**Progetto di ingegneria del software, anno accademico 2022-2023**

**Membri del team**

* Lorenzo Magni, matricola 1073257
* Saif Bouchemal, matricola 1074800
* Marianna Romelli, matricola 1072382

**Struttura del progetto**

…..

**Organizzazione del progetto**

…. Facciamo che il lavoro viene deciso settimana con settimana? Facciamo una issue su GitHub per tracciare il lavoro svolto da ognuno???

**TESTO PROGETTO**

Gioco spiegato bene per esteso e anche alcune caratteristiche per quanto riguarda chi può giocare, chi può fare cosa. Cosa fornisce il sistema? Funzionalità???

**QUALITà DEL SOFTWARE**

Il team si è prefissato ???????? l’obiettivo di sviluppare un software che rispetti i parametri e gli attributi di qualità definiti da McCall-Richards-Walters nel documento da loro redatto nel 1977, di seguito elencati e suddivisi in categorie come indicato dagli autori.

**Parametri riguardanti l’operatività del software**

* Correttezza - Il prodotto software da noi realizzato soddisfa i requisiti e le specifiche indicate dal cliente. Le funzionalità da noi non sviluppate ma solo progettate saranno portate avanti dal team di sviluppo che Mario assumerà dopo che avremo consegnato il prototipo.
* Affidabilità - Il software è affidabile in quanto revisionato a livello di team e sottoposto a una lunga fase di testing prima del rilascio.Usabilità - Il prodotto è semplice da utilizzare, infatti non sono richieste particolari abilità per poterne usufruire. L’uso dei servizi è facilitato su tutti i tipi di piattaforma grazie all’utilizzo della libreria Bootstrap che permette di avere componenti grafici semplici e facili da utilizzare.  
  I requisiti “base” includono:
  + Avere una connessione internet.
  + Avere un indirizzo email di proprietà per il processo di registrazione.
  + Sapere come effettuare un pagamento online.
* **Parametri riguardanti la revisione del software**
* Manutenibilità - Individuazione di errori ????
* Testabilità - possibile fare la fase di test sia manuali che automatici??
* Flessibilità - scelta di separare i lati front e back-end???
* **Parametri riguardanti la transizione verso un nuovo ambiente**
* Portabilità - È possibile utilizzare i servizi offerti dal nostro prodotto da qualsiasi dispositivo ????
* Riusabilità - Per la parte grafica usiamo qualcosa di già fatto??? Parte di codice preso da lavori già effettuati????
* Interoperabilità - Nel futuro il nostro prodotto software potrà essere integrato con altri servizi????
* Efficienza - Le risorse utilizzate dal prodotto sono limitate perché serve internet o un browser????
* Servono password o iscrizioni????
* Servono verifiche per creare account??? Sempre che ne facciamo
* Crittografia sulla chat (se riusciamo)
* **ARCHITETTURA SOFTWARE**
* L’architettura del nostro sistema è un’architettura client server
* L’architettura è sicura???? (non ha necessità di privilegi di admin ed è chiusa nella sandbox java?)
* La view è stata realizzata utilizzando OpenGL (LWGJL)
* SOFTWARE LIFE CYCLE
* Per il processo di sviluppo il team ha scelto un approccio di tipo Agile poiché meglio si adatta alla nostra metodologia di lavoro:
* Consideriamo importante il team, le abilità e le interazioni tra di noi. I lavori vengono assegnati in base alle capacità che ognuno possiede. Se uno della squadra si accorge che il compito va oltre le sue competenze, avvisa gli altri e se ne discute insieme.
* Nel team non c’è una struttura di tipo gerarchico: ci consideriamo tutti allo stesso livello. Questo ci porta ad essere più compatti e motivati.
* Consideriamo più importante un prodotto software funzionante che spendere troppo tempo sui documenti: a sviluppo iniziato, se ci accorgiamo che ci sono punti da implementare che non avevamo in precedenza considerato, li andiamo ad aggiungere poi alla documentazione.
* Utilizziamo la tecnica del pair programming: gli sviluppatori hanno l’opportunità di lavorare in coppia: uno dei due scrive il codice mentre l’altro funge da supervisore e revisore; i due possono cambiarsi di ruolo.
* Durante lo sviluppo del software utilizziamo la tecnica del timeboxing: suddividiamo lo sviluppo del codice in intervalli temporali entro i quali dobbiamo sviluppare determinate funzionalità.
* Lavoriamo per un pubblico generico che possa essere interessato al software. Ci mettiamo nei panni del pubblico per capire quali possano essere i requsiti e le preferenze. Una volta avviato il progetto si prendono in carico eventuali feedback da parte della community che utilizza il software.
* Come approccio di progettazione software per lo sviluppo del sistema ci siamo attenuti alla model-driven architecture (MDA), o architettura guidata dal modello.
* DISEGNO SOFTWARE
* Ci basiamo sul pattern Model-view-controller (MVC), allo scopo di suddividere il codice in parti che abbiano funzionalità distinte tra loro
* REQUISITI
* Uno dei passi più importanti prima di scrivere software è quello di analizzare i requisiti, ovvero definire con l’ipotetico cliente sia quelli funzionali (funzionalità del sistema) che quelli non funzionali (tempi di risposta, quantità di dati da immagazzinare ecc. ).
* Possiamo fare le funzionalità suddividendole in must have, should have e won’t have.
* PROJECT PLAN
* Immaginiamo che il testo del problema, contenente i requisiti scritti in linguaggio naturale, sia una richiesta da parte di un ipotetico gruppo di clienti.
* Cosa ci chiede questo cliente??? Perché ha bisogno del nostro gioco??? Cosa ne deve fare poi???
* La nostra azienda è una società di sviluppo software
* Persone coinvolte nel progetto: i clienti ipotetici, e il team di progettisti/sviluppatori, formato da noi?????? Di seguito saranno descritti i loro ruoli e le loro mansioni. (lo decidiamo alla fine)
* L'intera documentazione riguardante il progetto sarà riportata e tracciata su GitHub. In particolare, il team si incontra settimanalmente per analizzare il lavoro svolto dall'ultimo meet e pianificare le attività future, tracciando i task come issue su GitHub in modo che ogni persona del team sia al corrente di cosa stiano facendo gli altri (può andare????).
* PERSONE COINVOLTE
* Metodo agile???? …...