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.

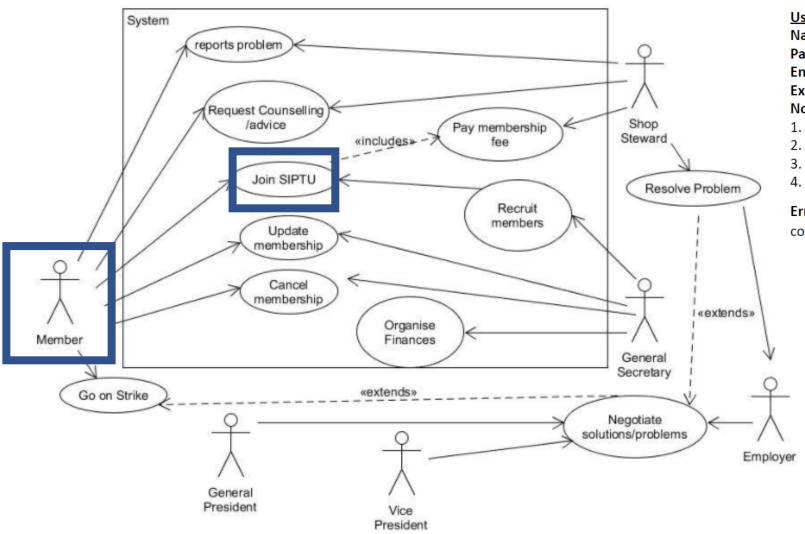
Further information on data protection and best practice when using videoconferencing software is available at https://www.tcd.ie/info_compliance/data-protection/.

© 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

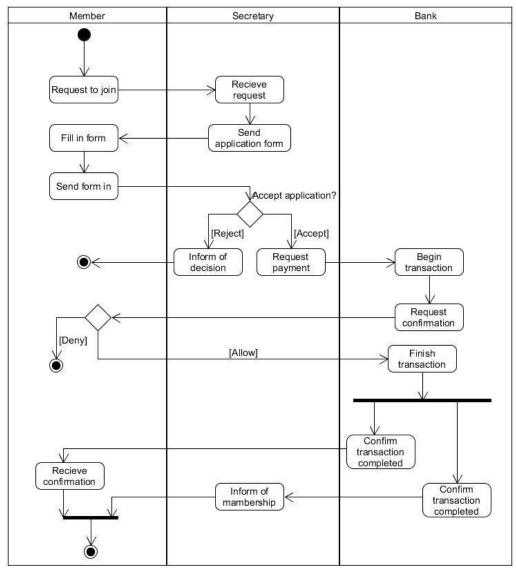
Entry condition: Potential member wants to join SIPTU **Exit condition:** Member has successfully joined SIPTU

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 Member Name: String Email: String Telephone: int HasPaved: Boolean Classes showing Sector - YearsofMembership: int NoOfMembers: int address: String Works in> inheritance Budget: int Occupation: String Gender, Char Subclasses: String[] - Type: String ReportProblem(): RenewMembership() SIPTU committee Transport Retired Member Shop Steward New Member Policies: String TypeOfTransport: String LengthOfMembership: int Company: String Employer: String Role: String - Revenue: int - DateOfRetirement: Date Employer: String Recruiter: String Salary: int Training: String DateJoined: Date GetAveragePay(); Represent(); StartOfTerm: Date Location: String Reasons: String LengthOfTerm: Int ProblemReported: boolean Problems: String HolidaysLeft: int Hours: int Fee: int EndOfTerm: Date Day: String MemberNumber: int StaffNumber: int Secretary Wage: double Income: int HoldMeetings(); Isa HasBenefits: boolean Expenditure: int FillOutForm(); MediaContacts: String 1 member of ▶ 1 OfferAdvice(): PayMembership() MembersRecruited: int Is a member of ▶ RepresentMembers(): ChangesMade: String - makingProfit: boolean ls a President 0..n member previousDivision: string - Achievements: string; monevAllocated: double of > MoneyAllocated: int Resolvelssue(); RepresentSIPTU(): [Resolves] > OrganiseFinances(): recruitmembers(); Vice President 0.ngl ▶ - Proposals: String Employer ◀ Informs Organizes > ResolveIssue(); Name: String; Negotiate(); [Resolves] ▶ Problems Telephone: int; 0..n Type: String - Email: String; [Resolves] ▶ Media Recruitment memberReported: String Company: String; Recruiters: String[] PressReleases: String DateReported: Date - EmployedMember: String: Location: String; MediaType: String DateResolved: Date Location: String: Government - IsResolved: Boolean Advertise(); NumberOfEmployees: int; ReleasePress(): [Resolves] - Sector: String Employer: String Sector: String; RecieveFeedback(); recruit(); - Minister: String ProblemID: int HireEmployees(); MemberID: int Resolveissues(); Fireemployees(); BeResolved(): CreateLegislation();

XML Document

</Member Info>

Member <- New Member

```
Shop Steward
                                                                                    Company: String
                                                                                    Employer: String
<Member Info>
                                                                                    Training: String
                                                                                    Location: String
<New Member Income="200">
                                                                                    ProblemReported: boolean
              <Member Age="20">
                                                                                    Hours: int
                                                                                    Day: String
              <Member.Name>Jack Douglas</Member.Name>
                                                                                    Wage: double
              <Member.Email>douglasj@tcd.ie</Member.Email>
                                                                                    HasBenefits: boolean
              <Member.Telephone>0871234567</Member.Telephone>
                                                                                   OfferAdvice():
                                                                                   RepresentMembers();
              <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>
```

Instantiation of New Member class

New Member

- Employer: String
- Recruiter: String
- Date Joined: Date
- Reasons: String
- Problems: String
- Fee: int
- MemberNumber: int
- Income: int

FillOutForm();
PavMembership()

New_Member

Name: Jack Douglas Email: 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

Member

Name: String

Email: String Telephone: int HasPayed: Boolean YearsofMembership: int address: String

Occupation: String Gender: Char Type: String

ReportProblem(); RenewMembership();

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,Ne
w_Member.Employer,New_Member.Recruiter,New_Member.Date_Joined,New_Member.Reasons_For_Joining,New Member.Problem Paparted Now Mamber Subscription Ease\>
<!ELEMENT Member.Name (#PCDATA)> 
                                                                                                                        Member
<!ELEMENT Member.Email (#PCDATA)>
                                                                                                                Name: String
<!ELEMENT Member.Telephone (#PCDATA)>
                                                                                                                 Email: String

    Telephone: int

<!ELEMENT Member.yearsOfMembership (#PCDATA)>
                                                                                                                HasPayed: Boolean
                                                                                                                YearsofMembership: int
<!ELEMENT Member.EirCode (#PCDATA)>
                                                                                                                address: String
<!ELEMENT Member.Occupation (#PCDATA)>
                                                                                                                Occupation: String
                                                                                                                Gender, Char
<!ELEMENT Member.Gender (#PCDATA)>

    Type: String

<!ELEMENT Member.typeOfMembership (#PCDATA)>
                                                                                                                ReportProblem();
<!ELEMENT Member.haspaid (#PCDATA)>
                                                                                                                RenewMembership():
<!ATTLIST New Member Age CDATA #REQUIRED>
<!ATTLIST Member.haspaid Iban CDATA #REQUIRED>
                                                                                                                       Shop Steward
                                                                                                                                                   New Member
<!ELEMENT New Member.Employer (#PCDATA)> <</pre>
                                                                                                                  Company: String
                                                                                                                                              Employer: String
                                                                                                                 Employer: String
                                                                                                                                              Recruiter: String
- Training: String
                                                                                                                                              DateJoined: Date
Location: String
                                                                                                                                              Reasons: String
                                                                                                                 ProblemReported: boolean
                                                                                                                                              Problems: String
<!ELEMENT New Member.Reasons For Joining (#PCDATA)>
                                                                                                                 Hours: int
                                                                                                                                              Fee: int
<!ELEMENT New Member.Problem Reported (#PCDATA)>
                                                                                                                 Day: String

    MemberNumber: int

                                                                                                                 Wage: double

    Income: int

<!ELEMENT New Member.Subscription Fees (#PCDATA)>

    HasBenefits: boolean

                                                                                                                                             FillOutForm();
<!ATTLIST New Member.Recruiter RecruitedMemberNumber CDATA #REQUIRED>
                                                                                                                 OfferAdvice():
                                                                                                                                             PayMembership(
<!ATTLIST New Member Income CDATA #REQUIRED>]>
                                                                                                                 RepresentMembers();
```

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:

```
<al>
<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 lban="20">True</member.haspaid>
<member.haspaid lban="20">True</member.haspaid>
<member.Employer>Irish Rail</member.Employer>
<member.Recruiter RecruitedMemberNumber="264">Member.Recruiter></member></member></member></member></mil></mr>
```

Thank You!