Il proprietario di un piccolo aeroporto desidera una nuova applicazione per la gestione dello stesso, che dovrà servire sia al personale per svolgere le funzioni amministrative, di manutenzione e di controllo, sia agli utenti per la ricerca e la prenotazione dei voli.

1. Introduzione

Si vuole progettare un software formato da: aerei, voli, prenotazioni, gate, personale amministrativo, di volo e di terra.

I voli sono caratterizzati da un ID, destinazione, data e ora di partenza e di arrivo, tipologia (diretto o con scalo), lista passeggeri, pilota e copilota. Ad ogni volo viene assegnato un aereo.

Gli aerei possono essere cargo o di linea. Sono descritti da un identificativo, dal numero massimo di passeggeri imbarcabili per gli aerei di linea, oppure dai kg supportati per gli aerei cargo. Si dovrà specificare se l’aereo è in attesa di fare rifornimento.

I passeggeri prenotano i voli accedendo ad un sistema dedicato con registrazione necessaria. Per completare la prenotazione dovranno inserire: informazioni generali (nome, cognome, data di nascita, carta d’identità, e-mail), certificazione Covid-19, numero di bagagli trasportati e se dispongono di priorità.

I gate, sono sette e sono identificati da una sigla crescente: G - #.

I lavoratori presso l'aeroporto si dividono in piloti, addetti manutenzione, addetti all’amministrazione, personale della torre di controllo. Ad eccezione dei piloti, gli altri lavoratori possono accedere tramite password a un sistema dedicato per compiere le funzioni.

Il decollo e l’atterraggio dei voli sono gestiti da una torre di controllo. All’atterraggio l’aereo in arrivo comunica alla torre le informazioni generali e gli viene assegnato un ID e il gate in cui sostare.

• Il sistema dovrà supportare il lavoro di amministrazione dell’aeroporto (assegnazione del personale di volo per ogni tratta, assegnazione del gate all’aereo, gestione della manutenzione), accessibile solamente tramite password dai dipendenti. La ricerca dei voli avrà libero accesso mentre al momento della prenotazione e per la modifica sarà necessario inserire la carta d’identità e l’identificativo del biglietto.

• Il sistema dovrà gestire l’elenco delle prenotazioni, dei voli e degli aerei registrati nell'aeroporto.

• Si vuole fornire ai passeggeri uno strumento di ricerca facile da utilizzare che mostri i voli e le date disponibili, specificando un aeroporto di arrivo.

• Il sistema deve gestire l’elenco degli aerei nell’aeroporto e le loro posizioni aggiornate al suo interno (gate in cui sono situati). Se i gate sono tutti occupati, nessun aereo può atterrare finché non se ne libera uno dedicato agli aerei della stessa tipologia.

• Dopo l’atterraggio gli addetti alla manutenzione riempiono sempre il serbatoio per il volo successivo.

Le persone responsabili del progetto sono gli studenti del corso di Ingegneria informatica dell'Università degli studi di Bergamo: Valentina Villa, Cristiano Gambirasio, Lorenzo Fagiani.

2. Modello del processo

Abbiamo scelto di seguire una modalità di sviluppo RAD perché adatto a progetti seguiti da poche persone. Assegniamo le priorità ai requisiti e stimiamo inizialmente degli slot temporali di una settimana in cui lavorare sui componenti.

3. Organizzazione del progetto

Applichiamo una gestione delle persone di tipo SWAT. Non essendoci particolari differenze di abilità, conoscenze ed esperienza, lavoriamo senza una gerarchia.

4. Standard, linee guida e procedure

Il codice sarà principalmente scritto con linguaggio JAVA. Il codice verrà generato da un modello UML tramite il tool Rebel di StarUML.

5. Gestione attività

Per avere una costante visione d’insieme del progetto faremo riunioni settimanali di sei ore in cui discuteremo dello stato del progetto, i successivi passi da eseguire e la suddivisione del lavoro.

6. Rischi

Non ci sono rischi dal punto di vista economico. E’ importante essere ben organizzati per rispettare i tempi di consegna, che costituiscono l’unico rischio legato al progetto.

7. Membri

Il team è composto da tre persone senza particolari differenze in termini di abilità, conoscenze ed esperienza.

8. Metodi e tecniche

Verrà utilizzato GitHub per organizzare il lavoro con branch ed issues, monitorare gli aggiornamenti un passo alla volta e lavorare sempre su una versione aggiornata del software.

Per la modellazione dell’architettura utilizzeremo UML tramite il programma StarUML e il tool Rebel.

9. Assicurazione qualità

Verranno eseguiti test sul programma durante lo sviluppo del codice mediante Junit e il tool Eclemma.

10. Suddivisione Lavoro

I requisiti sono stati stabiliti durante le prime riunioni. Il lavoro verrà suddiviso successivamente sfruttando GitHub, ma in seguito alla definizione dei componenti.

11. Risorse

Risorse software: i tool utilizzati sono Rebel per la generazione di codice ed Eclemma per il testing durante la stesura del codice.

Risorse temporali: schedulati incontri per sei ore a settimana. Lavoro individuale stimato: otto ore settimanali.

12. Budget e schedulazione

Il progetto non ha un budget e non né necessita. Le nuove versioni sviluppate saranno messe a disposizione settimanalmente.

13. Cambiamenti

Inizialmente avevamo predisposto due tipi di gate per gestire le due tipologie di aerei, successivamente abbiamo deciso di rendere i gate accessibili a tutti gli aerei indipendentemente dalla tipologia.

14. Consegna

La scadenza di consegna definitiva è prevista per il 27 gennaio 2022.