UNIVERSITÀ DEGLI STUDI DI SALERNO

FACOLTÀ DI INFORMATICA

DIPARTIMENTO DI INFORMATICA

CORSO DI LAUREA MAGISTRALE IN INFORMATICA

INGEGNERIA GESTIONE ED EVOLUZIONE SOFTWARE

ANALISI DI MANUTENZIONE

DOCENTE

Prof. Andrea De Lucia

CANDIDATA

Francesca Tassatone

Matr: 0522500568

Roberta Gesumaria

Matr: 0522500569

ANNO ACCADEMICO 2018-2019

Coordinatore del progetto

Nome
Prof. Andrea De Lucia

Partecipanti

Nome	Matricola
RG – Roberta Gesumaria	0522500569
FT – Francesca Tassatone	0522500568

Revision History

Data	Versione	Descrizione	Autori
18/12/2019	1.0	Prima stesura	Roberta Gesumaria Francesca Tassatone

Sommario

1. Scopo del documento

In questo documento saranno descritti gli obiettivi del processo di aggiunta di funzionalità alla gestione dei dipendenti nel progetto PentaMotor, in riferimento al documento di "Identificazione e classificazione delle modifiche richieste".
Verrà studiato come queste modifiche possano impattare sugli artefatti del sistema esistente, analizzando il rapporto costi/benefici e i possibili rischi derivanti dalla fase di progettazione.
Questo documento includerà uno studio di fattibilità e verranno pianificate le fasi successive di progettazione, implementazione e testing.

2. Panoramica del sistema attuale

Lo scopo di questo sistema è di consentire una facile gestione dei veicoli che sono in possesso dell'azienda.
In particolare, il sistema dovrà permettere ai dipendenti di poter inserire e ricercare veicoli dal database dell'azienda, gestendo e risolvendo casi critici in cui uno stesso veicolo è venduto contemporaneamente da due dipendenti in posti diversi.
Il sistema permettere, inoltre, al gestore dell'azienda di poter visualizzare e analizzare le statistiche relative alle vendite effettuate da ogni dipendente, in modo da avere una vista globale dell'andamento e di prenotare nuovi veicoli in caso di mancanza.

2.1. Attori e funzionalità

Il sistema prevede i seguenti attori con le relative funzionalità:

- Gestore:
  - Statistiche dipendenti;
  - Inserisci veicolo;
  - Modifica veicolo;
  - Elimina veicolo;
  - Gestione account.
- Dipendente:
  - Statistiche personali;
  - Prenota veicolo;
  - Visualizza veicolo;
  - Ricerca veicolo;
  - Vendita veicolo;
  - Gestione autenticazione.

Le funzionalità indicate per il dipendente possono essere svolte anche dal gestore, non viceversa.

2.2. Design, implementazione e testing del sistema attuale

2.2.1. Design

Il sistema PentaMotor è stato progettato e implementato utilizzando il modello MVC (Model, View, Controller) per suddividere la logica di presentazione dei dati dalla logica di business.

- Sistema di memorizzazione (Model): si occupa della gestione e dello scambio dei dati tra i sottosistemi;
- Presentazione (View): raccoglie e gestisce l'interfaccia grafica e gli eventi generati dall'utente;
- Business logico (Controller): si occupa della gestione della logica del sistema.

Il sistema software è stato sviluppato utilizzando JavaScript, HTML e Php.

Secondo la change request ci sarà un impatto su tutti e tre i layer. Per quanto riguarda il sistema di memorizzazione verranno effettuate modifiche alla tabella relativa ai dipendenti. Per il layer di presentazione, saranno aggiunte nuove interfacce, poiché alcune funzionalità non erano presenti in precedenza. Infine, per il business logico, saranno aggiunte le nuove funzionalità richiesta nella change request.

2.2.2. Implementazione

L'implementazione del software è stata effettuata in riferimento al modello descritto in precedenza, in particolare associando ad ogni layer un insieme di classi che sviluppano delle funzionalità richieste nella fase di analisi dei requisiti.

Secondo lo stile architetturale descritto in precedenza, possiamo dire che:

- Il livello Presentazione prevede un unico sottosistema, ovvero la gestione dell'interfaccia grafica e gli eventi generati dall'interazione con il sistema;
- Il livello Business Logico prevede a sua voluta una suddivisione in 4 sottosistemi:
    1. Gestione autenticazione;
    2. Gestione account;
    3. Gestione statistiche;
    4. Gestione veicoli;
- Il livello Sistema di memorizzazione prevede un unico sottosistema, ovvero la gestione e il salvataggio dei dati persistenti.

2.2.3. Testing

Data la mancanza di tutta la documentazione relativa al testing del sistema PentaMotor sarà necessario effettuare un lavoro ex-novo per quanto riguarda il testing.

Il testing che verrà effettuato si divide in due fasi:

1. Unit testing: la tecnica scelta riferisce alla metodologia black box, ovvero uno sviluppo incentrato sul controllo dell'input e dell'output previsti. La procedura inizia con l'individuazione delle classi di equivalenza, mediante le quali è possibile determinare il partizionamento di tutti i possibili input.
2. Integration testing: è stata usata la metodologia bottom-up, ovvero il testing dei moduli partendo dai livelli più bassi della gerarchia fino all'interfaccia grafica.

3. Analisi della modifica richiesta

Il processo di manutenzione che prevede l'aggiunta della funzionalità relativa alla gestione dei dipendenti licenziati, porterà all'aggiunta dei requisiti funzionali mantenendo invariati gli attori del sistema. La nuova funzionalità sarà, però, accessibile solamente al gestore. Inoltre, sarà necessario modificare alcune funzionalità già presenti. Lo stile dell'interfaccia grafica resterà invariato, aggiungendo semplicemente le schermate relative alla nuova funzionalità.

4. Individuazione della soluzione progettuale

4.1. Problematiche affrontate

Issue 1: Sono necessarie nuove funzionalità?

Proposal 1.1
Aggiunte funzionalità relative alla gestione dei dipendenti licenziati.

Criterion 1.1

Fornire nuove funzionalità per tenere traccia dei vecchi dipendenti e le loro statistiche relative alle vendite.

Argument 1.1
La nuova funzionalità introdotta permette una migliore gestione dei dipendenti.

Resolution 1.1
Analizzando l'argomentazione Argument 1.1, l'Issue 1 è stata risolta applicando la Proposal 1.1

Issue 2: Che tipologia di interfaccia grafica adottare?

Proposal 2.1
L'interfaccia manterrà lo stile del sistema attuale.
Proposal 2.2
L'interfaccia è minimale.

Criterion 2.1
L'interfaccia grafica deve essere *user-friendly.*

Argument 2.1
Un'interfaccia simile a quella precedente agevola l'utente nell'utilizzo della nuova funzionalità.
Argument 2.2
Un'interfaccia minimale permette di ridurre i tempi di sviluppo.

Resolution 2.1
Analizzando le argomentazioni Argument 2.1 e Argument 2.2, l'Issue 2 è stata risolta applicando la Proposal 2.1. Poiché ci sono poche interfacce da sviluppare.

Issue 3: Che tipologia di linguaggi di programmazione utilizzare per lo sviluppo delle nuove funzionalità

Proposal 3.1
Per l'implementazione dei moduli lato client si utilizzerà HTML, CSS e Javascript, mentre per l'implementazione dei moduli lato server si utilizzerà il linguaggio di programmazione PHP.

Criterion 3.1
L'implementazione deve essere facilitata dal riuso del codice.

Argument 3.1
Utilizzando la combinazione HTML, CSS e Javascript per l'implementazione dei moduli lato client, si assicura il funzionamento su ogni browser compatibile, indipendentemente dalla piattaforma hardware/software. I moduli del server sono implementati utilizzando il linguaggio PHP, che consente di eseguire classi PHP su richiesta del client.

Resolution 3.1
Analizzando le argomentazioni Argument 3.1, l'Issue 3 è stata risolta applicando la Proposal 3.1.

4.2. Soluzione individuata

La soluzione individuata consiste nell'unione di tutte le "resolution" ottenute nel paragrafo 4.1. In dettaglio, la soluzione dovrà avere le seguenti caratteristiche:

- Verranno aggiunte le funzionalità relative alla gestione dei dipendenti licenziati. Questo consentirà al gestore di visualizzare tutti i dipendenti che hanno lavorato per la propria concessionaria ed inoltre, di visualizzare le statistiche relative alle vendite di questi dipendenti;
- L'interfaccia deve essere quanto più simile possibile alla grafica del sistema attuale;
- Per l'implementazione dei moduli lato client si utilizzerà HTML, CSS e Javascript, mentre per l'implementazione dei moduli lato server si utilizzerà il linguaggio PHP.

4.3. Soluzione alternativa

La soluzione alternativa è rappresentata dalla proposta e argomentazione affrontata nel paragrafo 4.1, ma che non è stata selezionata per la soluzione adottata.

5. Identificazione dell'Impact Set

La soluzione individuata ha coinvolto diversi artefatti del sistema attuale. La tabella che segue descrive il Candidate Impact Set individuato, ovvero l'insieme di tutti gli artefatti che verranno modificati durante la fase di manutenzione. Inoltre, accanto ad ogni artefatto sarà associato un livello di impatto, ovvero tale livello farà riferimento al modo in cui la modifica va ad impattare sul sistema software attuale.

Per individuare le componenti del sistema impattate dalla modifica si utilizzerà un approccio top down, ovvero a partire dai documenti di alto livello si andrà ad individuare gli artefatti di più basso livello che andranno ad essere modificati durante la fase di manutenzione.

Dovendo aggiungere la funzionalità relativa alla gestione dei dipendenti, si avrà un impatto su tutti gli artefatti del progetto già esistente, partendo dal Requirement Analysis Document fino all' Object Design Document.

Nella tabella che segue verranno indicati gli artefatti dei documenti che saranno impattati. L'impatto della modifica verrà valutato utilizzando tre categorie:

- FORTE: se saranno necessarie pesanti modifiche nell'artefatto o se l'artefatto dovrà essere completamente sostituito;
- MEDIO: se saranno necessarie sostanziali modifiche all'artefatto, non facendo cambiare però la sua struttura in maniera eccessiva;
- DEBOLE: se saranno necessarie solo modifiche marginali.

Per quanto riguarda il RAD dopo l'aggiunta della specifica della nuova gestione (dipendenti) con i relativi requisiti funzionali, sarà necessario effettuare la stesura di tutti quegli artefatti relativi ai requisiti aggiunti, come scenari, use case e ecc. Inoltre, sarà necessario modificare la funzionalità relativa alla registrazione dei dipendenti. L'impatto complessivo relativo al RAD risulta essere di categoria MEDIA, poiché saranno necessarie modifiche ad ogni sezione del documento, senza però modificare del tutto la sua struttura.

Analizzando il Sistem Design Document gli artefatti che saranno impattati sono i seguenti:

Artefatto	Impatto	Descrizione

Decomposizione in sottosistemi	BASSO	Prevede l'aggiunta della gestione dipendente del livello di Business logico.
Gestione dati persistenti	FORTE	Prevede l'aggiunta di un attributo alla tabella relativa ai dipendenti. L'aggiunta di un'ulteriore tabella relativa al contratto di lavoro, con le relative modifiche alla tabella dipendente dovuta alla relazione con la nuova tabella.
Controllo degli accessi e sicurezza	DEBOLE	Prevede l'aggiunta della gestione dipendenti.
Servizi dei sottosistemi	MEDIO	Prevede l'aggiunta della gestione dei dipendenti.

Andando sempre più a basso livello, nell'Object Design Document, si può stilare una prima lista degli artefatti che si andranno a modificare.

Artefatto	Impatto	Descrizione
Packages	MEDIO	Aggiunta della nuova gestione relativa ai dipendenti.
Interfacce delle classi	MEDIO	Aggiunta delle classi relative alla nuova gestione.

Dopo l'analisi dell'Object Design, è possibile stilare una lista delle classi che saranno impattate dalla modifica. Nella tabella che segue sono indicati gli artefatti del codice che saranno impattati dalla modifica:

Artefatto	Impatto	Descrizione
Pagine Html relative all'attore gestore (tutti i file html nella cartella gestore)	BASSO	Modificare la sezione "Gestione account" in "Gestione Dipendenti" con l'aggiunta delle sezioni "in servizio", "licenziati" e "tutti".
File javascript e php relativi alla gestione dei dipendenti	BASSO	Modifiche de file gestore_account.js, gestore_dati.php per la nuova

(gestore_account.js, gestore_dati.php)		gestione dei dipendenti (che in precedenza era account).
gestore_account.html	BASSO	Modifica modal per l'assunzione e la modifica dei dati relativi ad un dipendente.

Nella seguente tabella verranno indicate le modifiche apportate al database.

Artefatto	Impatto	Descrizione
Tabella dipendente	BASSO	Inserimento di un nuovo attributo nella tabella dipendente.
Tabella contratto e stipulare	MEDIO	Creazione nuova tabella contratto (id, data_inizio, data_fine, tipo_contratto, retribuzione, attivo, allegati) e stipulare (id_contratto, codice_fiscale_dip) che rappresenta la tabella di relazione tra dipendente e contratto.

Nella seguente tabella verranno indicate i nuovi artefatti del codice prodotti.

Artefatto	Impatto	Descrizione
Creazione pagine html relative alle diverse visualizzazioni delle liste (gestore_dipendenti_inServizio.html, gestore_dipendenti_licenziati.html)	BASSO	Creazione pagine html per la visualizzazione della lista dipendenti in servizio e licenziati.

6. Studio di fattibilità

Nelle sezioni che seguono verrà indicato lo studio fattibilità riguardo ai costi, benefici e rischi relativi all'impatto della change request. Saranno valutati secondo le seguenti tre categorie:

- ALTA: se saranno necessarie sostanziali modifiche implementative che andranno ad impattare su tutto il sistema;

- MEDIA: se saranno necessarie modifiche che impatteranno solo su parte del sistema;
- BASSA: se saranno necessarie solo modifiche marginali.

6.1. Identificazione, descrizione e valutazione dei costi

Identificazione	Valutazione	Motivazioni
Aggiunta di nuove funzionalità	ALTO	La modifica prevede l'aggiunta della funzionalità relativa alla gestione Dipendente. Tale funzionalità verrà sviluppata partendo dalla struttura presente, comportando modifiche su tutto il sistema.
Mantenimento dello stesso stile del sistema attuale per l'interfaccia	BASSO	È uno degli obiettivi principali mantenere lo stesso stile per l'interfaccia per una migliore usabilità da parte degli utenti esterni. Partendo dal un sistema completo, non sarà complicato mantenere lo stile già presente.
Implementazione con linguaggi di programmazione HTML, CSS, Javascript e PHP	BASSO	Il team di sviluppo ha già una buona esperienza e conoscenza di tali tecnologie.
Livello di complessità dell'interfaccia utente	BASSO	Partendo da un sistema già completo e dovendo mantenere lo stesso stile dell' interfaccia, non sarà complicato aggiungere nuove pagine con un livello di complessità uguale/simile.
Gestione della sicurezza e della riservatezza	BASSO	La gestione della sicurezza è realizzata tramite autenticazione per mezzo di username e password, mentre la gestione della riservatezza non è prevista nel sistema attuale, né

		tantomeno nei requisiti che verranno aggiunti.
Organizzazione del lavoro nel team di sviluppo	BASSO	I componenti del team di sviluppo condividono abbastanza ore settimanali per lo sviluppo del sistema.
Testing funzionale e di accettazione	ALTO	Data la mancanza di un buon testing effettuato per il sistema ci saranno poche basi, per cui è richiesto maggior lavoro.

6.2. Identificazione, descrizione e classificazione dei benefici

Identificazione	Classificazione	Motivazioni
Facilità di distribuzione e manutenzione	ALTO	L'applicazione web si trova interamente sul server Altervista, per cui la pubblicazione o l'aggiornamento di risorse sul nodo server è automaticamente reso disponibile a tutti i nodi client.
Accesso multipiattaforma	ALTO	L'accesso all'applicazione web è indipendente dall'hardware e dal sistema operativo utilizzato dagli utenti.
Riduzione del costo di gestione	ALTO	L'uso di Internet come infrastruttura per un'applicazione web riduce notevolmente sia i costi di connettività che i costi di gestione dei client.
Scalabilità	ALTO	Il sistema software anche una volta modificato può crescere insieme alle esigenze del cliente senza particolari problemi.

6.3. Identificazione, descrizione e classificazione dei rischi

Identificazione	Probabilità	Motivazioni
Introduzione di nuovi fault	MEDIA	Con l'aggiunta di una nuova funzionalità si potrebbero facilmente introdurre nuovi fault. Il testing che si intente effettuare, però, aumenta le probabilità di riscontrare nuovi errori.
Malfunzionamento del Server	DEBOLE	Malfunzionamenti del Server possono interrompere drasticamente l'interazione con i nodi client.
Attacchi informatici	DEBOLE	L'uso del web aumenta la probabilità di essere esposti ad attacchi informatici.
Carico di lavoro eccessivo per il Server	DEBOLE	I client che comunicano con il Server potrebbero sbilanciare il carico di lavoro. Tuttavia, questo è un rischio marginale visto il bacino di utenza del sistema attuale.

6.4. Risultati previsti

Le modifiche da apportare al sistema software attuale sono definite secondo la soluzione individuata nel paragrafo 4.2 tenendo in considerazione costi, benefici e rischi di cui abbiamo discusso nel paragrafo 6.