**SOFTWARE DESIGN DESCRIPTIONS**

***for***

**[PROJECT NAME]**

***Prepared By***

[Grup Name]

[Grup Members]

[Due Date]

CHANGE HISTORY

*[The change history shall include the project name, version number of the report, date of release and a list of version numbers and dates of release of all previous versions of the Software Design Descriptions report.]*

PREFACE

*[The preface of the Software Design Descriptions shall describe the scope and context of the Software Design Descriptions report.]*

**TABLE OF CONTENTS**

[CHANGE HISTORY i](#_Toc528078218)

[PREFACE ii](#_Toc528078219)

[TABLE OF CONTENTS iii](#_Toc528078220)

[LIST OF FIGURES v](#_Toc528078221)

[LIST OF TABLES vi](#_Toc528078222)

[1 INTRODUCTION (Section 1 of the SRS) 1](#_Toc528078223)

[1.1 Purpose of the SRS 1](#_Toc528078224)

[1.2 Scope of the Product 1](#_Toc528078225)

[1.3 Definitions, acronyms and abbreviations 1](#_Toc528078226)

[1.4 References 1](#_Toc528078227)

[1.5 Overview 1](#_Toc528078228)

[2 OVERALL DESCRIPTION (Section 2 of the SRS) 2](#_Toc528078229)

[2.1 Product perspective 2](#_Toc528078230)

[2.1.1 System interfaces 2](#_Toc528078231)

[2.1.2 User interfaces 2](#_Toc528078232)

[2.1.3 Hardware interfaces 2](#_Toc528078233)

[2.1.4 Software interfaces 2](#_Toc528078234)

[2.1.5 Communication interfaces 2](#_Toc528078235)

[2.1.6 Memory Constraints 2](#_Toc528078236)

[2.1.7 Operations 3](#_Toc528078237)

[2.1.8 Site adaptation requirements 3](#_Toc528078238)

[2.2 Product Functions: 3](#_Toc528078239)

[2.3 User Characteristics: 3](#_Toc528078240)

[2.4 Constraints: 3](#_Toc528078241)

[2.5 Assumptions and Dependencies: 4](#_Toc528078242)

[2.6 Apportioning of Requirements: 4](#_Toc528078243)

[3 SPECIFIC REQUIREMENTS (Section 3 of the SRS) 5](#_Toc528078244)

[3.1 External Interfaces: 5](#_Toc528078245)

[3.2 Functions: 6](#_Toc528078246)

[3.3 Performance Requirements: 6](#_Toc528078247)

[3.4 Logical Database Requirements 7](#_Toc528078248)

[3.5 Design Constraints 7](#_Toc528078249)

[3.5.1 Standards Compliance 7](#_Toc528078250)

[3.6 Software System Attributes 7](#_Toc528078251)

[3.6.1 Reliability 7](#_Toc528078252)

[3.6.2 Availability 7](#_Toc528078253)

[3.6.3 Security 7](#_Toc528078254)

[3.6.4 Maintainability 7](#_Toc528078255)

[3.6.5 Portability 7](#_Toc528078256)

LIST OF FIGURES

*[List the name and page number of the figures used in the report]*

LIST OF TABLES

*[List the name and page number of the tables used in the report]*

# INTRODUCTION

## Purpose and Scope of the SDD

This document provides a representation and model of the software within the limitations indicated in the SRS document of the software.

SDD document has an important role in the development and maintenance of the system. Therefore, all the design information needed by project managers, quality assurance staff, configuration managers, software designers, programmers, tester, and maintainers are contained in the SDD document of the system.

The scope of the SDD document is to design all the design information needed. This document contains a structural overview of all modules, interfaces, and data. It also contains a detailed design of each module. The objective of this document is to provide sufficient design information to allow for successful implementation and testing of the project.

## Definitions and Acronyms

*In this subsection, all the necessary definitions and acronyms required to properly interpret the SDD will be specified.*

# REFERENCES

*In this subsection, a complete list of all documents referenced in the SDD will be specified.*

# DECOMPOSITION DESCRIPTION

*This subclause records the division of the software system into design entities. A* ***design entity*** *is an element (component) of a design that is structurally and functionally distinct from other elements and that is separately named and referenced. You should divide the system into separate components that can be considered, implemented, changed, and tested with minimal effect on other entities.*

*The attributes description for identification, type, purpose, function and subordinates should be included in this design view.*

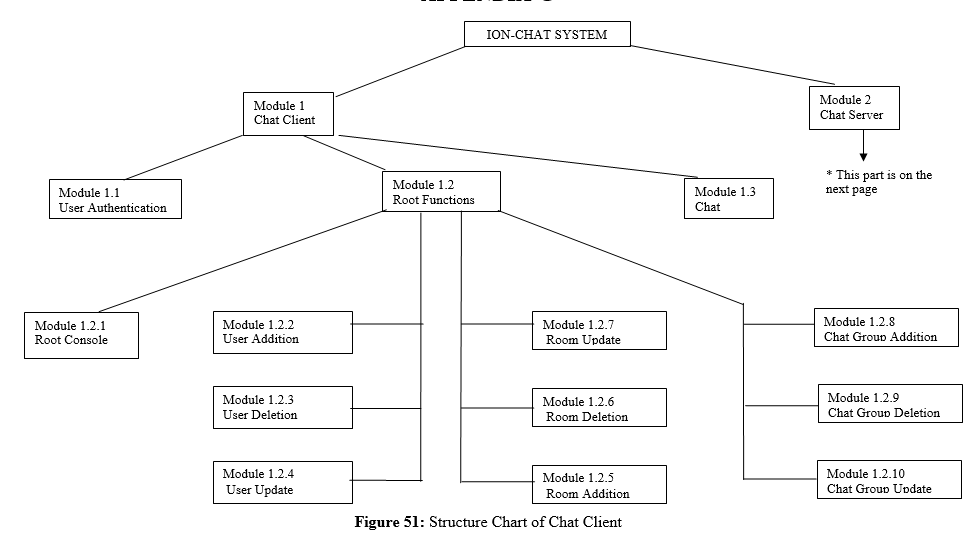
*Used by designers and maintainers to identify the major design entities of the system for purposes such as determining which entity responsible for performing specific functions. In addition, the information in the decomposition can be used by project management for planning, monitoring and control of a software project.*

*The primary graphical technique used to describe system decomposition is a hierarchical decomposition diagram. Then for each of your design entity fill the below table:*

|  |  |  |
| --- | --- | --- |
| **Identification** | **:** | *Give a hierarchical number and a descriptive name to your entity* |
| **Kind** | **:** | *A description of the kind of entity. It may simply name the kind of entity, such as subprogram, module, procedure, process, or data store.* |
| **Purpose** | **:** | *A description of why the entity exists.* |
| **Function** | **:** | *A statement of what the entity does.* |
| **Subordinates** | **:** | *The identification of all entities composing this entity.* |

*Example:*

|  |  |  |
| --- | --- | --- |
| **Identification** | **:** | Module 1 Chat\_Client |
| **Kind** | **:** | Packet |
| **Purpose** | **:** | To represent the collection of interfaces for users of the ION-CHAT SYSTEM. |
| **Function** | **:** | This module shall allow the users to enter the necessary inputs in order to login, to make basic operations about user, room or chat group and to chat in ION-CHAT SYSTEM, and to send the inputs entered to the server software. |
| **Subordinates** | **:** | Module 1.1 Client\_Site\_User\_Authentication  Module 1.2 Client\_Site\_Root\_Functions  Module 1.3 Client\_Site\_Chat |



# DEPENDENCY DESCRIPTION

This subclause specifies the relationships among entities, identifies the dependent entities, describes their coupling and identifies the required resources. It also defines the strategies for interactions among design entities and provides the information needed to easily perceive how, why, where and at what level system action occurs.

Provides an overall picture of how the system works in order to assess the impact of requirements and design changes.

Low module coupling and module cohesion are the methods. The relationship among design entities can also be shown by data flow diagrams, structure charts and transaction diagrams.

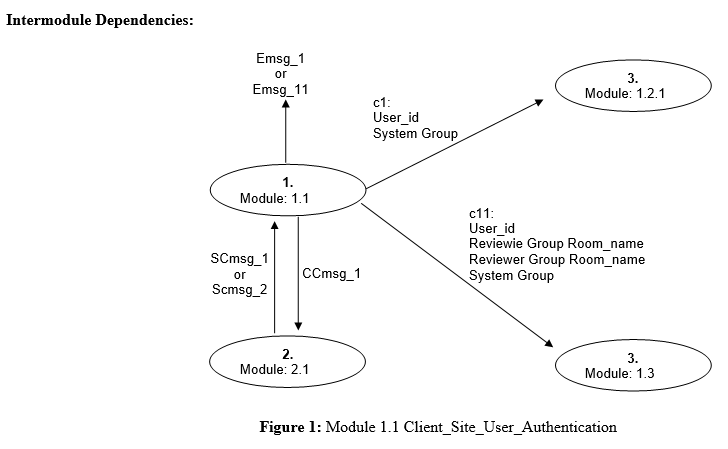
*For each of your design entity fill the below table:*

|  |  |  |
| --- | --- | --- |
| **Identification** | **:** | *Give a hierarchical number and a descriptive name to your entity* |
| **Type** | **:** | *A description of the kind of entity. It may simply name the kind of entity, such as subprogram, module, procedure, process, or data store.* |
| **Purpose** | **:** | *A description of why the entity exists.* |
| **Dependencies** | **:** | *The relationship among design entities can be represented by data flow diagrams, structure charts, or transaction diagrams.* |
| **Resources** | **:** | *The identification of all entities composing this entity.* |

*Example:*

|  |  |  |
| --- | --- | --- |
| **Identification** | **:** | Module 1.1 Client\_Site\_User\_Authentication |
| **Type** | **:** | Class |
| **Purpose** | **:** | To provide an interface, where the user shall enter the necessary inputs in order to login to the ION-CHAT SYSTEM, and to send the inputs entered to the server software. |
| **Dependencies** | **:** | The description of the data dependencies of this entity with other entities is given in Example Diagram. |
| **Resources** | **:** | java.applet.\* java.awt.\* java.awt.event.\* java.net.\* java.lang.\* |

**Example diagram:**

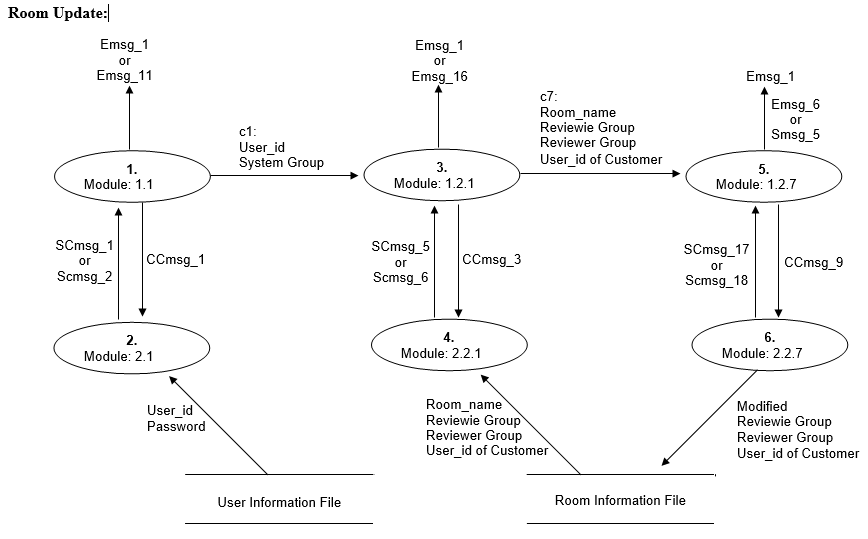


# INTERFACE DESCRIPTION

*The entity interface description shall provide the details of external and internal interfaces in order the designers and programmers to correctly use the functions provided by an entity. The attribute descriptions for identification, function and interfaces shall be included in that design view.*

*The interface description shall serve as a contract between designers and programmers. It shall provide them with an agreement needed before proceeding with the detailed design of entities about how cooperating entities shall interact. Each entity interface shall contain everything another designer or programmer needs to know to develop software that interacts with that entity.*

*The interface description should provide the language for communicating with each entity to include screen formats, valid inputs, and resulting outputs. For those entities that are data driven, a data dictionary should be used to describe the data characteristics. For this purpose you can give Data Flow Diagrams with valid inputs and resulting outputs.*



# DETAILED DESIGN DESCRIPTION

Contains the internal details of each design entity. These details include the attribute description for identification, processing and data.

Contains the details needed by programmers prior to implementation, used to aid in producing unit test plans.

Flowcharts shall be used to describe the details of each design entity.

