**System Design**

**Obbiettivi di design.**

**Criteri di performance:** Il sistema dovrà essere performante e reattivo in ogni situazione e garantire tempi di risposta ragionevoli per l'utente, in modo che egli non percepisca il sistema come bloccato e non provi frustrazione durante il suo utilizzo.

**Criteri di affidabilità:** il sistema deve essere robusto, ovvero ogni volta che l’utente compila degli input di testo, ad esempio il login/registrazione, l’input sarà controllato in modo da non permettere la sottomissione di input non ammesso. Inoltre, deve essere sicuro cioè, quando l’utente si registra o cambia la password, quest’ultima viene criptata e salvata all’interno del database, così da non permettere, in nessun modo di risalire alla password se non attraverso il codice di decifratura.

**Criteri di costi:** il sistema deve essere consegnato entro il secondo appello di is.

**Criteri di mantenimento:** Il codice deve essere comprensibile e facile da modificare in caso di futuri cambiamenti. Per far questo si userà Javadoc per commentare i metodi, il che aiuterà l’individuazione del metodo da modificare.

**Criteri di utente finale:** Il sito deve essere semplice dal punto di vista dell’usabilità. Con semplice si intende che il sito deve guidare l’utente in tutte le azioni che fa, aiutandolo attraverso un’interfaccia semplice con immagini, testi e bottoni, in modo da minimizzare il numero di errori che un utente può commettere.

**References.**

Vedere RAD.

**Proposed software architecture.**

**Subsystem decomposition.**