

Relazione progetto di Ingegneria del software 2019

Edoardo Zorzi, Elia Piccoli, Marian Statache

Luglio 2019

Contenuti

1	Ingegneria e sviluppo	2
---	---------------------------------	---

1 Ingegneria e sviluppo

1.1 Metodologia del processo di sviluppo

La dimensione del team di sviluppo — tre persone — e i diversi livelli di interesse relativi agli ‘ambiti’ di programmazione concernenti tale progetto, espressi inizialmente dai soggetti del team, hanno portato alla decisione di suddividere in modo moderatamente netto i compiti assegnati alle diverse persone, perlomeno nella fase iniziale, pre-design.

Specificatamente, la decisione è stata quella di assegnare a Marian il compito di creare e sviluppare la maggior parte delle interfacce grafiche di tutti i componenti costituenti il sistema nel suo complesso e quello di ideare e sviluppare la GUI generale, e in particolar modo di considerare i modi con cui gli utenti si aspettano di interagire con il sistema e quindi di sviluppare accordatamente le relative interfacce grafiche. Ad Elia ed Edoardo invece è stato assegnato il compito di ideare e sviluppare la back-end, l’architettura del sistema generale e le interfacce di comunicazione e interazione dei diversi componenti. Pur non avendo definito inizialmente la suddivisione specifica di questi compiti, nel corso dello sviluppo del sistema, a seguito della fase di progettazione dove si sono prese decisioni relative ai design patterns migliori da usare — in particolare il pattern MVC e il pattern Observer — le aree di progettazione sono state naturalmente divise nello sviluppo dei controllori e modelli dei componenti, assegnata ad Elia, e lo sviluppo e gestione dello stato centralizzato, secondo il pattern Observer, assegnata invece ad Edoardo. Data la fondamentale connessione tra tali due aree da un certo punto in poi lo sviluppo dell’architettura, e in particolar modo l’interfaccia comune tra stato e controllori, è stata eseguita in modo unico dai due soggetti che hanno seguito la tecnica di dual programming, secondo la quale il codice è stato scritto insieme, a turno, con l’input dell’altro: questo sia per coadiuvare la comprensione dell’architettura nel suo complesso, sia per permettere di sviluppare codice che sin da subito rispettasse i due diversi approcci di programmazione e che unisse nel modo più chiaro possibile le interfacce tra le due aree.

1.2 Casi d’uso

