

Project Plan

Federico Aguzzi - 1066519

Andrea Arici - 1060222

Fabio Crippa - 1068507

Iqbal Kpekpasi Fofana - 1068097

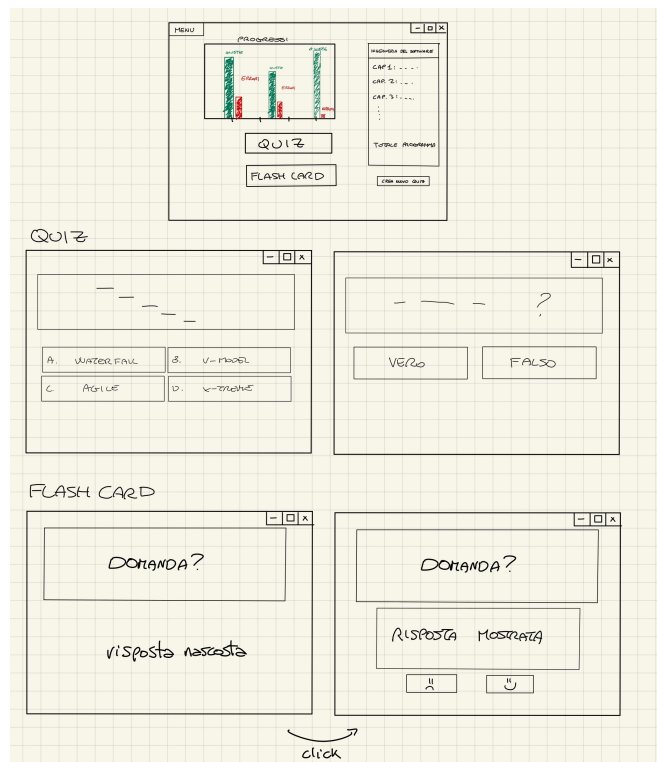
1 Introduzione

Il progetto si basa sulla realizzazione di una applicazione desktop che si occupa di testare le conoscenze dell'utente riguardo ingegneria del software.

L'idea è nata dall'utilizzo di *Kahoot* con il professore, e prende ispirazione anche dalle applicazioni del quiz per la patente di guida.

L'obiettivo dell'applicazione è di aiutare l'utente a prepararsi per gli esami, più nello specifico per l'esame di ingegneria del software, tramite quiz già preimpostati e *flashcard* che permettono di testare la propria preparazione sul programma.

Sarà possibile vedere l'andamento dei risultati tramite statistiche. Inoltre il programma permette di inserire nuovi corsi con i relativi quiz.



2 Modello di processo

- *Agile*, con sprint settimanali

3 Organizzazione del progetto

Aguzzi Federico si occupa di interfacce grafiche, gestione Github e diagrammi UML

Andrea Arici si occupa di interfacce grafiche e implementazione codice, diagrammi UML

Fabio Crippa si occupa di business logic, classi di unit test e diagrammi UML

Iqbal Kpekpasi Fofana si occupa del design grafico, scrivere le domande e le risposte dei quiz, creazione e funziononaltà delle classi quiz, creazione diagrammi UML

4 Standard, linee guida, procedure

Lo standard seguito è quello classico di Java:

- *CamelCase* per classi e interfacce
 - *esempio: NomeClasse*
- *DromedarCase* per variabili di funzione e metodi
 - *esempio: nomeVariabile, Object.nomeMetodo(args)*
- *CappedSnakeCase* per costanti
 - *esempio: NOME_COSTANTE*
- *m_dromedarCase* per i campi delle classi
 - *esempio: m_nomeVariabile*
- *Egyptian Brackets* per il codice
 - *Esempio:*

```
NomeFunzione(int x, int y) {  
  
}
```

5 Attività di gestione

Sprint settimanali, nel quale il team si riunisce per:

- *sprint retrospective*: discutere i progressi sul progetto
- *sprint planning*: le funzionalità da implementare

6 Rischi

- software non consegnato in tempo
- conoscenza non perfetta di alcune librerie di Java

7 Personale

- 4 persone per l'intera durata del progetto

8 Metodi e tecniche

- design del software: starUML
- UI del software: disegnata su carta
- ambiente di sviluppo: Eclipse
- linguaggio di programmazione: Java
- repository: Github
- testing: junit
- esecuzione: Windows, Mac OS
- coordinamento tra membri: Notion, Discord

9 Garanzia di qualità

- Sviluppo del software:
 - correttezza
 - affidabilità
 - efficienza
 - integrità
 - usabilità
- Revisione del software:
 - mantenibilità
 - testabilità
 - flessibilità

10 Pacchetti di lavoro

- diagrammi UML
- creazione di quiz
- creazione interfaccia grafica
- creazione delle classi

11 Risorse

Utilizzeremo 4 computer (3 PC, 1 Mac), uno per ogni membro del team

12 Budget e schedulazione

- costo del progetto: 0€
- tempo necessario: 5 settimane

13 Cambiamenti

Le proposte di cambiamento da parte di uno dei membri del team vengono discusse nelle riunioni settimanali insieme al resto del team per decidere la loro eventuale implementazione

14 Consegna del progetto

Consegna prevista per il 27/01/2022.