

## Sport Prediction System

Generated by Doxygen 1.9.1



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 CSVReader< M > Class Template Reference	5
3.1.1 Member Function Documentation	5
3.1.1.1 GetMatchDataFromCsvFile()	5
3.2 CSVWriter< M, P > Class Template Reference	6
3.3 EmailService Class Reference	6
3.3.1 Detailed Description	6
3.3.2 Constructor & Destructor Documentation	6
3.3.2.1 EmailService()	6
3.3.3 Member Function Documentation	7
3.3.3.1 SendEmail()	7
3.4 FootballMatch Class Reference	7
3.5 FootballMatchFactory Class Reference	8
3.6 FootballPrediction Class Reference	8
3.7 IMatchFactory< M > Interface Template Reference	9
3.8 Match Class Reference	9
3.8.1 Detailed Description	10
3.9 Member< P, M > Class Template Reference	10
3.9.1 Detailed Description	11
3.9.2 Member Function Documentation	11
3.9.2.1 SearchPredictionDone()	11
3.9.3 Member Data Documentation	11
3.9.3.1 Scores	12
3.10 Prediction Class Reference	12
3.11 PredictionGame Class Reference	12
3.11.1 Detailed Description	13
3.11.2 Constructor & Destructor Documentation	13
3.11.2.1 PredictionGame()	13
3.11.3 Member Function Documentation	13
3.11.3.1 Register()	13
3.11.3.2 SendDailyEmail()	14
3.11.3.3 Unsubscribe()	14
3.11.4 Property Documentation	14
3.11.4.1 PredictionGameID	14
3.12 Schedule< M > Class Template Reference	14
3.12.1 Detailed Description	15
3.13 Score Class Reference	15



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

CSVReader< M > . . . . .	5
CSVWriter< M, P > . . . . .	6
EmailService . . . . .	6
IMatchFactory< M > . . . . .	9
IMatchFactory< FootballMatch > . . . . .	9
FootballMatchFactory . . . . .	8
Match . . . . .	9
FootballMatch . . . . .	7
Member< P, M > . . . . .	10
Prediction . . . . .	12
FootballPrediction . . . . .	8
PredictionGame . . . . .	12
Schedule< M > . . . . .	14
Score . . . . .	15



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">CSVReader&lt; M &gt;</a>	5
<a href="#">CSVWriter&lt; M, P &gt;</a>	6
<a href="#">EmailService</a>	
Provides functionality to send emails using SMTP	6
<a href="#">FootballMatch</a>	7
<a href="#">FootballMatchFactory</a>	8
<a href="#">FootballPrediction</a>	8
<a href="#">IMatchFactory&lt; M &gt;</a>	9
<a href="#">Match</a>	
Abstract match class to set a frame for sport-specific kinds of matches	9
<a href="#">Member&lt; P, M &gt;</a>	
Represents a member participating in the Sport <a href="#">Prediction</a> System (SPS)	10
<a href="#">Prediction</a>	12
<a href="#">PredictionGame</a>	
Represents a prediction game in the Sport <a href="#">Prediction</a> System (SPS)	12
<a href="#">Schedule&lt; M &gt;</a>	
Generic class <a href="#">Schedule</a> which represents a tournament	14
<a href="#">Score</a>	15





## Chapter 3

# Class Documentation

### 3.1 CSVReader< M > Class Template Reference

#### Static Public Member Functions

- static void **SetMatchFactory** ([IMatchFactory](#)< M > match\_factory)
- static string[] [GetMatchDataFromCsvFile](#) (string PathToMatchDataCsvFile, int line\_number)  
*Reads the match data from a CSV file.*
- static List< M > **GetScheduleFromCsvFile** (string PathToCsvFile, SportsTypes sport\_type)

#### 3.1.1 Member Function Documentation

##### 3.1.1.1 GetMatchDataFromCsvFile()

```
static string [] CSVReader< M >.GetMatchDataFromCsvFile (  
    string PathToMatchDataCsvFile,  
    int line_number ) [inline], [static]
```

Reads the match data from a CSV file.

#### Parameters

<i>PathToMatchDataCsvFile</i>	The path to the CSV file containing match data.
<i>MatchID</i>	The unique identifier of the match.

#### Returns

An array of strings containing the match data.

The documentation for this class was generated from the following file:

- src/ClassLib/CSVReader.cs

## 3.2 CSVWriter< M, P > Class Template Reference

### Static Public Member Functions

- static void **UpdateSchedule** (string PathToCsvFile, List< M > schedule)
- static void **DeleteFile** (string PathToCsvFile)
- static void **WriteMemberData** (string PathToCsvFile, List< [Member](#)< P, M >> members)
- static void **TrackScoreData** (string PathToCsvFile, List< [Member](#)< P, M >> members)

The documentation for this class was generated from the following file:

- src/ClassLib/CSVWriter.cs

## 3.3 EmailService Class Reference

Provides functionality to send emails using SMTP.

### Public Member Functions

- [EmailService](#) ()  
*Initializes a new instance of the [EmailService](#) class.*
- void [SendEmail](#) (string Recipient, string Sender, string Subject, string Content)  
*Sends an email.*

### 3.3.1 Detailed Description

Provides functionality to send emails using SMTP.

### 3.3.2 Constructor & Destructor Documentation

#### 3.3.2.1 EmailService()

```
EmailService.EmailService ( ) [inline]
```

Initializes a new instance of the [EmailService](#) class.

#### Parameters

<i>smtpServer</i>	The SMTP server address.
<i>smtpPort</i>	The SMTP server port.
<i>username</i>	The username for SMTP authentication.
<i>password</i>	The password for SMTP authentication.

### 3.3.3 Member Function Documentation

#### 3.3.3.1 SendEmail()

```
void EmailService.SendEmail (
    string Recipient,
    string Sender,
    string Subject,
    string Content ) [inline]
```

Sends an email.

##### Parameters

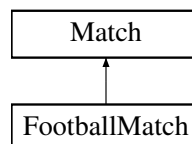
<i>recipient</i>	The recipient's email address.
<i>subject</i>	The subject of the email.
<i>content</i>	The content of the email.

The documentation for this class was generated from the following file:

- src/ClassLib/EmailService.cs

## 3.4 FootballMatch Class Reference

Inheritance diagram for FootballMatch:



### Public Member Functions

- [FootballMatch \(\)](#)  
Parameter free constructor to use generic type *M* in [CSVReader](#)
- **FootballMatch** (string PathToMatchDataCsvFile, int line\_number, SportsTypes sport\_type)
- override string **ToString** ()

### Properties

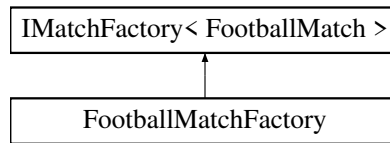
- string? **HomeTeam** [get]
- string? **AwayTeam** [get]
- byte? **ResultHomeTeamPenalties** [get]
- byte? **ResultAwayTeamPenalties** [get]

The documentation for this class was generated from the following file:

- src/ClassLib/FootballMatch.cs

### 3.5 FootballMatchFactory Class Reference

Inheritance diagram for FootballMatchFactory:



#### Public Member Functions

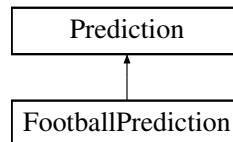
- [FootballMatch](#) **CreateMatch** (string PathToMatchDataCsvFile, int line\_number, SportsTypes sport\_type)

The documentation for this class was generated from the following file:

- src/ClassLib/MatchFactory.cs

### 3.6 FootballPrediction Class Reference

Inheritance diagram for FootballPrediction:



#### Public Member Functions

- **FootballPrediction** (uint MemberID, [FootballMatch](#) football\_match, DateTime predictionDate, byte prediction\_home, byte prediction\_away)
- void **ChangePrediction** (byte? NewPredictionHome, byte? NewPredictionAway)
- override string **ToString** ()

#### Properties

- byte **PredictionHome** [get, set]
- byte **PredictionAway** [get, set]

The documentation for this class was generated from the following file:

- src/ClassLib/FootballPrediction.cs

## 3.7 IMatchFactory< M > Interface Template Reference

### Public Member Functions

- M **CreateMatch** (string PathToMatchDataCsvFile, int line\_number, SportsTypes sport\_type)

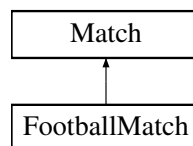
The documentation for this interface was generated from the following file:

- src/ClassLib/MatchFactory.cs

## 3.8 Match Class Reference

Abstract match class to set a frame for sport-specific kinds of matches.

Inheritance diagram for Match:



### Public Member Functions

- override int **GetHashCode** ()
- bool **Team1Won** ()
- bool **Team2Won** ()
- bool **Tie** ()

### Properties

- uint **MatchID** [get, protected set]  
*Unique identifier for the match.*
- DateTime **MatchDate** [get, protected set]  
*Date and time when the match takes place.*
- byte? **ResultTeam1** [get, protected set]  
*Result of team 1 in the match.*
- byte? **ResultTeam2** [get, protected set]  
*Result of team 2 in the match.*
- SportsTypes **SportsType** [get, protected set]
- string[] **MatchArray** [get, protected set]

### 3.8.1 Detailed Description

Abstract match class to set a frame for sport-specific kinds of matches.

This class provides the basic structure for any kind of sports match, including common properties like MatchID, MatchDate, and Results.

The documentation for this class was generated from the following file:

- src/ClassLib/Match.cs

## 3.9 Member< P, M > Class Template Reference

Represents a member participating in the Sport [Prediction](#) System (SPS).

### Public Member Functions

- [Member](#) (string forename, string surname, string emailaddress, string password)  
*Initializes a new instance of the [Member](#) class.*
- override int **GetHashCode** ()
- void [AddParticipatingSchedule](#) ([Schedule](#)< M > schedule, ScheduleTypes schedule\_type)  
*Adds a schedule to the member's list of participating schedules.*
- void [RemoveParticipatingSchedule](#) (ScheduleTypes schedule\_type)  
*Removes a schedule from the member's list of participating schedules.*
- void [AddPredictionToDo](#) ()  
*Adds a prediction to the member's list of predictions to do.*
- void [RemovePredictionToDo](#) (uint MatchID)  
*Removes a prediction from the member's list of predictions to do.*
- [Prediction SearchPredictionDone](#) (uint PredictionID)  
*Searches for a specific prediction in the member's list.*
- void **AddPrediction** ()
- void **CalculateScores** ()
- override string **ToString** ()

### Protected Attributes

- List< [Score](#) > [Scores](#)  
*List of Scores.*

## Properties

- uint **MemberID** [get]  
*Gets the unique ID of the member.*
- string? **forename** [get, set]
- string? **surname** [get, set]
- string **EmailAddress** [get]
- string **Password** [get]
- List< Schedule< M > > **ParticipatingSchedules** [get]  
*List of Schedules the member chose to participate predicting.*
- List< M > **PredictionsToDo** [get]  
*List of Matches, which need to be predicted on the specific day.*
- List< P > **PredictionsDone** [get]  
*List, which contains all Predictions where the match is already predicted, but a score was not calculated yet.*
- List< P > **ArchivedPredictions** [get]  
*List, which contains all Predictions where no score must be calculated anymore.*
- List< Score > **GetScores** [get]

### 3.9.1 Detailed Description

Represents a member participating in the Sport [Prediction](#) System (SPS).

#### Type Constraints

**P** : [Prediction](#)

**M** : [Match](#)

### 3.9.2 Member Function Documentation

#### 3.9.2.1 SearchPredictionDone()

```
Prediction Member< P, M >.SearchPredictionDone (
    uint PredictionID ) [inline]
```

Searches for a specific prediction in the member's list.

#### Returns

The prediction if found, otherwise null.

### 3.9.3 Member Data Documentation

### 3.9.3.1 Scores

```
List<Score> Member< P, M >.Scores [protected]
```

List of Scores.

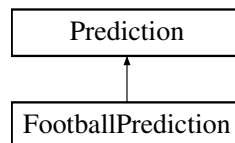
There is exactly one score for every schedule the member predicts.

The documentation for this class was generated from the following file:

- src/ClassLib/Member.cs

## 3.10 Prediction Class Reference

Inheritance diagram for Prediction:



### Public Member Functions

- **Prediction** (uint member\_id, Match predicted\_match, DateTime predictionDate)
- bool **ValidatePrediction** ()
- override int **GetHashCode** ()
- override string **ToString** ()

### Properties

- uint **PredictionID** [get]
- uint **MemberID** [get]
- Match **PredictedMatch** [get]
- DateTime **PredictionDate** [get, set]

The documentation for this class was generated from the following file:

- src/ClassLib/Prediction.cs

## 3.11 PredictionGame Class Reference

Represents a prediction game in the Sport Prediction System (SPS).



## Public Member Functions

- [PredictionGame](#) ([EmailService](#) emailService)  
*Initializes a new instance of the [PredictionGame](#) class.*
- void [Register](#) ([Member](#)< [Prediction](#), [Match](#) > member)  
*Registers a new member to the prediction game.*
- void [Unsubscribe](#) (int MemberID)  
*Unsubscribes a member from the prediction game.*
- void [SendDailyEmail](#) ()  
*Sends a daily email to all members with the matches that need to be predicted.*

## Properties

- uint [PredictionGameID](#) [get]  
*Gets the unique ID of the prediction game.*
- static List< [ScheduleTypes](#) > **ScheduleTypes** [get, set]

### 3.11.1 Detailed Description

Represents a prediction game in the Sport [Prediction](#) System (SPS).

### 3.11.2 Constructor & Destructor Documentation

#### 3.11.2.1 PredictionGame()

```
PredictionGame.PredictionGame (  
    EmailService emailService ) [inline]
```

Initializes a new instance of the [PredictionGame](#) class.

### 3.11.3 Member Function Documentation

#### 3.11.3.1 Register()

```
void PredictionGame.Register (  
    Member< Prediction, Match > member ) [inline]
```

Registers a new member to the prediction game.

### 3.11.3.2 SendDailyEmail()

```
void PredictionGame.SendDailyEmail ( ) [inline]
```

Sends a daily email to all members with the matches that need to be predicted.

### 3.11.3.3 Unsubscribe()

```
void PredictionGame.Unsubscribe (
    int MemberID ) [inline]
```

Unsubscribes a member from the prediction game.

## 3.11.4 Property Documentation

### 3.11.4.1 PredictionGameID

```
uint PredictionGame.PredictionGameID [get]
```

Gets the unique ID of the prediction game.

The documentation for this class was generated from the following file:

- src/ClassLib/PredictionGame.cs

## 3.12 Schedule< M > Class Template Reference

Generic class [Schedule](#) which represents a tournament.

### Public Member Functions

- **Schedule** (string PathToCsvFile, SportsTypes sport\_type, ScheduleTypes schedule\_type)
- List< [FootballMatch](#) > **GetFootballScheduleFromCsvFile** (string PathToCsvFile, SportsTypes sport\_type)
- List< M > **GetMatchesOnDay** ()

### Properties

- ScheduleTypes **ScheduleID** [get]
- List< M >? **Matches** [get]

### 3.12.1 Detailed Description

Generic class [Schedule](#) which represents a tournament.

It contains a list of all the matches which take place during the tournament.

Added to that it also contains a list of all the matches on the specific day of the tournament.

#### Type Constraints

**M** : [Match](#)

The documentation for this class was generated from the following file:

- `src/ClassLib/Schedule.cs`

## 3.13 Score Class Reference

### Public Member Functions

- **Score** (ScheduleTypes predicted\_schedule)
- void **IncrementAmountOfPoints** (uint points)
- uint **CalculateFootballScore** ([FootballPrediction](#) prediction)
- override string **ToString** ()

### Properties

- ScheduleTypes **ScoreID** [get]

The documentation for this class was generated from the following file:

- `src/ClassLib/Score.cs`

