Faculty of Engineering and Environment

CM0720 - System Analysis and Design with UML

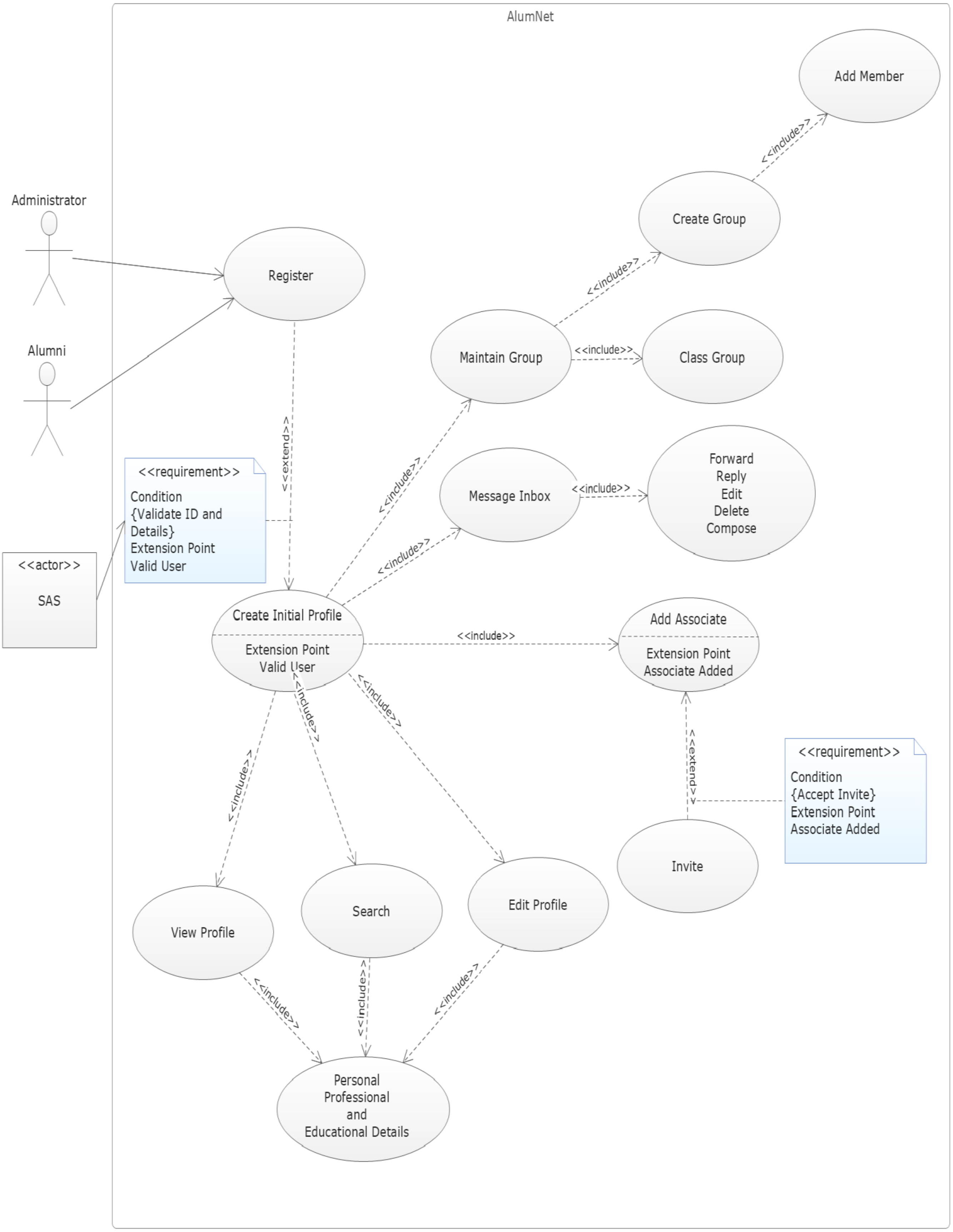
Assignment

Author: Pritesh Bhole

Student number: w14043450

Year: **2015/16**

Use Case Diagram


AlumNet
Administrator
Alumni
Register
<<extend>>
<<requirement>>
Condition
{Validate ID and
Details}
Extension Point
Valid User
<<actor>>
SAS
Create Initial Profile
Extension Point
Valid User
Maintain Group
<<include>>
Create Group
<<include>>
Add Member
<<include>>
Class Group
Message Inbox
<<include>>
Forward
Reply
Edit
Delete
Compose
<<include>>
Add Associate
Extension Point
Associate Added
<<extend>>
<<requirement>>
Condition
{Accept Invite}
Extension Point
Associate Added
Invite
View Profile
Search
Edit Profile
Personal
Professional
and
Educational Details

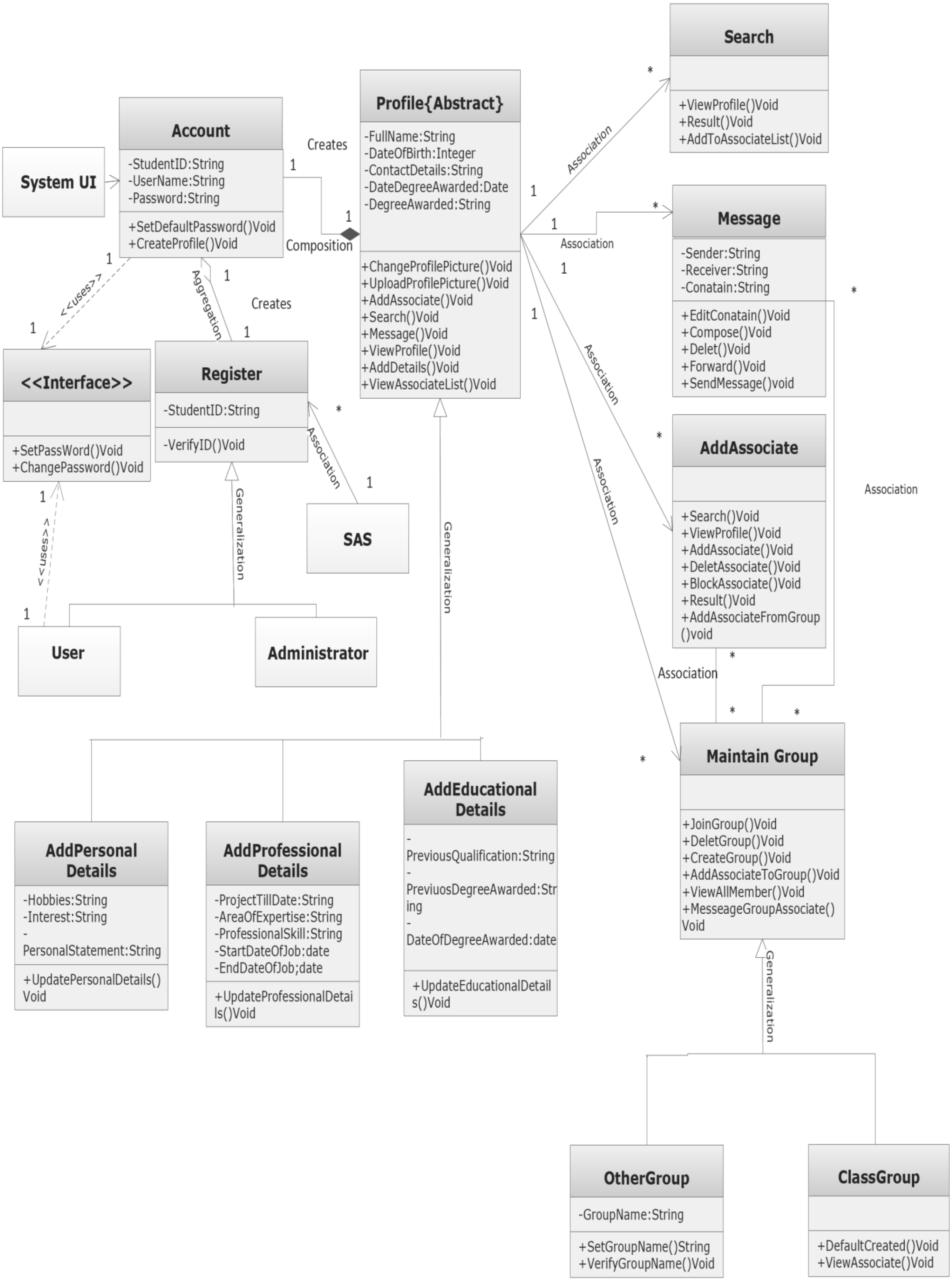
Use case is a method to analysis system in order to get system requirement. Use case diagram provides the functionality of a system that can be used by an external user (actor). In short use case diagram depicts what system needs to do.

In case of AlumnNet use of <<include>> and <<exclude>> was necessary as the system is huge and have lots of responsibilities. In use case <<extended>> is used at two places along with conditional requirement. First one between register and create initial profile. In this user is checked for authenticity by SAS and then proceeded to extension point of creating profile. This extended case is used to register valid user and condition that user must be register to create profile. Another one is while adding associate were user needs to invite associate and then there is a condition point, if associate accepts the invitation than will proceed to extension point associate added. Function of extend here is person can even reject invitation, thus providing the choice of communication. <<include>> function is used for all functionality such as search, view profile, edit profile as all of them are based on common details i.e. personal, professional, educational. As edit profile will include the function of editing these details and searching option will be based on these details, as well as if another person views profile he/she will see these details. As there can be many group one default class groups and other group which user can create, so both there functionality come under maintain group and also in new group user can add member hence it is in include function. For messaging functionality as user can edit, forward, send and delete a message. Profile will have all functionality in it hence all functionality is included in it.

Conclusion

Technique of extend and include are very helpful, but if generalization concept could have been used it would have made diagram less complicated.

Class Diagram


Search
+ViewProfile()Void
+Result()Void
+AddToAssociateList()Void
Profile{Abstract}
-FullName:String
-DateOfBirth:Integer
-ContactDetails:String
-DateDegreeAwarded:Date
-DegreeAwarded:String
+ChangeProfilePicture()Void
+UploadProfilePicture()Void
+AddAssociate()Void
+Search()Void
+Message()Void
+ViewProfile()Void
+AddDetails()Void
+ViewAssociateList()Void
Account
-StudentID:String
-UserName:String
-Password:String
+SetDefaultPassword()Void
+CreateProfile()Void
System UI
Creates
Composition
Aggregation
Creates
<<uses>>
<<Interface>>
+SetPassWord()Void
+ChangePassword()Void
Register
-StudentID:String
-VerifyID()Void
Association
SAS
Generalization
User
Administrator
Message
-Sender:String
-Receiver:String
-Conatain:String
+EditConatain()Void
+Compose()Void
+Delet()Void
+Forward()Void
+SendMessage()void
Association
AddAssociate
+Search()Void
+ViewProfile()Void
+AddAssociate()Void
+DeletAssociate()Void
+BlockAssociate()Void
+Result()Void
+AddAssociateFromGroup
()void
Maintain Group
+JoinGroup()Void
+DeletGroup()Void
+CreateGroup()Void
+AddAssociateToGroup()Void
+ViewAllMember()Void
+MesseageGroupAssociate()
Void
AddPersonal
Details
-Hobbies:String
-Interest:String
-
PersonalStatement:String
+UpdatePersonalDetails()
Void
AddProfessional
Details
-ProjectTillDate:String
-AreaOfExpertise:String
-ProfessionalSkill:String
-StartDateOfJob:date
-EndDateOfJob;date
+UpdateProfessionalDetai
ls()Void
AddEducational
Details
-
PreviousQualification:String
-
PreviuosDegreeAwarded:Str
ing
-
DateOfDegreeAwarded:date
+UpdateEducationalDetail
s()Void
OtherGroup
-GroupName:String
+SetGroupName()String
+VerifyGroupName()Void
ClassGroup
+DefaultCreated()Void
+ViewAssociate()Void
1
*

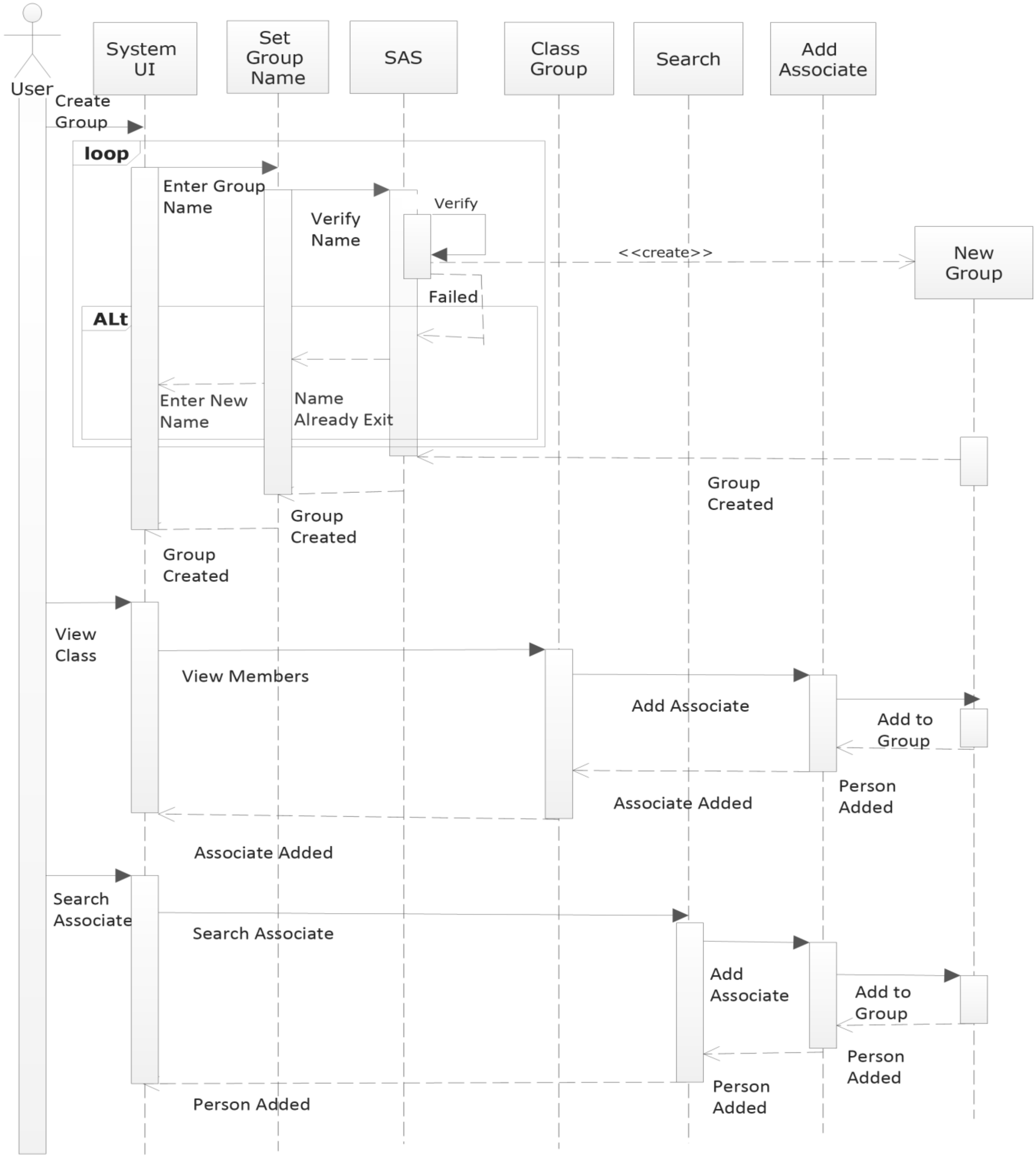
Class diagram describe the relationship between object and information structured used by system. It provides broader view of overall system. It forms the basis of system development. It also gives type of attribute and method that will be used while development of system. It gives inter-relationship between classes used in system like generalization, aggregation etc.

In case of AlumnNet starting with interface class it is used for security and confidentiality purpose. Firstly after successful registration an account would be created with default password later user can change it. As account can also be created by administration it would be threat to user privacy to have default password as administrator can even login. Therefore to prevent it interface class is provided which can only be accessed by user to change and set new password. During registration process data is verified by SAS. As one user can have only one account so the 1 to 1 relationship and without account user cannot create profile hence the composition relationship. Further profile class functionality is broken down into many small classes. To begin with details can be bifurcated into 3 category personal, professional, educational this is done to distinguish between details and make edit profile method more convenient and less complicated instead of putting all details in one method hence generalization is used. One user can send message to many other associate similarly one user can search, add or create many groups hence to 1 to many relationship and association. A user can even send message to an associate in group and can even add associate from the group to associate list. As user can send many associate invite, as well as message in group therefore many to many relationship with simple association is used. Association is used to connect these classes as they all together support extra functionality. Moreover, user can manage many groups which he/she can create, as well as default class group. Both the groups have many methods same hence the use of generalization. VerfyId is a private method it won't be visible to user as this will check user id internally with data base system of SAS. Making it private will provide security to the database system.

Conclusion

Overall, system is supporting all functionality and with use of concept like generalization, aggregation and composition it would be very easy to maintain and modify system in future like if user needs to add some more details in editing section about like places want to visit or something else he/she just can create another class next to educational details or could just add attribute appropriately.

Sequence Diagram


User
System UI
Set Group Name
SAS
Class Group
Search
Add Associate
New Group
Create Group
loop
Enter Group Name
Verify Name
Verify
<<create>>
Failed
ALt
Enter New Name
Name Already Exit
Group Created
Group Created
Group Created
View Class
View Members
Add Associate
Add to Group
Person Added
Associate Added
Associate Added
Search Associate
Search Associate
Add Associate
Add to Group
Person Added
Person Added
Person Added

Sequence diagram shows an interaction between objects in a particular scenario of use case. Basically it shows how things happen between two objects, what message is sent to whom and what replies are expected. If modified to an advance level it can show alternate path which message can take and also some loop can be used to check a particular condition.

In above scenario user communicate with system UI and select the functionality of creating new group after which he/she is prompted to enter group name. There is a loop in this case which make sure the uniqueness of group name and prompt user to enter group name again and again until it is unique and group with that name doesn't already exists. Replying takes an alternate path as specified in scenario. If user enters a name for group which doesn't already exist than new group with that name is created. Furthermore user will have option to add member in group through normal search or from default class group. An acknowledgement message will be sent to user in all cases regarding creation of group, as well as successfully adding of associate.

Task 4

General characteristic of good information

Information helps a person to get better understanding and knowledge about system, process or things. There is lot of information available in outside world, but to choose good information is important. Based on information many key decisions are made. It also helps to conclude result and to take final decision in some process. For designing process of AlumnNet domain following information is utmost important as this forms the basis to choose which functionality of system should be given more important than others. This information would be the base on which system would be build. This will also determine system success. Good information quality can be categorised as interactive, contextual and representational quality (Agarwal and Yiliyasi, 2010).

Representational Quality

- **Accessible and Understandable**

Accessibility stands for degree to which information is available or retrievable to use. If too much information is accessible than it might cause the problem of privacy and security, on the other hand if it is too much restricted than it will cause problem to user in understanding system or process. Accessibility refers to something that is easy to get. Information must be accessible so that user can know more about system. In case of AlumnNet if user wants to add another associate he/she should be accessible to at least basic information of that person, so that he/she knows that they are adding right person. Another important characteristic that should be maintained during development of AlumnNet system is easy to understand. It makes sure that data or process is understandable and comprehensible. User interface should be easy to use without much complication, or else users tend to become annoyed and it may frustrate them. Process such as adding information to profile or changing profile picture should be kept simple. According to (Roy, 2013) people tend to feel bored or discouraged if they find interface and layout complicated and dull. People are more likely to use website which are quick and easy to use.

Contextual Quality

- **Relevant**
  Relevancy stands for how much information is useful for particular task. Information must be relevant to the need for user or a situation it is needed. In case of AlumnNet system suppose a user wants to search an associate based on skills than system should only show associate which matches the skill searched for and not based on educational details or anything else.

Interactive Quality

- **Communication and Timely**
  Communication refers to ability of system to provide facilities to connect one user to another. Referring AlumnNet system messaging facility is provided for communicate between people. It is also necessary when message is sent it is received by correct person who was supposed to receive it. Timely refers to recent or up to date data or activity. To keep user up to date about associate is important this will help to keep system interactive. As main function of AlumnNet system is improving communication and networking between students and alumni these feature are crucial and utmost important. As this will help people to know recent activities about their associate and also they can communicate with them.

  Conclusion

  Above factors plays an important role in creating and maintaining AlumnNet system. Hence the properties mentioned above should be considered while design as well as developing AlumnNet system.

Task5

Modifiability is a quality attribute that take into account any future change or any up gradation made in system and ability of that system to accommodate that change. This is one of main factor in programming as it decides the flexibility and future scope of system. If this factor is not taken into account while designing a system it may lead to loss of both time and money to make any changes or further development is system. If object oriented approach is used in development process it has various technique and principles that take into account modifiability, as well as, maintenance factor while designing a system. To improve modifiability of system increasing cohesion and reducing coupling is important. Reduction in coupling can be achieved by encapsulation and abstraction while inheritance and polymorphism are used to split responsibilities and reduce coding time (Bachmann, Bass and Nord, 2007).

- Inheritance

  Inheritance is one of the most important principal concept of object orientated development approach it improves the flexibility and modifiability of system greatly and reduces complexity of classes, as existing and robust software are just extended rather than creating a new one from scratch. In inheritance a class derives attribute from another class which is called as parent class or super class and other class is called as child class or derived class. Causations while using inheritance is very important as overly or improper use can lead to reduce quality of system like system may become time consuming and difficult to maintain and modify (Bajeh et al., 2014)

  In AlumnNet system multiple inheritance is used as in many place concept of generalization is used. This helps the profile class to be less complex for example in case of details it has been divided into three smaller classes of details which derive some of its attribute from super class profile. So in future if the programmer wants to add some more attributes in details he/she can choose which class should be used to add which details like current city or personal email id can be added to personal details class. This will not only simply task of programmer to modify program, but it will also save time.

- Data encapsulation

  Principle behind encapsulation is just protecting anything that is prone to change. It helps in isolation of system functionality within a module to limit the effects of changes within the module. Therefore we don’t have to worry about bugs affecting the rest of program due to changes made. This also helps to keep data in one place and not scattered, so that during the time of modifiability it is easy to make changes. In AlumnNet system all attribute used are examples of data encapsulation. Thus any change in data type of any of attribute will not affect other attribute in other class or entire system. This makes it easy to modify system as stated in (Bachmann, Bass and Nord, 2007) cost of implementing encapsulation during programming stage reduces the cost and time of modification in future.

- Abstraction

  Abstraction is basically used to simply large design and making classes less complicated by hiding some of its methods especially in user interface. Encapsulation forms the basis of abstraction. In abstraction process abstract classes is sub divided in many sub classes this is called as procedural abstraction. Abstraction helps to improve modifiability of system by dividing big functionality into smaller functionality, so when a change has to be made it would be easy to amend small code rather than large code. In case of AlumnNet system profile class function has been divided into many other small classes like adding details or searching and associate. This will help in future if some changes have to be made in searching associate technique or any other functionality. Therefore abstraction can be re-implemented without making any change to abstraction that uses it. It also support for reuse of code, which saves both money and time and will ease the process of modifiability.

- Polymorphism

  Polymorphism helps in using one method again in different class with some modification in method. It is further enhancement to inheritance. Methods from parent class is overridden in child class with some new modification or new attribute and these changes only limit to that particular child class, so in any other child class can have some different changes. In case in AlumnNet system search method can be modified for different classes like in case of adding associate to associate list or adding associate to group, so in both cases search method can have more or less attributes based on which user search an associate i.e. search filter. It can also be used in case of view profile what specific details of user need to be shown in initial stage when user is not an associate and what after becoming associate. This will also help in improving security and privacy for user.

Conclusion

Thus the use of object oriented development technique and principal forms the basic if developer wants to enhance, upgrade or modify system architecture or functionality in future. As stated in (Zhao, Khomh and Zou, 2011) modifiability is one of the critical necessities for success of system or application.

Task 6

A person must behave in a professional manner while working. He/she must follow rules and regulations and also show an ethical and mannered behaviour in working environment. Some of professional behaviour is mentioned below and a person is expected to follow while analysing and developing AlumnNet system.

- Integrity

  Integrity is quality of being honesty and it is an important quality which brings trust with it. If a person is not honest he/she cannot be trusted with responsibility.

- Discipline

  Discipline refers to conducting or doing a work in a proper manner. Proper discipline can lead to proper planning which is important for development of any system. If discipline and planning is not maintained it will cause havoc while meeting the deadline of system.

- Use of other's intellectual property

  It's very un-professional to use other person work without their approval. A person should always respect other person work and should provide a credit if work is used.

- Professional competence.

  Competence is a necessary factor for growth and development at an individual level, as well as at an organizational level. It makes person to try harder and harder to achieve success and goals, but it should be in a health way and not just trying to put others down.

- Respect and knowledge of laws pertaining to professional work

  Employee must obey local, state, province law and should develop system with boundary and necessary security condition stated by authority.

- Honor contracts, agreements, and assigned responsibilities

  Keeping the words is important in any business and also providing what is promised. A person should always try to finish responsibilities and task assigned to him/her in given time and should be honest.

- Bribery

  It is illegal by law of many countries to accept bribery. This will not only let down company reputation, but it will also damage employee future.

- Be fair and impartial

  A person must not discriminate on the basis of sex, religious or superiority in organization. A fair treatment should be given to all.

- Privacy and confidentiality

  Developer must respect privacy and confidentiality of user and must develop a system which would provide minimum access to other people except user to their data.

  In case of AlumnNet system

  Analysis stands for process in which information is extracted and the person who does this job is called as analyst (Coad and Yourdon, 1991). Analyst should not manipulate and make false claim regarding research on functionality of system to ease work, but should be honest regarding finding of system needs and requirement and should work harder on getting knowledge what user exactly needs from system. This can be done by using various techniques like questionnaires which would help to get to know user better. Background reading and doing research will help to know about various faults about the system this will show integrity and will also help the analyst to enhance system further as they know the limitation of system. Discipline will help them to work in a professional manner so that they can meet deadline and can complete the task. This will also help to plan task in advance and to do preparation if an interview is to be conducted with the organization to get to know better about their requirement from system. If an analyst is using another person work as a basis for starting point he/she should obtain it legally. An analyst must make system such that it maintains rules and regulation of privacy and security. In case of AlumnNet analyst must make system in such a manner that administrator should not be allowed to browse personal data of user like pictures as it can lead to misuse of data. User should also be given privacy setting function such as what to display to whom and what not to, this all should be consider while creating system. Analyst should be trust worth and should keep words to complete given responsibility on time. Another important responsibility of an analyst is to keep documentation of work it is a part of discipline as in case if he/she leaves job than next analyst should be able to continue from where work is left. It would be very unprofessional if the previous analyst leaves no record of previous work and new analyst would have to start from scratch. After investigating and collecting proper documents needed for analysis he/she can start analysis process according to need. Identifying functionality requirement will be first step and then object and classes. To identify object and classes CRC technique and grammatical analysis of description is very useful. Further, basic fist cut use case diagram is constructed and based on that basic communication diagram is constructed. After that based on previous diagram advance level use case and communication diagram is done and then class diagram. Later communication is further enhanced with sequence diagram and finally class diagram is further enriched.

All above codes are reference from ACM (Acm.org, n.d.)

And techniques and principles are reference from (Bennett, McRobb and Farmer, 2010) and (Coad and Yourdon, 1991).

Reference

Acm.org, (n.d.). *Code of Ethics — Association for Computing Machinery*. [online] Available at: http://www.acm.org/about/code-of-ethics [Accessed 3 May 2015].

Agarwal, N. and Yiliyasi, Y. (2010). *Information Quality Challenges in Social Media*. 1st ed. [ebook] Available at: http://mitiq.mit.edu/ICIQ/Documents/IQ%20Conference%202010/Papers/3A1_IQChallengesInSocialMedia.pdf [Accessed 28 Apr. 2015].

Bachmann, F., Bass, L. and Nord, R. (2007). *Modifiability Tactics*. [online] Software Architecture Technology Initiative. Available at: http://www.sei.cmu.edu/reports/07tr002.pdf [Accessed 3 May 2015].

Bajeh, A., Basri, S., Jung, L. and Almomani, M. (2014). Empirical validation of object-oriented inheritance hierarchy modifiability metrics. *Proceedings of the 6th International Conference on Information Technology and Multimedia*.

Bennett, S., McRobb, S. and Farmer, R. (2010). *Object-oriented systems analysis and design using UML*. London: McGraw-Hill.

Coad, P. and Yourdon, E. (1991). *Object-oriented analysis*. Englewood Cliffs, N.J.: Yourdon Press.

Ifs.host.cs.st-andrews.ac.uk, (n.d.). *Object-oriented requirements analysis*. [online] Available at: http://ifs.host.cs.st-andrews.ac.uk/Books/SE9/Web/OORA/ [Accessed 3 May 2015].

Roy, S. (2013). *The Importance of User Interface Design For Employees - ROSSUL - UX and UI Design Agnecy*. [online] ROSSUL - UX and UI Design Agnecy. Available at: http://www.rossul.com/2013/blog/the-importance-of-user-interface-design-for-employees/ [Accessed 2 May 2015].

Zhao, X., Khomh, F. and Zou, Y. (2011). Improving the Modifiability of the Architecture of Business Applications. *2011 11th International Conference on Quality Software*.

Bibliography

Chou, A. and Chou, D. (n.d.). Information System Characteristics and Social Network Software Adoption. [online] Available at: http://www.swdsi.org/swdsi2009/papers/9k02.pdf [Accessed 1 May 2015].

Coromina, L. (n.d.). *Measurement Quality in Social Networks*. 1st ed. [ebook] Available at: http://grupsderecerca.uab.cat/egolab/sites/grupsderecerca.uab.cat.egolab/files/Coromina-PersonalNetworksUAB-5july1x.pdf [Accessed 2 May 2015].

Chai, K., Potdar, V. and Dillon, T. (2009). Content Quality Assessment Related Frameworks for Social Media. *Computational Science and Its Applications – ICCSA 2009*, pp.791-805.

Coskuncay, D. (2013). *Identifying the Factors Affecting Users� Adoption of Social Networking*. [online] Cscjournals.org. Available at: http://www.cscjournals.org/library/manuscriptinfo.php?mc=IJHCI-68 [Accessed 2 May 2015].

Bachmann, Felix., Bass, Len., & Nord, Robert. (2007). *Modifiability Tactics* (CMU/SEI-2007-TR-002). Retrieved May 03, 2015, from the Software Engineering Institute, Carnegie Mellon University website: http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=8299

Ifs.host.cs.st-andrews.ac.uk, (n.d.). *Object-oriented requirements analysis*. [online] Available at: http://ifs.host.cs.st-andrews.ac.uk/Books/SE9/Web/OORA/ [Accessed 3 May 2015].

intel code of conduct. (2015). 1st ed. [ebook] Available at: http://www.intel.co.uk/content/dam/www/public/us/en/documents/corporate-information/policy-code-conduct-corporate-information.pdf [Accessed 4 May 2015].

Lee, S., Ungson, G. and Russo, M. (2011). What determines an engaging website?: An empirical study of website characteristics and operational performance. *The Journal of High Technology Management Research*, 22(1), pp.67-79.

Mislove, A., Marcon, M., Gummadi, K., Druschel, P. and Bhattacharjee, B. (2007). Measurement and analysis of online social networks. *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement - IMC '07*.

Mai, J. (2013). The quality and qualities of information. *Journal of the American Society for Information Science and Technology*, 64(4), pp.675-688.

Marttila-Kontio, M., Hassinen, M. and Kontio, M. (2009). A Monolithic Program vs. Modifiability: Enhancing a Visual Data Flow Program with Object-Oriented Techniques. *2009 Fourth International Conference on Software Engineering Advances*.

Pace, J. and Zunino, A. (2012). Pattern-Based Modifiability Analysis of EJB Architectures. *2012 Sixth Brazilian Symposium on Software Components, Architectures and Reuse*.

Poels, G. and Dedene, G. (2001). Evaluating the effect of inheritance on the modifiability of object-oriented business domain models. *Proceedings Fifth European Conference on Software Maintenance and Reengineering*.

Scanniello, G., Gravino, C., Genero, M., Cruz-Lemus, J. and Tortora, G. (2014). On the impact of UML analysis models on source-code comprehensibility and modifiability. *TOSEM*, 23(2), pp.1-26.