Leonardo Billi - 331332

Chiara Lombardo - 330350

Indice

Documentazione

* Abstract-sommario del Progetto
* Obiettivo del Progetto
* Analisi dei Requisiti
* Glossario dei principali termini
* Requisiti di Sistema
* Descrizione testuale dell’architettura del sistema

Diagrammi UML

* Diagramma di classe
* Diagramma di un caso d’uso
* Diagramma di sequenza e diagramma di collaborazione
* Diagramma di stato
* Diagramma di attività

Casi di Test funzionali

* Black-box

**Documentazione**

**Abstract-sommario del Progetto**

Questo progetto prevede la creazione di un canale per la gestione delle consegne di compiti, in cui i partecipanti sono di due tipi: docenti e studenti.

Gli studenti possono ottenere l'elenco dei docenti, oppure caricare le foto di una prova scritta cartacea da consegnare al docente.

D’altra parte, i docenti possono visualizzare la lista degli studenti che hanno effettuato consegne, e visualizzare o scaricare le foto caricate dagli studenti.

Le consegne vengono registrate, con relativa data e ora, in un’apposita tabella del database lato server. Vengono poi salvate in un file di testo scaricabile per una facile revisione da parte del docente. Il docente inoltre può anche decidere eliminare dal database le foto e le informazioni relative ai compiti consegnatigli.

**Obiettivo del progetto**

L’obiettivo principale del progetto è fornire un canale interattivo efficiente ed affidabile per la gestione delle consegne dei compiti, consentendo una comunicazione diretta tra docenti e studenti, facilitando la raccolta delle prove scritte e mantenendo le informazioni delle consegne in modo sicuro all'interno di un database lato server.

**Analisi dei requisiti**

**Glossario dei principali termini**

* **Canale (Bot Telegram)**: Una piattaforma interattiva su cui avviene la comunicazione tra docenti e studenti per la gestione delle consegne di compiti scritti.
* **Studenti**: I partecipanti del canale che hanno il ruolo di studenti e inviano le foto delle loro prove scritte ai docenti per la valutazione.
* **Docenti**: I partecipanti del canale che hanno il ruolo di insegnanti o professori e ricevono le consegne dei compiti dagli studenti.
* **Autenticazione**: Processo mediante il quale l'identità di un utente (studente o docente) viene verificata e confermata, attraverso l’utilizzo di credenziali, per accedere al sistema.
* **Visualizzazione della Lista dei Docenti**: Funzionalità del bot disponibile per gli studenti che permette di ottenere l'elenco dei docenti.
* **Consegna**: Funzionalità del bot disponibile per gli studenti che permette di caricare le foto di una prova scritta e consegnarla a un docente specifico utilizzando il suo ID (= matricola).
* **Visualizzazione della Lista dei Compiti Consegnati**: Funzionalità del bot disponibile per i docenti che permette di visualizzare la lista degli ID degli studenti che hanno consegnato compiti.
* **Visualizzazione di un Compito**: Funzionalità del bot disponibile per i docenti che permette di visualizzare o scaricare le foto caricate dagli studenti relativi a un determinato ID studente.
* **Database**: Una struttura dati lato server che mantiene le informazioni relative ai docenti, agli studenti e alle consegne, inclusi dati come ID, password, data e ora.

**Requisiti di Sistema**

* **Autenticazione**: Il sistema deve consentire agli utenti (studenti o docenti) l’autenticazione in modo sicuro, impedendo l’accesso a utenti non autorizzati.
* **Visualizzazione della Lista dei Docenti**: Il sistema deve fornire agli studenti l’elenco dei docenti, in modo che gli studenti possano selezionare il docente destinatario della consegna.
* **Consegna**: Il sistema deve consentire agli studenti di caricare foto di una prova scritta cartacea e consegnarla al docente selezionato tramite il suo ID (= Cognome e Nome del docente).
* **Visualizzazione della Lista dei Compiti Consegnati**: Il sistema deve permettere ai docenti di visualizzare la lista degli ID degli studenti che hanno effettuato consegne, per facilitare la valutazione dei compiti.
* **Visualizzazione di un Compito**: Il sistema deve consentire ai docenti di visualizzare o scaricare le foto caricate dagli studenti relativi a un determinato ID studente.
* **Interfaccia Utente**: Il canale deve avere un’interfaccia utente intuitiva e semplice da utilizzare, sia per i docenti che per gli studenti.
* **Sicurezza**: Il sistema deve garantire un’adeguata sicurezza dei dati, compreso l’utilizzo di un’autenticazione per l’accesso e la protezione delle informazioni personali degli utenti.

**Descrizione testuale dell’architettura del sistema**

Il sistema è progettato per gestire la consegna di compiti scritti tra docenti e studenti.

L’architettura del sistema è costituita da diversi componenti interconnessi che collaborano per garantire un’efficace e sicura gestione delle consegne e delle interazioni tra gli utenti. Questi componenti comprendono un’interfaccia utente intuitiva, un’autenticazione sicura (per garantire la sicurezza dei dati) e una gestione dei dati tramite database.

L’architettura del sistema è progettata per essere scalabile, robusta ed efficiente, fornendo un canale di comunicazione diretto e sicuro tra docenti e studenti per la consegna dei compiti scritti.

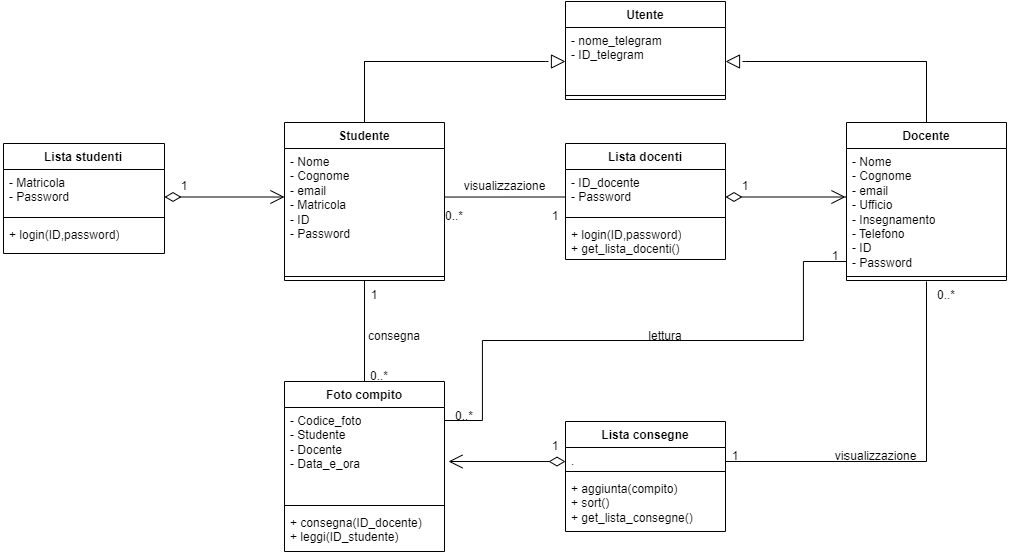
Un’attenzione particolare è dedicata allo sviluppo di un’interfaccia utente intuitiva e facile da utilizzare.

Il codice sorgente del sistema è sviluppato utilizzando il linguaggio di programmazione Python, servendosi della libreria Telegram per gestire le varie funzionalità del bot.

Per la generazione e gestione del database si usa la libreria SQLite (sqlite3)

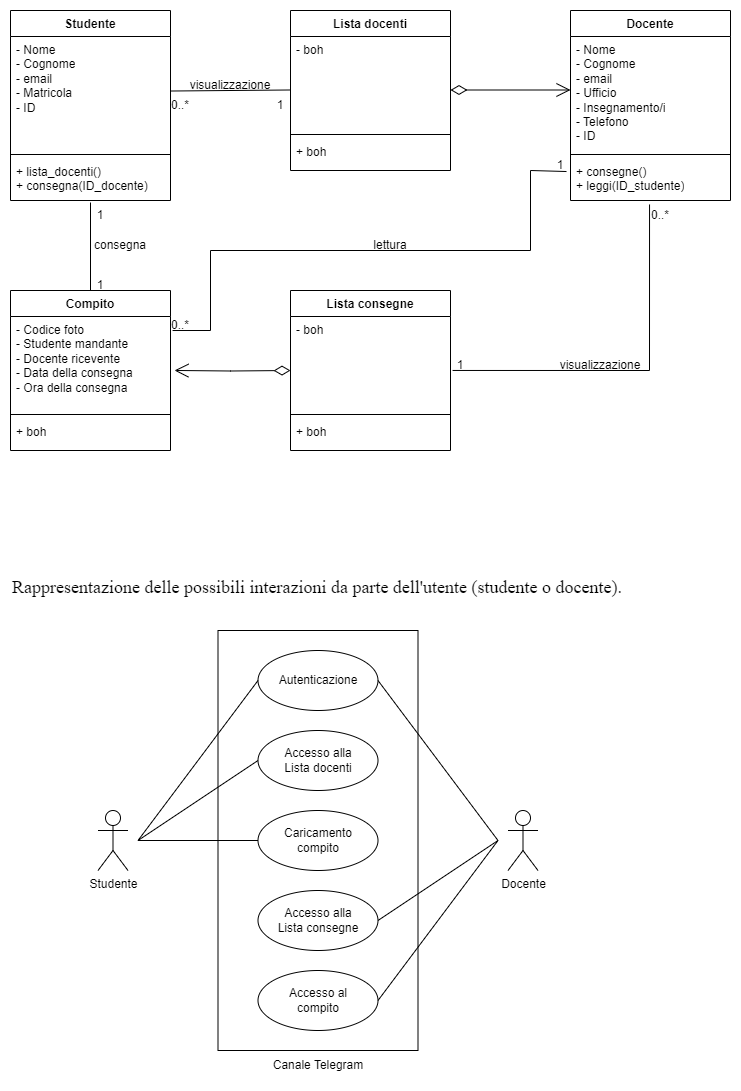
Viene inoltre utilizzata la libreria BeautifulSoup (bs4) per la selezione dei dati riguardanti i docenti dalla pagina web del sito del dipartimento, da inserire nella rispettiva tabella del database.

**Diagrammi UML**

**Diagramma di classe**

**Diagramma di un caso d’uso**

Rappresentazione delle possibili interazioni da parte dell’utente (studente o docente).



**Diagramma di sequenza e diagramma di collaborazione**

Diagramma di sequenza relativo al caso d’uso “consegna di un compito da parte di uno studente”.

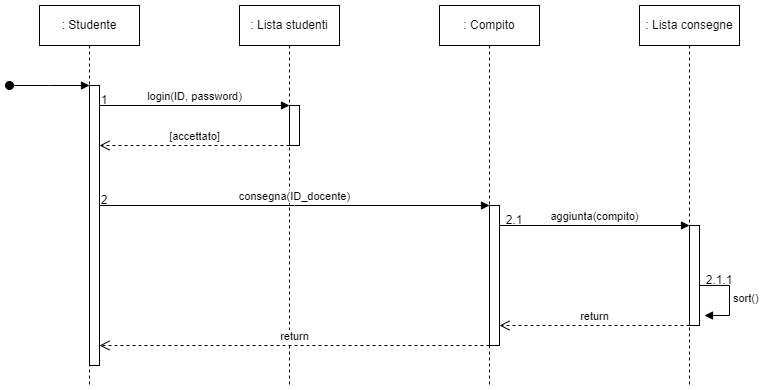
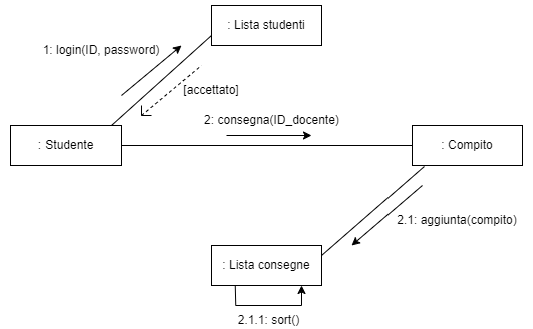
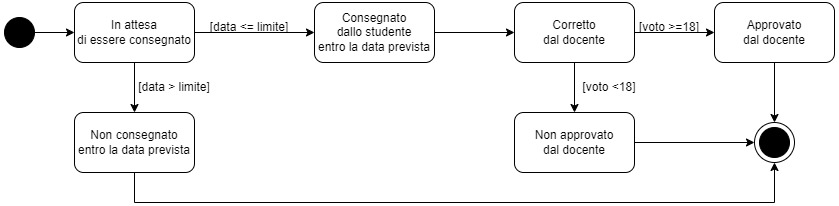


Diagramma di collaborazione relativo al caso d’uso “consegna di un compito da parte di uno studente”.



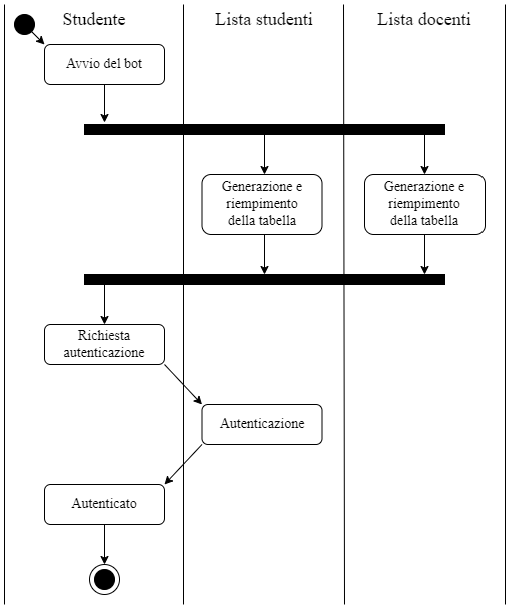
**Diagramma di stato di un elemento significativo del sistema**

Diagramma di stato relativo ad un “compito”.



**Diagramma di una attività del sistema**

Diagramma di attività relativo alla attività “avvio del bot e autenticazione” da parte di uno studente.



**Casi di Test funzionali**

**Black-box**

I casi di test funzionali rappresentano una parte fondamentale nell’analisi delle funzionalità di un progetto, in quanto valutano l’input e l’output del software nel suo ambiente operativo. Attraverso di essi, vengono verificati i comportamenti e le operazioni specifiche del sistema per garantire che esso soddisfi le aspettative e gli obiettivi stabiliti.

Le tecniche di progettazione dei test funzionali mirano a identificare una serie di casi di prova, in modo da verificare in modo approfondito diverse parti del sistema, evidenziando le interazioni chiave e le operazioni critiche per il suo corretto funzionamento.

I casi di test riguardano principalmente i diversi input potenziali forniti dagli utenti, sia studenti che docenti, al fine di convalidarne l’accuratezza. Questo processo mira a prevenire un uso improprio delle funzionalità del bot, garantendo un utilizzo corretto e adeguato, favorendo una minimizzazione di possibili errori o problemi di prestazioni.

Di seguito è fornito un insieme di casi di test funzionali per l’aspetto del progetto dell’autenticazione dell’utente, svolta attraverso un ID e una password.

**Valori**:

* Utente: char;
* ID: stringa alfanumerica di massimo 16 caratteri;
* Password: stringa alfanumerica di massimo 12 caratteri.

**Vincoli**:

* Utente: può assumere valore solo S (Studente) o D (Docente).
* ID:
* se l’utente è uno studente, l’ID è una stringa esclusivamente numerica di esattamente 6 cifre;



* se l’utente è un docente, l’ID è una stringa esclusivamente alfabetica di lunghezza variabile tra 5 e 12 caratteri compresi.



* Password: è una stringa alfanumerica di lunghezza compresa tra 4 e 8 caratteri.



**Tabella dei Test Funzionali (Black-box)**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Casi di test** | | | | |
| **Caso** | **Utente** | **ID** | **Password** | **Validità** |
| 1 | S | 331332 | Ciao01 | Valido |
| 2 | D | Milani | Ciao01 | Valido |
| 3 | D | **Rossi** | **Ciao** | Estremo valido |
| 4 | D | **Guastaveglia** | **Bianco01** | Estremo valido |
|  | | | | |
| 5 | **T** | 331350 | Ciao01 | Non valido |
| 6 | S | **Milani** | Ciao01 | Non valido |
| 7 | S | **33** | Ciao01 | Non valido |
| 8 | S | **33133233** | Ciao01 | Non valido |
| 9 | D | **331332** | Ciao01 | Non valido |
| 10 | D | **Lu** | Ciao01 | Non valido |
| 11 | D | **BiancoLombardo** | Ciao01 | Non valido |
| 12 | S | 331332 | **Ci** | Non valido |
| 13 | S | 331332 | **BiancoRosso** | Non valido |
| 14 | S | **3** | Ciao01 | Estremo non valido |
| 15 | S | **33133** | Ciao01 | Estremo non valido |
| 16 | S | **3313323** | Ciao01 | Estremo non valido |
| 17 | S | **3313323313323313** | Ciao01 | Estremo non valido |
| 18 | D | **L** | Ciao01 | Estremo non valido |
| 19 | D | **Luca** | Ciao01 | Estremo non valido |
| 20 | D | **RossoLombardo** | Ciao01 | Estremo non valido |
| 21 | D | **LombardoLombardo** | Ciao01 | Estremo non valido |
| 22 | S | 331332 | **L** | Estremo non valido |
| 23 | S | 331332 | **Leo** | Estremo non valido |
| 24 | S | 331332 | **Lombardo1** | Estremo non valido |
| 25 | S | 331332 | **Lombardo2001** | Estremo non valido |

Nella tabella sono riportati:

* 4 casi validi, di cui 2 casi estremi;
* 21 casi non validi, di cui 12 casi estremi. Per ognuno di questi è evidenziato in rosso il valore non valido.