PROJECT PLAN:

1. **Introduzione**

Si vuole realizzare un applicativo che gestisca i noleggi di auto. Gli utenti saranno in grado di accedere ad un’interfaccia dove è possibile selezionare il tipo di noleggio ed il tipo di auto desiderato. Viene offerta la possibilità di scegliere tra 3 tipi di noleggi: car sharing (durata variabile da qualche giorno ad una settimana), breve peiodo (fino a qualche mese), lungo termine (fino ad un anno). Gli utenti poi potranno scegliere tra diversi tipi di auto in base alle proprie esigenze. Verranno proposte auto di tipo: utilitarie, business, luxury.

Al fine di confermare ed inizializzare il noleggio verrà richiesta l’autenticazione presso il servizio e quindi verrà offerto un servizio di registrazione e login.

Al termine del noleggio agli utenti verranno accreditati punti in base ai km percorsi ed al tipo di noleggio effettuato.

1. **Modello di processo**

Metodo Agile – Extreme Programming

Modello che si basa su delle best practices tra cui design semplice, refactoring e programmazione in coppia.

Cinque principi:

* **Feedback rapido**: ogni volta che vengono aggiornate parti del sistema queste vengono controllate e testate dagli sviluppatori, in quanto si pensa di consegnare sempre una versione funzionante al cliente.
* **Semplicità**: si vuole evitare di crearsi ulteriore complessità, per cui si cerca di mantenere un design semplice e di aggiungere funzionalità solo quando necessario ed il codice sarà oggetto di refactoring.
* **Cambiamento incrementale:** non si prendono decisioni improvvise, i cambiamenti vengono effettuati in maniera graduale
* **Abbracciare il cambiamento:** non si fanno pianificazioni a lungo termine, i problemi più gravi vengono risolti il prima possibile.
* **Lavoro di qualità:** si cerca di offrire sempre un prodotto qualitativo

1. **Organizzazione del progetto**

Modello che utilizziamo: XP-Extreme Programming. Tutti lavorano su tutto, quindi l’organizzazione del team è un’organizzazione semplice: tutti i membri saranno al centro del progetto.

1. **Standard, linee guida, procedure**

Linguaggio di programmazione: Java

Linguaggio di modellazione: UML

Programmi e tool utilizzati:

* StarUML per la realizzazione dei modelli, con il plugin Rebel per la stesura dello scheletro del codice.
* Eclipse come ambiente di sviluppo, con il plugin Maven per la gestione delle dipendenze con le librerie e tra i diversi componenti.
* Github per la Configuration Management
* MySQL per la gestione e configurazione dei DB
* Postman per progettare, creare, testare ed iterare le proprie applicazioni.

1. **Attività di gestione**

**1° incontro:** Stesura project plan e organizzazione del lavoro

**2° incontro:** Controllo e modifiche diagrammi UML. Stesura codice sorgente

**3° incontro:** Controllo e modifiche diagrammi UML. Prima versione dell’applicazione compilata. Aggiornamento e modifiche al Project Plan.

**4° incontro:** Controllo codice backend e discussione riguardo l’implementazione del frontend

**5° incontro:** Discussione sulla parte relativa al frontend e sull’implementazione della pagina di login

**6° incontro:** Riunione di aggiornamento relativa agli ultimi aggiornamenti effettuati al codice ed attività di briefing riguardante i prossimi steps.

1. **Rischi**

Gli utenti potrebbero sempre preferire il car sharing agli altri due, facendo decadere il programma fedeltà.

Il software potrebbe non riuscire a gestire adeguatamente un elevato numero di richieste.

1. **Personale**

Team composto da tre persone:

* ALEXANDER RUBINO (MAT. 1064467)
* BENIAMINO INFANTE (MAT. 1063452)
* IBRAHIMA SARR (MAT. 1046446)

1. **Metodi e tecniche**

Il sistema verrà costruito in piccoli step. Tutto il team lavora su tutto e tutti avranno le stesse responsabilità ,rilasciando software che siano funzionanti. Anche per questo lo sviluppo sarà Test Driven, dove ogni versione o modifica del sistema verrà testata. Il Design sarà molto semplice, questo per garantire una semplicità nella modifica ed implementazione di nuovi componenti nel sistema. I test verranno svolti quotidianamente o dopo ogni modifica effettuata per verificare il corretto funzionamento del sistema.

1. **Garanzia di qualità**

Il sistema dovrà seguire il modello di qualità del software ISO 9126, si favorisce il prodotto finale al processo.

1. **Pacchetti di lavoro**

La modellazione del software attraverso i diversi diagrammi citati nel punto 12 viene suddivisa per i tre membri del team al fine di ottimizzare i tempi. Le fasi di stesura del codice e di testing verranno svolte da tutto il team come da programma dell’Extreme Programming.

1. **Risorse**

Vengono utilizzati comuni notebook non dall’elevata potenza di calcolo che comunque permettono di utilizzare i software e tools citati al punto 4.

1. **Budget e programma**

Il programma consiste in:

* Project Plan e organizzazione del progetto
* Modellazione attraverso i diagrammi
* Stesura del codice a partire dallo scheletro ottenuto dal Class Diagram
* Sviluppo del codice, realizzazione di metodi
* Testing del software
* Eventuale manutenzione prima del rilascio

1. **Cambiamenti**

I cambiamenti sono inevitabili. Utilizzando un modello di processo Agile ogni iterazione può includere cambiamenti

1. **Consegna**

Il sistema è pensato per fornire un servizio di noleggi pratico, affidabile e flessibile.

Il cliente sarà in grado di accedere al servizio tramite sito web.