**Requirements document**

1. **Full specification of the application**
   * What the system is supposed to do?
     + Ex: the robot is supposed to bring drinks, the robot is supposed to take orders
2. **Use case model**
   * Use case diagram
     + Use cases capture who (actor) does what (interaction) with the system, for what purpose (goal), without dealing with system internals. A complete set of use cases specifies all the different ways to use the system, and therefore defines all behaviour required of the system, bounding the scope of the system.
   * Use case text document
     + Using Use Case Template provided we need a table for each use case we have
     + Scenarios - A scenario is an instance of a use case, and represents a single path through the use case.
3. **Functional requirements** (global and local)
   * The intended behaviour of the system. The functions the system is required to perform.
     + Baseline functionality - necessary for any system to compete in that product domain (avoids obstacles)
     + Features - differentiate the system from competitors
4. **Non-functional requirements (global and local)**
   * Requirements related to Capacity, Performance, Security, Speed…
   * Covers the conditions that do not directly relate to the behaviour or functionality of the solution, but describes environmental conditions under which the solution must remain effective or qualities that the system must have.
   * Metrics, numbers? Wiki page?
     + Ex: Speed: The robot should take the order in a timely manner
     + Ex: Security: The robot shall not disclose any personal information of the customer when checking ID
5. **Assumptions** 
   * Assumptions for the robot and the environment it operates in
     + Ex: the robot is on 4 wheels, the robot cannot climb stairs, the robot understands speech





 