

# **Student Online Teaching Advice Notice**

**The materials and content presented within this session are intended solely for use in a context of teaching and learning at Trinity.**

**Any session recorded for subsequent review is made available solely for the purpose of enhancing student learning.**

**Students should not edit or modify the recording in any way, nor disseminate it for use outside of a context of teaching and learning at Trinity.**

**Please be mindful of your physical environment and conscious of what may be captured by the device camera and microphone during videoconferencing calls.**

**Recorded materials will be handled in compliance with Trinity's statutory duties under the Universities Act, 1997 and in accordance with the University's [policies and procedures](#).**

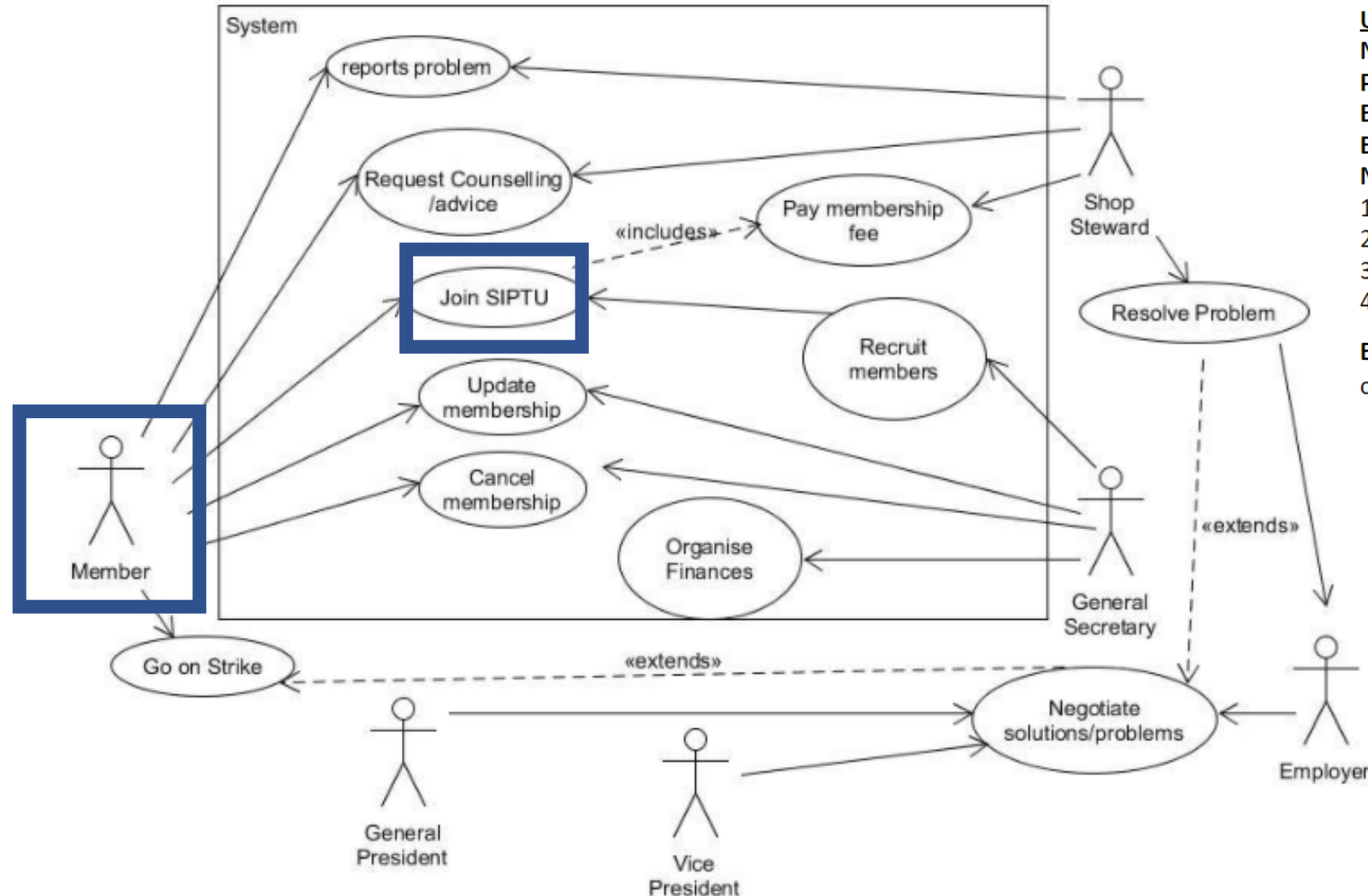
**Further information on data protection and best practice when using videoconferencing software is available at [https://www.tcd.ie/info\\_compliance/data-protection/](https://www.tcd.ie/info_compliance/data-protection/).**

**© Trinity College Dublin 2020**

# UML to XML

*Example: Trade Union SIPTU*

# UML Use Case Diagram – Trade Union SIPTU



## Use Case 1

**Name:** Join SIPTU

**Participating Actor:** Member

**Entry condition:** Potential member wants to join SIPTU

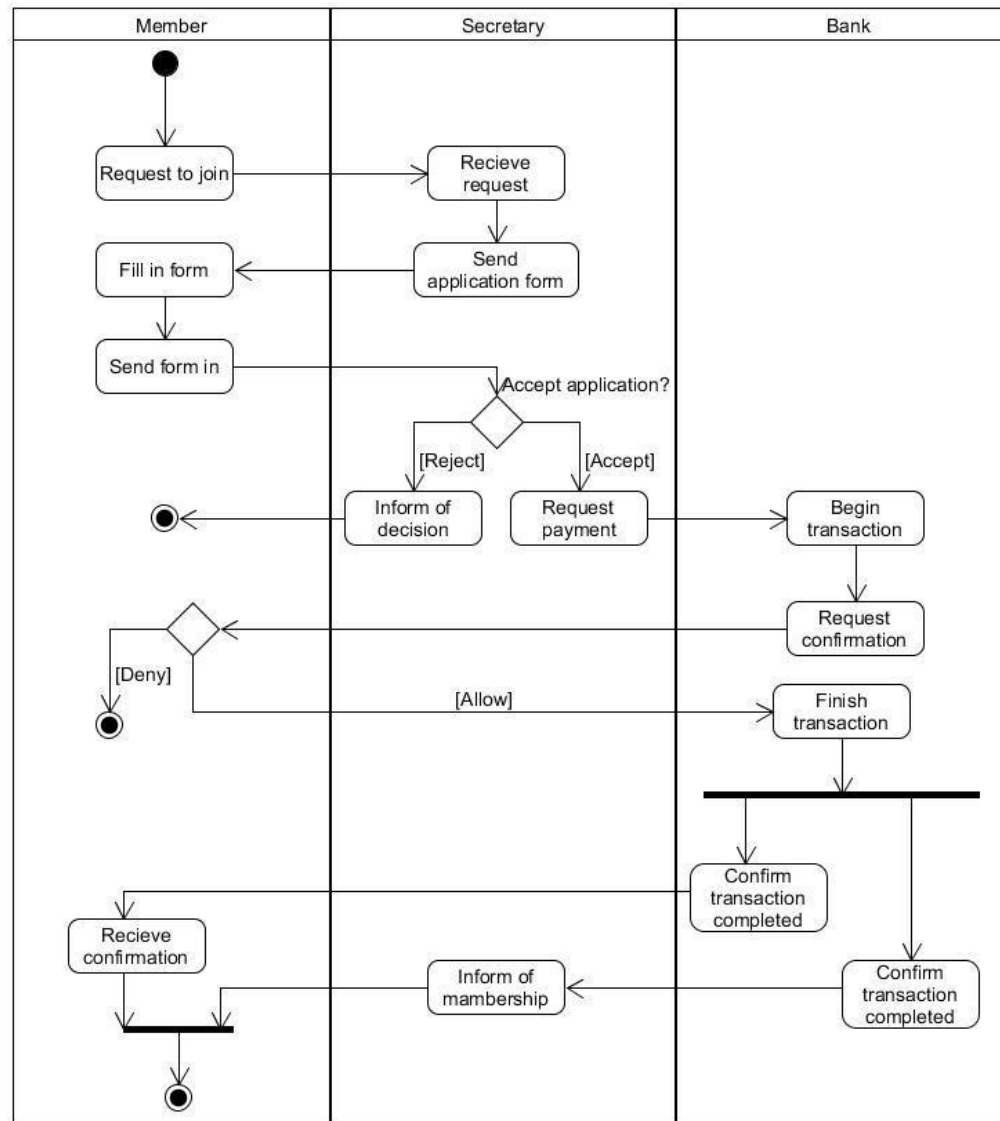
**Exit condition:** Member has successfully joined SIPTU

**Normal Scenario:**

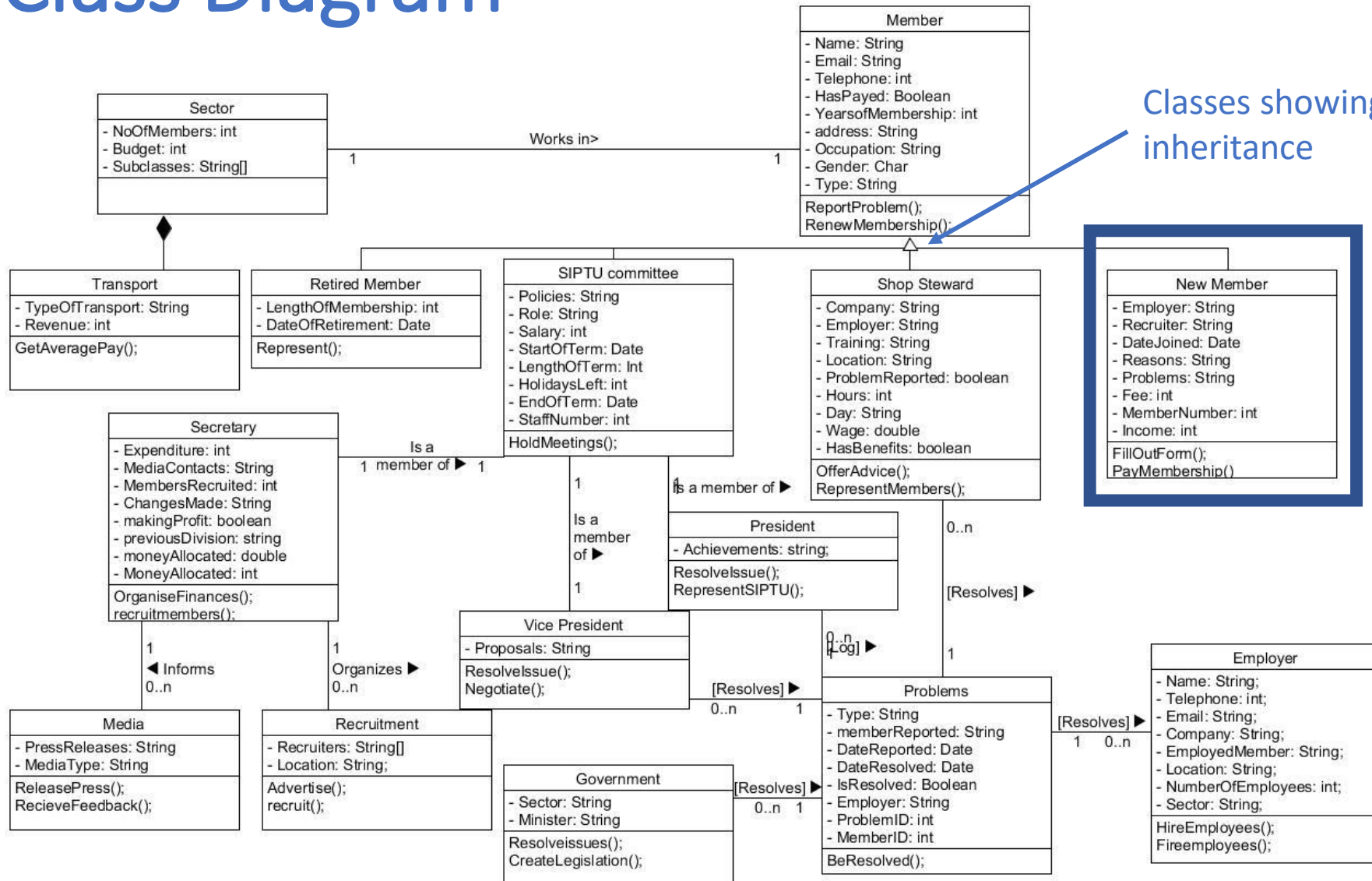
1. Potential Member goes online to SIPTU's website
2. They fill out and submit the application form
3. SIPTU receives the application and decide to let them join
4. Member meets the criteria and joins SIPTU

**Error Scenario:** The application is rejected as applicant does not meet the correct criteria.

# UML Activity Diagram – Join SIPTU



# UML Class Diagram

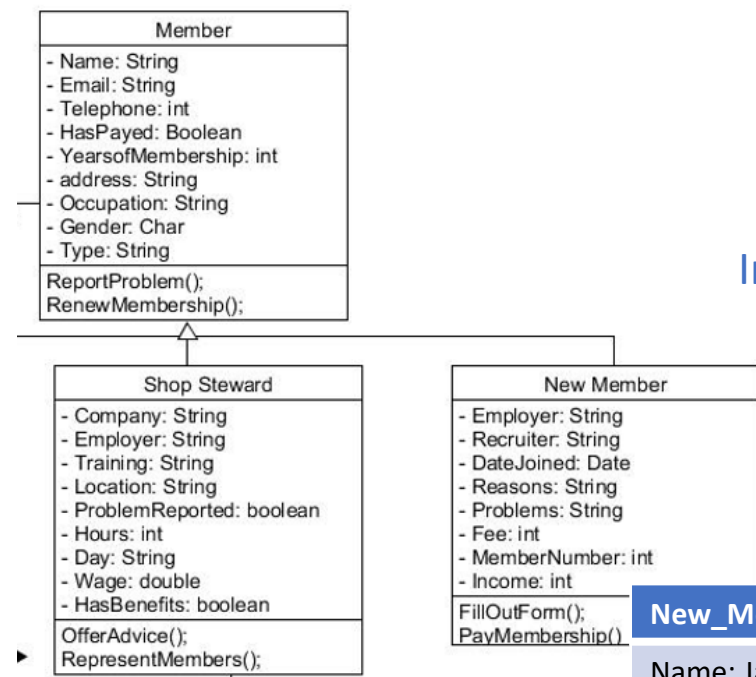




# XML Document

Member <- New Member

```
<Member_Info>
<New_Member Income="200">
  <Member Age="20">
    <Member.Name>Jack Douglas</Member.Name>
    <Member.Email>douglasj@tcd.ie</Member.Email>
    <Member.Telephone>0871234567</Member.Telephone>
    <Member.yearsOfMembership>0</Member.yearsOfMembership>
    <Member.EirCode>D1</Member.EirCode>
    <Member.Occupation>Train Driver</Member.Occupation>
    <Member.Gender>M</Member.Gender>
    <Member.typeOfMembership>New Member</Member.typeOfMembership>
    <Member.haspaid Iban="20">True</Member.haspaid>
  </Member>
  <New_Member.Employer>Irish Rail </New_Member.Employer>
  <New_Member.Recruiter RecruitedMemberNumber = "264">Martha Kelly</New_Member.Recruiter>
  <New_Member.Date_Joined>05/12/2017</New_Member.Date_Joined>
  <New_Member.Reasons_For_Joining>Compensation </New_Member.Reasons_For_Joining>
  <New_Member.Problem_Reported>Discrimination at work </New_Member.Problem_Reported>
  <New_Member.Subscription_Fees>20</New_Member.Subscription_Fees>
</New_Member>
</Member_Info>
```



Instantiation of New Member class

New\_Member

Name: Jack Douglas  
Email: [douglasj@tcd.ie](mailto:douglasj@tcd.ie)  
Telephone: 0871234567  
haspaid:False  
yearsOfMembership: 0  
Eircode: D1 \*  
Occupation: Train Driver  
Gender: Male  
typeOfMembership: New Member  
Employer: Irish Rail  
Recruiter: Martha Kelly  
DateJoined: 05/12/2017  
Reasons\_For\_Joining: Compensation  
Problems\_Reported: Discrimination at work  
Subscription\_Fees: 20  
MemberNumber 264  
Income: 30000  
Age= 20 \*

\*changed from/added to UML

# DTD Document

## Member <- New Member

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Member_Info
[<!ELEMENT Member_Info (New_Member*)>
<!ELEMENT New_Member
(Member.Name,Member.Email,Member.Telephone,Member.yearsOfMembership,Member.EirCode,Member.Occupation,Member.Gender,Member.typeOfMembership,Member.haspaid,New_Member.Employer,New_Member.Recruiter,New_Member.Date_Joined,New_Member.Reasons_For_Joining,New_Member.Problem_Reported,New_Member.Subscription_Fees)
<!ELEMENT Member.Name (#PCDATA)>
<!ELEMENT Member.Email (#PCDATA)>
<!ELEMENT Member.Telephone (#PCDATA)>
<!ELEMENT Member.yearsOfMembership (#PCDATA)>
<!ELEMENT Member.EirCode (#PCDATA)>
<!ELEMENT Member.Occupation (#PCDATA)>
<!ELEMENT Member.Gender (#PCDATA)>
<!ELEMENT Member.typeOfMembership (#PCDATA)>
<!ELEMENT Member.haspaid (#PCDATA)>
<!ATTLIST New_Member Age CDATA #REQUIRED>
<!ATTLIST Member.haspaid Iban CDATA #REQUIRED>
<!ELEMENT New_Member.Employer (#PCDATA)>
<!ELEMENT New_Member.Recruiter (#PCDATA)>
<!ELEMENT New_Member.Date_Joined (#PCDATA)>
<!ELEMENT New_Member.Reasons_For_Joining (#PCDATA)>
<!ELEMENT New_Member.Problem_Reported (#PCDATA)>
<!ELEMENT New_Member.Subscription_Fees (#PCDATA)>
<!ATTLIST New_Member.Recruiter RecruitedMemberNumber CDATA #REQUIRED>
<!ATTLIST New_Member Income CDATA #REQUIRED>]>
```

```
classDiagram
    class Member {
        Name: String
        Email: String
        Telephone: int
        HasPaid: Boolean
        YearsofMembership: int
        address: String
        Occupation: String
        Gender: Char
        Type: String
        ReportProblem()
        RenewMembership()
    }
    class ShopSteward {
        Company: String
        Employer: String
        Training: String
        Location: String
        ProblemReported: boolean
        Hours: int
        Day: String
        Wage: double
        HasBenefits: boolean
        OfferAdvice()
        RepresentMembers()
    }
    class NewMember {
        Employer: String
        Recruiter: String
        DateJoined: Date
        Reasons: String
        Problems: String
        Fee: int
        MemberNumber: int
        Income: int
        FillOutForm()
        PayMembership()
    }
    Member <|-- ShopSteward
    Member <|-- NewMember
```

The diagram illustrates the relationship between three classes: Member, Shop Steward, and New Member. Member is the base class, with Shop Steward and New Member as subclasses. Member has attributes: Name (String), Email (String), Telephone (int), HasPaid (Boolean), YearsofMembership (int), address (String), Occupation (String), Gender (Char), and Type (String). It has methods: ReportProblem() and RenewMembership(). Shop Steward has attributes: Company (String), Employer (String), Training (String), Location (String), ProblemReported (boolean), Hours (int), Day (String), Wage (double), and HasBenefits (boolean). It has methods: OfferAdvice() and RepresentMembers(). New Member has attributes: Employer (String), Recruiter (String), DateJoined (Date), Reasons (String), Problems (String), Fee (int), MemberNumber (int), and Income (int). It has methods: FillOutForm() and PayMembership(). Blue arrows connect the DTD elements to the corresponding attributes in the classes.

# XQuery

```
declare function local:all_members($c as xs:string?)
{for $s in
doc("C:\Users\gstephen\Desktop\WIP\information
Management\2021\XML\SethsXML\Seth's
ExampleRevised.xml")/Member_Info/New_Member
where contains($s,$c)
return
<New_Member>
{$s/Member.Name}
{$s/Member.Email}
{$s/Member.yearsOfMembership}
{$s/Member.Occupation}
{$s/Member.Gender}
{$s/Member.typeOfMembership}
{$s/Member.haspaid}
{$s/New_Member.Employer}
{$s/New_Member.Recruiter}
{$s/New_Member.DateJoined}
</New_Member>
};

<all>{local:all_members("Jack Douglas")}</all>
```

# Result:

```
<all>
  <New_Member>
    <Member.Name>Jack Douglas</Member.Name>
    <Member.Email>douglasj@tcd.ie</Member.Email>
    <Member.yearsOfMembership>0</Member.yearsOfMembership>
    <Member.Occupation>Train Driver</Member.Occupation>
    <Member.Gender>M</Member.Gender>
    <Member.typeOfMembership>New Member</Member.typeOfMembership>
    <Member.haspaid Iban="20">True</Member.haspaid>
    <New_Member.Employer>Irish Rail</New_Member.Employer>
    <New_Member.Recruiter RecruitedMemberNumber="264">Martha Kelly</New_Member.Recruiter>
  </New_Member>
</all>
```



# Thank You!