



Detailed Design Document

A Document Management System (DMS) for governmental/non-governmental institutions

VENTURE^{UP}

DMS

Studentët: Blerta Brahim
Mynevere Hyseni
Njomza Aliu

Table of Contents

1. Mid-Level Design Models

Moduli për regjistrimi i të dhënave(trajnime,partner,profesorë,student etj) në bazën e të dhënave

1.1. Data Access Layer Static Structure

- 1.1.1. Decomposition**
- 1.1.2. State**
- 1.1.3. Collaborations**
- 1.1.4. Responsibilities**
- 1.1.5. Interface**
- 1.1.6. Properties**
- 1.1.7. Transitions**
- 1.1.8. Relationships**

2. Low-Level Design Models

2.1. Packaging

- 3. Mapping Between Models**
- 4. Detailed Design Rationale**

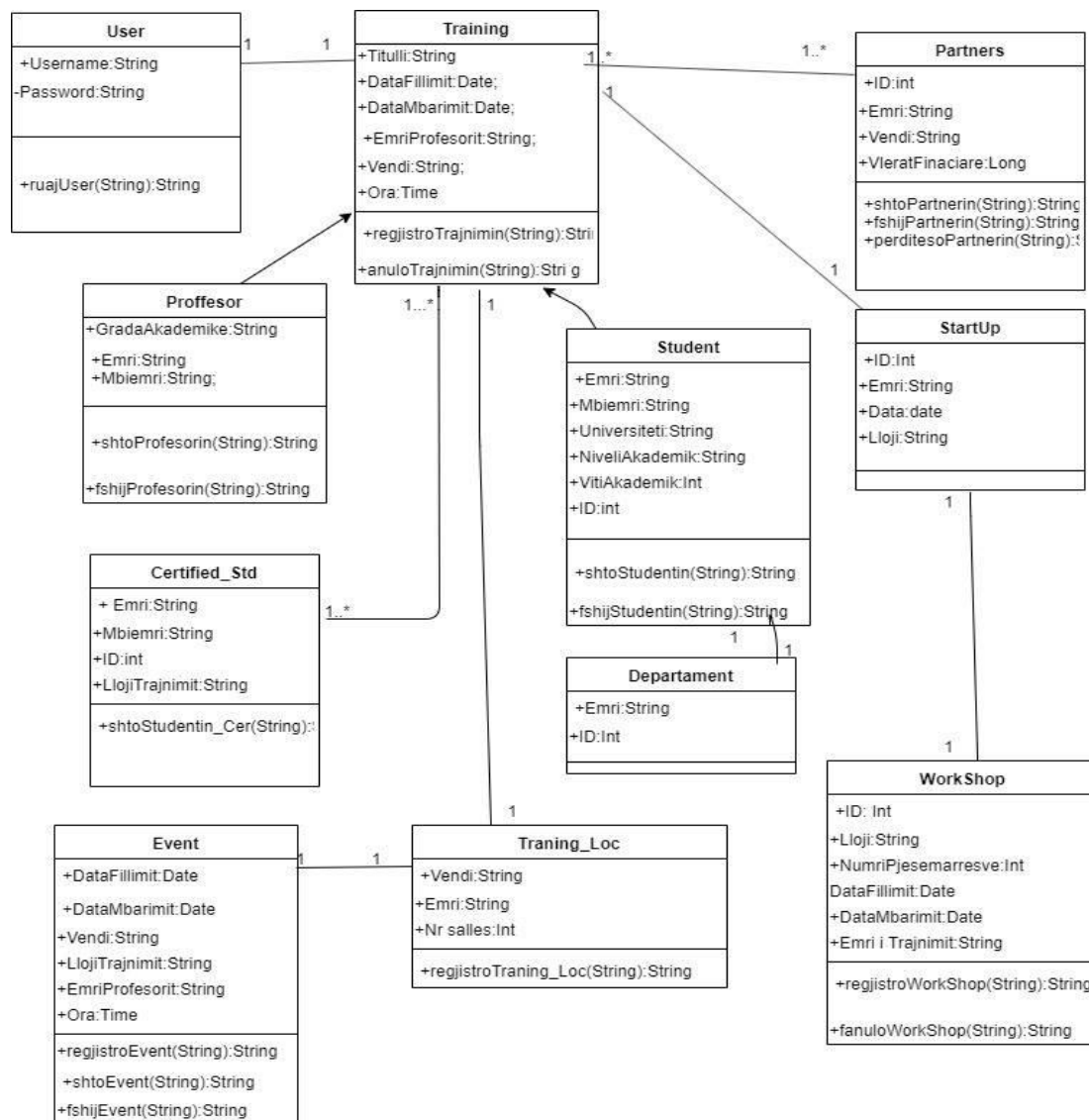
1. Mid-Level Design Models

Modeli i dizajnit Mid-Level shtjellon cdo shtresë arkitekturale dhe organizon secilin layer nëpër seksione. Ne do ta shtjellojmë më poshtë shtresën Data Access.

Moduli për regjistrimi i të dhënave(trajnime,partner,profesorë,student etj) në bazën e të dhënave

1.1. Data Access Layer Static Structure

Data Access Layer(DAL) është njëra prej shtresave të softuerit VentureUp DMS e cila siguron një qasje të thjeshtuar në të dhënat e ruajtura dhe në ruajtje të vazhdueshme të tyre në një bazë të dhënash. Data Access Layer ka një strukturë statike siq është paraqitur më poshtë.

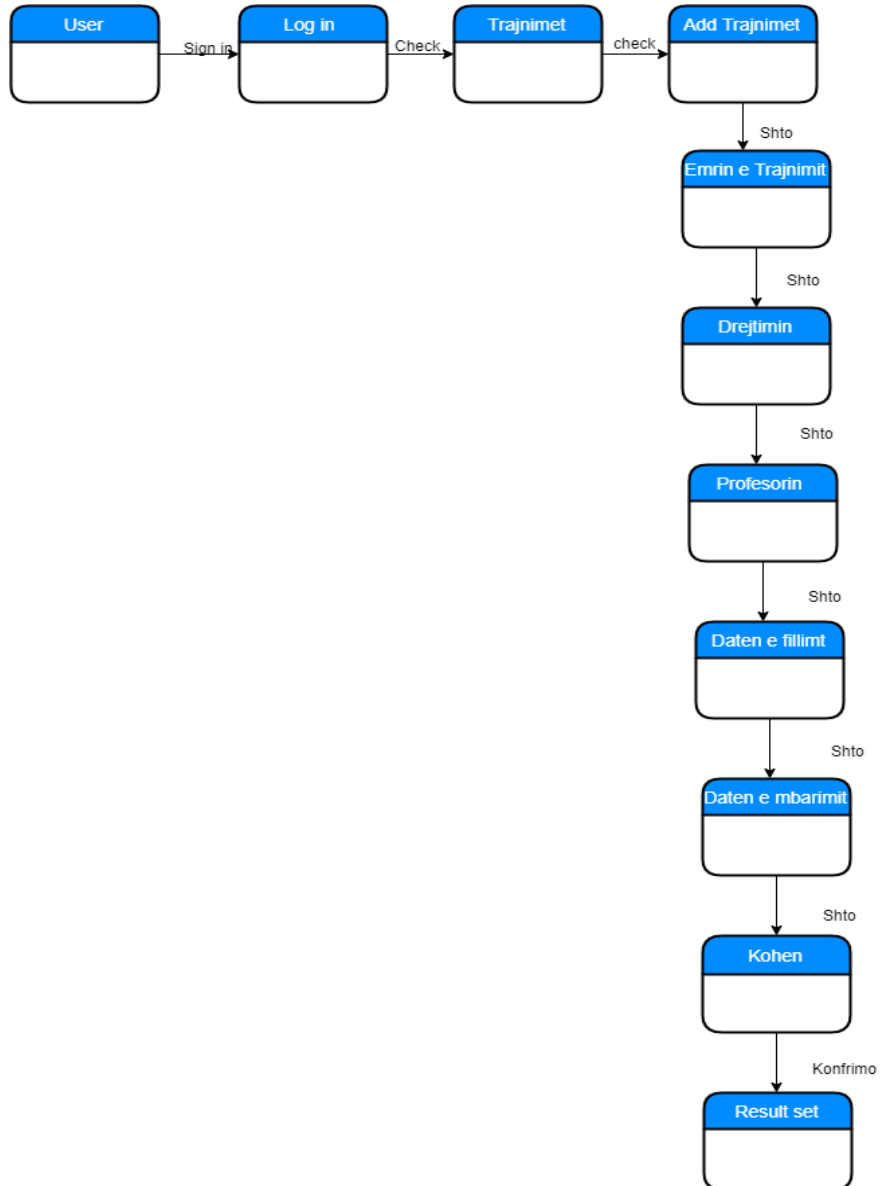


1.1.1. Decomposition

Module	Responsibilities
User	Përmbanë operacionet bazë të një përdoruesi. Të dhënat e përdoruesit i cili do të jetë i vetëm në këtë sistem. Të dhëna si: username, passëord. Si dhe operacione si ruajUserin.
Training	Përmbanë të gjitha të dhënat e një trajnimi si dhe operacionet që do të ndodhin në lidhje me atë trajnim. Të dhënat si data e fillimit, dita e mbarimit, profesori që është i caktuar për trajnim, vendi dhe ora. Operacione mund të jenë regjistroTrajnimin dhe anuloTrajnimin.
Partners	Përmban partnerët mbështetës të organizatës të cilët mundësojnë financimin e trajnimeve në organizatë. Të dhënat për këtë pjesë mund të jenë emri i partnerit dhe vlera financiare që ofron kurse operacionet shtoPartnerin, fshiPartnerin dhe përditësoPartnerin..
Professor	Përmban profesorët që mbajnë trajnimet dhe i organizojnë në këtë organizatë. Të dhënat për këtë pjesë mund të jenë emri, mbiemri dhe grada si dhe operacionet janë shtoProfesorin dhe fshijProfesorin.
Student	Përmban studentët që janë pjesë e trajnimeve si dhe të gjitha të dhënat për ta si emri, mbiemri, universiteti, niveli dhe viti akademik kurse operacione janë shtoStudentin dhe fshijStudentin.
Startup	Përmban të dhënat për startupet që organizon organizata me të dhënat dhe të gjitha vecoritë për startup, të dhëna këto si emri, dhe lloji i startupit.
Certified_Student	Përmban listen e studentëve që kanë përfunduar me sukses trajnimin dhe janë certifikuar kur ka përfunduar trajnimi me të dhënat emri dhe mbiemri i studentit dhe lloji i trajnimit dhe operacionet si shtoStudentin.
Department	Përmban të dhënat e departmetit të së cilit lloj është trajnimi.
Event	Përmban eventet që organizohen për të gjithë të interesuarit me një temë të caktuar dhe të dhënat për to si data e fillimit, data e mbarimit, vendi, lloji i trajnimit, emri i profesorit, ora e mbajtjes dhe operacionet si regjistroEventin, shtoEventin dhe fshijEventin.
Workshop	Përmban të dhënat për workshopet e organizuara, të dhëna këto si numri i pjesëmarrësve, data e fillimit dhe mbarimit, lloji i workshopit, emri i trajnimit me operacionet si regjistroworkshopin dhe anuloworkshopin.

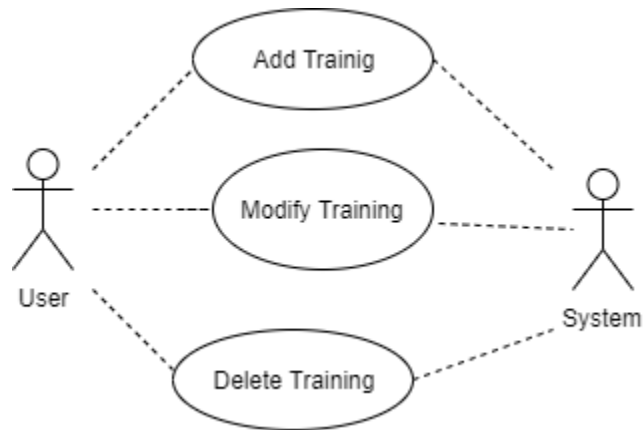
1.1.2. State

Përdoruesi do të çaset në sistem me anë të të cilit do të shtojë trajnimin e caktuar duke plotësuar secilën field që i kërkohet për të dhënat e trajnimit siç janë emri,drejtimi(pra lloji se cfarë teme përmban ai trajnim),profesori se cili do të jetë trajner,datën e fillimit, datën e mbarimit si dhe orën se kur do të mbahet ai trajnim.



1.1.3. Collaborations

Useri pasi të kycet në sistem mund të shtojë trajimin,modifikojë dhe të fshijë atë.



1.1.4. Responsibilities

Module	Responsibilities
User	Përmban të dhënat kryesore të userit, sic janë username dhe passëord.
Log-in	Pasi shfrytëzuesi ti jep të dhënat e tij ai mund të logohet.
Trajnimet	Pasi të logohet hapet dritarja e trajnimeve ku janë të gjitha trajnimet e shtuara më parë , ku në dritaren trajnimet është edhe një dritare tjerë e cila mundëson të shtohen trajnimet.
Add Trajnimin	Dritarja add trajnimin mundësonë regjistrimin e trajnimit duke i plotësuar të dhënat e tijë.
Emri i Trajnimit	Pasi të bëjmë add trajnimin duhet të tregohet emri i trajnimit i cili po regjistrohet.
Drejtimi	Kjo dritare mundëson shtimin e drejtimit të trajnimit se cilit drejtim i takonë.
Data e Fillimit	Data e Fillimit mundësonë shtimin e datës së fillimit të trajnimit.
Data e Mbarimit	Data e Mbarimit mundësonë shtimin e datës së mbarimit të trajnimit
Kohen	Koha mundësonë shtimin e kohës së trajnimit.
Result Set	Pasë të gjitha konfirmimeve apo kërkimeve ky na jep rezultatin e duhur.

1.1.5. Interface

Të gjitha operacionet që përmbajnë parametra gjuajnë `IllegalArgumentException` nëse parametrat nuk janë dhënë ashtu siq kërkohet.

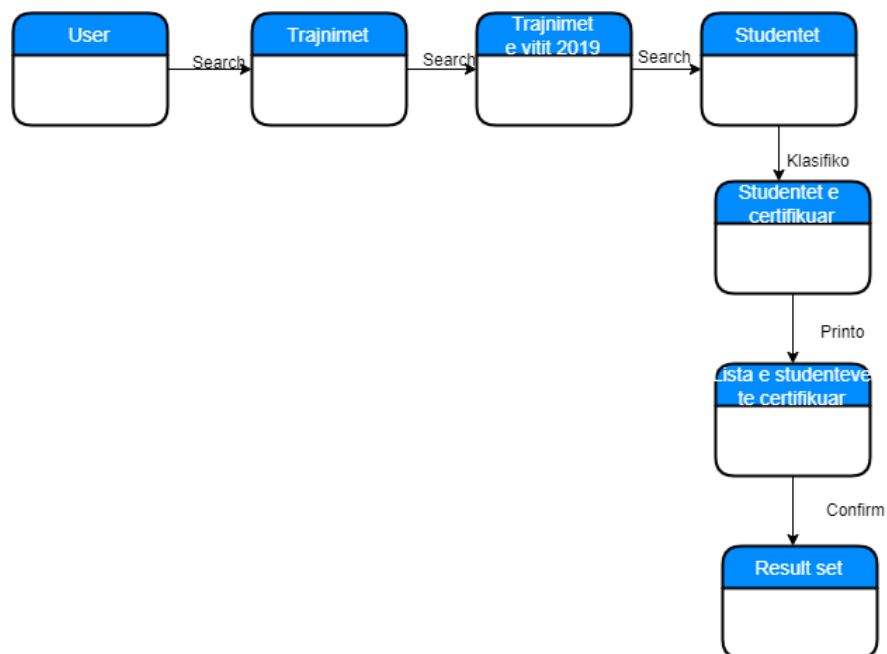
Use data	Syntax: <code>use(d:DataAccessObject)</code> Pre: d nuk duhet të jetë null Post: E dhëna është shfaqur të përdoruesi
Create data	Syntax: <code>create(t:Data)</code> Pre: t nuk duhet të jetë null Post: E dhëna(data) është ruajtur dhe inicializuar
Data access object parameters	Syntax: <code>create(d:Data)</code> <code>update(t:DataSource)</code> <code>read():DataSource</code> <code>delete(t:DataSource)</code> Pre: t duhet të jetë valid dhe jo null Post: Parametrat e <code>DataAccessObject</code> janë përdorur dhe të dhënat janë ruajtur.
Add a student	Syntax: <code>addStudent(id:String firstName:String lastName:String trainingId:String)</code> Pre: id nuk mund të jetë null Post: Shton një student në një trajnim të caktuar
Add a training	Syntax: <code>addTraining(id:String title:String type:String studentNr:String professorN:String)</code> Pre: id dhe title nuk mund të jetë null Post: Shton një trajnim të caktuar me një numër të kufizuar pjesëmarrësish.
Add a workshop	Syntax: <code>addworkshop(id:String name:String startDate:Date endDate:Date type:String trainingN:String)</code> Pre: id, startDate dhe endDate nuk duhet të jenë null Post: shton një workshop i cili do të jetë me data të caktuara
Add a event	Syntax: <code>addEvent(id:String name:String startDate:Date endDate:Date type:String location:String)</code> Pre: id nuk duhet të jetë null Post: shton një event i cili do të jetë në një kohë të caktuar dhe në një vend të caktuar
Add a professor	Syntax: <code>addProfessor(id:String firstName:String lastName:String grade:String training:String)</code> Pre: id nuk duhet të jetë null Post: shton një profesor të caktuar i cili do të ligjerojë një trajnim të caktuar
Review a certified student	Syntax: <code>setReview(id:String name:String training:String)</code> Pre: id nuk mund të jetë null Post: ruan një student si të certifikuar

1.1.6. Properties

Data Access Layer(DAL) është shtresa që na mundëson një qasje të thjeshtë në të dhënat e ruajtura si dhe një ruajtje e të dhënave të vazhdueshme në një bazë të dhënash. Te ky modul(regjistrimi i të dhënave)(trajnime,profesore,partner etj)(në bazën e të dhënave) shtresa kryesore që do të përdoret është DAL, në të cilën do të krijohen të gjitha interface dhe repository të këtij moduli e të cilat do ta mundësojnë funksionimin e tij.Interface dhe repository të cilat do të krijohen në DAL do ta bëjnë një performancë më të mirë në sistem si dhe besueshmëria e sistemit rritet.Proceset si insertimi, fshirja, përditësimi i të dhënave në një tabelë specifike në një bazë të dhënash siç është psh. Insertimi i një trajnimi do të mundësohet nga kjo shtresë(DAL) gjë që e bënë funksionale këtë pjesë të sistemit.

1.1.7. Transitions

Së pari përdoruesi logohet në sistem , pasi të logohet hap dritaren e trajnimeve dhe bënë kërkimin e trajnimeve të vitit 2019, pastaj kërkon studentët dhe klasifikon studentët e certifikuar gjatë mbajtjes së trajnimeve , pastaj printonë listën e studentëve dhe jep konfirmimin e kryerjes së detyrës.



1.1.8. Relationships

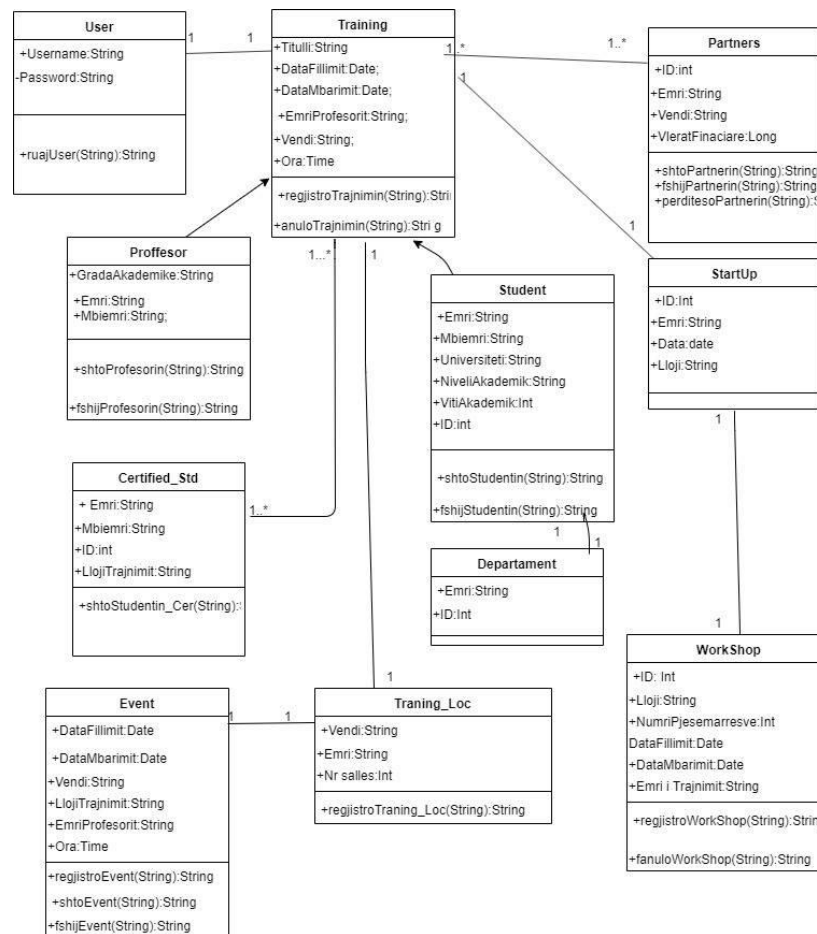
Siç e dijmë ekzistojnë relaçione në mes tabelave të bazës së të dhënave nga dhe në të cilën merren ose ruhen të dhënat. Secili trajnim do të ketë një profesorë të caktuar dhe cdo profesorë do të marrë pjesë si trajnues në një trajnim të caktuar. Poashtu trajnimi do të ketë shumë studentë si dhe secili student do të jetë pjesë e një trajnimi të caktuar.

Trajnimi do të ketë një ose më shumë partner si dhe partnerët do ta financojnë një ose më shumë trajnime.

Workshopi do të jetë pjese e një startup-i dhe secilit startup do ti takojë një trajnim i caktuar.

Një osë më shumë trajnimesh do të kenë një osë më shumë studentë të certifikuar.

Eventet e caktuara do të kenë një lokacion në të cilin do të mbahen dhe do të mbahen në bazë të një trajnimi të caktuar. Relacionet në mes klasave të këtij sistemi të gjitha janë të paraqitura në class diagramin më poshtë:



2. Low-Level Design Models

Low-Level Design Models është një proces i dizajnit në nivelin e komponenteve që ndjekin një proces hap pas hapi. Low-Level bënë hartimin logjik të kodit aktual të programit. Dizajni i nivelit të ulët është hartuar në bazë të nivelit të lartë të dizajnit. Loë-Level përshkruan diagramet e klasës me metodat dhe marrëdhëniet në mes të klasave dhe specifikave të sistemit. Ai përshkruan modulet në mënyrë që programuesi të mund ta kodojë drejtpërdrejt programin nga dokumenti. Faza e dizajnit të ulët është faza ku hartohen komponentet aktuale të sistemit.

2.1. Packaging

Paketat (packages) e Java-s do të përdoren për të implemtuar secilën paketë të shtresave të sistemit. Ato të cilat do ta bëjnë strukturimin e kodit, do ta bëjnë më të lehtë gjetjen e kodit i cili korrespondon me secilën pjesë të dizajnit. Specifikisht paketat e Java-s të cilat janë përdorur janë:

*Startup-kodet e modulit **startup layer***

*UI-kodet e modulit **user interface layer***

*DAL-kodet e modulit **data access layer***

*BLL-kodet e modulit **business layer***

IO-pjesen e kodit për input dhe output

Util- utility code modules

3. Mapping Between Models

Trajnimi	Profesori	Studenti	Partneri	Startup	Workshop	Event
Add Training button	Add Professor button	Add Student button	Add Partner button	Add Startup button	Add Workshop button	Add Event button
Save button	Save button	Save button	Save button	Save button	Save button	Save button
Update button			Update button	Update button	Update button	Update button
Delete button				Delete button	Delete button	Delete button
Search Field				Search Field	Search Field	Search Field

4. Detailed Design Rationale

Projekti do të jetë i përcaktuar dhe i ndarë në paketa(packages) sic janë shënuar në pjesën e loë-level design si dhe do ti plotësojë kërkesat e paraqitura në SRS.Projekti do të bazohet në disa parime. Ku parimet kryesore që e drejtojnë këtë projekt janë shënuar më poshtë:

- Ti përmbahemi kufizimeve të shtresave arkitekturale për të ruajtur integritetin e tyre
- Mbrojtja e informacioneve sensitive dhe zbërthimi i moduleve sa më shumë në mënyrë që ti bjëjmë sa më të ndryshueshme.
- Reflektimi nga modeli konceptual për ndertimi i modeleve të dizajnit sa më të mirë
- Ndërtimi i modeleve të dizajnit duke u bazuar në dokumentimet e shtresave.
- Dizajnimi i interface-ve sa më të mira që është e mundur.
- Nga modeli mid-level do të zgjidhim problemet me dizajnim dhe krijimin e një dizajni sa më të lehtë të kuptueshëm.
- Nga modeli loë-level do të kuptojmë se cilat paketa(packages) na duhen për ta bërë pjesën e implementimit të sistemit

Këto parime do të përdoren gjatë procesit të implemetimin të sistemit.Specifikisht në gjenerimin, evaulimin dhe përzgjedhjen e alternativave të dizajnit.Do të kemi një qasje me të lehtë kur të fillojmë me pjesen e kodit dhe të caktimit të detyrave hap pas hapi.Gjithcka do të jetë e analizuar në mënyrë të detajuar nesë plotësohen parimet e shënuara më lartë.