***UNIBite - PROJECT PLAN***

Università degli studi di Bergamo  
Ingegneria Informatica

Beccarelli Raissa Matr. 1086785  
Locatelli Giacomo Matr. 1086262  
Valceschini Marco Matr. 1086356

***Immagine che contiene testo, Carattere, Elementi grafici, simbolo

Descrizione generata automaticamente***

**1 Introduzione**

UNIBite è un progetto dedicato alla gestione delle mense universitarie. Si occupa di gestire la prenotazione di ordini da parte di studenti, docenti e persone esterne, per facilitare e velocizzare il servizio. Permette inoltre al personale di selezionare ogni giorno i piatti disponibili e di controllare le quantità di piatti richieste.

Il progetto è sviluppato tramite una schermata di login a cui studenti e docenti accedono, ottenendo la possibilità di ordinare il loro pasto e di pagarlo sulla base della fascia ISEE indicata nella fase di registrazione. Al contrario, le persone esterne non hanno la possibilità di fare il login e non hanno diritto agli sconti universitari.

Al personale è dedicata una pagina di login, che gli permette di accedere ad un’area di controllo riservata in cui si può gestire il servizio giornaliero.

Il progetto nasce dall’idea di velocizzare il servizio della mensa per renderlo più efficiente, diminuendo i tempi di attesa e facilitando il lavoro al personale.

**2 Modello di processo**

Il progetto si basa su un modello di tipo RAD (Rapid Application Development), per poter utilizzare uno sviluppo evolutivo tramite dei time box che permetteranno di avere i risultati desiderati entro dei tempi stabiliti e inamovibili. Tramite il RAD verrà utilizzata la tecnica di MoSCoW, che permetterà di suddividere i requisiti richiesti sulla base della loro priorità. Essi saranno definiti nel documento della specifica dei requisiti.

**3 Organizzazione del progetto**

Il progetto prevede un team composto da tre membri, organizzati sulla base del team SWAT (Skilled With Advanced Tools), tipico del modello RAD.

È stato scelto questo approccio organizzativo perché consente ai membri di svolgere le attività sulla base delle loro competenze, permettendo loro di affrontare anche altre attività, come il testing, con l’obiettivo di massimizzare la produttività. Inoltre, usando un modello di tipo RAD, il team SWAT è libero di poter decidere la durata delle caselle temporali e decidere quali funzionalità da implementare in ogni timebox.

Essendo un gruppo formato da sole tre persone con simili competenze, non viene definito un leader a capo del progetto che prende le decisioni, ma i tre membri prenderanno le loro decisioni di comune accordo.

**4 Standard, linee guida, procedura**

All’interno del progetto si necessita che ogni membro segua determinati standard e linee guida per avere un progetto di semplice comprensione per tutti.

* **Standard**

Lo standard su cui si basa il progetto è quello di Java Oracle, il quale presenta regole e direttive prefissate.

* **Linee guida**

I membri baseranno il progetto sulle linee guida determinate dai diagrammi UML (Unified Modeling Language), che saranno definiti nella documentazione allegata. Per facilitare la collaborazione tra membri utilizzeremo l’IDE di Eclipse e la sua estensione di WindowBuilder, e il tool GitHub, ovvero un software control management distribuito. Inoltre si introduce la possibilità di utilizzare la piattaforma di Vaadin per implementare la parte grafica.

* **Procedura**

Il progetto si comporrà delle seguenti fasi, ciascuna con relativo documento: ingegneria dei requisiti, architettura e progettazione, implementazione e testing.

La documentazione relativa al progetto verrà consegnata per sottoporla a revisione entro un mese dalla consegna dell’intero progetto. I membri si occuperanno di aggiornarla in caso di nuove modifiche necessarie.

**5 Attività di gestione**

Le attività relative alla gestione del progetto seguono gli obiettivi e le priorità stabilite all’interno del metodo MoSCoW. I membri si obbligano ad incontrarsi all’inizio e alla fine di ogni timebox, per definire le attività da svolgere e per confrontarsi sullo stato del progetto. Viene quindi redatta alla fine di ogni timebox una relazione di sintesi sullo stato di avanzamento.

Durante ogni timebox sarà utilizzata per visionare i progressi la Kanbard Board, che verrà implementata tramite la piattaforma GitHub. Essa permetterà ai membri di avere una visualizzazione chiara del flusso di lavoro, limitando la quantità di lavoro assegnata e tracciando il progresso anche in tempo reale.

**6 Rischi**

Il principale rischio a cui il progetto può andare incontro nel caso in cui ci sia un guasto del dispositivo hardware su cui è caricato il database locale, è la perdita dei dati di registrazione che gli utenti, nel tempo, hanno inserito nel sistema.

Un altro rischio a cui il sistema va incontro è l’informazione critica delle porzioni di piatti disponibili. Infatti, si potrebbe incorrere nell’errore umano del calcolo e della distribuzione delle porzioni, che non essendo precise potrebbero portare ad avere uno o più ordini che non possono essere soddisfatti. Al contrario potrebbe accadere che il numero di porzioni previste sia inferiore alla reale quantità di porzioni disponibili. Una soluzione a questo tipo di rischio potrebbe essere un aggiornamento in tempo reale delle porzioni disponibili.

**7 Personale**

Il personale è composto dai tre membri del progetto che dovranno contribuire per tutta la durata dello sviluppo sfruttando al massimo le proprie conoscenze e capacità. I tre membri collaboratori sono:

* Beccarelli Raissa
* Locatelli Giacomo
* Valceschini Marco

Non sarà necessario l’intervento di personale esterno.

**8 Metodi e tecniche**

Per quanto riguarda la gestione del controllo della versione e della configurazione utilizzeremo i tools di Git e GitHub, che permetteranno di mantenere uno storico delle modifiche effettuate e delle versioni create. Per organizzare al meglio il lavoro si farà utilizzo anche della Kanban Board, strumento di GitHub.

Nella fase di ingegneria dei requisiti verrà effettuato il processo di elicitazione dei requisiti, tramite la tecnica dell’analisi dello scenario, della derivazione da un sistema esistente (si prende spunto dall’idea dei totem all’interno dei fast food) e della prototipazione.

Come detto precedentemente, nella fase di implementazione baseremo l’ordine delle attività da seguire sul metodo MoSCoW.

Per quanto riguarda la fase di testing, si rimanda al file allegato “Testing.docx”.

**9 Garanzie di qualità**

Per garantire che il sistema sia di ottima qualità, esso sarà fondato su alcuni fattori di qualità di McCall. Il progetto, infatti, si basa sui criteri chiave di:

* Correttezza, in quanto vogliamo che esso risponda alle specifiche identificate all’interno della specifica dei requisiti e che possa quindi essere in grado di soddisfare l’obiettivo principale, ovvero il semplificare e velocizzare il servizio;
* Affidabilità, per fare in modo che il progetto sia sempre in grado di garantire il corretto svolgimento del servizio, senza creare errori che possano portare a problemi di livello economico-organizzativo (si veda il rischio dell’informazione critica);
* Integrità, per garantire che gli accessi riservati al personale all’interno del sistema di modifiche vengano eseguiti in modo sicuro, evitando quindi che persone non autorizzate accedano a quest’area. Si vuole inoltre che studenti e docenti accedano con le loro relative credenziali e che queste mantengano un livello di sicurezza adeguato;
* Usabilità, si vuole garantire che le interfacce siano di facile comprensione sia dal lato degli utenti che dal lato del personale, in modo da poter velocizzare il processo;
* Manutenibilità, per garantire che il sistema possa essere integrato con modifiche in caso di rilevamento di errori e bug, e per effettuare migliorie anche dopo la consegna del lavoro.

**10 Pacchetti di lavoro**

Le attività lavorative sono suddivise in slot lavorativi con diversi obiettivi determinati dai timebox e gestiti tramite la Kanban Board.

**11 Risorse**

Dal punto di vista dell’hardware sono utilizzati tre personal computer, su cui sono utilizzate la piattaforma Git e GitHub, ed è installato il software Eclipse IDE per la programmazione con linguaggio Java e con le sue estensioni di:

* Maven, per gestire il progetto software e per poterlo spostare da un ambiente ad un altro avendo sicurezza di avere sempre le stesse versioni delle librerie;
* WindowBuilder, per quanto riguarda la gestione dell’interfaccia grafica;
* Papyrus, per la creazione e la gestione dei diagrammi UML.

Per lo sviluppo del progetto, inoltre, si farà utilizzo di un database embedded locale tramite SQLite.

Per quanto riguarda lo sviluppo dell’interfaccia grafica, oltre all’utilizzo di WindowBuilder di Java, teniamo in considerazione anche la possibilità di utilizzare la piattaforma Vaadin.

Dal punto di vista delle risorse umane, il team sarà composto da tre membri che lavoreranno e collaboreranno per tutto il tempo necessario per la gestione e lo sviluppo del progetto, raggiungendo gli obiettivi e i requisiti fissati.

**12 Budget**

Il budget è basato sull’impiego del tempo a disposizione, suddividendo il lavoro così come segue:

* Decisione del progetto
* Definizione di un project plan
* Definizione della specifica dei requisiti, con la fase di elicitazione dei requisiti
* Definizione dell’architettura software
* Fase di progettazione/design
* Creazione dei diagrammi UML
* Scrittura del codice
* Testing

Per il progetto non è previsto un budget in termini monetari.

**13 Cambiamenti**

All’interno di un progetto con un modello di processo RAD, i cambiamenti vengono visti come delle modifiche che fanno parte dei processi evolutivi gestiti in ogni timebox, trattati quindi in modo leggero come proprio dei metodi agili. Ogni modifica sarà gestita e documentata nello storico dei commit e degli issues di GitHub.

**14 Consegna**

La consegna verrà effettuata, con la relativa documentazione allegata, tramite la piattaforma di GitHub tramite il repository condiviso.