

1. Introducere

Acest document a fost creat pentru a realiza o viziune realistica a specificatiilor produsului care consta in implementarea unui joc multiplayer, similar cu celebrul joc Bomberman.

1.1 Descriere generala a produsului

Produsul specificat in acest document este un joc destinat persoanelor de orice varsta, care vor sa interactioneze online cu prietenii sau sa scape de momentele de monotonie. Jocul va avea facilitatile de baza a jocului Bomberman insa orientat doar pe multiplayer online si chat.

1.2 Privire de ansamblu asupra functionalitatii produsului

Produsul trebuie sa fie capabil sa ofere urmatoarele functii:

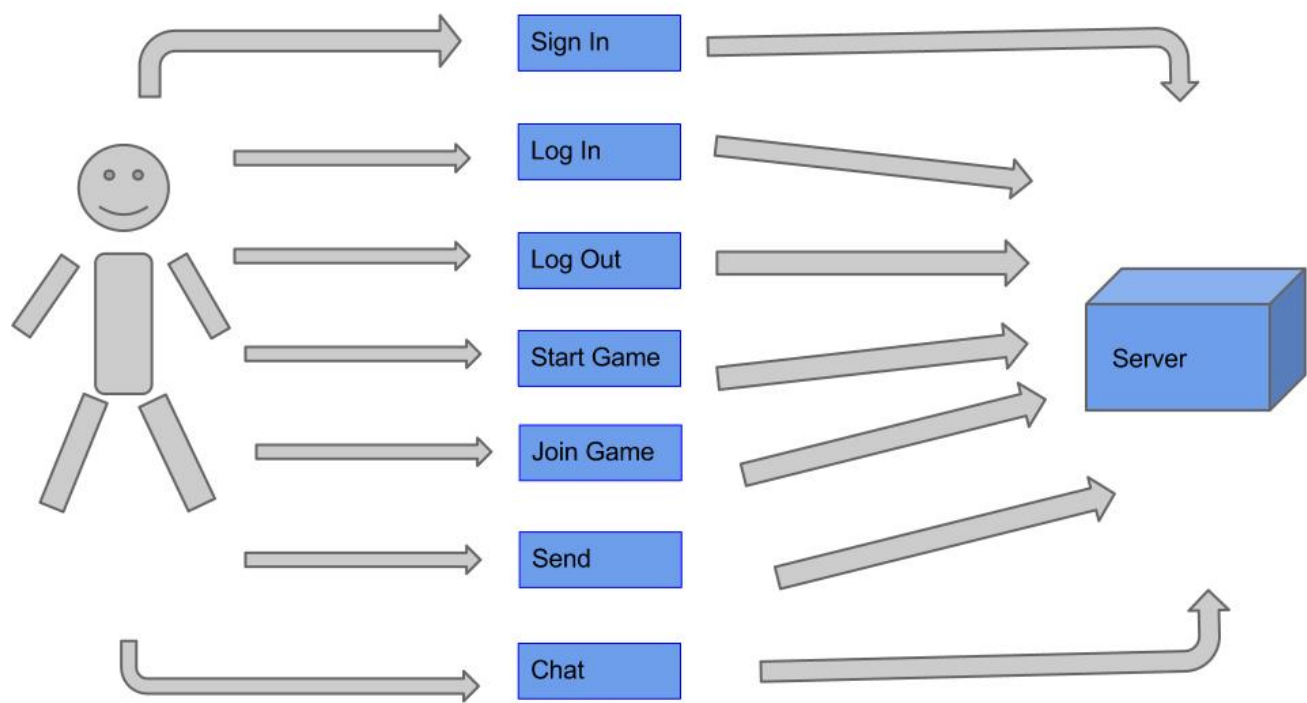
- posibilitatea de creare cont
- posibilitatea de autentificare in contul propriu
- posibilitatea de a concura contra altor concurenti
- posibilitatea de a interactiona cu ceilalti utilizatori prin intermediul unui chat
- posibilitatea afisarii unor statistici legate de contul fiecarui jucator

1.3 Software context

Aplicatia va fi dezvoltata open-source si poate fi utilizata gratuit de catre orice persoana. De asemenea orice developer isi poate aduce aportul la dezvoltare intrucat repository-ul va fi facut public pe GitHub dupa terminarea proiectului.

2. Cerinte functionale

2.1 Aplicatia client



2.1.1 Log In

Pentru a putea utiliza aplicatia, orice jucator trebuie intai sa se poata autentifica (login) folosind metoda clasica de autentificare, introducand user name si parola.

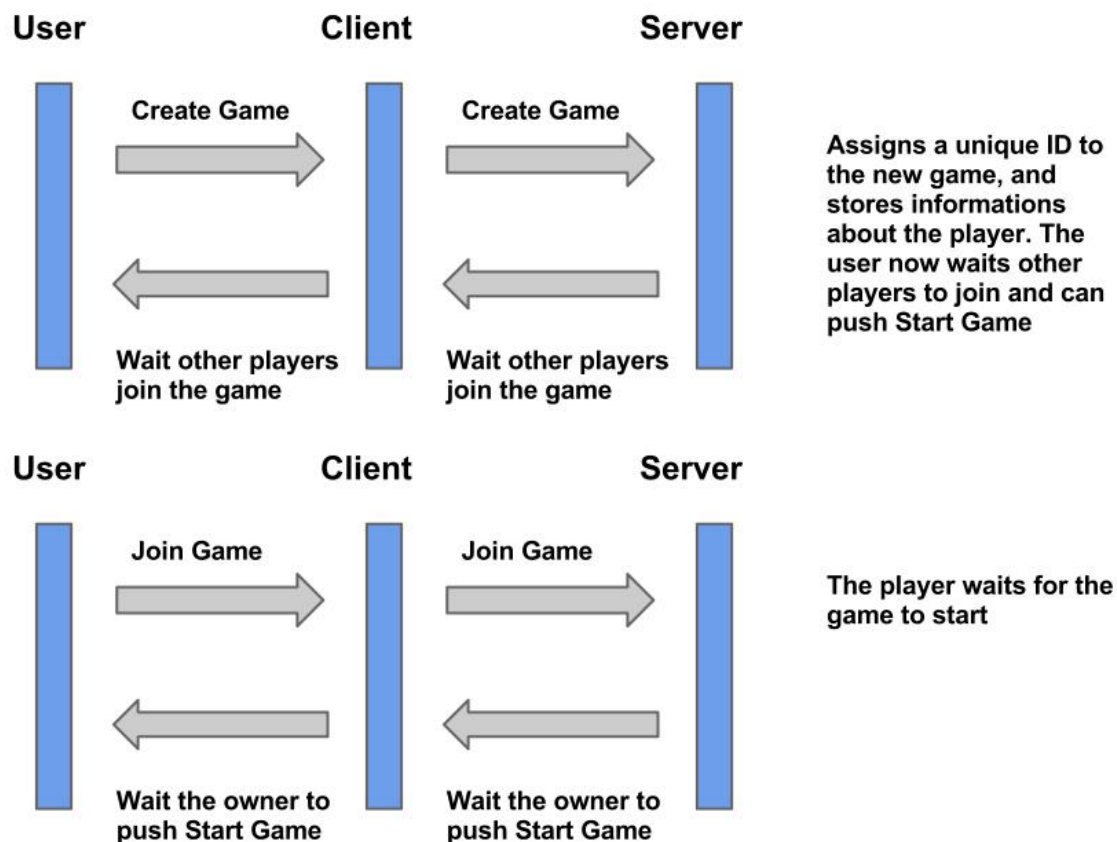
Pe parcursul autentificarii, aplicatia client va interoga serverul si va cere drepturile de acces asociate userului care face cererea. Apoi jucatorul are acces la restul facilitatilor oferite, fara nici o restrictie.

2.1.2 Log Out

Actiunea de deautentificare apare in situatia in care jucatorul initiaza actiunea (apasa butonul de log out) sau jucatorul inchide aplicatia client caz in care deautentificarea se realizeaza automat.

2.1.3 Sign In

Jucatorul care doreste sa utilizeze aplicatia, insa nu are pana in acest moment cont necesar autentificarii, isi poate crea un cont cu user name si parola. Pe parcursul inscrierii aplicatia client va trimite serverului numele si parola pentru stocare, serverul fiind capabil sa recunoasca din acest moment id-ul jucatorului si oferindu-i acestuia dreptul la joc.



2.1.4 Create Game

Dupa ce actiunea de autentificare a fost realizata, si utilizatorul este logat, acesta poate selecta optiunea de a incepe un joc nou, acest joc fiind memorat in baza de date a serverului cu un identificator unic. Apoi, clientul care a creat jocul va astepta pana cand alti utilizatori intra in jocul sau. Cu toate acestea, jocul in sine va fi rulat pe server, nu pe clientul care a creat jocul.

2.1.5 Join Game

Utilizatorii logati, daca nu doresc sa creeze un joc, vor avea o lista cu jocuri deja create de alti utilizatori pentru a putea intra in acesta. De exemplu, daca un utilizator doreste sa joace cu prietenii sai, va alege aceasta optiune.

2.1.6 Start Game

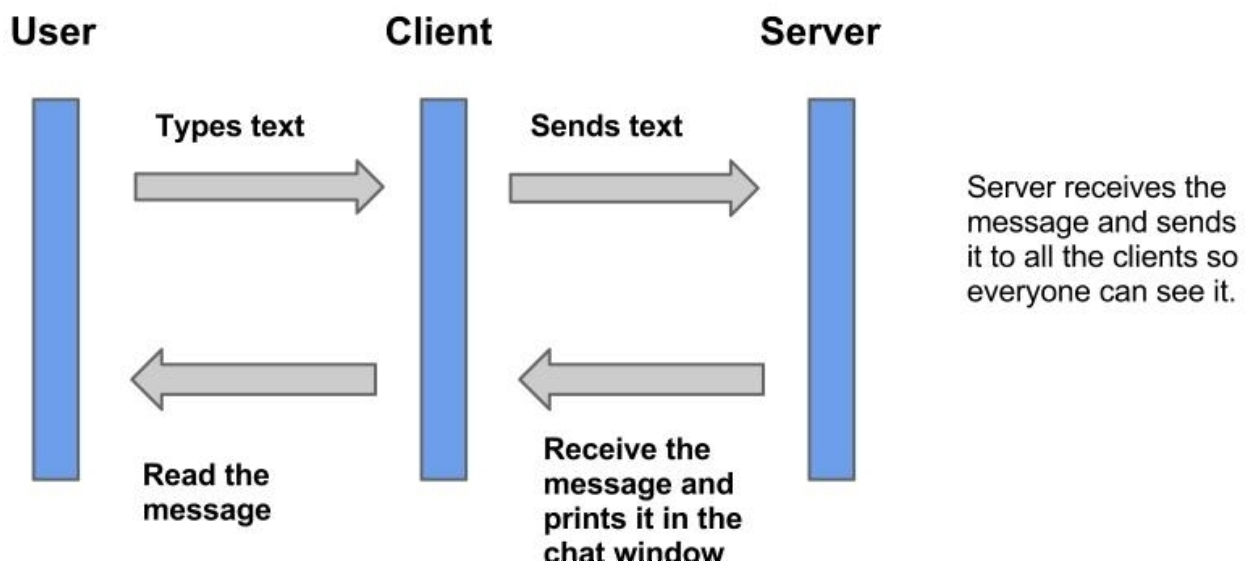
Dupa ce au intrat suficienti utilizatori in jocul creat, utilizatorul host poate selecta sa inceapa jocul. Toti utilizatorii vor incepe jocul iar dupa ce toate datele au fost incarcate in memorie, jocul va incepe pentru toti in acelasi timp.

2.1.7 Send

Pe parcursul jocului, fiecare utilizator va trimite mesaje serverului cu pozitia sa si actiunea pe care doreste sa o efectueze. Daca acesta primeste rezultat afirmativ, atunci va efectua actiunea dorita.

2.1.8 Chat

In afara de functionalitatile standard ale jocului, fiecare utilizator va avea la dispozitie un ecran de chat unde va putea comunica cu ceilalti utilizatori din jocul sau.



2.2 Aplicatia server

Aplicatia server va fi implementata in Java pentru ca ofera usurinta in lucrul cu stiva TCP/IP si cu obiectele de tip thread fiind limbajul cel mai potrivit.

Scopul serverului consta in asigurarea comunicarii cu aplicatiile client sau intre acestea.

Cazurile in care aplicatiile client trebuie sa comunice cu serverul sunt cele in care se realizeaza operatiile de autentificare, deautentificare, sau sign in. Cazurile cand clientii comunica intre ei prin server sunt de fapt momentele cand acestia sunt angrenati in joc si necesita schimb de informatii intre acestia. Serverul va astepta cereri in permanenta si va crea threaduri noi pentru a putea tria cererile si a le putea onora.

2.2.1 Log In

Serverul primeste cerere de autentificare de la un client pentru a primi drepturi de acces la joc. Serverul face o cautare in fisierul XML ce contine utilizatorii, in cazul in care utilizatorul va fi gasit, acesta va primi drepturi de acces, altfel serverul va notifica aplicatia client de lipsa utilizatorului. In primul caz, utilizatorul va fi inserat in lista de playeri online necesara clientilor pentru a fi informati ca pot interactiona cu noul venit.

2.2.2 Log Out

Serverul primeste cerere de deautentificare de la un client deja autentificat. Serverul va trimite confirmare clientului, si va sterge utilizatorul din lista de playeri online. Serverul va trimite notificare tuturor clientilor de modificarea din lista de playeri online (probabil nu va fi trimisa imediat clientilor care nu se afla in mijlocul unui joc - similar Facebook- dar va trebui trimisa imediat clientilor care sa afla in joc).

2.2.3 Sign In

Serverul primeste cerere de inscriere a unui utilizator, primind astfel username-ul si parola. Serverul va cauta in fisierul XML al utilizatorilor daca mai exista o astfel de persoana cu aceleasi date(username si parola). In caz afirmativ, serverul va trimite o notificare clientului avertizandu-l ca mai exista o persoana cu aceleasi date. In cazul in care doar username-ul e identic inregistrarea va fi facuta cu succes (probabil se va adauga un numar la username-ul persoanei inconjurat de paranteze rotunde pentru a putea distinge intre cei 2 useri insa acest lucru nu are o deosebita importanta intrucat nu se efectueaza operatii importante cum ar fi transferuri bancare sau mesaje). La fel se va intampla si pentru cazul cand nicio intrare nu are numele identic cu cel care efectueaza cererea.

2.2.4 Arhitectura sistemului

Produsul nostru ofera posibilitatea utilizatorilor de a juca un joc multi-player prin intermediul unei aplicatii client ce se afla pe PC-ul personal(optional implementare in browser daca mai este timp). Astfel produsul va consta din 2 aplicatii: aplicatia client si aplicatia server. Cele 2 vor comunica permanent pentru a actualiza datele intre clientii angrenati in joc, pentru crearea de conturi sau pentru autentificare/deautentificare. Dat fiind ca numarul de persoane inregistrate va fi destul de mic (de ordinul zecilor), nu va exista efectiv o baza de date, cel mai probabil va exista un fisier XML ce va contine informatiile asociate utilizatorilor.

Cel mai probabil, aplicatia server va rula pe un sistem diferit de cel al clientilor, oferind astfel usurinta in utilizare (evitand astfel necesitatea comunicarii adresei IP a serverului creat de unul din utilizatori, etc.)

2.2.5 Aplicatia server

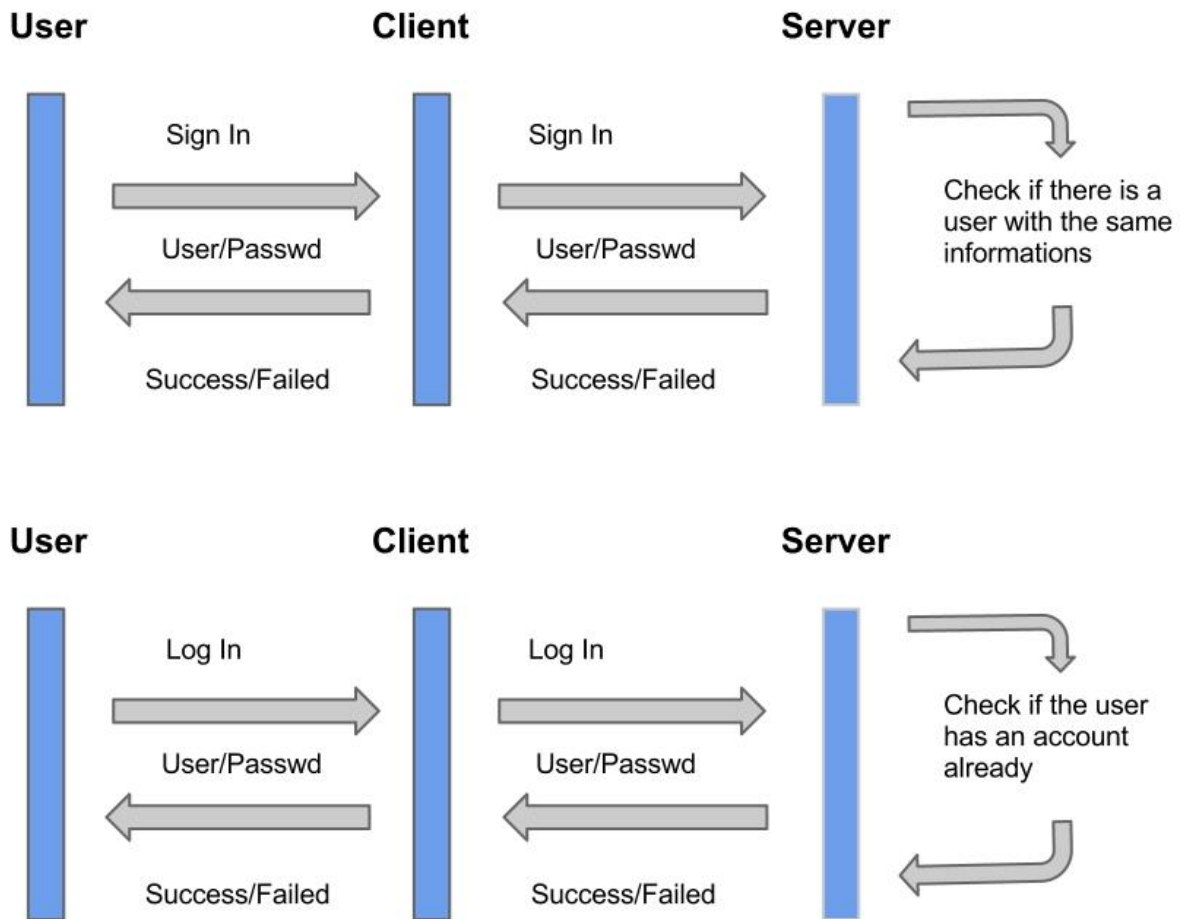
Serverul va astepta noi conexiuni pe un port dedicat, si va crea noi threaduri ce se vor ocupa de un anumit client. Astfel fiecare thread asociat unui client va astepta cereri de la utilizator pentru a fi rezolvate asignandu-se astfel cate un thread pentru fiecare cerere pentru a fi rezolvata.

Desigur serverul va necesita si un sistem de locking pentru sincronizare a threadurilor si pentru a proteja datele shareuite intre threaduri.

De asemenea serverul va retine starea pentru fiecare joc aflat in derulare, pe care o numim starea oficiala a jocului. Serverul va primi in permanenta actualizari pentru pozitia jucatorilor si va trimite notificari celorlalti jucatori despre starea actualizata a jocului.

Sistemul pe care ruleaza serverul va necesita performante de calcul destul de ridicate, pentru a asigura derularea fluenta a jocului.

Vom folosi probabil 2 tipuri de conexiune, TCP pentru cereri de autentificare, deautentificare, sign in, si UDP pentru actualizarea jocului, intrucat aceasta conexiune nu trebuie sa fie neaparat sigura (nu conteaza daca se pierde o parte din pachete, oricum frecventa de transmitere va fi destul de mare) ci rapida.



3. Cerinte non-functionale

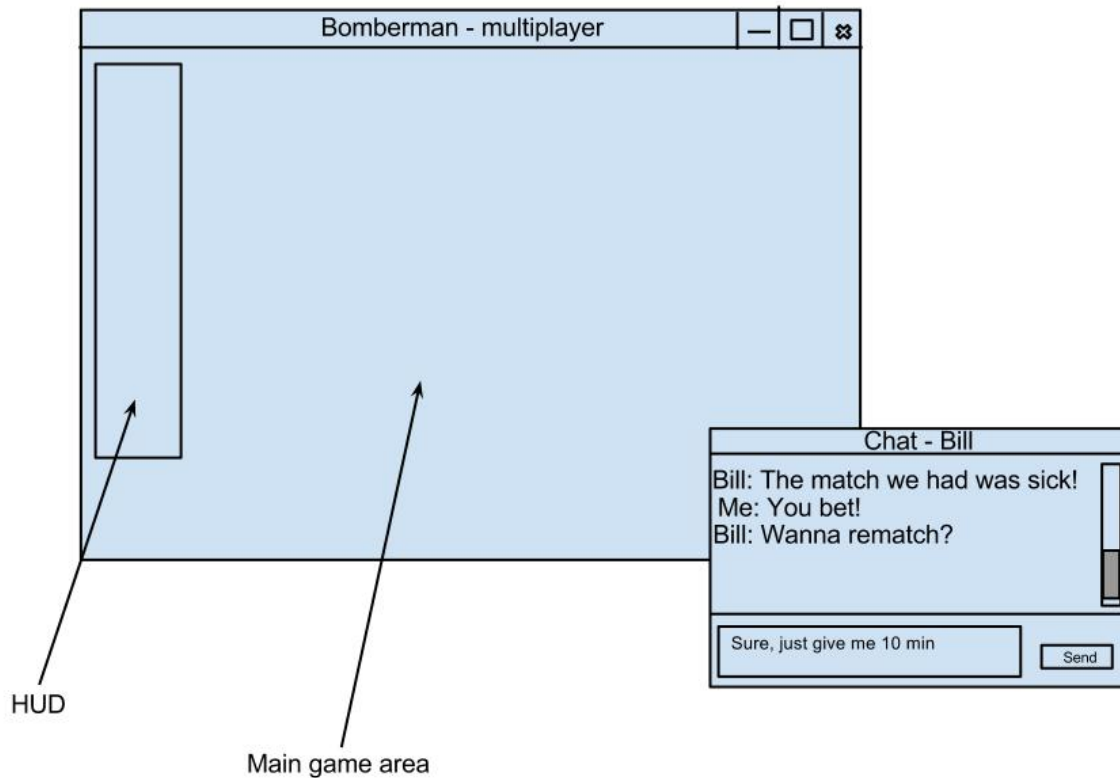
3.1 Cerinte interfata grafica

Interfata grafica va fi realizata in Python utilizand biblioteca Pygame pentru usurinta in implementare. Intrucat jocul va fi 2D procesarile facute nu vor fi foarte intense, acest fapt transformandu-se intr-un joc cu un timp de raspuns bun si cu un numar mare de cadre pe secunda desenate.

Jocul va avea 2 ferestre principale, fereastra jocului, in care se fac toate desenerile si fereastra de chat care va fi complet separata. Intrucat Bomberman este un joc destul de intens nu vrem ca chatul sa distraga jucatorii de la actiune.

Pe langa jocul care se desfasoara in interior, fereastra de joc va mai contine si meniuri activabile de un buton in care utilizator va putea configura diverse setari ca sunet, rezolutie sau brightness.

Un prototip al interfetei grafice:



3.2 Cerinte de performanta

Aplicatia client trebuie sa fie aiba un timp de raspuns foarte mic, sa fie capabil de reactie imediata mai ales avand in vedere si tipul jocului ce se bazeaza pe miscare (care trebuie sa fie aceeasi la fiecare client).

Pentru autentificare/deautentificare si inscriere aplicatia poate avea delay (nu mai mult de o secunda), pentru ca nu executa niste operatii importante, insa timpul de raspuns la modificarea pozitiei unui jucator de exemplu trebuie sa fie foarte mic (sub 0,5 sec) pentru a asigura cursivitate jocului(probabil in cazul in care jucatorul are o conexiune slaba la internet, aplicatia se va comporta ca orice joc de acest tip si anume freeze in starile pentru care se asteapta informatii de la server).

Sistemul ce va rula serverul nu va necesita un spatiu de stocare mare, deoarece aplicatia nu va accesa o baza de date, ci va interactiona cu un fisier XML datorita numarului restrans de utilizatori(pentru inceput), insa va necesita o conexiune foarte sigura si de mare viteza la

internet, pentru a rezolva prompt cererile utilizatorilor.

3.4 Securitate

Pentru securitatea utilizatorilor, parolele ar trebui alese folosind combinatii de litere, cifre si caractere. Datele transmise intre server si client vor respecta un protocol stabilit, insa nu va exista o criptare intrucat utilizatorii nu ofera date personale importante (optional - daca timpul va permite vom implementa si o astfel de criptare a datelor pentru a evita un atac de tip man-in-the-middle).

3.5 Administrarea serverului si mentenanta

Sistemul pe care ruleaza serverul nu are nevoie de cerinte speciale de mentenanta spre deosebire de un calculator obisnuit.

3.6 Portabilitate

Aplicatia va fi functionala pe sisteme ce ruleaza Windows, Linux, MacOS deoarece va fi scrisa in Python. Asa este.