GIMIGLIANO MARIANNA – 0000915343, DAGHIA MARTINA – 0000915665, ZAULI MARTINA – 0000915901

RELAZIONE E MANUALE D’USO

**DESCRIZIONE DEL METODO ADOTTATO:**

Il progetto e la sua gestione sono stati sviluppati secondo la metodologia Scrum.

* RUOLI

Abbiamo utilizzato i ruoli definiti all’interno dello Scrum Team(SCRUM MASTER, PRODUCT OWNER, DEVELOPMENT TEAM), e abbiamo lavorato in stretta connessione per assicurare un continuo e veloce flusso di informazioni. Abbiamo ricoperto questi ruoli a rotazione.

Essendo solo 3 componenti all’interno del gruppo il membro che svolgeva il ruolo di Scrum Master svolgeva contemporaneamente anche il ruolo di componente del Development Team. In questo modo siamo riuscite a lavorare più velocemente ed efficacemente.

* EVENTI

SPRINT PLANNING: prima dell’inizio di ogni sprint il team ha effettuato una riunione per popolare lo Sprint Backlog e quindi pianificare il lavoro da svolgere durate l’iterazione corrente. La decisone delle priorità dei task è stata decisa in base all’effort necessario e alle priorità già assegnate dal Product Owner.

DAILY SCRUM: ogni mattina abbiamo avuto un breve confronto per creare un piano giornaliero per decidere e sincronizzare le attività.

SPRINT REVIEW: abbiamo effettuato delle revisioni alla fine di ogni sprint per valutare se l’obiettivo prefissato era stato raggiunto e se sorgevano delle problematiche da risolvere.

SPRINT RETROSPECTIVE: alla fine di ogni sprint abbiamo svolto un’ulteriore analisi per valutare cosa continuare a fare, cosa smettere di fare e cosa migliorare nello sprint successivo per ottenere performance ancora più efficienti.

* MILESTONES

Implementazione delle varie pagine grafiche

Login degli utenti a seconda della tipologia (studente, docente, ecc.)

Collegamento dei metodi delle classi “database” con i metodi delle classi “greeting”

Completamento metodi Studente

Completamento metodi Docente

Completamento metodi Amministratore

Completamento metodi Segreteria

* ISSUES

Capire il funzionamento generale di GWT e la strutturazione dei package nel progetto.

Rendere compatibili le versioni di GWT, Java, librerie per rendere il progetto funzionante.

Capire il collegamento dei metodi delle classi “database” con i metodi delle classi “greeting”.

Strutturare i BTreeMap in modo efficace nel database.

Gestire alcuni errori a run-time dei quali non riuscivamo a risalire alla causa.

Come gestire l’invio dei voti alla Segreteria e solo successivamente la pubblicazione.

* USER STORIES

Seguendo quando progettato nella fase di Inception nel modello dei casi d’uso il Development Team ha creato le User Stories. Ha fatto in modo che fossero scritte in modo semplice e subito comprensibile. Queste sono state poi utili per creare i task da inserire all’interno dello Sprint Backlog.

Non è stato necessario suddividere ulteriormente i casi d’uso per creare le stories, in quanto questi erano già sufficientemente piccoli e facilmente risolvibili.

* CRITERI PER L’ASSEGNAZIONE DI PRIORITÀ AI TASK

Durante gli Sprint Planning il Development Team ha selezionato un sottoinsieme dei task in cima al Product Backlog. Per questi task ha cercato di assegnare delle priorità sulla base della stima dell’effort necessario e sulle necessità tecniche di svolgere alcuni task prima di altri.

Per cominciare è stata assegnata priorità più alta ai task riguardanti le parti grafiche dell’applicazione, in quanto costituivano la base per visualizzare i risultati delle feature successive. A seguire venivano le stories riguardanti le feature di ogni utente, nuovamente con priorità a quelle utili per visualizzare i risultati delle altre (es. visualizzare le informazioni sui corsi prima di gestire la modifica dei dati dei corsi).

* ARTEFATTI SCRUM

TRELLO – PRODUCT BACKLOG

Abbiamo utilizzato Trello come project tracker. È stato utilizzato come PRODUCT BACKLOG.

Per memorizzare il backlog ci siamo servite della bacheca offerta da Trello.

Prima dell’inizio dello sviluppo del progetto il Product Owner ha popolato il backlog con tutti gli elementi che sarebbero andati a comporre l’applicazione. Principalmente quindi è stato popolato con nuove feature da implementare. In seguito il Product Owner ha riordinato il backlog, spostando in cima gli item con maggiore priorità.

Tramite Trello sono stati quindi inseriti in ordine gli item da svolgere nella sezione “da fare”, quelli in lavorazione (cioè quelli inseriti nello sprint corrente) nella sezione “in esecuzione”, quelli completati (cioè quelli completati nei vari sprint) nella sezione “fatto”.

Il Product Owner ha aggiornato durante il tempo il backlog con nuove feature. Queste feature sono state proposte dal Development Team durante la fase di sviluppo, perché ritenute utili e una buona aggiunta all’applicazione.

\*(LINK ALLA PAGINA TRELLO NEL FILE “DIARIO DI PROGETTO”)

JIRA – SPRINT BACKLOG

Abbiamo utilizzato Jira come project tracker. È stato utilizzato come SPRINT BACKLOG.

Per memorizzare il backlog ci siamo servite della bacheca offerta da Jira.

Lo sviluppo del progetto ha richiesto 3 Sprint, ciascuno della durata di una settimana (7 giorni). Prima dell’inizio di ogni sprint, durante lo Sprint Planning, il Development Team ha popolato lo Sprint Backlog.

Tramite Jira sono stati quindi inseriti in ordine gli item …

\*(LINK ALLA PAGINA JIRA NEL FILE “DIARIO DI PROGETTO”)

**RELAZIONE SU SCELTE PROGETTUALI E DI INTERFACCIA**

Uniwebite è un software che simula il sito web dell’Università di Bologna.

È stato interamente sviluppato in Eclipse, con l’utilizzo di GWT e scritto in Java.

Il cuore del progetto è contuenuto nel package SRC mentre i test delle funzionalità dell’applicazione nel package Test.

GWT

Seguendo la strutturazione di un progetto GWT l’SRC è diviso ulteriormente in 3 ulteriori package.

Nel package CLIENT abbiamo sviluppato tutte le classi che implementano la grafica e quelle che la gestiscono. Compare quindi lo stesso nome di file due volte, uno NomeFile.ui.xml nel quale è disegnata la grafica, e uno NomeFile.java nel quale sono richiamati tutti gli eventi da gestire durante l’interazione dell’utente con la grafica. Nelle classi java vengono richiamati i metodi implementati nel package Server. In più qui si trovano la classe GreetingService.java che contiene le interfacce dei metodi per la comunicazione sincrona con il Server, e la classe GreetingServiceAsync.java che contiene le interfacce dei metodi per la comunicazione asincrona con il Server. Infine si trova UniWebsite.java che costituisce l’Entry Point dell’applicazione.

Nel package SERVER abbiamo sviluppato tutte le classi che implementano i metodi che gestiscono il database. È stata creata una classe, quindi un database, per ogni oggetto dell’applicazione. In più qui si trova la classe GreetingServiceImpl.java che realizza le interfacce delle classi nel package Client e per ogni metodo, richiama la sua effettiva implementazione nelle classi sopra citate.

Nel package SHARED abbiamo sviluppato tutte le classi che rappresentano gli oggetti dell’applicazione, con relativi costruttori, metodi getter e setter.

MAPDB

I dati sono resi persistenti nell’applicazione grazie all’utilizzo come database di MapDB. Abbiamo creato un database per ogni categoria di oggetti dell’applicazione. Per memorizzare effettivamente i dati abbiamo scelto la struttura dati BTreeMap. I BTreeMap sono realizzati come una mappa chiave-valore. Nel nostro caso sono state sfruttate così:

* BTreeMap<Email Utente, Utente>
* BTreeMap<Nome Corso, Corso>
* BTreeMap<Nome Esame, Esame>
* BTreeMap<Id Iscrizione Corso, Iscrizione Corso>
* BTreeMap< Id Iscrizione Esame, Iscrizione Esame>
* BTreeMap< Id Voto, Voto>

Sono state implementate queste classi “Iscrizione Corso” e “Iscrizione Esame” e i relativi database per trovare una soluzione che potesse risolvere il problema di mantenere delle liste di utenti per corsi ed esami.

TESTING

Tramite l’utilizzo di JUnit 5 e Mokito sono stati effettuati Unit Test su tutte le operazioni di inserimento, modifica, eliminazione di dati dal database per verificarne a mano a mano la correttezza.

**MANUALE DI INSTALLAZIONE DEL SOFTWARE E MANUALE UTENTE**

* INSTALLAZIONE DEL SOFTWARE

Per installare il software è necessario scaricare una copia del progetto sul proprio computer. In seguito è necessario aprire il progetto all’interno dell’IDE preferito. Bisogna tenere presente che per utilizzare il progetto è necessario aver scaricato GWT. Il software è stato sviluppato con gwt-2.9.0. Tutti gli altri jar che permettono il funzionamento del software sono già inclusi del progetto.

Per lanciare il software è necessario selezionare la voce “Run as” e la voce “GWT Development mode with Jetty”. In seguito, nella sezione console “Development Mode” comparirà il link con il quale aprire nel browser la homepage del sito universitario.

* MANUALE UTENTE

All’apertura il sito mostra la homepage dell’Università di Bologna.

Tramite i bottoni in alto è possibile:

* DIPARTIMENTI: Visualizzare i dipartimenti dell’università
* CONTATTI: Visualizzare i contatti dell’università
* LOGIN: Visualizzare il form per effettuare il login alla piattaforma

A seguito del login l’utente viene indirizzato alla pagina apposita sulla base della tipologia di utente.

FUNZIONI STUDENTE

All’apertura il sito mostra la homepage per gli utenti Studente.

Tramite i bottoni in alto è possibile:

* CORSI: Visualizzare i corsi disponibili e iscriversi ad un corso. Per fare ciò selezionare tramite la listbox il corso d’interesse e cliccare il bottone “Iscriviti”
* ESAMI: Visualizzare gli esami disponibili e iscriversi ad un esame. Per fare ciò selezionare tramite la listbox l’esame d’interesse e cliccare il bottone “Iscriviti”
* VOTI: Visualizzare i propri voti
* AREA PERSONALE: Visualizzare i propri dati personali
* LOGOUT: Effettuare il logout

FUNZIONI DOCENTE

All’apertura il sito mostra la homepage per gli utenti Docente.

Tramite i bottoni in alto è possibile:

* VISUALIZZA CORSI ED ESAMI: Visualizzare i propri corsi ed esami
* GESTISCI CORSI: Gestire i propri corsi

1. Creare un corso. Per fare ciò riempire tutti i campi con le informazioni e cliccare il bottone “Crea Corso”
2. Modificare un corso. Per fare ciò riempire i campi che si desidera modificare con le nuove informazioni e cliccare il bottone “Modifica Corso”
3. Eliminare un corso. Per fare ciò selezionare tramite la listbox il corso d’interesse e cliccare il bottone “Elimina Corso”

* GESTISCI ESAMI: Gestire i propri esami

1. Creare un esame. Per fare ciò riempire tutti i campi con le informazioni e cliccare il bottone “Crea Esame”
2. Modificare un esame. Per fare ciò riempire i campi che si desidera modificare con le nuove informazioni e cliccare il bottone “Modifica Esame”
3. Eliminare un esame. Per fare ciò selezionare tramite la listbox l’esame d’interesse e cliccare il bottone “Elimina Esame”

* GETISCI VOTI: Inviare voti alla segreteria. Per fare ciò selezionare tramite la listbox l’esame, selezionare tramite la listbox lo studente, inserire il voto e cliccare il bottone “Aggiungi Voto”
* AREA PERSONALE: Visualizzare i propri dati personali
* LOGOUT: Effettuare il logout

FUNZIONI SEGRETERIA

All’apertura il sito mostra la homepage per gli utenti Segreteria.

Tramite i bottoni in alto è possibile:

* VISUALIZZA STUDENTI: Visualizzare tutti gli studenti della piattaforma
* PUBBLICA VOTI: Pubblicare i voti inviati dai docenti. Per fare ciò selezionare tramite la listbox lo studente d’interesse e cliccare il bottone “Cerca”. In seguito, se lo studente ha voti non ancora pubblicati compariranno della listbox sottostante. Selezionare il voto d’interesse e cliccare il bottone “Pubblica Voto”
* LOGOUT: Effettuare il logout

FUNZIONI AMMINISTRATORE

All’apertura il sito mostra la homepage per l’utente Amministratore.

Tramite i bottoni in alto è possibile:

* VISUALIZZA ACCOUNT: Visualizzare tutti gli utenti della piattaforma
* CREA ACCOUNT: riempire tutti i campi con le informazioni e cliccare il bottone “Crea Studente”/”Crea Docente”/”Crea Segreteria”
* MODIFICA ACCOUNT: selezionare tramite la listbox l’utente d’interesse, riempire i campi che si desidera modificare con le nuove informazioni e cliccare il bottone “Modifica Studente”/” Modifica Docente”
* LOGOUT: Effettuare il logout