Scuola universitaria professionale della Svizzera italiana
Dipartimento tecnologie innovative
Istituto sistemi informativi e networking

SUPSI

Sk(a)etchiamo la noia

Personal Sketching System

(14 slides)

M. Boutaleb, M. Rosselli, N. Gregori

Introduzione

- Sfogo creatività e mitigare noia
- Persistere sketch e poterli cercare
- Design e struttura backend e frontend
- Risultati e compromessi architetturali
- Conclusioni

Motivazione e contesto

- Esprimere creatività
- Esplicare qualcosa
- Rappresentazioni veloci
- Estendere progetto

Stato dell’arte

- Soluzioni in commercio già esistenti
- Distribuzioni differenti per ogni piattaforma

Approccio al problema

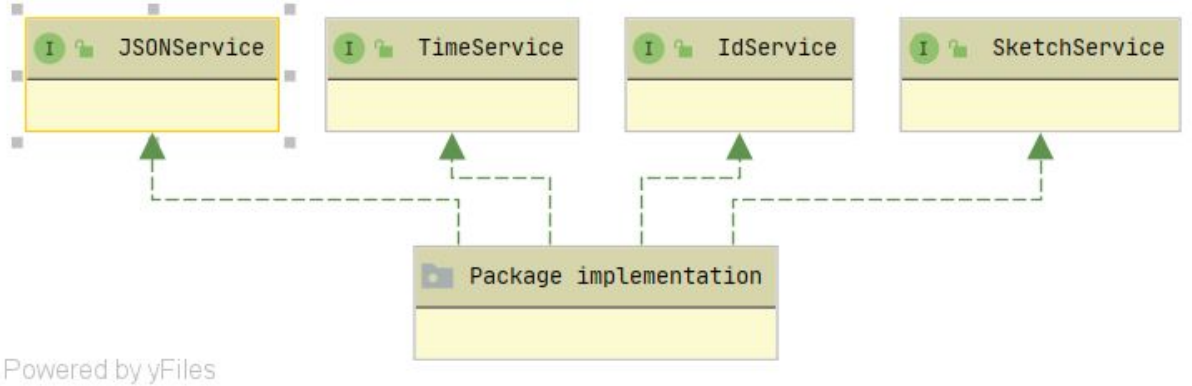
- Backend: persistenza dati e preferenze, implementazione servizi e API per frontend
- Frontend: client GUI

Software Patterns and Principles

- Software Design
- Singleton classes
- Coupling
- Cohesion
- Separation of concerns
- Interaction design
- Inheritance
- Inversion of Control
- Testing and Debugging

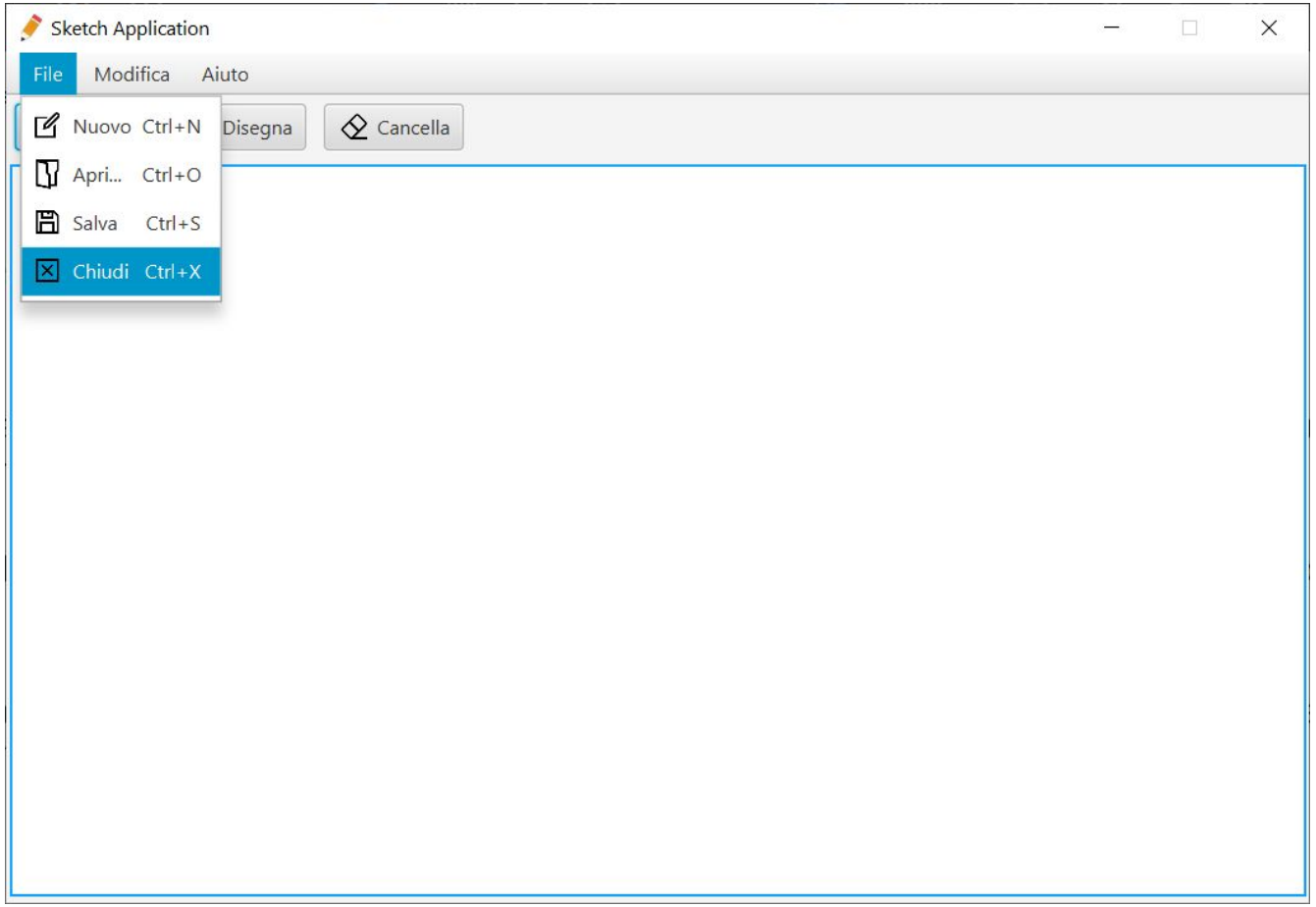
Coupling, Cohesion e Polimorfismo

- Coupling rilassato grazie al polimorfismo
- Maggior coesione

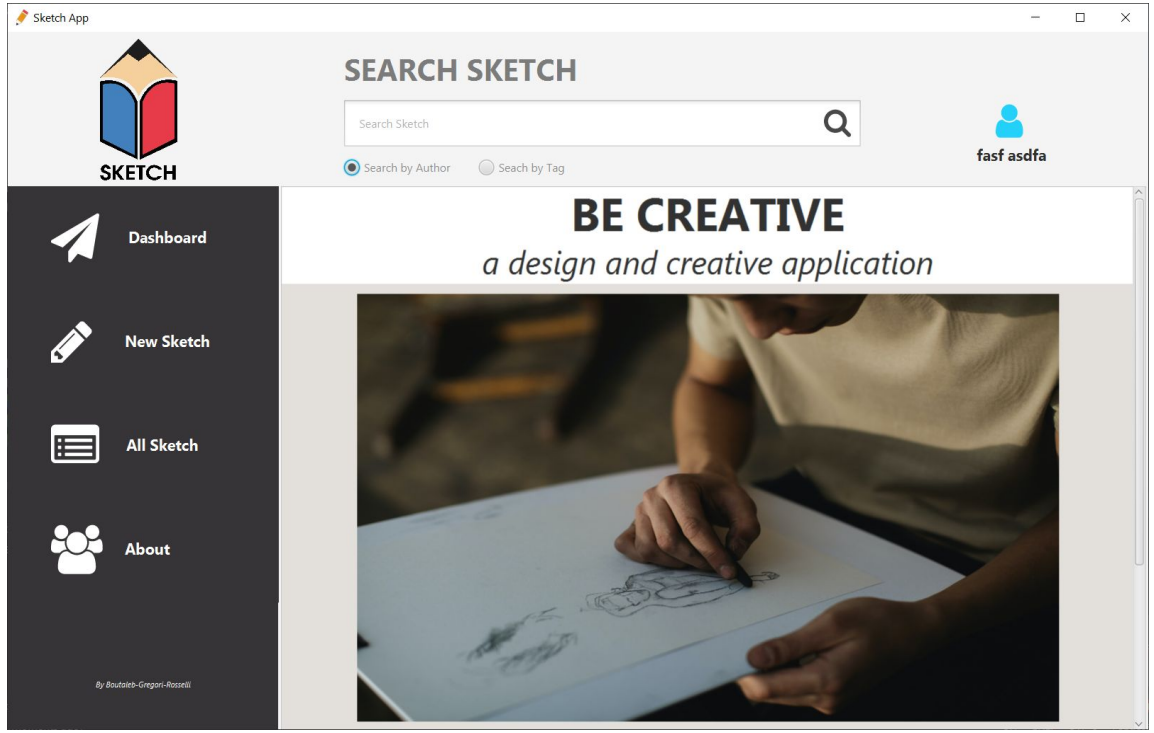


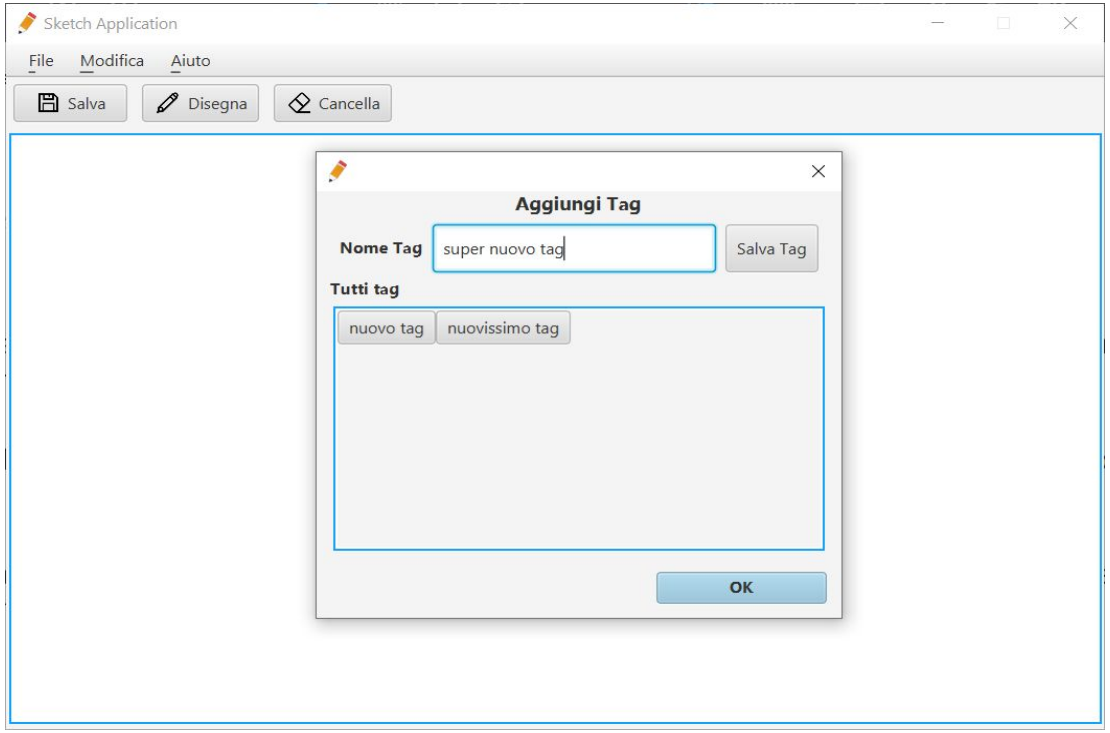
Interaction design

- GUI consistente
- Principio W.I.M.P
- Minima sorpresa



Old vs new version





Inversion of Control

```
public class GenericExtensionFilter implements FilenameFilter {
   private final String extension;

   public GenericExtensionFilter (final String extension) {
       this.extension = extension;
   }

   @Override
   public boolean accept(final File dir, final String name) {
       return name.endsWith(extension);
   }
}
```

```
final GenericExtensionFilter filterBySketch = new
GenericExtensionFilter( SKETCH_EXTENSION);
final GenericExtensionFilter filterByMetadata = new
GenericExtensionFilter( METADATA_EXTENSION);


final String
sktFiles[]=dir.list(filterBySketch);
```

Backend: struttura

Suddivisione in tre livelli:

- **Controller** interfacciamento con client GUI
- **Service** rappresentante i servizi di business
- **Repository** per persistenza dei dati e preferenze

Frontend: struttura

Suddivisione in tre livelli:

- **Bundles** controllo lingua e labels
- **Controller** con binding dinamico files .fxml
- **Utils** per costanti e metodi statici applicazione

Strumenti utilizzati

- Framework JavaFX
- Linguaggio FXML
- IntelliJ IDEA
- Scene Builder
- JSON
- Resource Bundle
- Maven
- Git

Considerazioni finali

- Progetto estendibile
- Custom GUI