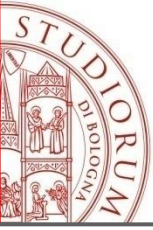


Progetto Ingegneria del Software A.A. 2022/23

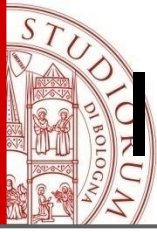
Marco Ferrati

PhD student, cycle XXXVIII, University of Bologna
marco.ferrati2@unibo.it



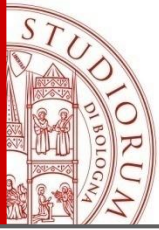
Progetto A.A. 2022/23

Progettare e realizzare un'applicazione web per lo scambio di carte da gioco



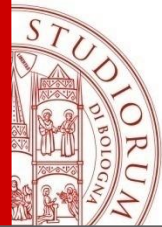
Informazioni di natura generale

- Progetto da realizzare in gruppi di 3/4 persone
- Documento riepilogativo



Principali funzionalità

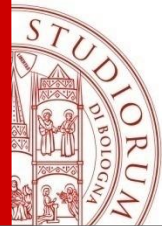
- **Ricerca e visualizzazione di carte**
- **Gestione delle carte possedute**
- **Gestione delle carte desiderate**
- **Gestione di deck**
- **Scambio**



Il catalogo delle carte (1/4)

L'applicazione supporta le carte di tre giochi:

- Magic: the gathering
- Pokémon
- Yu-Gi-Oh!



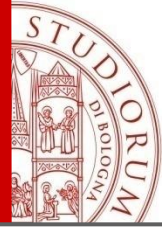
Il catalogo delle carte (2/4)

Il catalogo di carte dei tre giochi vi viene fornito in formato JSON

- 3 file JSON (uno per gioco)
- Il contenuto dei file deve essere letto e convertito in oggetti per poi essere utilizzati

In un file JSON si possono trovare:

- **Oggetti**
- **Coppie chiave-valore** (il valore può essere a sua volta un oggetto o una lista)
- **Liste**

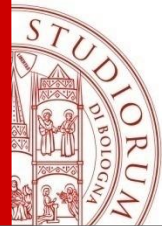


Il catalogo delle carte (3/4)

```
[
  {
    "key1": "string value",
    "key2": 1,
    "key3": {
      "key1": 2,
      "key2": "text"
    }
  },
  {
    "key1": "string value",
    "key2": 1,
    "key3": {
      "key1": 2,
      "key2": "text"
    }
  }
]
```

Parentesi {} => **oggetto**
Coppie chiave-valore separate da , (virgola)

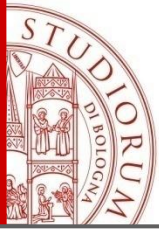
Parentesi [] => **lista**
elementi separati da , (virgola)



Il catalogo delle carte (4/4)

Le carte hanno caratteristiche diverse in base al gioco di appartenenza:

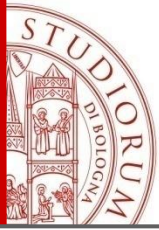
- Funzionalità compatibili con tutte le carte
- Visualizzazione delle carte differente



Browsing/Ricerca delle carte

Un visualizzatore dell'applicazione può sfogliare/fare una ricerca in ognuno dei tre cataloghi

Deve essere fornita la possibilità di applicare dei filtri alla ricerca



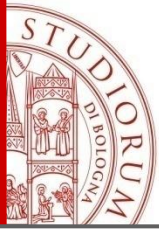
Dettaglio di una carta

Quando un visitatore apre la pagina di dettaglio di una carta vede

- Informazioni della carta
- Utenti che la possiedono
- Utenti che la desiderano

Se il visitatore ha effettuato il login avrà la possibilità di:

- Aggiungere la carta alla lista di carte che possiede
- Aggiungere la carta alla lista di carte che desidera

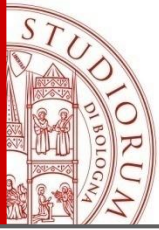


Utenti

L'applicazione prevede l'esistenza di utenti.

Sono quindi richieste le funzionalità essenziali di

- Registrazione
- Login
- Logout



Gestione delle carte

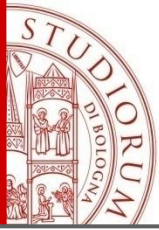
Ogni utente ha associati due gruppi di carte:

- **Carte possedute**
- **Carte desiderate**

Ogni carta posseduta/desiderata dall'utente ha due ulteriori proprietà:

- In quali **condizioni** si trova/desidera la carta (da rovinata a perfetta)
- Una **descrizione testuale** (dove l'utente può scrivere cosa vuole in cambio oppure che carte è disposto a offrire)

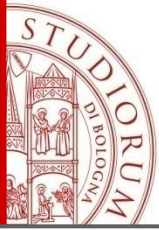
Le carte possedute devono poter essere ulteriormente organizzate in **deck**



Scambi (1/3)

Un utente può fare una richiesta di scambio ad un altro utente

Un utente può chiedere e offrire un qualsiasi numero di carte (numero > 0)

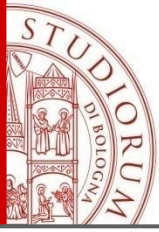


Scambi (2/3)

Quando un utente propone uno scambio, l'utente che riceve la proposta può

- Rifiutarla
- Accettarla

Nel caso la accetti il sistema deve effettuare lo scambio delle carte tra i due utenti



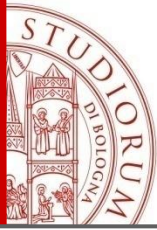
Scambi (3/3)

Un utente deve poter visualizzare la lista di scambi non ancora approvati/rifiutati

Reminder!

Si consiglia vivamente di rileggere più volte le specifiche di progetto.





Modalità di realizzazione

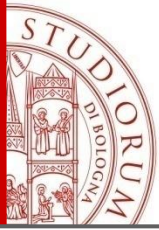
Inception

- Modellazione dei casi d'uso
- Modello di dominio
- Glossario e altri artefatti che il gruppo ritiene utili

Construction

SCRUM:

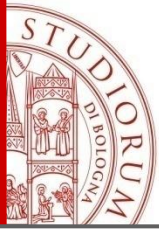
- Membri del team, scrum master e product owner (documentare nel Diario);
- Casi d'uso identificati nella fase di inception vengono posti nel backlog e poi raffinati nei diversi diversi scenari relativi al caso d'uso, i cosiddetti "use case slice"
- **Eventi standard:** Sprint planning, Daily scrum, Sprint review e Sprint retrospective;
- **Artefatti:** Product backlog, Sprint backlog, Burn down chart.



Artefatti (1/3)

Artefatti della fase di inception:

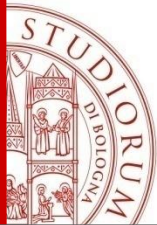
- Modello dei casi d'uso
- Modello di dominio
- Glossario



Artefatti (2/3)

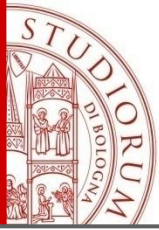
Diario del progetto: (da compilare in maniera schematica)

- Inizio e fine delle iterazione e degli sprint
- Chi ha interpretato quale ruolo durante lo sprint
- Stato dei backlog a inizio e fine sprint e il criterio (come viene popolato)



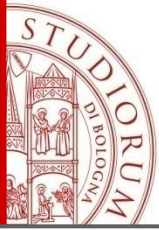
Artefatti (3/3)

- **Manuale utente:** guida accessibile all'utilizzo dell'applicazione per utenti inesperti
- **Manuale dello sviluppatore:** istruzioni su come ottenere il codice sorgente e come avviare il prodotto. Inoltre deve contenere una panoramica di come sono state realizzate le varie funzionalità



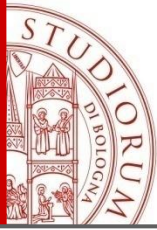
Requisiti tecnici (1/2)

- Linguaggi e librerie:
 - **Java**
 - **GWT** → pagine web
 - **MapDB** → persistenza dei dati
 - **Gson** → parsing file JSON
 - **JUnit** → unit testing
- **Maven**



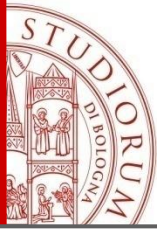
Requisiti tecnici (2/2)

- **Git** → versionamento
- **GitHub:**
 - Remote repository
 - Board SCRUM (**GitHub Projects** e **GitHub Issues**)
 - Continuous Integration (**GitHub Actions**)



Consegna e discussione (1/2)

- Le date delle discussione (comunicate almeno un mese prima) verranno fissate nei mesi di:
 - Febbraio
 - Aprile
 - Giugno
 - Luglio
 - Ottobre
 - Novembre
 - Dicembre
- La data ultima per la consegna è una settimana prima della data della discussione
- Eventuali date straordinarie possono essere aggiunte

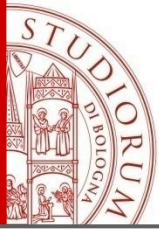


Consegna e discussione (2/2)

- Creare una release su GitHub contenente tutti i file
- Inviare una mail a marco.ferrati2@unibo.it (aggiungendo in cc tutti i componenti del gruppo)
- Invitare al vostro repository GitHub l'utente "jjocram"

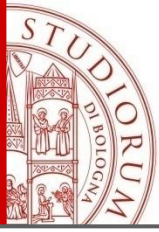
- Alla discussione, dopo aver discusso la documentazione, il gruppo dovrà fare una demo del software

- Le specifiche sono valide fino a Dicembre 2023. Dopo l'appello di Dicembre verranno cambiate le specifiche



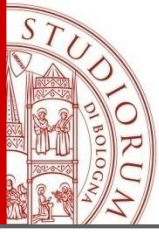
Criteri di valutazione

- La realizzazione delle specifiche funzionali
- L'organizzazione e la leggibilità del codice
- L'adozione di una metodologia e l'utilizzo appropriato degli strumenti di testing, versioning, bug tracking e code reviewing, project tracking
- L'adeguatezza della documentazione allegata al progetto e in particolare della descrizione del processo di sviluppo adottato
- La discussione del progetto.



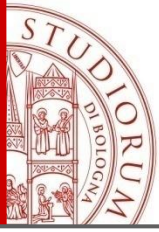
Errori ricorrenti (1/2)

- Modelli concettuali (diagramma di dominio) non rispecchiano il codice.
- Modelli concettuali consegnati (soprattutto diagramma di dominio) “non finali”:
 - Mancano le nomi di associazioni
 - Cardinalità associazioni
 - Attributi classi con tipo ed eventualmente cardinalità



Errori ricorrenti (2/2)

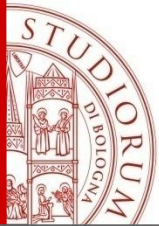
- Documenti richiesti dimenticati
- Tool per lo sviluppo/gestione non usati bene o non usati
- Confusione tra diagramma dei casi d'uso e diagramma di dominio.



Contact

Per qualsiasi dubbio o chiarimento riguardanti le specifiche del progetto non esitate a contattarmi all'indirizzo email:

marco.ferrati2@unibo.it



Grazie e buon lavoro