ARCHITECTURE AND DESIGN
CHAT PROTOCOL
- ### VERSION : 1.0
- ### NAME : CHAT PROTOCOL
- ### DATE : 26/11/2019

1. **INTRODUCTION**
The purpose of the architecture/design document is to explain the organization of the code. A well-written architecture document will make it easier for new programmers to become familiar with the code.
The architecture/design document should identify major system components and describe their static attributes and dynamic patterns of interaction.
Software architecture and design are typically expressed with a mix of UML models(class and sequence diagrams being the two most common) and prose. Dataflow diagrams are also helpful for understanding the interaction between components and overall flow of data through the system
ABOUT THIS TEMPLATE
This template suggest one way of documenting a software system's architecture/design. You aren't required to include every section in this template nor all the content in the sections you do include.
However, the document you do submit should pass the following checklist:

- Are design objectives clearly stated? For example, if performance is more important the reusability, this should be made clear at the start of the design specification.
- Does the architecture partition the implementation into clearly defined subsystem or modules with well-defined interfaces?
- Does the architecture express in a clear way the main patterns ofncmunication between subsystem and modules?
- Does the architecture satisfy the requirements?
- Is the architecture traceable to requirements?
- Any models created should either be expressed with well-known modelling language, or if a well-known modeling language isn't used, the syntax and semantics of the symbols that are used should be defined.

This document describes the architecture and design for the CHAT PROTOCOL application being develop for GROUP 5. Classroom messaging application.
The purpose of this document is to describe the and design of the CHAT PROTOCOL application in a way that addresses the interest and concerns of all major stakeholders.
For this application the major stakeholders are:
- Users and the customer - they want assurances that the architecture will provide for system functionality and exhibit desiderabile non-functional quality requirements such as usability, reliability, etc.
- Developers - they want an architecture that will minimize complexity and development effort.
- Project manager - the project manager is responsible for assigning tasks and coordinating development work. He or she wants an architecture that divides the system into components of roughly equal size and Complexity Tha can be developed simultaneously with minimal dependencies.For this to happen, the modules need well-defined interfaces. Also, because most individuals specialize in a particular skill or technology, modules should be designed around specific expertise. For example, all UI logic might be encapsulated in one module. Another might have all business logic.
- Maintenance Programmers - they want assurance that the system will be easy to evolve and maintain on into the future.

The architecture and design for a software is complex and individual stakeholders often have specialized interest. There is no one diagram or model that can easily

express a system's architecture and design. For this reason, software architecture and design is often presented in terms of multiple views or prospectives [IEEE Std. 147]. Here the architecture of the CHAT PROTOCOL application is described from 4 different perspectives [1995 Krutchen] :
1. Logical View - major components, their attributes and operations. This view also includes relationship between components and their interaction. When doing OO design, class diagrams and sequence diagrams are often used to express the logical view.
2. Process View - the threads of control and processed used to execute the operations identified in the logical view.
3. Development View - how system modules map to development organization.
4. Use Case View - the use case view is used to both motivate and validate design activity. At the start of design the requirements define the functional objectives for the design. Use cases are also used to validate suggested designs. It should be possible to walk through a use case scenario and follow the interaction between high-level components. The components should have all the necessary behavior to conceptually execute a use case.

2. **DESIGN GOALS**
There is no absolute measure for distinguishing between good and bad design. The value of a design depends on stakeholder priorities. For example, depending on the circumstances, an efficient design might be better than a maintainable one, or vise versa.
Therefore, before presenting a design it is good practice to state the design priorities. The design that is offered will be judged according to how well it satisfies the stated priorities.
The design priorities for the CHAT PROTOCOL application are:
- The design should minimize complexity and development effort
- The design should fell the general functioning of the software.

3. **SYSTEM BEHAVIOR**
The use case view is used to both drive the design phase and validate the output of the design phase. The architecture description presented here starts with a review of the expect system behavior in order to set the stage for the architecture description that follows. For a more detailed account of software requirements, see the requirements document. The CHAT PROTOCOL application uses datagram packets through the UDP protocol to send packets (messages) to the user with whom you want to communicate.

4. **LOGICAL VIEW**
The logical view describes the main functional components of the system. This includes modules, the static relationship between modules, and their dynamic patterns of interaction. In this section the modules of the system are first expressed in terms of high-level components(architecture) and progressively refined into more detailed components and eventually classes with specific attributes and operations.
4.1. **HIGH-LEVEL DESIGN (ARCHITECTURE)**

The high-level view or architecture consist of CHAT PROTOCOL major component :
- The **Application Control Logic** is the main driver of the application. It presents informations to the user and reacts to user inputs.

5. **USE CASE VIEW**
The application will be used to simplify the exchange of messages between classmates and teachers.