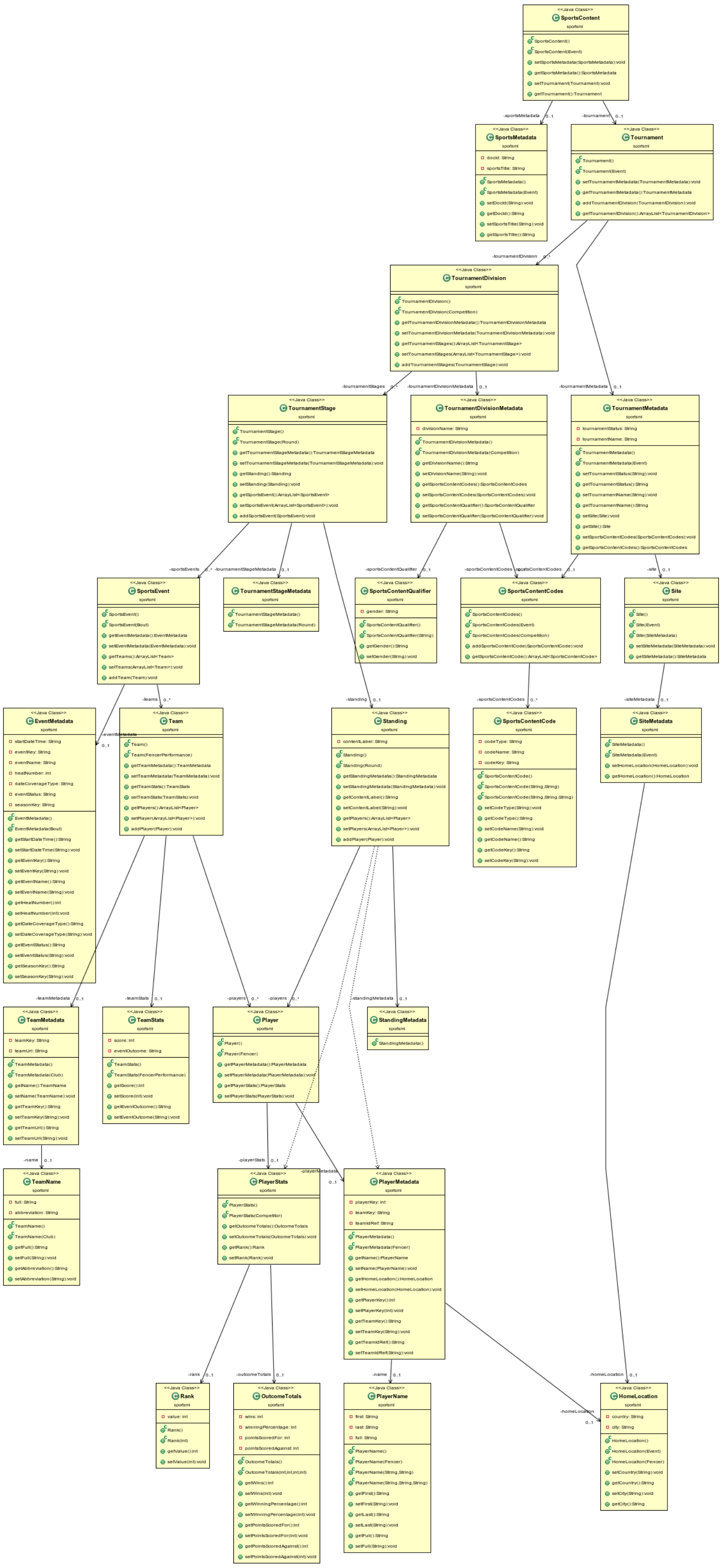


Figure 5.6: sportsml Class Diagram - Generated from source code

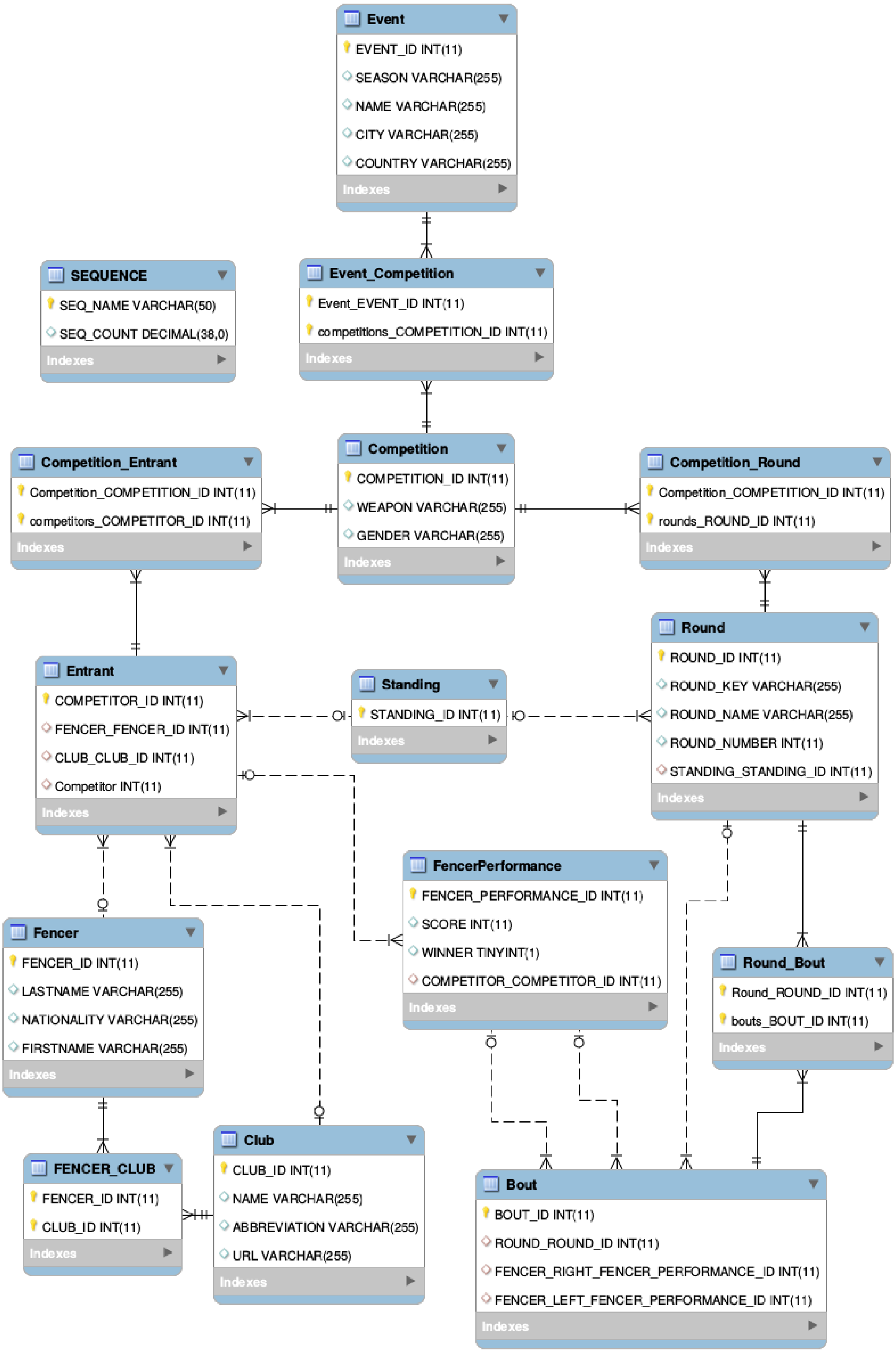


Figure 5.7: Database ER Model - as generated by MySQL Workbench

6 Design Decisions

6.1 SOAP vs. REST

6.2 Axis2 vs. Jersey

6.3 Development Environment

- Disabled EclipseLink caching during development

7 Problems Encountered

7.1 Tomcat Garbage Collection

I found out that if an application is re-deployed to Tomcat (for example after making some changes during development) the existing class instances remain in the Java 'PermGen' memory and are not garbage-collegted as you might expect. Over time this memory space will fill up and eventually a java.lang.OutOfMemoryError exception will be thrown. This had me confused for a while but after a bit of Googling I found that the easiest solution is simply to restart Tomcat. It is possible to increase the PermGen space however this will only delay the problem.

7.2 XML Element Ordering

The SportsML schema dictates that elements be ordered correctly - Jersey doesn't (and indeed can't) guarentee the order in which elements are rendered. To get around this it's possible to define the order by using the $@XmlType\{propOrder = \{\ldots\}\}$ structure.

7.3 Development Environment

- Disabled EclipseLink caching during development

8 *Source Code*

8.1 Project Setup Files

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <properties>
    <slf4jVersion>1.6.1</slf4jVersion>
    <jUnitVersion>4.11</jUnitVersion>
    <javadocVersion>2.10.3</javadocVersion>
  </properties>
  <modelVersion>4.0.0</modelVersion>
  <groupId>net.fencingarchive</groupId>
  <artifactId>FencingArchive</artifactId>
  <version>0.0.24-SNAPSHOT</version>
  <packaging>war</packaging>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-war-plugin</artifactId>
        <version>2.6</version>
        <configuration>
          <webXml>web\WEB-INF\web.xml</webXml>
          <archive>
            <manifest>
              <addDefaultImplementationEntries>true</addDefaultImplementationEntries>
              <addDefaultSpecificationEntries>true</addDefaultSpecificationEntries>
            </manifest>
          </archive>
          <archiveClasses>true</archiveClasses>
          <webResources>
            <resource>
              <!-- this is relative to the pom.xml directory -->
              <directory>static</directory>
            </resource>
          </webResources>
        </configuration>
      </plugin>
      <plugin>
        <groupId>org.apache.tomcat.maven</groupId>
        <artifactId>tomcat7-maven-plugin</artifactId>
        <version>2.2</version>
        <configuration>

          <!-- Enable this section for local deployment -->

          <url>http://localhost:8080/manager/text</url>
          <username>autodeploy</username>
          <password>autodeploy-password</password>


          <!-- Enable this section for Azure deployment -->
          <!-- <url>http://fencingarchive.westus.cloudapp.azure.com/manager/text</url>
            <username>fencingarchive</username> <password>Suwr1sp1</password> -->
        </configuration>
      </plugin>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-release-plugin</artifactId>
        <version>2.5.3</version>
      </plugin>
    </plugins>
  </build>
  <dependencies>
    <dependency>
      <groupId>com.sun.jersey</groupId>
      <artifactId>jersey-server</artifactId>
      <version>1.17</version>
    </dependency>
    <dependency>
      <groupId>com.sun.jersey</groupId>
      <artifactId>jersey-json</artifactId>
      <version>1.17</version>
    </dependency>
    <dependency>
      <groupId>com.sun.jersey</groupId>
      <artifactId>jersey-client</artifactId>
      <version>1.17</version>
    </dependency>
    <dependency>
      <groupId>com.sun.jersey</groupId>
      <artifactId>jersey-servlet</artifactId>
      <version>1.17</version>
    </dependency>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>${jUnitVersion}</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-api</artifactId>
      <version>${slf4jVersion}</version>
    </dependency>
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-simple</artifactId>
      <version>${slf4jVersion}</version>
    </dependency>
    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>4.3.5.Final</version>
    </dependency>
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>5.1.6</version>
    </dependency>
    <dependency>
      <groupId>org.eclipse.persistence</groupId>
      <artifactId>javax.persistence</artifactId>
      <version>2.0.0</version>
      <scope>compile</scope>
    </dependency>
    <dependency>
      <groupId>org.eclipse.persistence</groupId>
      <artifactId>eclipselink</artifactId>
      <version>2.0.0</version>
      <scope>compile</scope>
    </dependency>
  </dependencies>
  <repositories>
    <repository>
      <id>mvnrepository</id>
      <name>MVN Repository</name>
      <url>http://mvnrepository.com/</url>
    </repository>
    <repository>
      <id>EclipseLink Repo</id>
      <url>http://www.eclipse.org/downloads/download.php?r=1&amp;nf=1&amp;file=/rt/eclipselink/maven.repo</url>
    </repository>
  </repositories>
  <reporting>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-javadoc-plugin</artifactId>
        <version>2.10.3</version>
        <configuration>
          <bootclasspath>${java.home}/lib/rt.jar</bootclasspath>
          <reportOutputDirectory>${project.reporting.outputDirectory}</reportOutputDirectory>
          <destDir>javadoc</destDir>
          <quiet>true</quiet>
        </configuration>
      </plugin>
    </plugins>
  </reporting>
</project>
```

Listing 8.1: pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
      http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  version="2.5">

  <servlet>
    <servlet-name>FencingArchiveAPI</servlet-name>
    <servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer</servlet-class>
    <init-param>
      <param-name>com.sun.jersey.config.property.packages</param-name>
      <param-value>net.fencingarchive</param-value>
    </init-param>
    <init-param>
      <param-name>com.sun.jersey.api.json.POJOMappingFeature</param-name>
      <param-value>true</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>FencingArchiveAPI</servlet-name>
    <!-- This defines where the API will be hosted.
     This can't be set to /* because then the static resources (XSDs, JavaDocs etc)
     won't be available. -->
    <url-pattern>/api/*</url-pattern>
  </servlet-mapping>
</web-app>
```

Listing 8.2: web.xml

8.2 Package: net.fencingarchive

```
package net.fencingarchive;


import javax.ws.rs.Consumes;
import javax.ws.rs.GET;
import javax.ws.rs.POST;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;

import net.fencingarchive.db.Database;
import sportsml.SportsContent;

// import org.codehaus.jettison.json.JSONException;
// import org.codehaus.jettison.json.JSONObject;

/**
 * FencingArchive API Service Class
 *
 * By default this will be deployed to http://127.0.0.1:8080/FencingArchive/ meaning that the API
 * will be accessible at http://127.0.0.1:8080/FencingArchive/api/fencingarchive/
 *
 * @author matt
 *
 */
```

```java
// Path is left empty because it's not needed and will just lengthen an already-quite-long URL.
@Path("")
public class FencingArchiveApi {
  public final static String CURRENT_VERSION = FencingArchiveApi.class.getPackage()
      .getImplementationVersion();

  /**
   * Access the profile of a fencer, e.g. http://127.0.0.1:8080/FencingArchive/api/fencer/123
   *
   * @param id The ID of the fencer
   * @return Fencer object
   */
  @Path("/fencer/{id}")
  // @XmlHeader("<?xml-stylesheet type=\"text/xsl\" href=\"sportsml-html.xsl\"?>")
  @GET
  @Produces(MediaType.APPLICATION_XML)
  public Fencer getProfile(@PathParam("id") int id) {
    Fencer fencer = new Fencer();
    fencer.setFirstName("Fencer " + id);
    return fencer;
  }

  /**
   * Placeholder for adding fencer
   *
   * @param id The ID of the fencer
   * @return Fencer object
   */
  @Path("/fencer/{id}")
  @POST
  @Consumes(MediaType.APPLICATION_XML)
  @Produces(MediaType.APPLICATION_XML)
  public Fencer addProfile(@PathParam("id") int id) {
    Fencer fencer = new Fencer();
    fencer.setFirstName("Fencer " + id);
    return fencer;
  }

  /**
   * Access the details of a competition, e.g.
   * http://127.0.0.1:8080/FencingArchive/api/competition/456
   *
   * @param id The ID of the competition
   * @return Competition object
   */
  @Path("/event/{id}")
  /*
   * The following line sets up an XML Stylesheet - this is how the data is rendered in
   * human-readable format but useful to switch off during dev.
   *
   * TODO: For some reason enabling this knocks off namespaces :/
   */
  // @XmlHeader("<?xml-stylesheet type=\"text/xsl\" href=\"http://127.0.0.1:8080/FencingArchive/sportsml-html.xsl\"?>")
  @GET
  @Produces(MediaType.APPLICATION_XML)
  public SportsContent getEvent(@PathParam("id") int id) {
    // Start a database connection
    Database db = new Database();

    Event event = (Event) db.readObject(Event.class, id);

    SportsContent sportsContent = new SportsContent(event);
    return sportsContent;
  }

  /**
   * Access the details of a competition, e.g.
   * http://127.0.0.1:8080/FencingArchive/api/competition/456
   *
   * @param id The ID of the competition
   * @return Competition object
   * @throws Exception
   */
  @Path("/event/create")
  /*
   * The following line sets up an XML Stylesheet - this is how the data is rendered in
   * human-readable format but useful to switch off during dev.
   *
   * TODO: For some reason enabling this knocks off namespaces :/
   */
  // @XmlHeader("<?xml-stylesheet type=\"text/xsl\" href=\"http://127.0.0.1:8080/FencingArchive/sportsml-html.xsl\"?>")
  @GET
  @Produces(MediaType.APPLICATION_XML)
  public SportsContent createEvent() throws Exception {
    // Start a database connection
    Database db = new Database();

    // Create Base Event
    Event event = new Event();
    event.setName("Test Open");
    event.setCity("London");
    event.setCountry("UK");
    event.setSeason("2015-16");

    // Create Competitions and Rounds and add them to the event
    Competition me = new Competition();
    me.setWeapon("epee");
    me.setGender("male");

    Round pouleRound = new Round();
    pouleRound.setKey("test");
    pouleRound.setName("Poules Round 1");
    pouleRound.setRoundNumber(1);

    Fencer bob = new Fencer();
    bob.setFirstName("Bob");
    bob.setLastName("Harris");
```

```
        bob.setNationality(”GBR”);

        Fencer fred = new Fencer();
        fred.setFirstName(”Fred”);
        fred.setLastName(”Dibnah”);
        fred.setNationality(”NED”);

        Fencer george = new Fencer();
        george.setFirstName(”George”);
        george.setLastName(”Smith”);
        george.setNationality(”USA”);

        Club kes = new Club();
        kes.setAbbreviation(”KES”);
        kes.setName(”King Edward VI Grammar School”);
        kes.setUrl(”http://kes.sch”);

        Club shakespeares = new Club();
        shakespeares.setAbbreviation(”SHKS”);
        shakespeares.setName(”Shakespeare’s Swords”);
        shakespeares.setUrl(”http://shakespeareswords.org”);

        bob.addClub(kes);
        bob.addClub(shakespeares);
        fred.addClub(kes);
        george.addClub(shakespeares);

        Competitor competitorBob = new Competitor();
        competitorBob.setFencer(bob);
        competitorBob.setClub(kes);

        Competitor competitorFred = new Competitor();
        competitorFred.setFencer(fred);
        competitorFred.setClub(kes);

        Competitor competitorGeorge = new Competitor();
        competitorGeorge.setFencer(george);
        competitorGeorge.setClub(shakespeares);

        me.addCompetitors(competitorBob);
        me.addCompetitors(competitorFred);
        me.addCompetitors(competitorGeorge);

        FencerPerformance bobPerf = new FencerPerformance();
        bobPerf.setCompetitor(competitorBob);
        bobPerf.setScore(15);
        bobPerf.setWinner(true);

        FencerPerformance fredPerf = new FencerPerformance();
        fredPerf.setCompetitor(competitorFred);
        fredPerf.setScore(10);
        fredPerf.setWinner(false);

        Bout bout1 = new Bout();
        bout1.setFencer_left(bobPerf);
        bout1.setFencer_right(fredPerf);
        bout1.setRound(pouleRound);
        /*
```

```
     * FencerPerformance bobPerf2 = new FencerPerformance(); bobPerf2.setFencer(bob);
     * bobPerf2.setScore(15); bobPerf2.setWinner(false); bobPerf2.setClub(kes);
     */
    FencerPerformance georgePerf = new FencerPerformance();
    georgePerf.setCompetitor(competitorGeorge);
    georgePerf.setScore(15);
    georgePerf.setWinner(true);

    Bout bout2 = new Bout();
    bout2.setFencer_left(bobPerf);
    bout2.setFencer_right(georgePerf);
    bout2.setRound(pouleRound);

    pouleRound.addBout(bout1);
    pouleRound.addBout(bout2);

    Standing pouleRoundStanding = new Standing();
    pouleRoundStanding.addStanding(1, competitorBob);
    pouleRoundStanding.addStanding(2, competitorFred);
    pouleRoundStanding.addStanding(3, competitorGeorge);

    pouleRound.setStanding(pouleRoundStanding);

    me.addRound(pouleRound);
    event.addCompetition(me);

    // Round round2 = new Round();
    // round2.setKey("test-me-poule-2");
    // round2.setName("Poules Round 2");
    // round2.setRoundNumber(2);
    // me.addRound(round2);

    // Write the competition to the database
    // db.writeObject(Competition.class, me);
    db.writeObject(Event.class, event);
    // db.writeObject(Bout.class, bout2);
    // db.writeObject(Competition.class, me);

    // Write the event to the database
    // db.writeObject(Event.class, event);

    // Generate SportsML
    SportsContent sportsContent = new SportsContent(event);
    return sportsContent;
  }

  /**
   * Get version information
   *
   * @return String object
   */
  @Path("/version")
  @GET
  @Produces(MediaType.APPLICATION_XML)
  public String getMyVersion() {
    return new String("test");
```

```java
    // return FencingArchiveApi.CURRENT_VERSION;
  }
}
```

Listing 8.3: net.fencingarchive.FencingArchiveApi()

```java
package net.fencingarchive;

import java.io.Serializable;
import java.util.HashSet;
import java.util.Set;

import javax.persistence.Basic;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.JoinTable;
import javax.persistence.ManyToMany;
import javax.persistence.Table;
import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;

@Entity
@Table(name = "Fencer")
@XmlRootElement
public class Fencer implements Serializable {
  /**
   * Unique database ID for this object
   */
  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  @Column(name = "FENCER_ID")
  private int fencer_id;

  @Basic(optional = false)
  private String firstName;

  @Basic(optional = false)
  private String lastName;

  @Basic(optional = false)
  private String nationality;

  @ManyToMany(cascade = CascadeType.PERSIST)
  @JoinTable(name = "FENCER_CLUB", joinColumns = @JoinColumn(name = "FENCER_ID",
      referencedColumnName = "FENCER_ID"), inverseJoinColumns = @JoinColumn(name = "CLUB_ID",
      referencedColumnName = "CLUB_ID"))
  private Set<Club> clubs;

  public Fencer() {
    this.clubs = new HashSet<Club>();
  }

  @XmlElement(name = ”name”)
  public String getName() {
    StringBuilder name = new StringBuilder();
    name.append(this.getFirstName());
    name.append(” ”);
    name.append(this.getLastName());
    return name.toString();
  }

  /**
   * @return the clubs
   */
  public Set<Club> getClubs() {
    return clubs;
  }

  /**
   * @param clubs the clubs to set
   */
  public void setClubs(Set<Club> clubs) {
    this.clubs = clubs;
  }

  /**
   * @param clubs the clubs to set
   */
  public void addClub(Club club) {
    this.clubs.add(club);
  }

  /**
   * @return the fencer_id
   */
  public int getFencer_id() {
    return fencer_id;
  }

  /**
   * @param fencer_id the fencer_id to set
   */
  public void setFencer_id(int fencer_id) {
    this.fencer_id = fencer_id;
  }

  public String getFirstName() {
    return this.firstName;
  }

  /**
   * @return the lastName
   */
  public String getLastName() {
    return this.lastName;
  }

  /**
   * @param lastName the lastName to set
   */
  public void setLastName(String lastName) {
    this.lastName = lastName;
  }

  /**
   * @param firstName the firstName to set
   */
  public void setFirstName(String firstName) {
    this.firstName = firstName;
  }

  /**
   * @return the nationality
   */
  public String getNationality() {
    return nationality;
  }

  /**
   * @param nationality the nationality to set
   * @throws Exception
   */
  public void setNationality(String nationality) throws Exception {
    // Constrain this field to 3 characters and force upper case.
    if (nationality.length() > 3) {
      throw new Exception("Nationality must be 3 characters or fewer");
    }
    this.nationality = nationality.toUpperCase();
  }

}
```

Listing 8.4: net.fencingarchive.Fencer()

```java
package net.fencingarchive;

import java.io.Serializable;

import javax.persistence.Basic;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.ManyToOne;
import javax.persistence.Table;

/**
 * This class represents a single performance of a fencer, i.e. their participation in a bout.
 *
 * @author matt
 *
 */
@Entity
@Table(name = "FencerPerformance")
```

```
public class FencerPerformance implements Serializable {
  /**
   * Unique database ID for this object
   */
  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  @Column(name = ”FENCER_PERFORMANCE_ID”, nullable = false)
  private int fencerPerformanceId;

  @ManyToOne(cascade = CascadeType.PERSIST)
  private Competitor competitor;

  @Basic(optional = false)
  private int score;

  @Basic(optional = false)
  private boolean winner;

  public FencerPerformance() {

  }

  /**
   * @return the fencerPerformanceId
   */
  public int getFencerPerformanceId() {
    return fencerPerformanceId;
  }

  /**
   * @param fencerPerformanceId the fencerPerformanceId to set
   */
  public void setFencerPerformanceId(int fencerPerformanceId) {
    this.fencerPerformanceId = fencerPerformanceId;
  }

  /**
   * @return the fencer
   */
  public Competitor getCompetitor() {
    return this.competitor;
  }

  /**
   * @param fencer the fencer to set
   */
  public void setCompetitor(Competitor competitor) {
    this.competitor = competitor;
  }

  /**
   * @return the score
   */
  public int getScore() {
    return score;
  }

  /**
```

```
   * @param score the score to set
   */
  public void setScore(int score) {
    this.score = score;
  }

  /**
   * @return the winner
   */
  public boolean isWinner() {
    return winner;
  }

  /**
   * @param winner the winner to set
   */
  public void setWinner(boolean winner) {
    this.winner = winner;
  }

}
```

Listing 8.5: net.fencingarchive.FencerPerformance()

```
package net.fencingarchive;

import java.io.Serializable;

import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "Club")
public class Club implements Serializable {
  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  @Column(name = "CLUB_ID")
  private int clubId;

  @Basic(optional = false)
  private String name;

  @Basic(optional = false)
  private String abbreviation;

  @Basic(optional = true)
  private String url;

  public Club() {

  }

  /**
```

```
   * @return the name
   */
  public String getName() {
    return name;
  }

  /**
   * @param name the name to set
   */
  public void setName(String name) {
    this.name = name;
  }

  /**
   * @return the abbreviation
   */
  public String getAbbreviation() {
    return abbreviation;
  }

  /**
   * @param abbreviation the abbreviation to set
   */
  public void setAbbreviation(String abbreviation) {
    this.abbreviation = abbreviation;
  }

  /**
   * @return the url
   */
  public String getUrl() {
    return url;
  }

  /**
   * @param url the url to set
   */
  public void setUrl(String url) {
    this.url = url;
  }

  /**
   * @return the clubId
   */
  public int getClubId() {
    return clubId;
  }

  /**
   * @param clubId the clubId to set
   */
  public void setClubId(int clubId) {
    this.clubId = clubId;
  }
}
```

Listing 8.6: net.fencingarchive.Club()

```java
package net.fencingarchive;

import java.io.Serializable;
import java.util.HashSet;
import java.util.Set;

import javax.persistence.Basic;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity
@Table(name = "Event")
public class Event implements Serializable {
  /**
   * Unique database ID for this object
   */
  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  @Column(name = "EVENT_ID", nullable = false)
  private int eventId;

  @Basic(optional = false)
  private String name;

  @Basic(optional = true)
  private String city;

  @Basic(optional = true)
  private String country;

  @Basic(optional = true)
  private String season;

  @OneToMany(cascade = CascadeType.PERSIST)
  private Set<Competition> competitions;

  public Event() {
    this.competitions = new HashSet<Competition>();
  }

  public void setName(String newName) {
    this.name = newName;
  }

  public String getName() {
    return this.name;
  }

  public void setCity(String newCity) {
    this.city = newCity;
  }

  public String getCity() {
    return this.city;
  }

  public void setCountry(String newCountry) {
    this.country = newCountry;
  }

  public String getCountry() {
    return this.country;
  }

  /*
   * public void addCompetition(Competition newCompetition) {
   * newCompetition.setEventId(this.getEventId()); this.competition.add(newCompetition);
   *
   * }
   *
   * public ArrayList<Competition> getCompetition() { return this.competition; }
   */
  /**
   * @return the season
   */
  public String getSeason() {
    return season;
  }

  /**
   * @param season the season to set
   */
  public void setSeason(String season) {
    this.season = season;
  }

  /**
   * @return the event_id
   */
  public int getEventId() {
    return eventId;
  }

  /**
   * @param event_id the event_id to set
   */
  public void setEventId(int eventId) {
    this.eventId = eventId;
  }

  /**
   * @return the competitions
   */
  public Set<Competition> getCompetitions() {
    return competitions;
  }

  /**
```

```
   * @param competitions the competitions to set
   */
  public void setCompetitions(Set<Competition> competitions) {
    this.competitions = competitions;
  }

  /**
   * @param competitions the competitions to set
   */
  public void addCompetition(Competition competition) {
    this.competitions.add(competition);
  }

}
```

Listing 8.7: net.fencingarchive.Event()

```
package net.fencingarchive;

import java.io.Serializable;
import java.util.HashSet;
import java.util.Set;

import javax.persistence.Basic;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity
@Table(name = "Competition")
public class Competition implements Serializable {
  /**
   * Unique database ID for this object
   */
  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  @Column(name = "COMPETITION_ID", nullable = false)
  private int competition_id;

  @Basic(optional = false)
  private String weapon;

  @Basic(optional = false)
  private String gender;

  @OneToMany(cascade = CascadeType.PERSIST)
  private Set<Round> rounds;

  private Set<Competitor> competitors;

  public Competition() {
    this.rounds = new HashSet<Round>();
    this.competitors = new HashSet<Competitor>();
  }

  /**
   * @return the weapon
   */
  public String getWeapon() {
    return weapon;
  }

  /**
   * @param weapon the weapon to set
   * @throws Exception
   */
  public void setWeapon(String weapon) throws Exception {
    weapon = weapon.toLowerCase();
    if (weapon.equals("epee") || weapon.equals("foil") || weapon.equals("sabre")) {
      this.weapon = weapon;
    } else {
      throw new Exception("weapon must be one of epee, foil, or sabre");
    }
  }

  /**
   * @return the gender
   */
  public String getGender() {
    return gender;
  }

  /**
   * @param gender the gender to set
   */
  public void setGender(String gender) {
    this.gender = gender;
  }

  /**
   * @return the rounds
   */
  public Set<Round> getRounds() {
    return this.rounds;
  }

  /**
   * @return the competition_id
   */
  public int getCompetition_id() {
    return competition_id;
  }

  /**
   * @param competition_id the competition_id to set
   */
  public void setCompetition_id(int competition_id) {
    this.competition_id = competition_id;
  }

  /**
   * @return the eventId
   */
  public int getEventId() {
    // return eventId;
    return 0;
  }

  /**
   * @param eventId the eventId to set
   */
  public void setEventId(int eventId) {
    // this.eventId = eventId;
  }

  /**
   * @param rounds the rounds to set
   */
  public void setRounds(Set<Round> rounds) {
    this.rounds = rounds;
  }

  /**
   * @param rounds the rounds to set
   */
  public void addRound(Round round) {
    this.rounds.add(round);
  }

  /**
   * @return the competitors
   */
  public Set<Competitor> getCompetitors() {
    return competitors;
  }

  /**
   * @param competitors the competitors to set
   */
  public void setCompetitors(Set<Competitor> competitors) {
    this.competitors = competitors;
  }

  /**
   * @param competitors the competitors to set
   */
  public void addCompetitors(Competitor competitor) {
    this.competitors.add(competitor);
  }

  /**
   * @param competitors the competitors to set
   */
  public void addCompetitors(Fencer fencer, Club club) {
    Competitor competitor = new Competitor();
    competitor.setFencer(fencer);
    competitor.setClub(club);
    this.competitors.add(competitor);
  }

}
```

Listing 8.8: net.fencingarchive.Competition()

```
package net.fencingarchive;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.ManyToOne;
import javax.persistence.Table;

/**
 * This class represents an individual's entry into a competition, along with the club that they
 * represented at that competition. This is needed because although the Fencer object is aware of
 * the clubs that a fencer is a member of, it doesn't know about which club a fencer was
 * representing when they participated in individual competitions/events.
 *
 * @author matt
 *
 */
@Entity
@Table(name = "Entrant")
public class Competitor {
  /**
   * Unique database ID for this object
   */
  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  @Column(name = "COMPETITOR_ID", nullable = false)
  private int competitorId;

  @ManyToOne(cascade = CascadeType.PERSIST)
  private Fencer fencer;

  @ManyToOne(cascade = CascadeType.PERSIST)
  private Club club;

  public Competitor() {

  }

  /**
   * @return the fencer
   */
  public Fencer getFencer() {
    return fencer;
  }

  /**
```

```java
   * @param fencer the fencer to set
   */
  public void setFencer(Fencer fencer) {
    this.fencer = fencer;
  }

  /**
   * @return the club
   */
  public Club getClub() {
    return club;
  }

  /**
   * @param club the club to set
   */
  public void setClub(Club club) {
    this.club = club;
  }

}
```

Listing 8.9: net.fencingarchive.Competitor()

```java
package net.fencingarchive;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.Set;

import javax.persistence.Basic;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.OneToOne;
import javax.persistence.Table;

/**
 * @author matt
 *
 */
@Entity
@Table(name = ”Round”)
public class Round implements Serializable {
  /**
   * Unique database ID for this object
   */
  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  @Column(name = ”ROUND_ID”, nullable = false)
  private int round_id;

  @Basic(optional = true)
  private String round_key;

  @Basic(optional = false)
  private String round_name;

  @Basic(optional = false)
  private int round_number;

  @OneToMany(cascade = CascadeType.PERSIST)
  private Set<Bout> bouts;

  @OneToOne(cascade = CascadeType.PERSIST)
  private Standing standing;

  public Round() {
    this.bouts = new HashSet<Bout>();
  }

  /**
   * @return the key
   */
  public String getKey() {
    return this.round_key;
  }

  /**
   * @param key the key to set
   */
  public void setKey(String key) {
    this.round_key = key;
  }

  /**
   * @return the name
   */
  public String getName() {
    return this.round_name;
  }

  /**
   * @param name the name to set
   */
  public void setName(String name) {
    this.round_name = name;
  }

  /**
   * @return the bouts
   */
  public Set<Bout> getBouts() {
    return this.bouts;
  }

  /**
   * @param bouts the bouts to set
   */
  public void setBouts(ArrayList<Bout> bouts) {
    // this.bouts = bouts;
  }

  /**
   * @return the roundNumber
   */
  public int getRoundNumber() {
    return this.round_number;
  }

  /**
   * @param roundNumber the roundNumber to set
   */
  public void setRoundNumber(int round_number) {
    this.round_number = round_number;
  }

  /**
   * @param bouts the bouts to set
   */
  public void setBouts(Set<Bout> bouts) {
    this.bouts = bouts;
  }

  /**
   * @param bouts the bouts to set
   */
  public void addBout(Bout bout) {
    this.bouts.add(bout);
  }

  /**
   * @return the standing
   */
  public Standing getStanding() {
    return standing;
  }

  /**
   * @param standing the standing to set
   */
  public void setStanding(Standing standing) {
    this.standing = standing;
  }

}
```

Listing 8.10: net.fencingarchive.Round()

```java
package net.fencingarchive;

import java.io.Serializable;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.ManyToOne;
import javax.persistence.Table;

/**
 * Representation of an individual bout. Note that the concept of a 'left' and 'right' fencer is a
 * real-world concept (i.e. one fencer will be on the left hand side of the piste and the other will
 * be on the right).
 *
 * TODO: It feels like defining a 'right' and 'left' fencer here is kinda wrong - shouldn't they
 * just be objects in a collection (of max size 2)?
 *
 * @author matt
 *
 */
@Entity
@Table(name = "Bout")
public class Bout implements Serializable {

  /**
   * Unique database ID for this object
   */
  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  @Column(name = "BOUT_ID", nullable = false)
  private int bout_id;

  /**
   * Fencer object representing the fencer on the left
   */
  @ManyToOne(cascade = CascadeType.PERSIST)
  private FencerPerformance fencerPerformanceLeft;

  /**
   * Fencer object representing the fencer on the right
   */
  @ManyToOne(cascade = CascadeType.PERSIST)
  private FencerPerformance fencerPerformanceRight;

  /**
   * Round within a competition that this bout is a member of
   */
  @ManyToOne(cascade = CascadeType.PERSIST)
  private Round round;

  /**
   *
   */
  public Bout() {

  }

  /**
   * @return the bout_id
   */
  public int getBout_id() {
    return bout_id;
  }

  /**
   * @param bout_id the bout_id to set
   */
  public void setBout_id(int bout_id) {
    this.bout_id = bout_id;
  }

  /**
   * @return the fencer_left
   */
  public FencerPerformance getFencer_left() {
    return fencerPerformanceLeft;
  }

  /**
   * @param fencer_left the fencer_left to set
   */
  public void setFencer_left(FencerPerformance fencer_left) {
    this.fencerPerformanceLeft = fencer_left;
  }

  /**
   * @return the fencer_right
   */
  public FencerPerformance getFencer_right() {
    return fencerPerformanceRight;
  }

  /**
   * @param fencer_right the fencer_right to set
   */
  public void setFencer_right(FencerPerformance fencer_right) {
    this.fencerPerformanceRight = fencer_right;
  }

  /**
   * @return the round
   */
  public Round getRound() {
    return round;
  }

  /**
   * @param round the round to set
   */
  public void setRound(Round round) {
    this.round = round;
  }
}
```

Listing 8.11: net.fencingarchive.Bout()

```
package net.fencingarchive;

import java.io.Serializable;
```

```java
import java.util.HashMap;
import java.util.Map;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.MapKey;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity
@Table(name = "Standing")
public class Standing implements Serializable {

  /**
   * Unique database ID for this object
   */
  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  @Column(name = "STANDING_ID", nullable = false)
  private int standingId;

  // TODO: Make this capable of holding duplicate positions
  // @ElementCollection
  // @MapKeyColumn(name = "Position")
  // @Column(name = "Fencer")
  // @CollectionTable(name = "Fencer_Standing_Map", joinColumns = @JoinColumn(name = "STANDING_ID"))
  @OneToMany
  @JoinColumn(name = "Competitor")
  @MapKey(name = "competitorId")
  private Map<Integer, Competitor> standings;

  public Standing() {
    this.standings = new HashMap<Integer, Competitor>();
  }

  public void addStanding(int position, Competitor competitor) {
    this.standings.put(position, competitor);
  }

  public Competitor getFencersInPosition(int position) {
    return this.getStandings().get(position);
  }

  /**
   * @return the standings
   */
  public Map<Integer, Competitor> getStandings() {
    return standings;
  }

  /**
   * @param standings the standings to set
   */
  public void setStandings(Map<Integer, Competitor> standings) {
    this.standings = standings;
  }

  /**
   * @return the standingId
   */
  public int getStandingId() {
    return standingId;
  }

  /**
   * @param standingId the standingId to set
   */
  public void setStandingId(int standingId) {
    this.standingId = standingId;
  }
}
```

Listing 8.12: net.fencingarchive.Standing()

8.3 Package: sportsml

```
package sportsml;

import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlType;

import net.fencingarchive.Event;


@XmlRootElement(name = "sports-content")
@XmlType(propOrder = {"sportsMetadata", "tournament"})
public class SportsContent {
  private SportsMetadata sportsMetadata;
  private Tournament tournament;

  public SportsContent() {

  }

  public SportsContent(Event event) {
    /*
     * SportsMetadata sportsMetadata = new SportsMetadata();
     * sportsMetadata.setDocId(competition.getName().toLowerCase().replace
    (" ", "-"));
     * sportsMetadata.setSportsTitle(competition.getName());
     *
     * TournamentMetadata tournamentMetadata = new TournamentMetadata();
     *
     * Tournament tournament = new Tournament(tournamentMetadata);
     *
     *
     * this.setSportsMetadata(sportsMetadata); this.setTournament(
    tournament);
```

```
     */
    this.sportsMetadata = new SportsMetadata(event);
    this.tournament = new Tournament(event);
  }

  public void setSportsMetadata(SportsMetadata newSportsMetadata) {
    this.sportsMetadata = newSportsMetadata;
  }

  @XmlElement(name = "sports-metadata")
  public SportsMetadata getSportsMetadata() {
    return this.sportsMetadata;
  }

  public void setTournament(Tournament newTournament) {
    this.tournament = newTournament;
  }

  @XmlElement(name = "tournament")
  public Tournament getTournament() {
    return this.tournament;
  }
}
```

Listing 8.13: sportsml.SportsContent()

```
package sportsml;

import javax.xml.bind.annotation.XmlAttribute;
import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlType;

import net.fencingarchive.Event;


@XmlRootElement(name = "sports-metadata")
@XmlType(propOrder = {"sportsTitle"})
public class SportsMetadata {
  private String docId;
  private String sportsTitle;

  public SportsMetadata() {

  }

  public SportsMetadata(Event event) {
    this.setDocId(event.getName().toLowerCase().replace(" ", "-"));
    this.setSportsTitle(event.getName());
  }

  public void setDocId(String newDocId) {
    this.docId = newDocId;
  }

  @XmlAttribute(name = "doc-id")
  public String getDocId() {
    return this.docId;
```

```java
  }

  public void setSportsTitle(String newSportsTitle) {
    this.sportsTitle = newSportsTitle;
  }

  @XmlElement(name = "sports-title")
  public String getSportsTitle() {
    return this.sportsTitle;
  }
}
```

Listing 8.14: sportsml.SportsMetadata()

```java
package sportsml;

import java.util.ArrayList;

import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlType;

import net.fencingarchive.Competition;
import net.fencingarchive.Event;


@XmlRootElement(name = "tournament")
@XmlType(propOrder = {"tournamentMetadata", "tournamentDivision"})
public class Tournament {
  private TournamentMetadata tournamentMetadata;
  private ArrayList<TournamentDivision> tournamentDivision;

  public Tournament() {
    this.tournamentDivision = new ArrayList<TournamentDivision>();
  }

  public Tournament(Event event) {
    // Initialise Class Variables
    this.tournamentDivision = new ArrayList<TournamentDivision>();

    // Create Tournament Metadata based on Event Object
    this.setTournamentMetadata(new TournamentMetadata(event));

    // Add Tournament Divisions based on Event Object
    for (Competition competition : event.getCompetitions()) {
      this.addTournamentDivision(new TournamentDivision(competition));
    }
  }

  public void setTournamentMetadata(TournamentMetadata
   newTournamentMetadata) {
    this.tournamentMetadata = newTournamentMetadata;
  }

  @XmlElement(name = "tournament-metadata")
  public TournamentMetadata getTournamentMetadata() {
    return this.tournamentMetadata;
  }

  public void addTournamentDivision(TournamentDivision newTournamentDivision) {
    this.tournamentDivision.add(newTournamentDivision);
  }

  @XmlElement(name = "tournament-division")
  public ArrayList<TournamentDivision> getTournamentDivision() {
    return this.tournamentDivision;
  }
}
```

Listing 8.15: sportsml.Tournament()

```java
package sportsml;

import javax.xml.bind.annotation.XmlAttribute;
import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlType;

import net.fencingarchive.Event;


@XmlRootElement(name = "tournament-metadata")
@XmlType(propOrder = {"site", "sportsContentCodes"})
public class TournamentMetadata {
  private String tournamentStatus;
  private String tournamentName;
  private Site site;
  private SportsContentCodes sportsContentCodes;

  public TournamentMetadata() {

  }

  public TournamentMetadata(Event event) {
    this.setTournamentStatus("post-event");
    this.setTournamentName(event.getName());
    this.setSite(new Site(event));
    this.setSportsContentCodes(new SportsContentCodes(event));
  }

  public void setTournamentStatus(String newTournamentStatus) {
    this.tournamentStatus = newTournamentStatus;
  }

  @XmlAttribute(name = "tournament-status")
  public String getTournamentStatus() {
    return this.tournamentStatus;
  }

  public void setTournamentName(String newTournamentName) {
    this.tournamentName = newTournamentName;
  }

  @XmlAttribute(name = "tournament-name")
  public String getTournamentName() {
    return this.tournamentName;
  }

  public void setSite(Site newSite) {
    this.site = newSite;
  }

  @XmlElement(name = "site")
  public Site getSite() {
    return this.site;
  }

  public void setSportsContentCodes(SportsContentCodes
   newSportsContentCodes) {
    this.sportsContentCodes = newSportsContentCodes;
  }

  @XmlElement(name = "sports-content-codes")
  public SportsContentCodes getSportsContentCodes() {
    return this.sportsContentCodes;
  }
}
```

Listing 8.16: sportsml.TournamentMetadata()

```
package sportsml;

import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;

import net.fencingarchive.Event;


@XmlRootElement(name = "site")
public class Site {
  private SiteMetadata siteMetadata;

  public Site() {

  }

  public Site(Event event) {
    this.setSiteMetadata(new SiteMetadata(event));
  }

  public Site(SiteMetadata newSiteMetadata) {
    this.setSiteMetadata(newSiteMetadata);
  }

  public void setSiteMetadata(SiteMetadata newSiteMetadata) {
    this.siteMetadata = newSiteMetadata;
  }

  @XmlElement(name = "site-metadata")
  public SiteMetadata getSiteMetadata() {
    return this.siteMetadata;
  }
}
```

Listing 8.17: sportsml.Site()

```java
package sportsml;

import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;

import net.fencingarchive.Event;


@XmlRootElement(name = "site-metadata")
public class SiteMetadata {
  private HomeLocation homeLocation;

  public SiteMetadata() {

  }

  public SiteMetadata(Event event) {
    this.setHomeLocation(new HomeLocation(event));
  }

  public void setHomeLocation(HomeLocation newHomeLocation) {
    this.homeLocation = newHomeLocation;
  }

  @XmlElement(name = "home-location")
  public HomeLocation getHomeLocation() {
    return this.homeLocation;
  }
}
```

Listing 8.18: sportsml.SiteMetadata()

```java
package sportsml;

import javax.xml.bind.annotation.XmlAttribute;
import javax.xml.bind.annotation.XmlRootElement;

import net.fencingarchive.Event;
import net.fencingarchive.Fencer;


@XmlRootElement(name = "home-location")
public class HomeLocation {
  private String country;
  private String city;

  public HomeLocation() {

  }

  public HomeLocation(Event event) {
    this.setCountry(event.getCountry());
    this.setCity(event.getCity());
  }

  public HomeLocation(Fencer fencer) {
    this.setCountry(fencer.getNationality());
  }

  public void setCountry(String newCountry) {
    this.country = newCountry;
  }

  @XmlAttribute(name = "country")
  public String getCountry() {
    return this.country;
  }

  public void setCity(String newCity) {
    this.city = newCity;
  }

  @XmlAttribute(name = "city")
  public String getCity() {
    return this.city;
  }
}
```

Listing 8.19: sportsml.HomeLocation()

```java
package sportsml;

import java.util.ArrayList;

import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;

import net.fencingarchive.Competition;
import net.fencingarchive.Event;


@XmlRootElement(name = "sports-content-codes")
public class SportsContentCodes {
  private ArrayList<SportsContentCode> sportsContentCodes;

  public SportsContentCodes() {
    this.sportsContentCodes = new ArrayList<SportsContentCode>();
  }

  public SportsContentCodes(Event event) {
    this.sportsContentCodes = new ArrayList<SportsContentCode>();
    this.addSportsContentCode(new SportsContentCode("season", event.
   getSeason()));
  }

  public SportsContentCodes(Competition competition) {
    this.sportsContentCodes = new ArrayList<SportsContentCode>();
    this.addSportsContentCode(new SportsContentCode("sport-category",
   competition.getWeapon()
        .toLowerCase(), competition.getWeapon()));
  }

  public void addSportsContentCode(SportsContentCode newSportsContentCode) {
    this.sportsContentCodes.add(newSportsContentCode);
  }

  @XmlElement(name = "sports-content-code")
  public ArrayList<SportsContentCode> getSportsContentCode() {
    return this.sportsContentCodes;
  }
}
```

Listing 8.20: sportsml.SportsContentCodes()

```java
package sportsml;

import javax.xml.bind.annotation.XmlAttribute;

public class SportsContentCode {
  private String codeType;
  private String codeName;
  private String codeKey;

  public SportsContentCode() {

  }

  public SportsContentCode(String newCodeType, String newCodeName) {
    this.setCodeType(newCodeType);
    this.setCodeName(newCodeName);
  }

  public SportsContentCode(String newCodeType, String newCodeKey, String newCodeName) {
    this.setCodeType(newCodeType);
    this.setCodeKey(newCodeKey);
    this.setCodeName(newCodeName);
  }

  public void setCodeType(String newCodeType) {
    this.codeType = newCodeType;
  }

  @XmlAttribute(name = "code-type")
  public String getCodeType() {
    return this.codeType;
  }

  public void setCodeName(String newCodeName) {
    this.codeName = newCodeName;
  }

  @XmlAttribute(name = "code-name")
  public String getCodeName() {
    return this.codeName;
  }

  /**
   * @return the codeKey
   */
  @XmlAttribute(name = ”code−key”)
  public String getCodeKey() {
    return codeKey;
  }

  /**
   * @param codeKey the codeKey to set
   */
  public void setCodeKey(String codeKey) {
    this.codeKey = codeKey;
  }

}
```

Listing 8.21: sportsml.SportsContentCode()

8.4 Acknowledgments

I would like to acknowledge the assistance of the many people that have helped me with this project, inluding my tutor Prof Peter Smith, Katie Rhodes from British Fencing, members of the SportsML Yahoo mailing list and, most of all my long-suffering wife Abby and our dog Dotty - I'll start paying a bit more attention to you both again now!

Matt Carus
Sept 2016
The Open University

8 References

British Fencing. *Competitive Fencing*. British Fencing, 2010. URL `http://www.surreyfencing.com/files/BFA_Competitive_Fencing_Guide_Nov2010.pdf`. [Accessed Feb 3, 2016].

Engarde Escrime. Functionalities of the 3 versions of engarde, undated. URL `http://engarde-escrime.com/siteTemplate.php?lang=en&page=Engarde3Versions.php`. [Accessed Feb 4, 2016].

Fencing Time. Fencing time features, undated. URL `https://www.fencingtime.com/Home/Features`. [Accessed Feb 4, 2016].

J. Fèvre. Adding fencing to sportsml, 2016. [email].

S. Potts. Adding fencing to sportsml, 2016. [email].

Appendices

A Sample Fencing XML

This is just a placeholder file, need to include Engarde XML and FencingTime XML files here

A.1 Engarde XML

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<sports-content xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://iptc.org/std/SportsML/2008-04-01/"
  >
  <!--
      This sample competition is loosely based on the data available in human-readable format at
        http://www.britishfencing.com/uploads/files/scottish_open_2016_mens_foil.htm

      Ignore all IDs and timestamps - these are made up for this example

      Many of the repeated sections are omitted for brevity

      The data is not necessarily consistent, no weight should be given to the actual values, these
      are representative only.
  -->
  <sports-metadata doc-id="XYZ">
    <sports-title>Scottish Open</sports-title>
  </sports-metadata>
  <tournament>
    <tournament-metadata tournament-status="post-event" tournament-name="Scottish Open">
      <site>
        <site-metadata>
          <home-location country="UK" city="Edinburgh"/>
        </site-metadata>
      </site>
      <sports-content-codes>
        <!-- Fencing seasons follow the UK academic year, running from Sept 1st to Aug 31st the following year -->
        <sports-content-code code-type="season" code-name="2015-16"/>
      </sports-content-codes>
    </tournament-metadata>
    <!--
        The first level of division refers to the age-group/weapon/gender combination of the competition
        This is the Open, Foil, Male one
    -->
    <tournament-division>
      <!--
      In the context of fencing, Open refers to an 'age category', others might be u15, u17, veteran etc.
      Younger age categories may also be referred to by a french name such as 'poussin' for u9.
      The full list is here: https://fr.wikipedia.org/wiki/Cat%C3%A9gorie_(sports)#Cat.C3.A9gories_en_athl.C3.A9tisme
      It is possible to have multiple age groups in a competition although this example only has one.
  -->
  <tournament-division-metadata division-name="Open">
    <!--
        TODO: add qualifier for weapon
        <sports-content-qualifier weapon="foil"/>

        Will sports-content-code work?

        <sports-content-codes>
          <sports-content-code code-type="weapon" code-key="foil" code-name="Foil"/>
        </sports-content-codes>

    -->
    <sports-content-qualifier gender="male"/>
  </tournament-division-metadata>
  <!--
      A tournament round is e.g. the first round of poules, second round of poules, first round of DE etc.
      There may or may not be elimination after each round of poules but will be elimination after each round
      of DE. (Note though, need to take into account plate competitions and repechage)
  -->
  <!--
      Note: is it worth splitting the stages between poules and DE? There are different rules for each (e.g.
      score needed to win)
  -->
  <tournament-stage>
    <tournament-stage-metadata stage-status="post-event" stage-key="1" stage-name="Poules Round 1"/>
    <!--
        Standings are the results at the END of this round, however they must come before the bouts data to
        conform with the schema
    -->
    <standing content-label="Poules Round 1">
      <!-- Is this an error in the XSD? Do I really need an empty metadata element here?
           See sportsml-core.xsd Line 2982, no minOccurs attribute -->
      <standing-metadata/>

      <player>
        <player-metadata player-key="fencing.fencer:1" team-key="fencing.club:1" team-idref="HOLY">
          <name last="Cook" first="Keith" full="Keith Cook"/>
          <home-location country="GBR"/>
        </player-metadata>
        <player-stats>
```

```
          <rank value="1"/>
          <sports-property formal-name="victories" value="6"/>
          <sports-property formal-name="victories-rate" value="1.00"/>
          <sports-property formal-name="hits-scored" value="30"/>
          <sports-property formal-name="hits-received" value="6"/>
          <!--
              Indicator can be positive and negative, and is normally displayed with a leading plus(+)
              or minus(-) sign. Should the plus (+) sign be included here? Also, indicator can always
              be calculated from the two elements above, should it even be included here?
           -->
          <sports-property formal-name="indicator" value="24"/>
        </player-stats>
      </player>

      <player>
        <player-metadata player-key="fencing.fencer:23" team-key="fencing.club:23" team-idref="WALL">
          <name last="Woollard" first="Jonathan" full="Jonathan Woollard"/>
          <home-location country="GBR"/>
        </player-metadata>
        <player-stats>
          <rank value="2"/>
          <sports-property formal-name="victories" value="5"/>
          <sports-property formal-name="victories-rate" value="1.00"/>
          <sports-property formal-name="hits-scored" value="25"/>
          <sports-property formal-name="hits-received" value="3"/>
          <sports-property formal-name="indicator" value="22"/>
        </player-stats>
      </player>

      <player>
        <player-metadata player-key="fencing.fencer:24" team-key="fencing.club:24" team-idref="SALEH">
          <name last="Russell" first="Iain" full="Iain Russell"/>
          <home-location country="GBR"/>
        </player-metadata>
        <player-stats>
          <rank value="3"/>
          <sports-property formal-name="victories" value="5"/>
          <sports-property formal-name="victories-rate" value="1.00"/>
          <sports-property formal-name="hits-scored" value="25"/>
          <sports-property formal-name="hits-received" value="7"/>
          <sports-property formal-name="indicator" value="18"/>
        </player-stats>
      </player>

      <player>
        <player-metadata player-key="fencing.fencer:26" team-key="fencing.club:26" team-idref="SALLH">
          <name last="Douglas" first="Jack" full="Jack Douglas"/>
        </player-metadata>
        <player-stats>
          <rank value="4"/>
          <sports-property formal-name="victories" value="5"/>
          <sports-property formal-name="victories-rate" value="1.00"/>
          <sports-property formal-name="hits-scored" value="25"/>
          <sports-property formal-name="hits-received" value="9"/>
          <sports-property formal-name="indicator" value="16"/>
        </player-stats>
      </player>
      <!-- Other fencers omitted for brevity -->
    </standing>

    <!-- A sports event is one bout within that round -->
    <!--
        TODO: Need some way of defining which poule this is for, encoding it in the event-key just doesn`t
        seem right. Will sports-content-code work?
    -->
    <sports-event>
      <event-metadata start-date-time="yyyymmddThhmmss+hhmm"
        event-key="XYZ.poule-1.bout-1"
        event-name="Scottish Open Mens Foil 2016"
        date-coverage-type="event" event-status="post-event" season-key="2015-16"/>
      <player>
        <player-metadata player-key="fencing.fencer:519" team-key="fencing.club:26" team-idref="SALLH">
          <name last="McEwan" first="Mike" full="Mike McEwan"/>
        </player-metadata>
        <!-- In fencing the victor`s score is ordinarily marked as 'V', rather than the actual score they got.
            This is to cope with the situation in epee where the score is 4-4 (or 14-14) and a double is scored,
            giving both fencers the maximum score. Another point is fenced and the winner gets 'V' and the looser
            5 (or 15). This does have an impact on calculations used to determine positions. -->
        <player-stats score="3"/>
      </player>
      <player>
        <player-metadata player-key="fencing.fencer:16275" team-key="fencing.club:26" team-idref="SALLH">
          <name last="Bradie" first="Angus" full="Angus Bradie"/>
        </player-metadata>
        <player-stats score="V"/>
      </player>
    </sports-event>

    <sports-event>
      <event-metadata start-date-time="yyyymmddThhmmss+hhmm"
        event-key="XYZ.poule-1.bout-2"
        event-name="Scottish Open Mens Foil 2016"
        date-coverage-type="event" event-status="post-event" season-key="2015-16"/>
      <player>
        <player-metadata player-key="fencing.fencer:519" team-key="fencing.club:26" team-idref="SALLH">
          <name last="McEwan" first="Mike" full="Mike McEwan"/>
        </player-metadata>
        <player-stats score="V"/>
      </player>
      <player>
        <player-metadata player-key="fencing.fencer:16275" team-key="fencing.club:26" team-idref="SALLH">
          <name last="Crawford" first="Angus" full="Angus Crawford"/>
        </player-metadata>
        <player-stats score="0"/>
      </player>
    </sports-event>

    <!-- other bouts omitted for brevity -->
  </tournament-stage>

  <!-- This particular competition only had one round of poules so the next round is the DE -->
  <tournament-stage>
    <!-- This is an example of a DE round -->
    <tournament-stage-metadata stage-status="post-event" stage-key="2" stage-name="DE Last 32"/>
    <!--
        These are the standings at the END of this round of DE, as such fencers that have already been
        eliminated will not change position (i.e. if we‘re at the quarter final, 8 fencers are left in
        the competition so their positions can change, but everything from 9 down will be the same as
        it was for the last round.
    -->
    <standing content-label="Semi-Final">
      <standing-metadata/>

      <player>
        <player-metadata player-key="fencing.fencer:1" team-key="fencing.club:1" team-idref="HOLY">
          <name last="Cook" first="Keith" full="Keith Cook"/>
          <home-location country="GBR"/>
        </player-metadata>
        <player-stats>
          <rank value="1"/>
        </player-stats>
      </player>

      <player>
        <player-metadata player-key="fencing.fencer:23" team-key="fencing.club:23" team-idref="WALL">
          <name last="Woollard" first="Jonathan" full="Jonathan Woollard"/>
          <home-location country="GBR"/>
        </player-metadata>
        <player-stats>
          <rank value="2"/>
        </player-stats>
      </player>

      <player>
        <player-metadata player-key="fencing.fencer:24" team-key="fencing.club:24" team-idref="SALEH">
          <name last="Russell" first="Iain" full="Iain Russell"/>
          <home-location country="GBR"/>
        </player-metadata>
        <player-stats>
          <rank value="3"/>
        </player-stats>
      </player>

      <player>
        <player-metadata player-key="fencing.fencer:26" team-key="fencing.club:26" team-idref="SALLH">
          <name last="Douglas" first="Jack" full="Jack Douglas"/>
        </player-metadata>
        <player-stats>
          <rank value="4"/>
        </player-stats>
      </player>
      <!-- Other fencers omitted for brevity -->

    </standing>

    <sports-event>
      <event-metadata start-date-time="yyyymmddThhmmss+hhmm"
        event-key="XYZ.de-last16.bout-1"
        event-name="Scottish Open Mens Foil 2016"
        date-coverage-type="event" event-status="post-event" season-key="2015-16"/>
      <player>
        <player-metadata player-key="fencing.fencer:1" team-key="fencing.club:1" team-idref="HOLY">
          <name last="Cook" first="Keith" full="Keith Cook"/>
          <home-location country="GBR"/>
        </player-metadata>
        <player-stats score="V"/>
      </player>
      <player>
        <player-metadata player-key="fencing.fencer:26" team-key="fencing.club:26" team-idref="SALLH">
          <name last="Douglas" first="Jack" full="Jack Douglas"/>
        </player-metadata>
        <player-stats score="10"/>
      </player>
    </sports-event>

    <sports-event>
      <event-metadata start-date-time="yyyymmddThhmmss+hhmm"
        event-key="XYZ.de-last16.bout-1"
        event-name="Scottish Open Mens Foil 2016"
        date-coverage-type="event" event-status="post-event" season-key="2015-16"/>
      <player>
        <player-metadata player-key="fencing.fencer:23" team-key="fencing.club:23" team-idref="WALL">
          <name last="Woollard" first="Jonathan" full="Jonathan Woollard"/>
          <home-location country="GBR"/>
        </player-metadata>
        <player-stats score="3"/>
      </player>
      <player>
```

```
            <player-metadata player-key="fencing.fencer:24" team-key="fencing.club:24" team-idref="SALEH">
              <name last="Russell" first="Iain" full="Iain Russell"/>
              <home-location country="GBR"/>
            </player-metadata>
            <player-stats score="V"/>
          </player>
        </sports-event>

        <!-- Semi-final so only two bouts -->

      </tournament-stage>
      <!-- Final would feature here, along with the final standing that is the overall result of the competition. -->

    </tournament-division>
    <!--
        Additional tournament-division elements would feature here, in this particular competition they would be for:
        Womens Foil
        Mens Epee
        Womens Epee
        Mens Sabre
        Womens Sabre
    -->
  </tournament>
</sports-content>
```

Listing A.1: ../SportsML-G2/testing/example-2-open.xml

A.2 FencingTime XML

```
<?xml version="1.0" encoding="iso-8859-1"?>
<sports-content xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://iptc.org/std/SportsML/2008-04-01/"
  >
  <!--
      This sample competition is loosely based on the data available in human-readable format at
        http://www.britishfencing.com/uploads/files/scottish_open_2016_mens_foil.htm

      Ignore all IDs and timestamps - these are made up for this example

      Many of the repeated sections are omitted for brevity

      The data is not necessarily consistent, no weight should be given to the actual values, these
      are representative only.
  -->
  <sports-metadata doc-id="XYZ">
    <sports-title>Scottish Open</sports-title>
  </sports-metadata>
  <tournament>
    <tournament-metadata tournament-status="post-event" tournament-name="Scottish Open">
```

```
      <site>
        <site-metadata>
          <home-location country="UK" city="Edinburgh"/>
        </site-metadata>
      </site>
      <sports-content-codes>
        <!-- Fencing seasons follow the UK academic year, running from Sept 1st to Aug 31st the following year -->
        <sports-content-code code-type="season" code-name="2015-16"/>
      </sports-content-codes>
    </tournament-metadata>
    <!--
        The first level of division refers to the age-group/weapon/gender combination of the competition
        This is the Open, Foil, Male one
    -->
    <tournament-division>
      <!--
          In the context of fencing, Open refers to an 'age category', others might be u15, u17, veteran etc.
          Younger age categories may also be referred to by a french name such as 'poussin' for u9.
          The full list is here: https://fr.wikipedia.org/wiki/Cat%C3%A9gorie_(sports)#Cat.C3.A9gories_en_athl.C3.A9tisme
          It is possible to have multiple age groups in a competition although this example only has one.
      -->
      <tournament-division-metadata division-name="Open">
        <!--
            TODO: add qualifier for weapon
            <sports-content-qualifier weapon="foil"/>

            Will sports-content-code work?

            <sports-content-codes>
              <sports-content-code code-type="weapon" code-key="foil" code-name="Foil"/>
            </sports-content-codes>

        -->
        <sports-content-qualifier gender="male"/>
      </tournament-division-metadata>
      <!--
          A tournament round is e.g. the first round of poules, second round of poules, first round of DE etc.
          There may or may not be elimination after each round of poules but will be elimination after each round
          of DE. (Note though, need to take into account plate competitions and repechage)
      -->
      <!--
          Note: is it worth splitting the stages between poules and DE? There are different rules for each (e.g.
          score needed to win)
      -->
      <tournament-stage>
        <tournament-stage-metadata stage-status="post-event" stage-key="1" stage-name="Poules Round 1"/>
        <!--
            Standings are the results at the END of this round, however they must come before the bouts data to
            conform with the schema
        -->
        <standing content-label="Poules Round 1">
          <!-- Is this an error in the XSD? Do I really need an empty metadata element here?
               See sportsml-core.xsd Line 2982, no minOccurs attribute -->
          <standing-metadata/>

          <player>
            <player-metadata player-key="fencing.fencer:1" team-key="fencing.club:1" team-idref="HOLY">
              <name last="Cook" first="Keith" full="Keith Cook"/>
              <home-location country="GBR"/>
            </player-metadata>
            <player-stats>
              <rank value="1"/>
              <sports-property formal-name="victories" value="6"/>
              <sports-property formal-name="victories-rate" value="1.00"/>
              <sports-property formal-name="hits-scored" value="30"/>
              <sports-property formal-name="hits-received" value="6"/>
              <!--
                  Indicator can be positive and negative, and is normally displayed with a leading plus(+)
                  or minus(-) sign. Should the plus (+) sign be included here? Also, indicator can always
                  be calculated from the two elements above, should it even be included here?
               -->
              <sports-property formal-name="indicator" value="24"/>
            </player-stats>
          </player>

          <player>
            <player-metadata player-key="fencing.fencer:23" team-key="fencing.club:23" team-idref="WALL">
              <name last="Woollard" first="Jonathan" full="Jonathan Woollard"/>
              <home-location country="GBR"/>
            </player-metadata>
            <player-stats>
              <rank value="2"/>
              <sports-property formal-name="victories" value="5"/>
              <sports-property formal-name="victories-rate" value="1.00"/>
              <sports-property formal-name="hits-scored" value="25"/>
              <sports-property formal-name="hits-received" value="3"/>
              <sports-property formal-name="indicator" value="22"/>
            </player-stats>
          </player>

          <player>
            <player-metadata player-key="fencing.fencer:24" team-key="fencing.club:24" team-idref="SALEH">
              <name last="Russell" first="Iain" full="Iain Russell"/>
              <home-location country="GBR"/>
            </player-metadata>
```

```
          <player-stats>
            <rank value="3"/>
            <sports-property formal-name="victories" value="5"/>
            <sports-property formal-name="victories-rate" value="1.00"/>
            <sports-property formal-name="hits-scored" value="25"/>
            <sports-property formal-name="hits-received" value="7"/>
            <sports-property formal-name="indicator" value="18"/>
          </player-stats>
        </player>

        <player>
          <player-metadata player-key="fencing.fencer:26" team-key="fencing.club:26" team-idref="SALLH">
            <name last="Douglas" first="Jack" full="Jack Douglas"/>
          </player-metadata>
          <player-stats>
            <rank value="4"/>
            <sports-property formal-name="victories" value="5"/>
            <sports-property formal-name="victories-rate" value="1.00"/>
            <sports-property formal-name="hits-scored" value="25"/>
            <sports-property formal-name="hits-received" value="9"/>
            <sports-property formal-name="indicator" value="16"/>
          </player-stats>
        </player>
        <!-- Other fencers omitted for brevity -->
      </standing>

      <!-- A sports event is one bout within that round -->
      <!--
          TODO: Need some way of defining which poule this is for, encoding it in the event-key just doesn‘t
          seem right. Will sports-content-code work?
      -->
      <sports-event>
        <event-metadata start-date-time="yyyymmddThhmmss+hhmm"
          event-key="XYZ.poule-1.bout-1"
          event-name="Scottish Open Mens Foil 2016"
          date-coverage-type="event" event-status="post-event" season-key="2015-16"/>
        <player>
          <player-metadata player-key="fencing.fencer:519" team-key="fencing.club:26" team-idref="SALLH">
            <name last="McEwan" first="Mike" full="Mike McEwan"/>
          </player-metadata>
          <!-- In fencing the victor‘s score is ordinarily marked as 'V', rather than the actual score they got.
               This is to cope with the situation in epee where the score is 4-4 (or 14-14) and a double is scored,
               giving both fencers the maximum score. Another point is fenced and the winner gets 'V' and the looser
               5 (or 15). This does have an impact on calculations used to determine positions. -->
          <player-stats score="3"/>
        </player>
        <player>
          <player-metadata player-key="fencing.fencer:16275" team-key="fencing.club:26" team-idref="SALLH">
            <name last="Bradie" first="Angus" full="Angus Bradie"/>
          </player-metadata>
          <player-stats score="V"/>
        </player>
      </sports-event>

      <sports-event>
        <event-metadata start-date-time="yyyymmddThhmmss+hhmm"
          event-key="XYZ.poule-1.bout-2"
          event-name="Scottish Open Mens Foil 2016"
          date-coverage-type="event" event-status="post-event" season-key="2015-16"/>
        <player>
          <player-metadata player-key="fencing.fencer:519" team-key="fencing.club:26" team-idref="SALLH">
            <name last="McEwan" first="Mike" full="Mike McEwan"/>
          </player-metadata>
          <player-stats score="V"/>
        </player>
        <player>
          <player-metadata player-key="fencing.fencer:16275" team-key="fencing.club:26" team-idref="SALLH">
            <name last="Crawford" first="Angus" full="Angus Crawford"/>
          </player-metadata>
          <player-stats score="0"/>
        </player>
      </sports-event>

      <!-- other bouts omitted for brevity -->
    </tournament-stage>

    <!-- This particular competition only had one round of poules so the next round is the DE -->
    <tournament-stage>
      <!-- This is an example of a DE round -->
      <tournament-stage-metadata stage-status="post-event" stage-key="2" stage-name="DE Last 32"/>
      <!--
          These are the standings at the END of this round of DE, as such fencers that have already been
          eliminated will not change position (i.e. if we‘re at the quarter final, 8 fencers are left in
          the competition so their positions can change, but everything from 9 down will be the same as
          it was for the last round.
      -->
      <standing content-label="Semi-Final">
        <standing-metadata/>

        <player>
          <player-metadata player-key="fencing.fencer:1" team-key="fencing.club:1" team-idref="HOLY">
            <name last="Cook" first="Keith" full="Keith Cook"/>
            <home-location country="GBR"/>
          </player-metadata>
          <player-stats>
            <rank value="1"/>
          </player-stats>
        </player>
```

```

          <player>
            <player-metadata player-key="fencing.fencer:23" team-key="fencing.club:23" team-idref="WALL">
              <name last="Woollard" first="Jonathan" full="Jonathan Woollard"/>
              <home-location country="GBR"/>
            </player-metadata>
            <player-stats>
              <rank value="2"/>
            </player-stats>
          </player>

          <player>
            <player-metadata player-key="fencing.fencer:24" team-key="fencing.club:24" team-idref="SALEH">
              <name last="Russell" first="Iain" full="Iain Russell"/>
              <home-location country="GBR"/>
            </player-metadata>
            <player-stats>
              <rank value="3"/>
            </player-stats>
          </player>

          <player>
            <player-metadata player-key="fencing.fencer:26" team-key="fencing.club:26" team-idref="SALLH">
              <name last="Douglas" first="Jack" full="Jack Douglas"/>
            </player-metadata>
            <player-stats>
              <rank value="4"/>
            </player-stats>
          </player>
          <!-- Other fencers omitted for brevity -->

        </standing>

        <sports-event>
          <event-metadata start-date-time="yyyymmddThhmmss+hhmm"
            event-key="XYZ.de-last16.bout-1"
            event-name="Scottish Open Mens Foil 2016"
            date-coverage-type="event" event-status="post-event" season-key="2015-16"/>
          <player>
            <player-metadata player-key="fencing.fencer:1" team-key="fencing.club:1" team-idref="HOLY">
              <name last="Cook" first="Keith" full="Keith Cook"/>
              <home-location country="GBR"/>
            </player-metadata>
            <player-stats score="V"/>
          </player>
          <player>
            <player-metadata player-key="fencing.fencer:26" team-key="fencing.club:26" team-idref="SALLH">
              <name last="Douglas" first="Jack" full="Jack Douglas"/>
            </player-metadata>
            <player-stats score="10"/>
          </player>
        </sports-event>

        <sports-event>
          <event-metadata start-date-time="yyyymmddThhmmss+hhmm"
            event-key="XYZ.de-last16.bout-1"
            event-name="Scottish Open Mens Foil 2016"
            date-coverage-type="event" event-status="post-event" season-key="2015-16"/>
          <player>
            <player-metadata player-key="fencing.fencer:23" team-key="fencing.club:23" team-idref="WALL">
              <name last="Woollard" first="Jonathan" full="Jonathan Woollard"/>
              <home-location country="GBR"/>
            </player-metadata>
            <player-stats score="3"/>
          </player>
          <player>
            <player-metadata player-key="fencing.fencer:24" team-key="fencing.club:24" team-idref="SALEH">
              <name last="Russell" first="Iain" full="Iain Russell"/>
              <home-location country="GBR"/>
            </player-metadata>
            <player-stats score="V"/>
          </player>
        </sports-event>

        <!-- Semi-final so only two bouts -->

      </tournament-stage>
      <!-- Final would feature here, along with the final standing that is the overall result of the competition. -->

    </tournament-division>
    <!--
        Additional tournament-division elements would feature here, in this particular competition they would be for:
        Womens Foil
        Mens Epee
        Womens Epee
        Mens Sabre
        Womens Sabre
    -->
  </tournament>
</sports-content>
```

Listing A.2: ../SportsML–G2/testing/example–2–open.xml

B Sample SportsML

B.1 Original Sample SportsML file of Fencing Data

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<sports-content xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://iptc.org/std/SportsML/2008-04-01/"
  >
  <!--
      This sample competition is loosely based on the data available in human-readable format at
        http://www.britishfencing.com/uploads/files/scottish_open_2016_mens_foil.htm

      Ignore all IDs and timestamps - these are made up for this example

      Many of the repeated sections are omitted for brevity

      The data is not necessarily consistent, no weight should be given to the actual values, these
      are representative only.
  -->
  <sports-metadata doc-id="XYZ">
    <sports-title>Scottish Open</sports-title>
  </sports-metadata>
  <tournament>
    <tournament-metadata tournament-status="post-event" tournament-name="Scottish Open">
      <site>
        <site-metadata>
          <home-location country="UK" city="Edinburgh"/>
        </site-metadata>
      </site>
      <sports-content-codes>
        <!-- Fencing seasons follow the UK academic year, running from Sept 1st to Aug 31st the following year -->
        <sports-content-code code-type="season" code-name="2015-16"/>
      </sports-content-codes>
    </tournament-metadata>
    <!--
        The first level of division refers to the age-group/weapon/gender combination of the competition
        This is the Open, Foil, Male one
    -->
    <tournament-division>
      <!--
          In the context of fencing, Open refers to an 'age category', others might be u15, u17, veteran etc.
          Younger age categories may also be referred to by a french name such as 'poussin' for u9.
        The full list is here: https://fr.wikipedia.org/wiki/Cat%C3%A9gorie_(sports)#Cat.C3.A9gories_en_athl.C3.A9tisme
        It is possible to have multiple age groups in a competition although this example only has one.
    -->
    <tournament-division-metadata division-name="Open">
      <!--
          TODO: add qualifier for weapon
          <sports-content-qualifier weapon="foil"/>

          Will sports-content-code work?

          <sports-content-codes>
            <sports-content-code code-type="weapon" code-key="foil" code-name="Foil"/>
          </sports-content-codes>

      -->
      <sports-content-qualifier gender="male"/>
    </tournament-division-metadata>
    <!--
        A tournament round is e.g. the first round of poules, second round of poules, first round of DE etc.
        There may or may not be elimination after each round of poules but will be elimination after each round
        of DE. (Note though, need to take into account plate competitions and repechage)
    -->
    <!--
        Note: is it worth splitting the stages between poules and DE? There are different rules for each (e.g.
        score needed to win)
    -->
    <tournament-stage>
      <tournament-stage-metadata stage-status="post-event" stage-key="1" stage-name="Poules Round 1"/>
      <!--
          Standings are the results at the END of this round, however they must come before the bouts data to
          conform with the schema
      -->
      <standing content-label="Poules Round 1">
        <!-- Is this an error in the XSD? Do I really need an empty metadata element here?
             See sportsml-core.xsd Line 2982, no minOccurs attribute -->
        <standing-metadata/>

        <player>
          <player-metadata player-key="fencing.fencer:1" team-key="fencing.club:1" team-idref="HOLY">
            <name last="Cook" first="Keith" full="Keith Cook"/>
            <home-location country="GBR"/>
          </player-metadata>
          <player-stats>
            <rank value="1"/>
            <sports-property formal-name="victories" value="6"/>
            <sports-property formal-name="victories-rate" value="1.00"/>
            <sports-property formal-name="hits-scored" value="30"/>
            <sports-property formal-name="hits-received" value="6"/>
            <!--
                Indicator can be positive and negative, and is normally displayed with a leading plus(+)
                or minus(-) sign. Should the plus (+) sign be included here? Also, indicator can always
                be calculated from the two elements above, should it even be included here?
             -->
            <sports-property formal-name="indicator" value="24"/>
          </player-stats>
        </player>

        <player>
          <player-metadata player-key="fencing.fencer:23" team-key="fencing.club:23" team-idref="WALL">
            <name last="Woollard" first="Jonathan" full="Jonathan Woollard"/>
            <home-location country="GBR"/>
          </player-metadata>
          <player-stats>
            <rank value="2"/>
            <sports-property formal-name="victories" value="5"/>
            <sports-property formal-name="victories-rate" value="1.00"/>
            <sports-property formal-name="hits-scored" value="25"/>
            <sports-property formal-name="hits-received" value="3"/>
            <sports-property formal-name="indicator" value="22"/>
          </player-stats>
        </player>

        <player>
          <player-metadata player-key="fencing.fencer:24" team-key="fencing.club:24" team-idref="SALEH">
            <name last="Russell" first="Iain" full="Iain Russell"/>
            <home-location country="GBR"/>
          </player-metadata>
          <player-stats>
            <rank value="3"/>
            <sports-property formal-name="victories" value="5"/>
            <sports-property formal-name="victories-rate" value="1.00"/>
            <sports-property formal-name="hits-scored" value="25"/>
            <sports-property formal-name="hits-received" value="7"/>
            <sports-property formal-name="indicator" value="18"/>
          </player-stats>
        </player>

        <player>
          <player-metadata player-key="fencing.fencer:26" team-key="fencing.club:26" team-idref="SALLH">
            <name last="Douglas" first="Jack" full="Jack Douglas"/>
          </player-metadata>
          <player-stats>
            <rank value="4"/>
            <sports-property formal-name="victories" value="5"/>
            <sports-property formal-name="victories-rate" value="1.00"/>
            <sports-property formal-name="hits-scored" value="25"/>
            <sports-property formal-name="hits-received" value="9"/>
            <sports-property formal-name="indicator" value="16"/>
          </player-stats>
        </player>
        <!-- Other fencers omitted for brevity -->
      </standing>

      <!-- A sports event is one bout within that round -->
      <!--
          TODO: Need some way of defining which poule this is for, encoding it in the event-key just doesn‘t
          seem right. Will sports-content-code work?
      -->
      <sports-event>
        <event-metadata start-date-time="yyyymmddThhmmss+hhmm"
          event-key="XYZ.poule-1.bout-1"
          event-name="Scottish Open Mens Foil 2016"
          date-coverage-type="event" event-status="post-event" season-key="2015-16"/>
        <player>
          <player-metadata player-key="fencing.fencer:519" team-key="fencing.club:26" team-idref="SALLH">
            <name last="McEwan" first="Mike" full="Mike McEwan"/>
          </player-metadata>
          <!-- In fencing the victor‘s score is ordinarily marked as ’V’, rather than the actual score they got.
               This is to cope with the situation in epee where the score is 4-4 (or 14-14) and a double is scored,
               giving both fencers the maximum score. Another point is fenced and the winner gets ’V’ and the looser
               5 (or 15). This does have an impact on calculations used to determine positions. -->
          <player-stats score="3"/>
        </player>
        <player>
          <player-metadata player-key="fencing.fencer:16275" team-key="fencing.club:26" team-idref="SALLH">
            <name last="Bradie" first="Angus" full="Angus Bradie"/>
          </player-metadata>
          <player-stats score="V"/>
        </player>
      </sports-event>

      <sports-event>
        <event-metadata start-date-time="yyyymmddThhmmss+hhmm"
          event-key="XYZ.poule-1.bout-2"
          event-name="Scottish Open Mens Foil 2016"
          date-coverage-type="event" event-status="post-event" season-key="2015-16"/>
        <player>
          <player-metadata player-key="fencing.fencer:519" team-key="fencing.club:26" team-idref="SALLH">
            <name last="McEwan" first="Mike" full="Mike McEwan"/>
          </player-metadata>
          <player-stats score="V"/>
        </player>
        <player>
          <player-metadata player-key="fencing.fencer:16275" team-key="fencing.club:26" team-idref="SALLH">
            <name last="Crawford" first="Angus" full="Angus Crawford"/>
          </player-metadata>
          <player-stats score="0"/>
        </player>
      </sports-event>

      <!-- other bouts omitted for brevity -->
    </tournament-stage>

    <!-- This particular competition only had one round of poules so the next round is the DE -->
    <tournament-stage>
      <!-- This is an example of a DE round -->
      <tournament-stage-metadata stage-status="post-event" stage-key="2" stage-name="DE Last 32"/>
      <!--
          These are the standings at the END of this round of DE, as such fencers that have already been
          eliminated will not change position (i.e. if we're at the quarter final, 8 fencers are left in
          the competition so their positions can change, but everything from 9 down will be the same as
          it was for the last round.
      -->
      <standing content-label="Semi-Final">
        <standing-metadata/>

        <player>
          <player-metadata player-key="fencing.fencer:1" team-key="fencing.club:1" team-idref="HOLY">
            <name last="Cook" first="Keith" full="Keith Cook"/>
            <home-location country="GBR"/>
          </player-metadata>
          <player-stats>
            <rank value="1"/>
          </player-stats>
        </player>

        <player>
          <player-metadata player-key="fencing.fencer:23" team-key="fencing.club:23" team-idref="WALL">
            <name last="Woollard" first="Jonathan" full="Jonathan Woollard"/>
            <home-location country="GBR"/>
          </player-metadata>
          <player-stats>
            <rank value="2"/>
          </player-stats>
        </player>

        <player>
          <player-metadata player-key="fencing.fencer:24" team-key="fencing.club:24" team-idref="SALEH">
            <name last="Russell" first="Iain" full="Iain Russell"/>
            <home-location country="GBR"/>
          </player-metadata>
          <player-stats>
            <rank value="3"/>
          </player-stats>
        </player>

        <player>
          <player-metadata player-key="fencing.fencer:26" team-key="fencing.club:26" team-idref="SALLH">
            <name last="Douglas" first="Jack" full="Jack Douglas"/>
          </player-metadata>
          <player-stats>
            <rank value="4"/>
          </player-stats>
        </player>
        <!-- Other fencers omitted for brevity -->

      </standing>

      <sports-event>
        <event-metadata start-date-time="yyyymmddThhmmss+hhmm"
           event-key="XYZ.de-last16.bout-1"
           event-name="Scottish Open Mens Foil 2016"
           date-coverage-type="event" event-status="post-event" season-key="2015-16"/>
        <player>
          <player-metadata player-key="fencing.fencer:1" team-key="fencing.club:1" team-idref="HOLY">
            <name last="Cook" first="Keith" full="Keith Cook"/>
            <home-location country="GBR"/>
          </player-metadata>
          <player-stats score="V"/>
        </player>
        <player>
          <player-metadata player-key="fencing.fencer:26" team-key="fencing.club:26" team-idref="SALLH">
            <name last="Douglas" first="Jack" full="Jack Douglas"/>
          </player-metadata>
          <player-stats score="10"/>
        </player>
      </sports-event>

      <sports-event>
        <event-metadata start-date-time="yyyymmddThhmmss+hhmm"
           event-key="XYZ.de-last16.bout-1"
           event-name="Scottish Open Mens Foil 2016"
           date-coverage-type="event" event-status="post-event" season-key="2015-16"/>
        <player>
          <player-metadata player-key="fencing.fencer:23" team-key="fencing.club:23" team-idref="WALL">
            <name last="Woollard" first="Jonathan" full="Jonathan Woollard"/>
            <home-location country="GBR"/>
          </player-metadata>
          <player-stats score="3"/>
        </player>
        <player>
          <player-metadata player-key="fencing.fencer:24" team-key="fencing.club:24" team-idref="SALEH">
            <name last="Russell" first="Iain" full="Iain Russell"/>
            <home-location country="GBR"/>
            </player-metadata>
            <player-stats score="V"/>
          </player>
        </sports-event>

        <!-- Semi-final so only two bouts -->

      </tournament-stage>
      <!-- Final would feature here, along with the final standing that is the overall result of the competition. -->

    </tournament-division>
    <!--
        Additional tournament-division elements would feature here, in this particular competition they would be for:
        Womens Foil
        Mens Epee
        Womens Epee
        Mens Sabre
        Womens Sabre
    -->
  </tournament>
</sports-content>
```

Listing B.1: ../SportsML-G2/testing/example-2-open.xml

B.2 Sample SportsML file after Advice from SportsML Developers

```
<?xml version="1.0" encoding="iso-8859-1"?>
<?xml-stylesheet type="text/xsl" href="sportsml-html.xsl"?>
<sports-content xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://iptc.org/std/SportsML/2008-04-01/"
  >
  <!--
      This sample competition is loosely based on the data available in human-readable format at
        http://www.britishfencing.com/uploads/files/scottish_open_2016_mens_foil.htm

      Ignore all IDs and timestamps - these are made up for this example

      Many of the repeated sections are omitted for brevity

      The data is not necessarily consistent, no weight should be given to the actual values, these
      are representative only.
  -->
  <sports-metadata doc-id="XYZ">
    <sports-title>Scottish Open</sports-title>
  </sports-metadata>
  <tournament>
    <tournament-metadata tournament-status="post-event" tournament-name="Scottish Open">
      <site>
        <site-metadata>
          <home-location country="UK" city="Edinburgh"/>
        </site-metadata>
      </site>
      <sports-content-codes>
        <!-- Fencing seasons follow the UK academic year, running from Sept 1st to Aug 31st the following year -->
        <sports-content-code code-type="season" code-name="2015-16"/>
      </sports-content-codes>
    </tournament-metadata>
    <!--
        The first level of division refers to the age-group/weapon/gender combination of the competition
        This is the Open, Foil, Male one
    -->
    <tournament-division>
      <!--
          In the context of fencing, Open refers to an 'age category', others might be u15, u17, veteran etc.
          Younger age categories may also be referred to by a french name such as 'poussin' for u9.
          The full list is here: https://fr.wikipedia.org/wiki/Cat%C3%A9gorie_(sports)#Cat.C3.A9gories_en_athl.C3.A9tisme
          It is possible to have multiple age groups in a competition although this example only has one.
      -->
      <tournament-division-metadata division-name="Open">
        <sports-content-codes>
          <sports-content-code code-type="sport-category" code-key="foil" code-name="Foil"/>
        </sports-content-codes>
        <sports-content-qualifier gender="male"/>
      </tournament-division-metadata>
      <!--
          A tournament round is e.g. the first round of poules, second round of poules, first round of DE etc.
          There may or may not be elimination after each round of poules but will be elimination after each round
          of DE. (Note though, need to take into account plate competitions and repechage)
      -->
      <!--
          Note: is it worth splitting the stages between poules and DE? There are different rules for each (e.g.
          score needed to win)
      -->
      <tournament-stage>
        <tournament-stage-metadata stage-status="post-event" stage-key="1" stage-name="Poules Round 1"/>
        <!--
            Standings are the results at the END of this round, however they must come before the bouts data to
            conform with the schema
        -->
        <standing content-label="Poules Round 1">
          <!-- Is this an error in the XSD? Do I really need an empty metadata element here?
                See sportsml-core.xsd Line 2982, no minOccurs attribute -->
        <standing-metadata/>

        <team>
          <team-metadata team-key="HOLY" home-page-url="http://example.com">
            <name full="Holyrood" abbreviation="HOLY"/>
          </team-metadata>
          <team-stats/>
          <player>
            <player-metadata player-key="fencing.fencer:1" team-key="fencing.club:1" team-idref="HOLY">
              <name last="Cook" first="Keith" full="Keith Cook"/>
              <home-location country="GBR"/>
            </player-metadata>
            <player-stats>
              <outcome-totals wins="6" winning-percentage="100" points-scored-for="30" points-scored-against="6"/>
              <rank value="1"/>
            </player-stats>
          </player>
        </team>

        <team>
          <team-metadata team-key="WALL" home-page-url="http://example.com">
            <name full="Wallace Fencing Academy" abbreviation="WALL"/>
          </team-metadata>
          <team-stats/>
          <player>
            <player-metadata player-key="fencing.fencer:23" team-key="fencing.club:23" team-idref="WALL">
              <name last="Woollard" first="Jonathan" full="Jonathan Woollard"/>
              <home-location country="GBR"/>
            </player-metadata>
            <player-stats>
              <outcome-totals wins="5" winning-percentage="100" points-scored-for="25" points-scored-against="3"/>
              <rank value="3"/>
            </player-stats>
          </player>
        </team>

        <team>
          <team-metadata team-key="WFIFE" home-page-url="http://example.com">
            <name full="West Fife Fencing Club" abbreviation="WFIFE"/>
          </team-metadata>
          <team-stats/>
          <player>
          <player-metadata player-key="fencing.fencer:24" team-key="fencing.club:24" team-idref="WFIFE">
            <name last="Russell" first="Iain" full="Iain Russell"/>
            <home-location country="GBR"/>
          </player-metadata>
          <player-stats>
            <outcome-totals wins="5" winning-percentage="100" points-scored-for="25" points-scored-against="7"/>
            <rank value="2"/>
          </player-stats>
          </player>

          <player>
            <player-metadata player-key="fencing.fencer:26" team-key="fencing.club:26" team-idref="SALLH">
              <name last="Douglas" first="Jack" full="Jack Douglas"/>
            </player-metadata>
            <player-stats>
              <outcome-totals wins="5" winning-percentage="100" points-scored-for="25" points-scored-against="9"/>
              <rank value="4"/>
            </player-stats>
          </player>
        </team>
        <!-- Other fencers omitted for brevity -->
      </standing>

      <!-- A sports event is one bout within that round -->
      <!-- Note that the 'heat-number' attribute is actually the poule number and will be non-unique -->
      <sports-event>
        <event-metadata start-date-time="yyyymmddThhmmss+hhmm"
          event-key="XYZ.poule-1.bout-1"
          event-name="Scottish Open Mens Foil 2016"
          heat-number="1"
          date-coverage-type="event"
          event-status="post-event"
          season-key="2015-16"/>
        <player>
          <player-metadata player-key="fencing.fencer:519" team-key="fencing.club:26" team-idref="SALLH">
            <name last="McEwan" first="Mike" full="Mike McEwan"/>
          </player-metadata>
          <!-- In fencing the victor‘s score is ordinarily marked as 'V', rather than the actual score they got.
                This is to cope with the situation in epee where the score is 4-4 (or 14-14) and a double is scored,
                giving both fencers the maximum score. Another point is fenced and the winner gets 'V' and the looser
                5 (or 15). This does have an impact on calculations used to determine positions. -->
          <player-stats score="3"/>
        </player>
        <player>
          <player-metadata player-key="fencing.fencer:16275" team-key="fencing.club:26" team-idref="SALLH">
            <name last="Bradie" first="Angus" full="Angus Bradie"/>
          </player-metadata>
          <player-stats score="V"/>
        </player>
      </sports-event>

      <sports-event>
        <event-metadata start-date-time="yyyymmddThhmmss+hhmm"
          event-key="XYZ.poule-1.bout-2"
          event-name="Scottish Open Mens Foil 2016"
```

```
          heat-number="1"
          date-coverage-type="event"
          event-status="post-event"
          season-key="2015-16"/>
        <player>
          <player-metadata player-key="fencing.fencer:519" team-key="fencing.club:26" team-idref="SALLH">
            <name last="McEwan" first="Mike" full="Mike McEwan"/>
          </player-metadata>
          <player-stats score="V"/>
        </player>
        <player>
          <player-metadata player-key="fencing.fencer:16275" team-key="fencing.club:26" team-idref="SALLH">
            <name last="Crawford" first="Angus" full="Angus Crawford"/>
          </player-metadata>
          <player-stats score="0"/>
        </player>
      </sports-event>

      <!-- other bouts omitted for brevity -->
    </tournament-stage>

    <!-- This particular competition only had one round of poules so the next round is the DE -->
    <tournament-stage>
      <!-- This is an example of a DE round -->
      <tournament-stage-metadata stage-status="post-event" stage-key="2" stage-name="DE Last 32"/>
      <!--
          These are the standings at the END of this round of DE, as such fencers that have already been
          eliminated will not change position (i.e. if we‘re at the quarter final, 8 fencers are left in
          the competition so their positions can change, but everything from 9 down will be the same as
          it was for the last round.
      -->
      <standing content-label="Semi-Final">
        <standing-metadata/>

        <player>
          <player-metadata player-key="fencing.fencer:1" team-key="fencing.club:1" team-idref="HOLY">
            <name last="Cook" first="Keith" full="Keith Cook"/>
            <home-location country="GBR"/>
          </player-metadata>
          <player-stats>
            <rank value="1"/>
          </player-stats>
        </player>

        <player>
          <player-metadata player-key="fencing.fencer:23" team-key="fencing.club:23" team-idref="WALL">
            <name last="Woollard" first="Jonathan" full="Jonathan Woollard"/>
            <home-location country="GBR"/>
          </player-metadata>
          <player-stats>
            <rank value="2"/>
          </player-stats>
        </player>

        <player>
          <player-metadata player-key="fencing.fencer:24" team-key="fencing.club:24" team-idref="SALEH">
            <name last="Russell" first="Iain" full="Iain Russell"/>
            <home-location country="GBR"/>
          </player-metadata>
          <player-stats>
            <rank value="3"/>
          </player-stats>
        </player>

        <player>
          <player-metadata player-key="fencing.fencer:26" team-key="fencing.club:26" team-idref="SALLH">
            <name last="Douglas" first="Jack" full="Jack Douglas"/>
          </player-metadata>
          <player-stats>
            <rank value="4"/>
          </player-stats>
        </player>
        <!-- Other fencers omitted for brevity -->

      </standing>

      <sports-event>
        <event-metadata start-date-time="yyyymmddThhmmss+hhmm"
          event-key="XYZ.de-last16.bout-1"
          event-name="Scottish Open Mens Foil 2016"
          date-coverage-type="event" event-status="post-event" season-key="2015-16"/>
        <player>
          <player-metadata player-key="fencing.fencer:1" team-key="fencing.club:1" team-idref="HOLY">
            <name last="Cook" first="Keith" full="Keith Cook"/>
            <home-location country="GBR"/>
          </player-metadata>
          <player-stats score="V"/>
        </player>
        <player>
          <player-metadata player-key="fencing.fencer:26" team-key="fencing.club:26" team-idref="SALLH">
            <name last="Douglas" first="Jack" full="Jack Douglas"/>
          </player-metadata>
          <player-stats score="10"/>
        </player>
      </sports-event>

      <sports-event>
        <event-metadata start-date-time="yyyymmddThhmmss+hhmm"
          event-key="XYZ.de-last16.bout-1"
          event-name="Scottish Open Mens Foil 2016"
          date-coverage-type="event" event-status="post-event" season-key="2015-16"/>
        <player>
          <player-metadata player-key="fencing.fencer:23" team-key="fencing.club:23" team-idref="WALL">
            <name last="Woollard" first="Jonathan" full="Jonathan Woollard"/>
            <home-location country="GBR"/>
          </player-metadata>
          <player-stats score="3"/>
        </player>
        <player>
          <player-metadata player-key="fencing.fencer:24" team-key="fencing.club:24" team-idref="SALEH">
            <name last="Russell" first="Iain" full="Iain Russell"/>
            <home-location country="GBR"/>
          </player-metadata>
          <player-stats score="V"/>
        </player>
      </sports-event>

      <!-- Semi-final so only two bouts -->

    </tournament-stage>
    <!-- Final would feature here, along with the final standing that is the overall result of the competition. -->

  </tournament-division>
  <!--
      Additional tournament-division elements would feature here, in this particular competition they would be for:
      Womens Foil
      Mens Epee
      Womens Epee
      Mens Sabre
      Womens Sabre
  -->
</tournament>
</sports-content>
```

Listing B.2: ../SportsML–G2/testing/example–2–open–2.xml

C Build Procedure

This section is mainly for my own benefit and documents how to do a full build (including building this report document) of the FencingArchive project.

1. Generate Javadocs

```
cd ~/workspace/FencingArchive
javadoc -d ./static/javadoc -sourcepath ./src/main/java net.fencingarchive
javadoc -d ./static/javadoc -sourcepath ./src/main/java sportsml
```

Listing C.1: Javadoc Generation

2. Maven build of FencingArchive project (either versioned or not, deployed or not)

```
mvn {clean} {release:update-versions} package {tomcat7:redeploy}
```

Listing C.2: Maven Build

3. Generate effective class diagrams
   (a) In Eclipse, navigate to TM470/src/ObjectAid and create new ObjectAid class diagram. All options should be selected, also select the option to save the diagram as png along with the ucls file.
   (b) Drag and drop all of the *.java* files from the *net.fencingarchive* package into the newly-created diagram.
   (c) Move the classes such that relationships don't overlap classes or other relationships. Relationships can be re-routed by right-clicking on the relationship and selecting *AutoRoute*.
   (d) Repeat stages 3a to 3c for each package that a class diagram is needed for.
   (e) Update *models.tex* to include all class diagram png files.
4. Generate effective database ER diagram
   (a) Open MySQL Workbench and connect to the correct MySQL instance and database.
   (b) Select *Database-ReverseEngineer*
   (c) Follow the wizard.
   (d) Rearrange the diagram to remove overlaps, if necessary. As much as possible, try to follow the layout of the associated class diagram to make for more straight-forward visual comparison.
5. Make sure that *source−code.tex* contains all the source files that need to be included in the report.

6. Download some output XML from the API to a local file to include in the final report. Make sure that this file is included in the relevant *tex* file.

```
cd ~/workspace/TM470/...
wget -O event-1.xml http://127.0.0.1:8080/FencingArchive/api/event/1
```

Listing C.3: API Downloads