Aplicatia este construita in jurul unei arhitecturi de tip-ul MVC (Model-View-Controller).In momentul in care user-ul intra in aplicatie, va fi randat View-ul de start in care sunt randate doua link-uri,unul este folosit pentru a redirectiona user-ul aplicatiei catre pagina de logare,celalalt link redirectionand user-ul catre pagina de inregistrare pentru crearea unui cont.Cele doua link-uri sunt implementate utilizand functia ActionLink,aceasta fiind o functie tip extensie asociata obiectului HtmlHelper, avand rolul de a redirectiona user ul catre un controller si o metoda asociata controllerului.Link-ul de LogIn redirectioneaza user ul catre Controller ul Home,metoda de LogIn,aceasta fiind o metoda care randeaza View ul de LogIn. View-ul de Login contine o forma cu doua input uri (username si password) si un buton.Randarea input urilor este realizata cu ajutor ul metodelor specifice ASP.NET,fiind extensii de metode care sunt asociate obiectului Html.De exemplu metoda LabelFor afiseaza label ul inputului,metoda EditorFor randeaza inputul.Formular ul de HTML este trimis,dupa introducerea credentialelor si apasarea butonului de submit,cu ajutorul metodei BeginForm care trimite datele catre metoda de Authenticate in controler ul Home.   
 Formular ul de logare este trimis in metoda Authenticate,care are ca si scop inceperea unei sesiuni si redirectionarea user ului catre contul asociat si view ul corespunzator (Individual sau Company).In aceasta metoda este apelata si metoda Validate din obiectul LoginValidation care verifica in baza de date daca exista un account(fie individual fie company) asociat acelor credentiale.In cazul in care exista,variabilei ValidCredentials ii este setata valoarea true,valoarea este intoarsa in metoda de Authenticate,aceasta facand redirectionarea user ului.In cazul in care nu a fost gasit un user cu aceste credentiale variabilei ValidCredentials ii este setata valoarea false,se face din now redirectionarea catre pagina de Login,fiind afisat mesaj ul de eroare ”Username or password invalid !”.Mesajul de eroare este stocat in variabila TempData si asociat modelului cu ajutorul metodei ModelState.AddError.In front end este afista prin intermediul metodei ValidationMessage.  
 View-ul de SignUp contine un drop-down care este implementat utilizand functia EnumDropDownList,utilizatorul putand alege inregistrarea fie ca o persoana fizica,fie ca o persoana juridica.In momentul in care un user isi creeaza un cont este apelata metoda InsertLogIn care face un insert in baza de date in entitatea Account si in entitatea de Individual sau Company. Pentru accesarea si maparea bazei de date este utilizat EntityFramework 6.Acesta este utilizat impreuna cu Linq,extensiile Linq avand ca si rol sortarea si gestionarea seturilor de date intoarse de metodele asociate EntityFramework.Operatiile de inserare a user ilor sunt realizate utilizand o tranzactie,in acest moment asociindu-se si un guid unic account ului si data in care a fost creat cont ul.  
 In cazul unei companii,dupa inregistrarea contului,user-ul este redirectionat catre actiunea SesionCompany din controller ul User.Aceasta metoda primeste ca si parametru un obiect de tipul LogInModel care contine proprietatea User,numele user ului fiind key word ul unic dupa care este realizat un querry in baza de date,fiind aduse in interfata datele asociate companiei,inclusiv datele financiare.Datele personale ale companiei cat si datele financiare sunt trimise in view ul SesionCompany prin intermediul proprietatii ViewBag.  
 View-ul SesionCompany contine diferite elemente asociate userului de tip company,ca de exemplu:tabel in care sunt afisate datele financiare din ultimul an al companiei,pe baza carora sunt calculati indicatori financiari precum Capitalizarea,Dividend-ul,EarningPerShare(EPS),PriceBookValue(PBV),grafic cu evolutia pretului actiunii,buton de inserare a fisierului CSV pentru a incarca datele financiare,formular pentru introducerea partii din capitalul social care va fi transformata in actiuni si numarul de actiuni care va fi distribuit in piata,date generale despre listarea companiei: data listarii,pretul actiunii,numele companiei,simbolul etc.Pentru a se lista,user-ul trebuie sa introduca datele financiare din ultimul an,prin intermediul unui fiserul de tip CSV uploadat in front end.In moment-ul in care user ul uploadeaza fiserul,se apeleaza metoda UploadCompanyDetails din controller-ul User,metoda care primeste un obiect de tipul HttpPostedFileBase.Aceasta metoda salveaza fisier-ul in locatia ~/App\_Data/uploads si apeleaza metoda Read din obiectul FileProcessing.Aceasta metoda parseaza tot fisierul CSV,mapeaza datele pe proprietatile obiectului CompanyFinancialDetails si realizeaza o lista de obiecte de tipul CompanyFinancialDetails,fiecare obiect reprezentand o linie in fisier ul CSV,linia reprezentand din punct de vedere bussiness un an sau un trimestru calendaristic.Dupa parsarea fisierului si realizarea unei liste cu obiecte se trimite lista in metoda InsertDetails impreuna cu username-ul user-ului.Metoda deschide o conexiune la baza de date prin intermediul EntityFramework si apeleaza metoda DocumentInsert din obiectul SqlScripts care insereaza datele financiare incarcate de user in baza de date.Dupa inserarea datelor,in cazul in care nu apare nici o exceptie,user ul este redirectat in acelasi view,SesionCompany,in care va putea observa datele financiare incarcate prin intermediul fisier ului csv intr-un tabel.  
 Dupa incarcarea datelor in fisier-ul CSV,user-ul are posibilitatea sa listeze compania asociata prin intermediul unui formular in care specifica procent-ul din capitalul propriu pe care vrea sa il listeze si numarul de actiuni in care sa imparta acest capital.Formular-ul pe care il completeaza user-ul este un view partial numit CompanyListing,acesta continand un buton care la apasare apeleaza metoda ListCompany si careia ii trimite un parametru de tip Listing care contine 3 proprietati:cui-ul companiei,procent-ul de listare din capitalul propriu si numarul de actiuni in care se doreste impartirea acestui capital.Aceasta apeleaza metoda UpdateCompanyforListing care primeste ca si parametru obiectul Listing si cui-ul companiei.Aceasta metoda are ca scop calcularea indicatorilor financiari,inserarea acestora in baza de date prin intermediul functiei InsertCompanyIndicators,updatarea datei la care a inceput tranzactionarea prin intermediul functiei UpdateCompanyBeginTransactionDate si inserarea in baza de date in tabela Ask a unei noi inregistrari care reprezinta listarea efectiva pe piata de capital.In tabela ask vor fi introduse numarul de actiuni listate,numele companiei,data listarii,pretul actiunii si un parametru de tip bool numit IsIPO care specifica daca inregistrarea in entitatea ask,practic actiunea care este vanduta,este vanduta direct de companie (IsIPO este true) sau de o persoana fizica (IsIPO este false).Pretul actiunii este calculat dupa formula ,unde :

* Cp : capitalul propriu;
* PL : procentul de listare;
* Na : numarul de actiuni care vor fi listate pe piata

Indicatorii financiari calculati :

* E/PS – earning per share :
* P/BV – price per book value :
* C – capitalizarea :
* P/ER – price earning ration :

In momentul in care compania se listeaza,view-ul de listare este ascuns si este afisat in front end in dreptul field ului Stare:„Listata”.   
 In cazul persoanelor fizice,dupa ce se logheaza prin intermediul view-ului de LogIn,li se va afisa o pagina in care este prezentat portofoliul acestora sub forma unui grafic,companiile listate pe platforma sub forma unui tabel,un formular in care pot modifica ordinele deschise in aplicatie,si un view in care pot vedea detaliile contului precum debit,numar de actiuni achizitionate si profitul portofoliului.  
 Toate companiile care sunt listate in aceasta platforma sunt prezentate intr-un tabel in front end.Datele tabelului sunt aduse in front end prin intermediul functiei SesionIndividual din controller ul User.In interiorul acestei functii este apelata functia FindListedCompanies din obiectul Context care deschide o conexiune cu baza de date si returneaza un obiect de tipul IEnumerable<Company>,fiecare obiect din aceasta Lista fiind o companie listata.In cadrul acestui tabel sunt afisate date precum numele companiei,pretul de piata,numarul total de actiuni,numarul de actiuni care au ramas in piata la oferta initiala de IPO.Langa fiecare inregistrare din tabel se afla un buton Details care duce catre pagina actiunii (companiei) selectate.Butonul are in spate o metoda din ASP.NET numita URL.Action care apeleaza metoda CompanyDetails din controller-ul User,careia ii trimite un obiect de tip Transaction care are ca si parametrii CNP ul individului si cui-ul companiei selectate.Aceasta metoda redirectioneaza user-ul catre View-ul CompanyDetails trimitand prin intermediul proprietatii ViewBag doua liste cu obiecte de tip ASK si BID prin intermediul metodelor FindASKForListing si FindBIDForListing.Acestea aduc toate ordinele de vanzare ( ASK ) si cumparare ( BID ) pentru compania respectiva si le afiseaza in doua tabele separate.Pentru fiecare tip de ordin,de vanzare si cumparare,exista un formular in care se specifica doua fielduri (cantitate si pret) pe care utilizatorul le completeaza pentru a vinde sau a cumpara o anumita cantitate dintr o actiune la un anumit pret.Fiecare formular are un buton la care este asociata o actiune de tip AJAX care trimite un obiect de tip Transaction catre metoda User/BUY sau User/SELL in momentul in care user ul plaseaza un ordin.  
 Pentru ordinele de cumparare,in momentul in care este plasat un ordin se apeleaza metoda BUY din controller ul User care la randul ei apeleaza metoda ValidateBuy.In aceasta metoda este apelata metoda CheckUserDebitOrder care verifica daca user ul are destui bani in cont pentru a realiza un ordin de cumparare.Aceasta metoda verifica daca suma tuturor ordinelor de cumparare in asteptare impreuna cu noul ordin de cumparare este mai mica sau egala cu suma de bani din contul clientului.Daca aceasta este mai mica se intelege faptul ca user-ul are destui bani in cont pentru a introduce un ordin de cumparare,acesta fiind introdus in baza de date in tabela de BID.In cazul in care suma este mai mare inseamna ca user ul nu are bani suficienti in cont,ordinul de cumparare nefiind introdus in baza de date.  
 Pentru ordinele de vanzare,in momentul in care se plaseaza un ordin,se apeleaza metoda SELL din controller ul User,metoda care la randui ei apeleza metoda ValidateSell.In aceasta metoda se apeleaza metoda CheckCountAskPortfolio care verifica daca user ul are destule actiuni in portofoliu pentru a plasa ordinul de vanzare.Metoda verifica daca cantitatea tuturor ordinelor de vanzare pe actiunea respectiva aflate in asteptare si asociate accountului respectiv plus cantitatea pe care o introduce pentru vanzare in acel ordin este mai mica sau egala cu cantitatea de actiuni detinuta de user in portofoliu pentru actiunea respectiva.Daca cantitatea este mai mare inseamna ca user ul incearca sa vanda o cantitate de actiuni mai mare decat cea pe care o detine in portofoliu si astfel ordinul de vanzare nu va fi introdus in baza de date.  
 In momentul in care un ordin de vanzare sau de cumparare este introdus in baza de date se apeleaza metoda Order,care este o metoda recursiva si are rolul de a asocia ordinele din tabelele de ASK si BID pentru a realiza o tranzactie intre un cumparator si un vanzator.Aceasta primeste un parametru de tip Transaction care este folosit de alte 2 metode care sunt apelate in interiorul metodei Order:OrderOrderedBID care intoarce o lista de obiecte de tipul BID si OrderOrderedASK care intoarce o lista de obiecte de tipul ASK.Metoda OrderOrderedBID are rolul de a intoarce o lista de inregistrari din tabela BID ( pentru compania pe care user ul o selecteaza din tabelul de companii listate) ordonate descrescator dupa pret si crescator dupa data crearii.OrderOrderedASK are rolul de a intoarce o lista de inregistrari din tabela ASK ordonate crescator dupa pret si crescator dupa data crearii ordinului(field ul CreatedOn).Motivul utilizarii acestor metode este maparea ordinelor de BID si ASK dupa urmatoarea logica : se preia din baza de date ordinul BID cel mai vechi cu cel mai mare pret si ordinul ASK cel mai vechi cu cel mai mic pret.Din aceste 2 liste se preia primul rezultat gasit folosind metoda extensie Linq First().Inainte de a realiza o tranzactie de cumparare vanzare se realizeaza 2 validari:se valideaza faptul ca ordinul de vanzare si cel de cumparare nu apartin aceluiasi user (in cazul in care apartin se preia urmatorul ordin din ASK care nu este asociat user ului din ordinul de BID) si se verifica ca pretul ordinului de cumparare (BID) sa fie mai mare sau egal cu pretul ordinului de vanzare.In cazul in care aceste 2 validari sunt realizate,se trece la urmatorul pas care reprezinta o verificare daca vanzatorul este o persoana juridica (companie) sau persoana fizica (individ).Aceasta verificare se realizeaza dupa field ul IsIPO care este inserat cu fiecare ordin ASK inserat in baza.In cazul in care ask ul este introdus de o firma,field ul IsIPO va avea valoarea true,altfel va avea valoarea false.Daca field-ul IsIPO are valoarea true se apeleaza metoda UpdateCompanyTransaction,altfel se apeleaza metoda UpdateIndividualPortfolio.Motivul acestei validari este functionalitatea platformei care permite companiilor sa se inregistreze si sa se listeze.In cazul in care un individ cumpara actiuni de la o companie tranzactiile financiare necesare sunt diferite fata de cazul in care ar fi cuparat de la o persoana fizica.Din acest motiv a fost necesara o separare a fluxurilor.  
 Metoda UpdateCompanyTransaction primeste 2 parametrii,obiectele bid si ask selectate prin metodele descrise anterior.In continuare se varifica cantitatea de cumparare si vanzare din cel 2 ordine,existand 3 cazuri posibile:

* Cantitatea din ordinul ASK (numarul de actiuni vandute de companie) este mai mare decat cantitatea din ordinul BID (numarul de actiuni pe care individul vrea sa le achizitioneze):

In acest caz se realizeaza urmatoarele tranzactii:

1. Se face diferenta intre cantitatea ask si bid si se face update pe ordin cu cantitatea ask ramasa;
2. Debitul companiei (depozitul in bani al companiei) va creste cu cantitatea ordinului bid inmultita cu pretul ordinului ask;
3. Cantitatea de actiuni oferita de companie sub forma de IPO scade cu cantitatea din BID;
4. Avand in vedere ca ordinul BID a fost epuizat (intreaga cantitate a fost cumparata si tranzactia realizata),ordinul de cumparare va fi sers din tabela de BID.
5. Tranzactia realizata va fi introdusa in tabela Transaction,fiind introduse detalii precum CNP ul cumparatorului,CUI ul vanzatorului,data realizarii tranzactiei,pretul si cantitatea.

* Cantitatea din ordinul ASK este egala cu cantitatea din ordin ul BID:

1. Debitul companiei creste cu cantitatea actiunilor din ordinul de bid inmultita cu pretul ordinului ASK.
2. Cantitatea de actiuni oferita de companie sub forma de IPO scade cu cantitatea din BID;
3. Ambele ordine sunt eliminate din tabelele de ASK si BID
4. Transactia este inserata in tabelul Transactions.

* Cantitatea din ordinul ASK este mai mica decat cantitatea din ordinul BID:

1. Se face diferenta dintre cantitatea BID si ASK,cu aceasta diferenta se face update pe ordinul de BID.
2. Debitul companiei creste cu rezultatul inmultirii dintre cantitatea din ordinul ask si pretul ask.
3. Cantitatea de actiuni oferita de companie sub forma de IPO scade la 0.
4. Ordinul ASK este eliminat din tabela ASK.
5. Tranzactia realizata este introdusa in tabelul Transactions.

In continuare pretul de piata al actiunii devine pretul ASK al tranzactiei.Deasemenea in aceasta functie se apeleaza si functia UpdatePotfolio care face update pe contul si portofoliul individului care a cumparat actiunea:in cazul in care nu exista un portofoliu (in baza de date in entitatea Portofolio nu exista un portofoliu asociat acelui individ ) se creeaza un nou portofoliu cu actiunea cumparata.In cazul in care exista un portofoliu pe numele individului va creste cantitatea de actiuni asociata respectivului pertofoliu cu cantitatea de actiuni cumparata.Deasemenea debitul individului va scadea cu rezultatul inmultirii dintre pretul actiunii si cantitatea cumparata.  
 In cazul functiei UpdateIndivdualPotfolio logica este aproape lafel,cu 2 exceptii:

* In cazul cresterii debitului (suma de bani din cont),va creste debitul contului individului (nu al companiei) deoarece actiunea a fost cumparata de o persoana fizica,de la o persoana fizica.
* Suma de actiuni din tranzactie va fi scazuta din portofoliul individului care vinde actiunea (spre deosebire de companie,unde se scadea din cantitatea totatala de actiuni oferita sub forma de IPO)