

UniBg Notes

Design e Architettura

Progetto di Ingegneria del software

Bilal Drissi 1087161

Daniele Iania 1085982

Mattia Nicastro 1086903

a.a 2024/2025



INDICE

Introduzione	1
Architettura del Sistema	2
Dettagli del Design	3
Misurazioni del Codice	4
Conclusioni	5

1. Introduzione

Questo documento descrive il design del sistema software UniBg Notes, progettato per agevolare gli studenti nella gestione e condivisione dei propri appunti universitari. L'obiettivo è fornire una guida chiara e condivisa per l'implementazione del sistema, basata sui requisiti definiti.

Le principali funzionalità del progetto includono:

- Condivisione di appunti con altri utenti tramite funzionalità di upload, ricerca e download.
- Gestione di profili utente per personalizzare l'esperienza.
- Accesso sicuro tramite autenticazione con credenziali.
- Condivisione di pensieri personali.

Questo documento è strutturato per fornire una visione completa delle scelte architetturali e progettuali, con un focus su semplicità, manutenibilità e scalabilità del sistema.

2. Architettura del Sistema

2.1 Panoramica

L'architettura del sistema segue il pattern MVC (Model-View-Controller) per separare la logica applicativa dalla presentazione e garantire una maggiore modularità.

Tecnologie principali:

- **Frontend:** Vaadin per l'interfaccia utente.
- **Backend:** Java per la logica applicativa.
- **Database:** SQLite per la gestione dei dati.

2.2 Componenti principali

1. Frontend:

- Composto da pagine web dinamiche gestite da Vaadin.
- Pagine:
 - Login
 - Registrazione
 - Recupero password
 - Caricamento appunti
 - Ricerca appunti
 - Condivisione di post
 - Profilo

2. Backend:

- Gestisce la logica di esecuzione e le operazioni sul database.

3. Database:

- Schema logico:
 - **users**(ID, username, email, password, name, surname, image, role)
 - **faculties**(ID, name)
 - **courses**(ID, id_faculty, name)
 - **note_types**(ID, type)
 - **notes**(ID, id_user, id_course, id_type, path, description, dateTime)
 - **posts**(ID, id_user, text, date)
 - **comments**(ID, id_user, id_post, text, date)
 - **documents**(ID, uploaded_by, path)

3. Dettagli del Design

3.1 Autenticazione e Autorizzazione

- Gli utenti devono registrarsi fornendo email e password.
- Password archiviate utilizzando un algoritmo di hashing sicuro.
- Accesso regolato tramite sessioni.

3.2 Modulo di Ricerca

- Implementato con filtri per nickname, dipartimento, corso di studi, ricerca per testo.
- Risultati ordinati per data di caricamento, utilizzando query SQL.

3.3 Gestione dei Documenti

- Funzionalità principali:
 - Caricamento appunti personali.
 - Download di appunti di altri utenti.

3.4 Interfaccia Utente

- Design responsive per desktop e mobile.
- Utilizzo di layout a griglia verticale/orizzontale per garantire una distribuzione chiara delle informazioni.

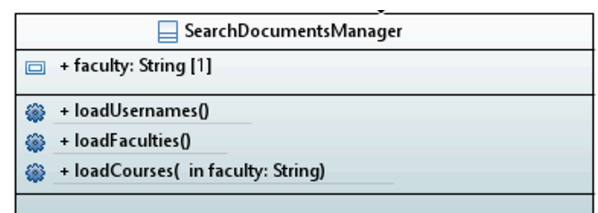
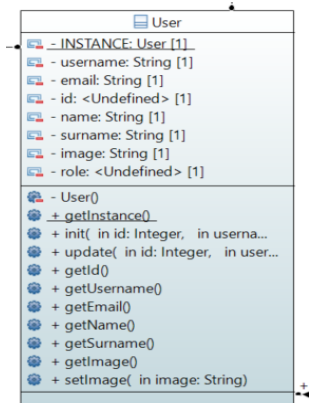
3.5 Design Patterns

- **Singleton**

Siccome è necessaria una sola istanza condivisa, in tutto il sistema, della classe *Utente*, si è deciso di utilizzare il design pattern *Singleton*. Questo evita la duplicazione di istanze e migliora l'efficienza garantendo una gestione coerente dello stato globale.

- **Delegation**

L'utilizzo di questo pattern risulta appropriato qualora si desideri sfruttare in modo più efficiente un metodo già presente in un'altra classe. In questo contesto, i compiti specifici della classe *SearchDocumentsManager* vengono delegati a un'altra classe, riducendo la complessità del



sistema e garantendo una chiara separazione delle responsabilità. Il *Delegation Pattern* semplifica la manutenzione del codice, favorisce la riusabilità e agevola l'adozione del principio Single Responsibility.

4. Misurazioni del Codice

4.1 Complessità basata sulle Linee di Codice

Source File ^	Total Lines	Source Code Lines	Source Code Lines [%]	Comment Lines	Comment Lines [%]	Blank Lines	Blank Lines [%]
Comment.java	40	31	78%	0	0%	9	22%
CommentValidator.java	13	10	77%	0	0%	3	23%
DatabaseManager.java	652	314	48%	194	30%	144	22%
FileManager.java	45	32	71%	1	2%	12	27%
LoadNotesPage.java	181	141	78%	7	4%	33	18%
LoginPage.java	103	66	64%	13	13%	24	23%
MaxLengthException.java	8	6	75%	0	0%	2	25%
NavBar.java	80	78	87%	0	0%	12	13%
Note.java	56	42	75%	0	0%	14	25%
NotesComponent.java	140	108	77%	0	0%	32	23%
PasswordManagerPage.java	132	111	84%	3	2%	18	14%
Post.java	90	63	70%	1	1%	26	29%
PostComponent.java	158	123	78%	3	2%	32	20%
PostsPage.java	85	67	71%	10	11%	18	19%
ProfilePage.java	138	105	76%	6	4%	27	20%
PublishBar.java	55	43	78%	1	2%	11	20%
RegistrationPage.java	80	60	75%	3	4%	17	21%
RootRoute.java	20	14	70%	0	0%	6	30%
SearchDocumentsPage.java	114	87	85%	0	0%	17	15%
SearchDocumentsManager.java	25	15	60%	0	0%	10	40%
Session.java	34	27	79%	0	0%	7	21%
TestDatabaseManager.java	123	103	84%	0	0%	20	16%
TestUtils.java	104	89	86%	0	0%	15	14%
UniNotesApplication.java	14	9	64%	0	0%	5	36%
UniNotesApplicationTests.java	13	9	69%	0	0%	4	31%
User.java	78	64	82%	0	0%	14	18%
Utils.java	122	91	75%	2	2%	29	24%
Total:	2723	1918	70%	244	9%	561	21%

Extension ^	Count	Size SUM	Size MIN	Size MAX	Size AVG	Lines	Lines MIN	Lines MAX	Lines AVG	Lines CODE
cmd (CMD files)	1x	7kB	7kB	7kB	7kB	149	149	149	149	127
db (DB files)	1x	65kB	65kB	65kB	65kB	242	242	242	242	228
gitattributes (GITATTRIBUTES files)	1x	0kB	0kB	0kB	0kB	2	2	2	2	2
gitignore (GITIGNORE files)	1x	0kB	0kB	0kB	0kB	34	34	34	34	26
html (HTML files)	1x	0kB	0kB	0kB	0kB	23	23	23	23	17
java (Java classes)	27x	99kB	0kB	25kB	3kB	2723	8	852	100	1918
js (JS files)	38x	6.778kB	0kB	6.096kB	178kB	173158	1	159112	4556	162591
map (MAP files)	2x	30kB	0kB	29kB	15kB	2	1	1	1	2
pdf (PDF files)	4x	52kB	0kB	26kB	13kB	646	0	323	161	640
properties (Java properties files)	2x	1kB	0kB	0kB	0kB	22	3	18	11	6
sql (SQL files)	2x	2kB	0kB	1kB	1kB	124	62	62	62	86
sqlite (SQLITE files)	1x	0kB	0kB	0kB	0kB	0	0	0	0	0
ts (TS files)	15x	58kB	0kB	43kB	3kB	1626	1	1196	108	1178
tsx (TSX files)	1x	0kB	0kB	0kB	0kB	12	12	12	12	9
txt (Text files)	3x	0kB	0kB	0kB	0kB	16	0	16	5	7
xml (XML configuration file)	1x	3kB	3kB	3kB	3kB	116	116	116	116	108
Total:	101x	7.102kB	79kB	6.302kB	294kB	178895	654	161959	5582	

4.2 Complessità Ciclomatica

- La complessità ciclomatica media dei metodi è compresa tra 3 e 5, con alcuni casi di valori superiori compresi tra 6 e 7.

4.3 Copertura dei Test

- Test automatici: 20% delle funzionalità coperte.
- Test manuale: 100% delle interazioni coperte.

5. Conclusioni

Il progetto UniBg Notes rappresenta un sistema intuitivo e scalabile per la gestione e condivisione di appunti accademici. Le scelte progettuali garantiscono una buona

manutenibilità, mentre l'uso di tecnologie moderne permette una rapida implementazione.

Alcuni passi per il futuro potrebbero essere:

- Ottimizzazione delle query per la ricerca.
- Implementazione di un'API per estendere le funzionalità a terze parti.
- Aggiunta verifiche profili utenti.