Architecture design metod :

Per derivare una prima architettura funzionante partendo dai requirements specificati nell’apposito documento, si segue il criterio Generalized Model di Hofmeister et al. (2007).

Questo architecture design method è particolarmente utile per il progetto in questione, in quanto è strutturato utilizzando la terminologia e i ragionamenti di SCRUM, che è il life-cycle agile usato per l’applicazione e il suo sviluppo.

Diagram

Description automatically generated

In particolare, usa lo stesso concetto di backlog per elencare i diversi problemi che devono essere gestiti, e da esso è possibile scegliere gli elementi ritenuti più importanti in particolare iterazione.

Il backlog ha in input i requirements del progetto con particolare attenzione ai requisiti di qualità; il contesto, che contiene idee aggiuntive sul come implementare un sistema che soddisfi i requisiti e, infine, i risultati della valutazione dell’architettura dell’iterazione precedente.

Nel progetto verranno presi in considerazioni questi tre fattori per determinare le scelte di design.

Decisioni di design:

A continuazione vengono elencate le scelte di design più importanti in linguaggio naturale. Rappresentazioni più tecniche sono disponibili nei diversi schemi UML.

Rappresentazione architettura con IEEE 1471

Le rappresentazioni più concrete dell'architettura dell'applicativo vengono descritte attraverso degli schemi UML. Questi possono essere categorizzati come delle View indirizzate a diversi Stakeholder e definite tramite diversi Viewpoint, come formalizzato dallo standard IEEE 1471.

In particolare, i Viewpoint utilizzati per definire le View sono stati scelti tra quelli proposti da Bass et al. (2003).

Module View

I Viewpoint nella categoria module costituiscono una rappresentazione statica del sistema. Per questo progetto la module view più importante è Class.

Le view costruite seguendo le specifiche del viewpoint Class descrivono il sistema in termini delle relazioni di eredità degli elementi. Il class diagram UML del progetto mette a disposizione questa informazione, assieme ad altre precisazioni dovute alla natura object-oriented del linguaggio di programmazione.

Component and connector View

Il tipo di viewpoint scelto in questa categoria è il process viewpoint. Questo definisce il sistema in termini di comunicazione e sincronizzazione di processi.

Come tutti gli altri viewpoint in questa categoria, offre una descrizione dinamica del software e, nel caso particolare del process viewpoint, è particolarmente utile per valutare performance e availability del sistema.

In termini di UML, questa informazione è disponibile principalmente tramite il Sequence diagram. Questo schema spiega come i diversi elementi (classi) del sistema comunicano tra di loro attraverso procedure calls (di tipo sincrono e asincrono).

Allocation View

Questa categoria di viewpoint descrive la relazione tra il sistema e il mondo che lo circonda. Di conseguenza, può essere utilizzato per definire come viene assegnato hardware al software o come quest’ultimo viene mappato al file system.

Per il progetto in questione, questo tipo di problematiche vengono gestite automaticamente dai framework utilizzati e quindi non sono competenza degli stakeholder che altrimenti ne farebbe uso, come i programmatori e i maintaner.

I viewpoint di tipo work assignment potrebbero essere utile per rappresentare graficamente la distribuzione del lavoro, ma in questo sprint vengono saltati per brevità.