**UTSAAH**

**Enabling the Special Kids**

Submitted By:

|  |  |  |
| --- | --- | --- |
| Name | SAPID | ROLL NO. |
| Sweata Prasad | 500070960 | R174218060 |
| Divyanshi Kesarwani | 500066711 | R174218015 |

Contents

[OBJECTIVE 4](#_Toc37620829)

[SUBOBJECTIVE 4](#_Toc37620830)

[OUTCOMES 4](#_Toc37620831)

[Adaptation Of Manifesto For Agile Software Development 5](#_Toc37620832)

[Adaptation of Principles Of Agile Manifesto 6](#_Toc37620833)

[INTRODUCTION 8](#_Toc37620834)

[ENTITY RELATIONSHIP DIAGRAM 9](#_Toc37620835)

[DATA FLOW DIAGRAM 10](#_Toc37620836)

[Process Flow Diagram: - 13](#_Toc37620837)

[13](#_Toc37620838)

[SCRUM METHOLOGY FOR PROVIDING SHELTER TO SPECIAL KIDS 14](#_Toc37620839)

[SCRUM: 14](#_Toc37620840)

[SCRUM FRAMEWORK 15](#_Toc37620841)

[15](#_Toc37620842)

[SPRINT 16](#_Toc37620843)

[SPRINT PLANNING 17](#_Toc37620844)

[SPRINT REVIEW 17](#_Toc37620845)

[SPRINT RETRO 18](#_Toc37620846)

[Sprint Backlog 18](#_Toc37620847)

[Product Backlog 18](#_Toc37620848)

[The Product Owner 19](#_Toc37620849)

[The Scrum Master 19](#_Toc37620850)

[The Development Team 20](#_Toc37620851)

[Why to use Scrum? 22](#_Toc37620852)

[Reasons to use Scrum Methodology In Our Application 22](#_Toc37620853)

[i. The Scrum Team members have courage to do the right thing and work on tough problems. Everyone in our team will focus on the work of the Sprint and the goals. **Our development team and stakeholders will have transparency** about all the work and the challenges with performing the work. 22](#_Toc37620854)

[ABSTRACT VIEW – 23](#_Toc37620855)

[State chart diagram 25](#_Toc37620856)

[Sequence diagram 27](#_Toc37620857)

[**Basic Sequence Diagram Notations** 27](#_Toc37620858)

[**Types of Messages in Sequence Diagrams** 29](#_Toc37620859)

[CLASS DIAGRAM 32](#_Toc37620860)

[Class Name 32](#_Toc37620861)

[Attributes: 32](#_Toc37620862)

[Relationships 33](#_Toc37620863)

[REFRENCES 36](#_Toc37620864)

# OBJECTIVE

* Our project Utsaah aims to reduce number of child trafficking in India.
* It aims to give better life to special kids and helps those who are willing to adopt children.

## SUBOBJECTIVE

* Its gives an open platform for those who want to help but are being unknown about such applications.
* It aims to provide safety and a routine checkup to the adopted child.

## OUTCOMES

* It will provide a proper livelihood for the special kids.
* Our project will keep record of the kids before and after their adoption.

# Adaptation Of Manifesto For Agile Software Development

1. Individuals and interactions over processes and tools: People will be our main asset for our project. We will focus on team communication and learning from feedback will meet all there needs
2. Working software over comprehensive documentation: We will focus more on development of software rather then focusing on documentation so the essential is produced without overloading the documentation.
3. Customer collaboration over contract negotiation: Our project aims to satisfy users needs more effectively through the quality of our software and effective teamwork & management.
4. Responding to change over following a plan: We in our project will be ready to make changes through analyzed user wants for the better development of our software by providing time-to-time update.

# Adaptation of Principles Of Agile Manifesto

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software:- Our Project will aims on working Software at regular interval, rather than waiting for long period of time.

2.Welcomechanging requirements, even late in development. Agile processes harness change for   
the customer's competitive advantage:-We will be ready to make any changes and avoid delay when our software will require a feature change.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a   
preference to the shorter timescale:- We will ensure regular delivery of the required working software.

4. Business people and developers must work together daily throughout the project:-We will ensure that there is teamwork and cooperation between the team members throughout the project is developed.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done:- Throughout the process of development of our project we will motivate our team and come out with best of our knowledge.

6. The most efficient and effective method of conveying information to and within development   
team is face-to-face conversation:-We will convey information face-to-face as it is more successful while developing software.

7. Working software is the primary measure of progress:- Our aim is to deliver working software on time which will measure our progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely:- We will maintain the delivering speed of our working software and will repeat it in each release.

9. Continuous attention to technical excellence and good design enhances agility:-We will make our software in such a way which ensures that it can sustain with changes and can improve our software.

10. Simplicity--the art of maximizing the amount of work not done--is essential:-We will simplify the design of our software so as to make it user friendly.

11. The best architectures, requirements, and designs emerge from self-organizing teams:- We will provide the best ideas from our own team. Our team members are able to cover all aspects of systems development and work together creating functional requirement, designs, architectures and implementation plans.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly:- We as a team will work efficiently by using proper skills and through process and self improvement.

# INTRODUCTION

Utsaah, an initiative to humanize existing institutions under the beggary prevention law. Our platform aims to anonymously get the details of the special kids found abandoned. It also gives open platform to society and different institute to adopt children. Our Application is a help for those children who can be adopted in good families for their livelihood. It’s a help to those who are willing to adopt children. It aims to remove the child labor from the society. It aims to reduce human trafficking.

# ENTITY RELATIONSHIP DIAGRAM

Views

NGO

USER

# DATA FLOW DIAGRAM

DFD LEVEL 0 : -

FAMILY

USER

Details Child info

Location Adoption

(Yes/No)

DFD LEVEL 1:-

Child Detail

Ngo Detail

Feedback

Family

Users

DFD LEVEL 2: -

Registration

Volunteer registers

Volunteer

According to location volunteer

Visits

Child will be

Pickup

sent to Ngo

Child

NGO

Child location

Assigns volunteer

Ngo employee

Check ups

Family

Monthly Inspection

Feedback

Volunteer give feedback

# Process Flow Diagram: -

# 

no

Child information is send

Child is taken care till child is adopted

Family then visits

yes

Wants to adopt child

Family

Personal information

Enter Info

Ngo picks the child

no

yes

Want to enter detail anonymously

Find child

# SCRUM METHOLOGY FOR PROVIDING SHELTER TO SPECIAL KIDS

## 

## SCRUM:

* The term ‘scrum’ is an abbreviation from scrummage (transferred sense of a “noisy throng”). Scrummage or scrimmage is an alteration of skirmish. Scrumming is often used to describe a tightly packed disorderly crowd.
* According to The Scrum Guide; “When the values of commitment, courage, focus, openness and respect are embodied and lived by the Scrum Team, the Scrum pillars of **transparency**, **inspection**, and **adaptation** come to life and build **trust** for everyone.”
* Companies are increasingly realizing that the old, sequential approach to developing new products simply won’t get the job done. So, rather than passing work down in a sequential manner, Scrum is the metaphor introduced for moving work forward as a team.
* This methodology encapsulated into multiple iterations, or sprints to create a working application. In this way, Scrum builds the application incrementally, with each increment adding and improving features and functionality created by its previous phases.
* The core of Scrum is only five phrase, starting with three pillars and two good ideas that enable the team to improve its focus:

1. Demo or deliver every Sprint.
2. Team decides (The team self-organizes; management ‘butts out’).
3. Inspect and adapt every day.
4. Scrum Master.
5. Product Owner.

# SCRUM FRAMEWORK

# 

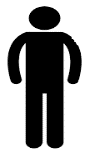
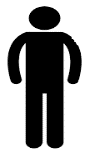
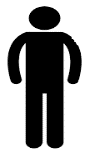
STAKEHOLDER

SPRINT BACKLOG

Stakeholders

SPRINT BACKLOG

Input from ngo and users



PRODUCT BACKLOG

Scrum

master

Team member



Product owner

FINAL

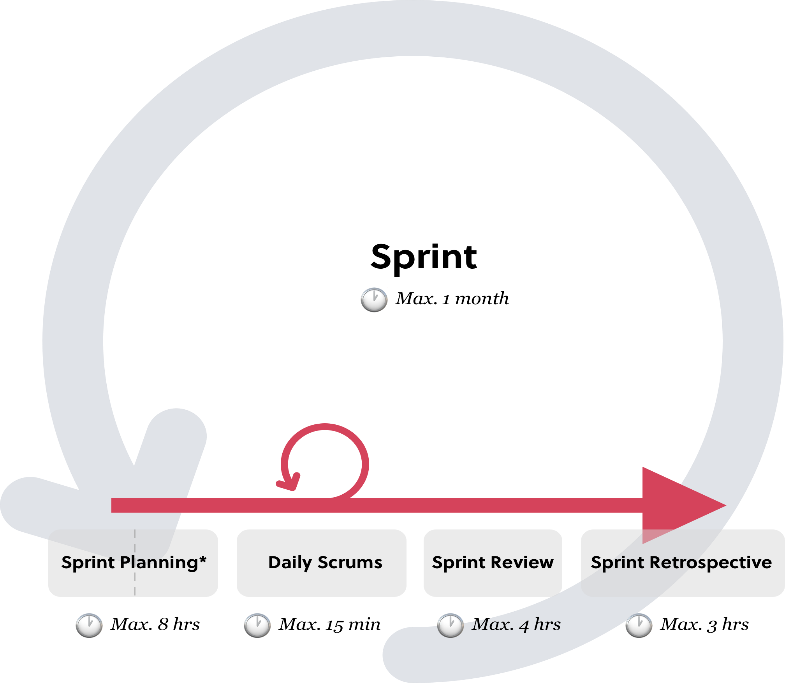
OUTCOME

TASK BREAKOUT

Sprint backlog

## SPRINT

* “Each Sprint has a goal of what is to be built, a design and flexible plan that will guide building it, the work, and the resultant product increment.”  
  — The Scrum Guide.



* Every Sprint, the team gets together on regular, consistent intervals. Consistency reduces complexity. In repetitive (iterative) cycles Increments are created. This routineenables continuous improvement; transparency*,*inspectionand adaptation.

* With Scrum, teams don’t idly sit around a table, but actively work together to do inspections and create actionable short term plans. Counter to disciplinary segregation, they are cross-functional and collaborate throughout the day when needed.
* The Sprint and its events have a Time box. A time box is an agreement made upfront about the duration of a certain activity. The maximum duration for a Sprint in Scrum is a calendar month. Sprints have consistent durations throughout a development effort.
* The time boxes for Sprint Events only have a maximum duration, no minimum. This means they might end sooner. Over time Scrum Teams should learn to have more effective and meaningful events.

## SPRINT PLANNING

* “The Development Team may also invite other people to attend to provide technical or domain advice.” — The Scrum Guide
* A Scrum Master might think it is their responsibility everyone attends and participates. Indeed, poor attendance and participation reduces transparency and introduces risk. However, the value of attendance and participation, whilst enabling a comfortable, enjoyable, calm, and respectful environment is the best way for a Scrum Master to motivate team members to participant.
* For the Sprint Planning we require:

1. Valuable input and feedback from the Sprint Review (may already have been processed into the Product Backlog).
2. A refined [Product Backlog](https://medium.com/serious-scrum/the-product-backlog-7aec7daf844f).
3. At least one high priority process improvement identified in the previous Retrospective meeting.

* By the end of the Sprint Planning, the Development Team should be able to explain to the Product Owner and Scrum Master how it intends to work as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment.

## SPRINT REVIEW

* The purpose of the Sprint Review isn’t to provide a status update or a presentation to stakeholders. It is to collect and process feedback on the actual Increment.
* “This is an informal meeting, not a status meeting, and the presentation of the Increment is intended to elicit feedback and foster collaboration.” — The Scrum Guide
* A Sprint Review is held at the end of the Sprint prior to the Sprint Retrospective.  Sprint Review serves to benefit transparency and this requires participants to be open about all the challenges too.
* During the Sprint Review first and foremost task is that the Scrum Team processes feedback on the Product Increment. Not just from stakeholders, but also from its own team members. They collectively discuss these and align on it.

## SPRINT RETRO

* In the process of‘ Retro the Scrum Team **inspects** itself. They create a plan on how to **adapt**during the next Sprint. The Sprint Retrospective is an opportunity for the team to develop [transparency](https://medium.com/serious-scrum/empiricism-transparency-33adad8fbba2) and to inspect and adapt.
* A Sprint Retrospective is held at the end of the Sprintafter the Sprint Review. Anytime is a good time to improve. You **don’t**have to wait for the Retro to implement or align on improvements.
* Each iteration in Scrum involves transparency, inspection and adaptation, not just on the product, but also the team itself and its work environment. A Sprint is not a Sprint if it didn’t include a Sprint Retrospective.

## Sprint Backlog

* It is a **forecast**containinga set of Product Backlog items selected for the Sprint, a **plan** for delivering the product Increment and realizing the Sprint Goal, and contains at least one high priority process **improvement**(from the retrospective).
* The emergence of sprint backlog occurs as the Development Team works through the plan and learns more about the work needed to achieve the Sprint Goal.
* The Sprint Backlog makes visible all the work that the Development Team identifies as necessary to meet the Sprint Goal.

# Product Backlog

* A product backlog is an ordered list of the work to be done in order to create, maintain and sustain a product. Managed by the Product Owner. The Product Backlog makes unrealized (potential) value transparent. It is the domain of The Product Owner.
* The Product Backlog is organized in Product Backlog items (PBI). Product Backlog items have the attributes of a description, order, estimate, and value.
* The Product Owner is the sole person responsible for managing the Product Backlog. The Development Team is structured and empowered to determine themselves how to best go about meeting a need.
* A Product Backlog may include projections too, as long as these are based on past performance of the team and the rate of progress made to date, and only as the Backlog currently stands. These will change as the backlog changes over time.

# The Product Owner

* The role in Scrum accountable for maximizing the value of a product, primarily by incrementally managing and expressing business and functional expectations for a product to the Development Team(s).
* The role of Product Owner is indeed a very powerful role, it is also somewhat restricted.
* The primary focus of the Product Owner is to maximize the value the Development Team performs, that isn’t to say the Development Team should have no awareness or responsibility towards maximizing value.
* The Product Owner will try to understand the market, the users and Stakeholders as best as possible, in order to be able to determine what would be most valuable to them.

## **The Scrum Master**

* The scrum master plays the role within a Scrum Team accountable for guiding, coaching, teaching and assisting a Scrum Team and its environments in a proper understanding and use of Scrum.
* As a Scrum Master, you do not only help better everyone’s understanding of the guide, you also help them improve in their role. A Scrum Master will explain to stakeholders (and Product Owner) the value of attending reviews and the value of the team being able to adapt to new insights, changing conditions and newly discovered complexities.
* When it comes to the Product Backlog and determining priorities, a Scrum Master might point out to a Product Owner that might not be the most transparent way to prioritize, especially if there are dozens of ‘Must Haves’ for a Development Team to choose from.
* The Scrum Master will help the team be as productive, creative and adaptable as it can be. He will inspire team members to take on challenges together instead of single-handedly.
* The Scrum Master will facilitate the Scrum Events and other interactions when requested or needed, and helps team members to adhere to the Sprint Cadence and time-boxes.

# The Development Team

* The development team plays the role within a Scrum Team accountable for managing, organizing and doing all development work required to create a releasable Increment of product every Sprint.
* Scrum is a teamwork .A team that is trusted and empowered to self-organize, is best equipped to handle complex dynamic challenges. By removing titles for Development Team members, Scrum really drives home that the whole team is collectively responsible. Scrum Developers apply themselves to achieving the set goals, even if this involves adapting oneself.
* Development Teams are structured and empowered by the organization to organize and manage their own work. The resulting synergy optimizes the Development Team’s overall efficiency and effectiveness*.*

# Why to use Scrum?

[**SCRUM**](https://www.scrum.org/) is a software development model that allows step by step to set up a plan successfully. Its agility is to encourage teams to learn through experiences, self-organize while working on a problem, and reflect on their wins and losses to continuously improve. Some of the reasons are highlighted below:

1. Scrum is used by many software development teams, its **principles and lessons can be applied to all kinds of teamwork**.  Scrum describes a set of meetings, tools, and roles that work in concert to help teams structure and manage their work.
2. Scrum **helps to save time** and with a responsible team, our team can achieve wonders. The ability of product owner to manage tasks in an organized way and scrum master to make plan necessary to reach the development team, make it a crucial tool to save time and money as well.
3. **Transparency plays a great role** on various levels of Scrum. Because of transparency all the stakeholders are informed where the project is at, it helps discovering weaknesses and it makes the effective teamwork possible, which makes Scum so efficient.
4. **Scrum helps to change accordingly** as it provides the possibility of adjusting to the our application through which not only the idea of an organized production of tasks is conceived, but also a plan that goes hand in hand with the company in charge of the project and its characteristics.

# Reasons to use Scrum Methodology In Our Application

# The Scrum Team members have courage to do the right thing and work on tough problems. Everyone in our team will focus on the work of the Sprint and the goals. **Our development team and stakeholders will have transparency** about all the work and the challenges with performing the work.

1. With Scrum approach, there is greater **customer satisfaction**. A key reason for this is that the users are getting useable portions of completed product quicker. They can then try out what they received and report back their findings and give feedback accordingly.
2. To prevent deviation from the desired process or end product, people need to inspect what is being created, and how, at regular intervals. Inspection should occur at the point of work but should not get in the way of that work. **Scrum Reviews & Retrospectives Offer Inspection Opportunities.**Scrum teams inspect their completed work and their process at the end of every iteration during the sprint reviews and sprint retrospectives.
3. In Scrum obtaining **early feedback is crucial for the project to adapt to the client’s needs**. However, many times the requirements are scarce, or the client only gives a sketch of what he/she wants. Therefore, it is important to make periodic deliveries to have the possibility of adding functionalities to the product in an incremental way. **Periodic deliveries provide space for continuous improvement**while allowing us to manage customer expectations better and to adapt to his/her needs.

# ABSTRACT VIEW –

Agile methodology that utilizes iterative development and prototyping are widely used in variety of industry projects as a light weight development method which can satisfy to the changes of requirements .Software engineering techniques have been employed for many years to create software products. The selections of appropriate software development methodologies for a given project, and tailoring the methodologies to a specific requirement have been a challenge since the establishment of software development as a discipline. In the late 1990’s, the general trend in software development techniques has changed from traditional waterfall approaches to more iterative incremental development approaches with different combination of old concepts, new concepts, and metamorphosed old concepts. Nowadays, the aim of most software companies is to produce software in short time period with minimal costs, and within unstable, changing environments that inspired the birth of Agile. Agile software development practice have caught the attention of software development teams and software engineering researchers worldwide during the last decade but scientific research and published outcomes still remains quite scarce. Every agile approach has its own development cycle that results in technological, managerial and environmental changes in the software companies. This paper explains the values and principles of ten agile practices that are becoming more and more dominant in the software development industry. Agile processes are not always beneficial, they have some limitations as well, and this paper also discusses the advantages and disadvantages of Agile processes.

Keywords - Agile Software Development, Agile Methodologies,XP, Scrum, Kanban.

# State chart diagram

User find child

User decides to help the child

User Registers

If user want to help by adding personal detail

Location of child

Enter detail

If want to help anonymously

Ngo volunteer picks the child

Child is taken care

If child is not adopted

Family wants to adopt

If family adopts child

Feedback is collected

Regular checkups are done

Child gets adopted

Transition: -

An arrow that makes path for the different state of objects.

Initial state: -

Represents the

Object’s initial state.

Final State: -

It represents

the object final state.

# Sequence diagram

## **Basic Sequence Diagram Notations**

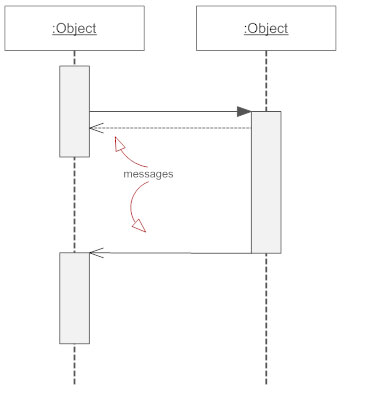
**Class Roles or Participants**  
Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

Object symbol - Sequence diagram

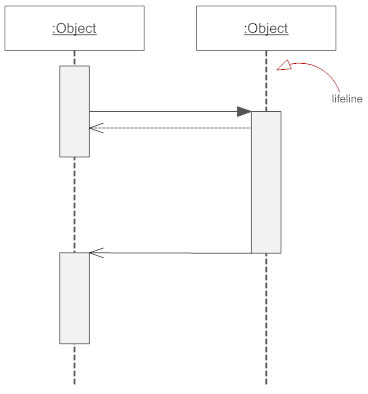
**Activation or Execution Occurrence**  
Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin gray rectangle placed vertically on its lifeline.



**Messages**  
Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks. For message types, see below.



**Lifelines**  
Lifelines are vertical dashed lines that indicate the object's presence over time.



**Destroying Objects**  
Objects can be terminated early using an arrow labeled "<< destroy >>" that points to an X. This object is removed from memory. When that object's lifeline ends, you can place an X at the end of its lifeline to denote a destruction occurrence.

**Loops**  
A repetition or loop within a sequence diagram is depicted as a rectangle. Place the condition for exiting the loop at the bottom left corner in square brackets [ ].

## **Types of Messages in Sequence Diagrams**

**Synchronous Message**  
A synchronous message requires a response before the interaction can continue. It's usually drawn using a line with a solid arrowhead pointing from one object to another.

Synchronous message - Sequence diagram

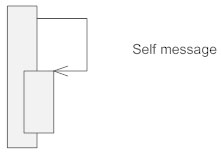
**Asynchronous Message**  
Asynchronous messages don't need a reply for interaction to continue. Like synchronous messages, they are drawn with an arrow connecting two lifelines; however, the arrowhead is usually open and there's no return message depicted.

Simple messageAsyncrhonous message

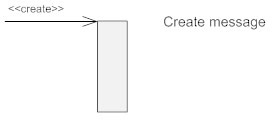
**Reply or Return Message**  
A reply message is drawn with a dotted line and an open arrowhead pointing back to the original lifeline.

Reply messages

**Self Message**  
A message an object sends to itself, usually shown as a U shaped arrow pointing back to itself.



**Create Message**  
This is a message that creates a new object. Similar to a return message, it's depicted with a dashed line and an open arrowhead that points to the rectangle representing the object created.



**Delete Message**  
This is a message that destroys an object. It can be shown by an arrow with an x at the end.

Delete message

**Found Message**  
A message sent from an unknown recipient, shown by an arrow from an endpoint to a lifeline.

Found message

**Lost Message**  
A message sent to an unknown recipient. It's shown by an arrow going from a lifeline to an endpoint, a filled circle or an x.

Lost message

child

Family

NGO

USER

User sends location of child NGO send detail of child

X

Message is sent message is deleted if family does not respond

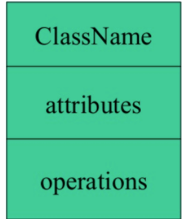
Volunteer are send to pick the child family visits the ngo

# CLASS DIAGRAM

Essential elements of UML class diagram are:

1. Class Name
2. Attributes
3. Operations

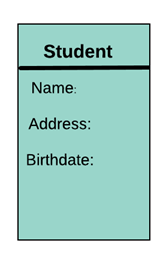
### Class Name

[](https://www.guru99.com/images/1/051818_1150_UMLClassDia1.png)

The name of the class is only needed in the graphical representation of the class. It appears in the topmost compartment. A class is the blueprint of an object which can share the same relationships, attributes, operations, & semantics. The class is rendered as a rectangle, including its name, attributes, and operations in sperate compartments.

### Attributes:

An attribute is named property of a class which describes the object being modeled. In the class diagram, this component is placed just below the name-compartment.

[](https://www.guru99.com/images/1/051818_1150_UMLClassDia2.png)

### Relationships

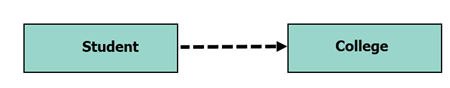
There are mainly three kinds of relationships in UML:

1. Dependencies
2. Generalizations
3. Associations

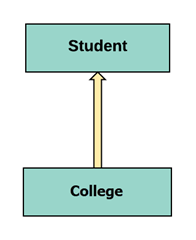
**Dependency**

A dependency means the relation between two or more classes in which a change in one may force changes in the other. However, it will always create a weaker relationship. Dependency indicates that one class depends on another.

In the following example, Student has a dependency on College

[](https://www.guru99.com/images/1/051818_1150_UMLClassDia4.png)

**Generalization:**

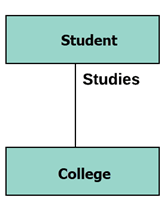
[](https://www.guru99.com/images/1/051818_1150_UMLClassDia5.png)

A generalization helps to connect a subclass to its superclass. A sub-class is inherited from its superclass. Generalization relationship can't be used to model interface implementation. Class diagram allows inheriting from multiple superclasses.

In this example, the class Student is generalized from Person Class.

**Association:**

This kind of relationship represents static relationships between classes A and B. For example; an employee works for an organization.in this example, the relationship between student and college is shown which is studies.

[](https://www.guru99.com/images/1/051818_1150_UMLClassDia6.png)

+detailvo ()

+name

+phone number

+address

+age

Volunteer

+name

+noofmembers  
+valid identification

+address

+familydetail ()

Family

+ngodetail ()

+adoption ()

+name

+location

+blocks  
+nofchildren

+detail ()

+noofadoption

Ngo

+detailuser ()

+name

+email id

+phone number

+password

+location

User

+detail ()

+name

+age

+blood group

+gender

+allergy

+weight

+height

Child

Take care of child

Contacts the ngo visit ngo for Ngo

Adoption shares

detail of child

Ngo assign

Picks child

child checkups

# REFRENCES

* <https://www.geeksforgeeks.org/>
* <https://www.tutorialspoint.com/>
* <https://www.guru99.com/>
* <https://www.researchgate.net/figure/Statechart-Diagram-of-ATM_fig3_220299153>