CLup

# Customer Line-up

1. **INTRODUCTION**
   1. PURPOSE  
      The purpose of this document is to build a more concrete foundation of what the system-to-be will be. It will also define the general behaviour and specific limitations of the system. This document is primarily addressed to the programmers and mostly includes technical language.
   2. SCOPE

The scope of the design document is to define the system’s behaviour in general cases and some critical scenarios, and to design the architecture of the system-to-be so as to provide a time-efficient, logical allocation of the components and the interaction between these components.

The document is not only limited to the architecture and behaviour of the components, but it also extends in part to the implementation and testing plan, where one possible course of action is explained, user interface design of user applications and requirements traceability relating to the Requirements and Specifications Document (RASD).

* 1. DEFINITIONS, ACRONYMS, ABBREVIATIONS
  2. REVISION HISTORY
  3. REFERENCE DOCUMENTS
  4. DOCUMENT STRUCTURE

1. **ARCHITECTURAL DESIGN**
   1. Overview: high-level components and their interaction

The architecture of the application is structured according to three logic layers:

* *Presentation Layer (P)* manages the presentation logic, namely the interaction with the user. It comprises a GUI (Graphic User Interface) that makes the application’s functionalities more understandable to the user.
* *Business Logic* or *Application Layer* (*A*) handles all the functions to provide to the user and manages the exchange of information between the user interface and the data source.
* *Data Access Layer* (*D*) provides access to the stored data. The implementation of the access logic should be both easy and structurally robust to guarantee a correct abstraction from the specific database and provide a model easy to use.

In order to guarantee as much flexibility and scalability as possible, the system is based on a 4-tier architecture (Client, Web Server, Application Server, Database Server) with a thin client. Since the application should be easy to use and executable in several different devices, the use of a thin client prevents a heavy computation load client side, carrying out all the heavy operation at server side. The user can directly communicate with the application server through the installed app in his device, while the store manager can access the functionalities devoted to him through a web application communicating with the web server.

* 1. Component view
  2. Deployment view
  3. Runtime view: You can use sequence diagrams to describe the way components interact to accomplish specific tasks typically related to your use cases
  4. Component interfaces
  5. Selected architectural styles and patterns: Please explain which styles/patterns you used, why, and how
  6. Other design decisions

1. USER INTERFACE DESIGN: Provide an overview on how the user interface(s) of your system will look like; if you have included this part in the RASD, you can simply refer to what you have already done, possibly, providing here some extensions if applicable
2. REQUIREMENTS TRACEABILITY: Explain how the requirements you have defined in the RASD map to the design elements that you have defined in this document.
3. IMPLEMENTATION, INTEGRATION AND TEST PLAN: Identify here the order in which you plan to implement the subcomponents of your system and the order in which you plan to integrate such subcomponents and test the integration.
4. EFFORT SPENT: In this section you will include information about the number of hours each group member has worked for this document.
5. REFERENCES