

PROJECT PLAN

• **INTRODUZIONE**

Il sistema implementa un servizio di prenotazione voli offerto da una nota compagnia low cost. I servizi offerti sono:

- Un operatore può inserire/cancellare/modificare i voli (orari, date e arrivi/destinazioni)
- Il sistema tiene traccia del personale di volo assegnato a ogni tratta
- Visualizzazione tratte andata/ritorno su località direttamente collegate
- Visualizzazione tratte andata/ritorno su località indirettamente collegate (con scalo)
- Prenotazione dei voli con relativi optional (solo bagaglio a mano e/o in stiva, scelta posto) per clienti registrati e non
- Biglietto a costo maggiore se si sceglie il posto, minore se il posto è assegnato casualmente
- Possibilità di acquisto del Biglietto Prioritario, che permette al cliente di saltare la fila (solo per bagagli a mano e con costo superiore)
- Il sistema invia un messaggio al cliente se il suo volo ha subito una variazione di ora/data oppure è stato cancellato (anche se l'aeroporto di arrivo/partenza è cambiato)
- Possibilità di accumulare buoni per sconti su voli (solo per clienti registrati)
- Il sistema riconosce l'idoneità ad alcune agevolazioni (studenti, famiglie, gruppi)
- Differenziare il costo del bagaglio in stiva in base al peso di questo
- Possibilità di prenotare al momento dell'acquisto del biglietto il pasto, che verrà pagato meno rispetto a comprarlo direttamente sul volo
- SEZIONE AGGIUNTIVA Noleggio Auto: il cliente può (anche senza aver acquistato un volo) noleggiare per un periodo un'auto in un dato aeroporto (solo per clienti registrati)
- Possibilità di scegliere auto differenti a tariffe diverse
- Possibilità di accumulare punti per buoni che vengono sommati a quelli già nel portafogli voli

• **PROCESS MODEL**

Il modello di sviluppo scelto è quello di sviluppo incrementale, dove per ogni versione viene seguito il modello di sviluppo Waterfall.

La prima versione del sistema è quella in cui l'utente può prenotare voli con:

- Bagaglio a mano e/o bagaglio in stiva di prezzo uguale (a prescindere dal peso)
- Il posto è assegnato dal sistema
- Il volo non può essere modificato/cancellato ma solo inserito dall'operatore
- Il sistema permette di far visualizzare all'utente il suo storico dei voli tramite l'inserimento del suo identificatore univoco CF (NO REGISTRAZIONE)

La seconda versione del sistema è quella in cui l'utente può prenotare voli con:

- Possibilità di avere prezzo diverso a seconda del peso del bagaglio in stiva
- (differenziazione tra Biglietto Standard e Biglietto Prioritario)
- Possibilità di avere il Biglietto Prioritario salta coda
- Possibilità di prenotare il pasto con sconto
- L'utente può registrarsi per cominciare ad accumulare punti per buoni nel suo portafogli virtuale
- L'utente può modificare/cancellare il volo (se ha il Biglietto Prioritario)

La terza versione del sistema è quella in cui l'utente può prenotare voli con:

- Se l'operatore cancella/modifica un volo, il sistema invia un messaggio con relativo rimborso a tutti i clienti passeggeri
- SERVIZIO CAR SHARING

La quarta versione del sistema è quella in cui l'utente può prenotare voli con:

- Possibilità di scegliere auto differenti a tariffe diverse

- Possibilità di accumulare punti per buoni che vengono sommati a quelli già nel portafogli voli

• ORGANIZZAZIONE DEL PROGETTO

All'interno del team di sviluppo non vi è una distinzione gerarchica: l'organizzazione è di tipo Agile Team, sia perché il team è piccolo (composto da 3 persone) sia perché ogni membro partecipa attivamente allo sviluppo in ogni sua fase.

• STANDARD, LINEE GUIDA E PROCEDURE

Verrà utilizzato lo standard MVC, con separazione delle classi nel Model, delle pagine html nel view e dei metodi di backend nel Controller.

La versione di java utilizzata è la 8.

I nomi delle classi sono Pascal Case (NomeClasseProva).

I nomi dei metodi sono Camel Case (nomeMetodoProva).

I nomi dei package sono Snake Case (nome_package_prova).

I nomi delle variabili sono Camel Case (nomeVariabileProva).

• ATTIVITA DI MANAGEMENT

Utilizziamo Github per tenere traccia dell'avanzamento dei task e del raggiungimento dei goal. In particolare per ogni versione del sistema creeremo un Project con template Basic Kanban in cui terremo traccia dello stato di avanzamento dell'implementazione di ogni Issue (fase To do, Progress, Done)

• RISCHI

Gli unici rischi possibili sono che non verranno implementate tutte e 4 le versioni del Software a causa di mancanza di tempo.

• STAFFING

Qui di seguito il numero di ore impiegate fin'ora per la realizzazione del sistema:

- 15/12/2021 (2.30h * 2 persone) inizio definizione punti 1 e 2 del PP
- 22/12/2021 (3.40h * 3 persone) apprendimento utilizzo Github + Java Spring e completamento PP
- 27/12/2021 (h * 3 persone)

• METODI E TECNICHE

Per ogni versione le tecniche e i metodi utilizzati in ogni fase di sviluppo (modello Waterfall) sono:

- Requirements : abbiamo stilato una documentazione dei requisiti in linguaggio naturale (vedi punto 2 del Project Plan)
- Design : la definizione dei componenti e la loro interazione è descritta dal Class Diagram
- Implementation : Pattern MVC per separazione dei componenti, approccio Bottom-Up per la loro implementazione
- Testing : verrà effettuato per ogni versione il Coverage Test
- Maintenance : in fase di sviluppo verrà valutata la possibilità di effettuare Refactoring su classi e metodi

• GARANZIE DI QUALITA

La qualità del prodotto è garantita da una completa documentazione della fase di testing con alcuni punti dello standard IEEE 928.

- **WORK PACKAGES**

Viene utilizzato Github per l'assegnamento dei task ai vari membri del gruppo (tramite creazione e assegnazione di un Issue).

In particolare tutte le persone del gruppo partecipano allo sviluppo del design dell'applicazione.

Per la fase di coding 1 persona si dedicherà al Model, 2 al View e 3 al Controller (eventualmente alcuni membri possono partecipare allo sviluppo di più moduli). Per la fase di testing invece saranno necessarie 2 persone e 1 che effettua la verifica finale.

- **RISORSE**

Il team è composto da 3 persone, ognuna delle quali si dedica ad ognuna delle fasi di sviluppo di ogni versione e alla stesura della relativa documentazione di ciò che ha fatto.

Per ogni fase verranno utilizzati i seguenti tools:

- Requirements : Word
- Design : StarUml (Class Diagram)
- Implementation : Eclipse + Spring, MySql per DB
- Testing : JUnit4 + Eclemma
- Maintenance : Eclipse

- **BUDGET E SCHEDULE**

La consegna del Project Plan è prevista per il 27 Dicembre, mentre la deadline per il completamento del progetto è fissata a qualche giorno prima della data d'esame. Non sono previsti costi di sviluppo.

- **MODIFICHE**

Anche per la documentazione delle versioni verrà usato Github, che permette di tenere traccia dello storico cambiamenti che ogni membro del team ha apportato al progetto.

- **DISTRIBUZIONE**

Nella prima scadenza verrà consegnato il pdf del Project Plan. Invece nell'ultima deadline (quella poco prima dell'esame) verrà consegnato l'intero zip del progetto software e in allegato i diagrammi UML (Use Case, Class, State Machine, Sequence e Activity).