



Inxhinierimi softverik

Pjesa 7 – Dizajni: Arkitektura dhe metodologjia

Prof. Asoc. Dr. Ermir Rogova

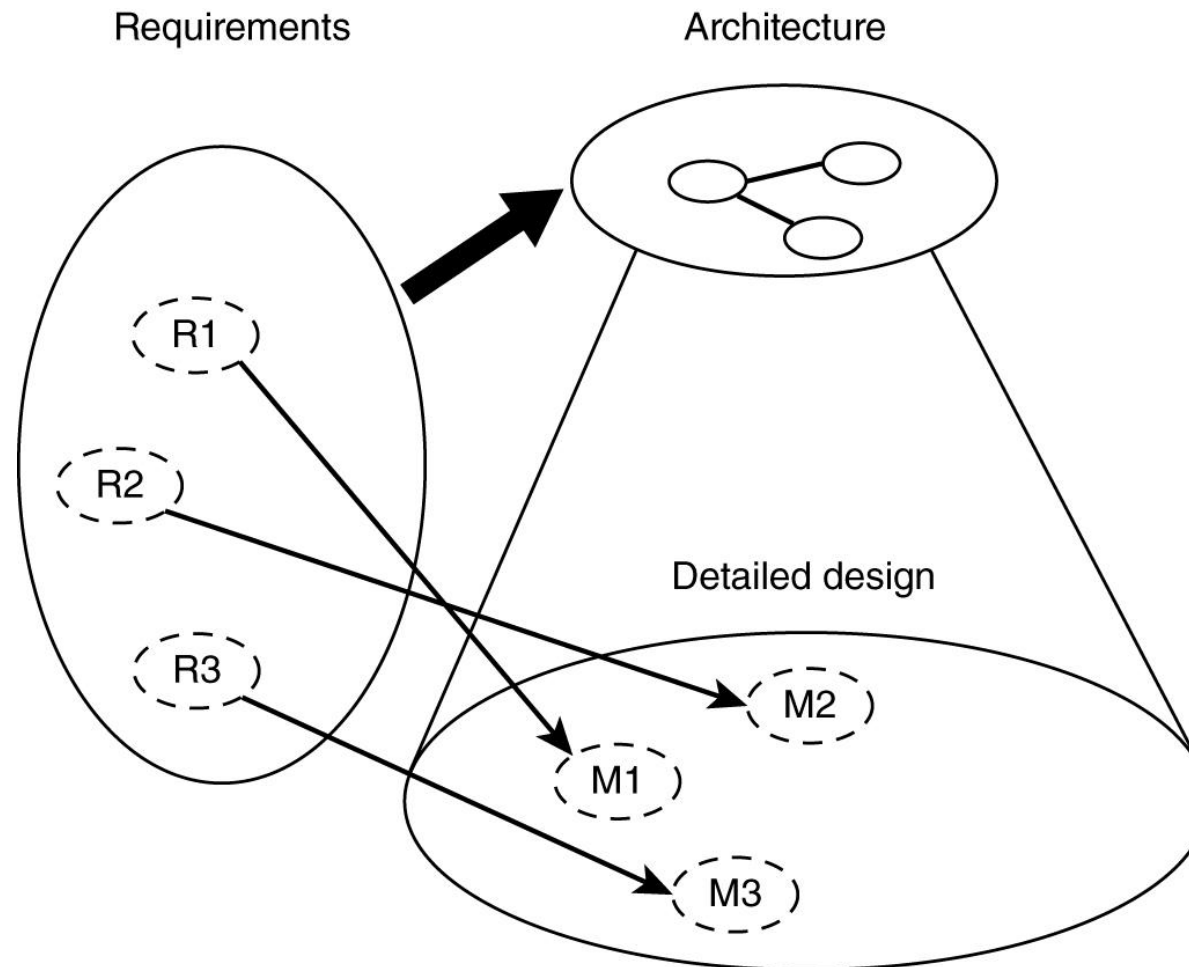
Temat mbi Dizajnin

- Dizajni arkitektural kundrejt atij të detajuar
- Stile, taktika dhe referenca
- Teknikat themelore për dizajn të detajuar
- Çështje themelore me dizajnin e ndërfaqes së shfrytëzuesit (UI)

Dizajni

- Fillon kryesisht prej kërkesave (duke evoluar prej funksionaliteteve dhe tipareve të tjera jo-funksionale).
- Si do të strukturohet një zgjidhje softverike?
 - Cilët janë komponentët kryesorë—(funksional)?
 - Zakonisht vijnë direkt prej funksionaliteteve (use cases).
 - Si ndërlidhen këto komponente?
 - Ekziston mundësia e ri-organizimit të komponenteve (kompozimi/dekompozimi).
- Dy nivelet kryesore të dizajnit:
 - Arkitektural (nivel i lartë)
 - Dizajn i detajuar

Marrëdhënia ndërmjet Arkitekturës dhe Dizajnit



Dizajni i detajuar
Vjen nga
Kërkesat dhe
Arkitektura

Arkitektura Softverike

- Struktura(t) e zgjidhjes, përbëhen nga:
 - Elementet e mëdha softverike
 - Tiparet e tyre të dukshme nga jashtë
 - Marrëdhëniet ndërmjet elementeve
- Çdo sistem softverik ka një arkitekturë.
- Mund të ketë shumë struktura!
 - Shumë mënyra të organizimit të elementeve , varsisht nga perspektiva
- Tiparet e jashtme të komponenteve (moduleve)
 - Ndërfaqet e komponenteve (moduleve)
 - Interaksionet e komponenteve (moduleve)

Pamjet (Views)

- Pamja (View) – përfaqësim i një strukture të sistemit
- 4+1 views (sipas Krutchen)
 - Logical (OO decomposition – key abstractions)
 - Process (run-time, concurrency/distribution of functions)
 - Subsystem decomposition
 - Physical architecture
 - +1: use cases
- Klasifikim tjetër (Bass, Clements, Kazman)
 - Module
 - Run-time
 - Allocation (mapping to development environment)
- Pamje të ndryshme për njerëz të ndryshëm

Stilet/mostrat(patterns) Arkitekturnale

Shërbejnë si pikënisje e ripërdorshme për aktivitete dizajnuese.

- Gypat dhe filtrat (Pipes and filters)
- Të shtyra nga ngjarjet (Event driven)
- Klient-server
- Model-view-controller (MVC)
- Shtresorë (Layered)
- Me fokus në DB (Database centric)
- Tre nivele (Three tier)

Stili arkitektural Pipe-Filter

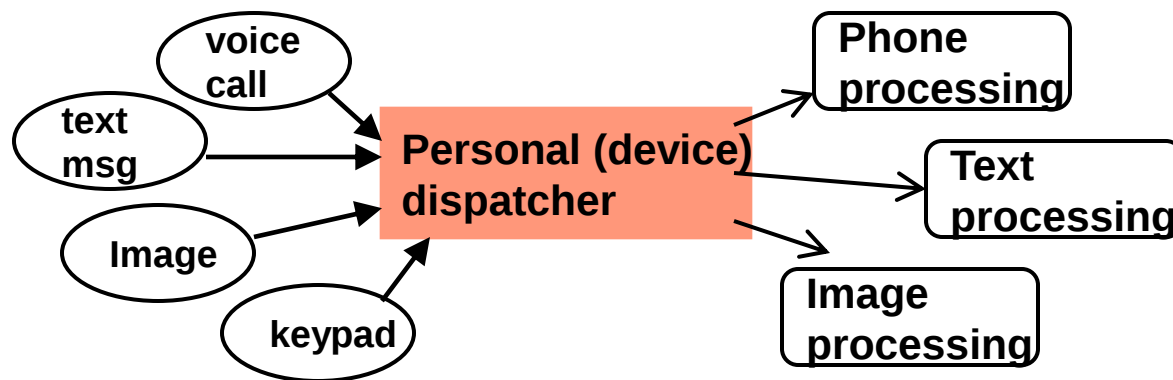
- Zgjidhja e dizajnit të nivelit të lartë dekompozohet në dy pjesë gjenerike (filtrat dhe gypat):
 - Filtri është një shërbim i cili transformon një rrëke të dhënave hyrëse në një rrëke të dhënave dalëse.
 - Gypi është një mekanizëm apo kanal përmes të cilit të dhënat rrjedhin prej një filtri në tjetrin.



Problemeve të cilat kërkojnë [procesim në grup](#) ju përshtatet kjo arkitekturë: p.sh., [payrolli](#), [kompilers](#), [financat mujore](#).

Të shtyera nga ngjarjet (Event Driven)

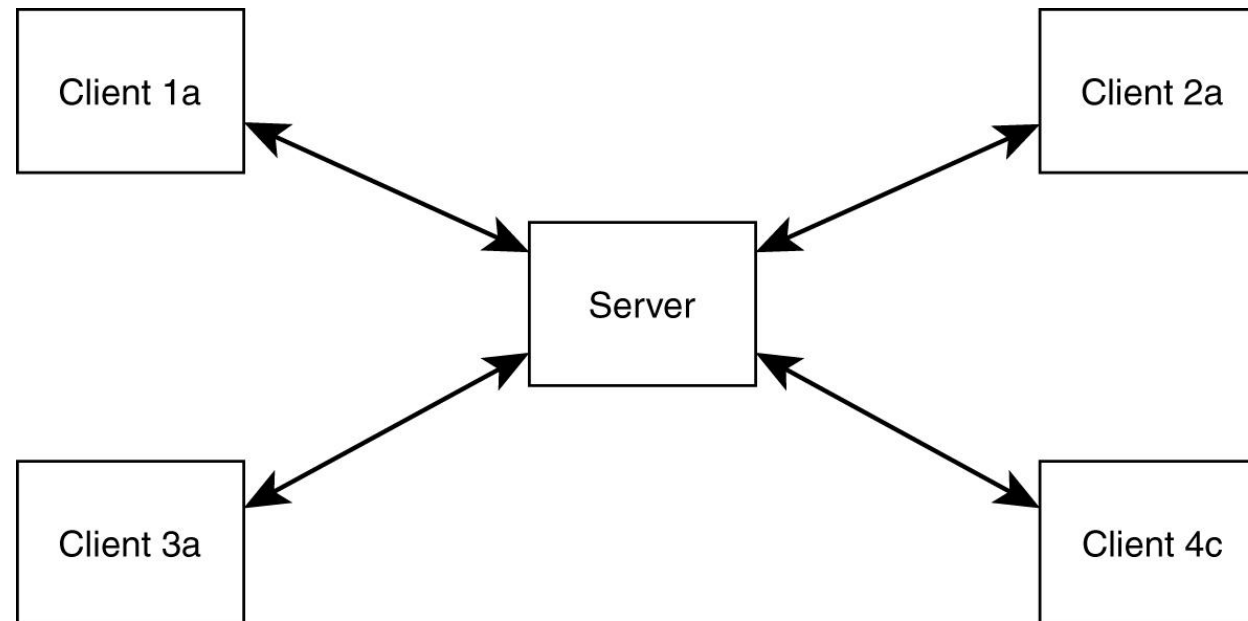
- Zgjidhja e dizajnit të nivelit të lartë bazohet në një event dispatcher, i cili menagjon ngjarjet dhe funksionalitetet që mvaren prej atyre ngjarjeve. Këto kanë tiparet në vijim:
 - Ngjarjet mund të jenë një njoftim i thjeshtë ose mund të përfshijë të dhëna
 - Ngjarjet mund të prioritizohen ose të bazohen në kufizime siç është koha.
 - Ngjarjet mund të kërkojnë procesim sinkron ose asinkron.
 - Ngjarjet mund të “registrohen” apo “çregistrohen” nga komponentet.



Problemet që ju përshtatet kjo arkitekturë përfshijnë sisteme në kohë reale si: **kontrolli i aeroplanit, monitori mjeksor, monitori shtëpiak, lojrat, etj.**

Stili themelor Klient-Server

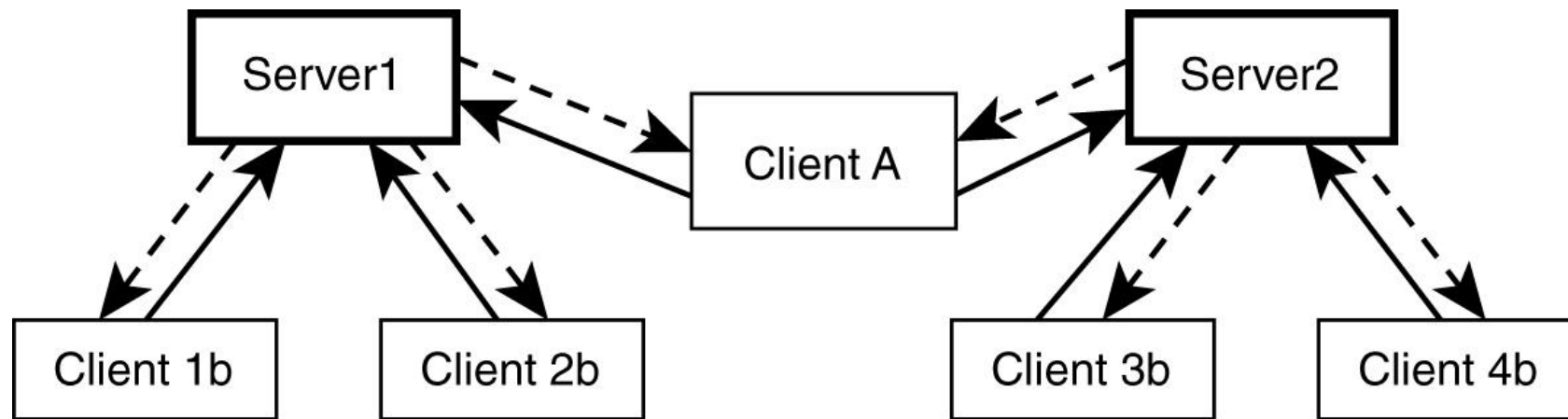
Aplikacioni ndahet në komponentën klient and komponentën server.



(a)

Stili Klient-Server

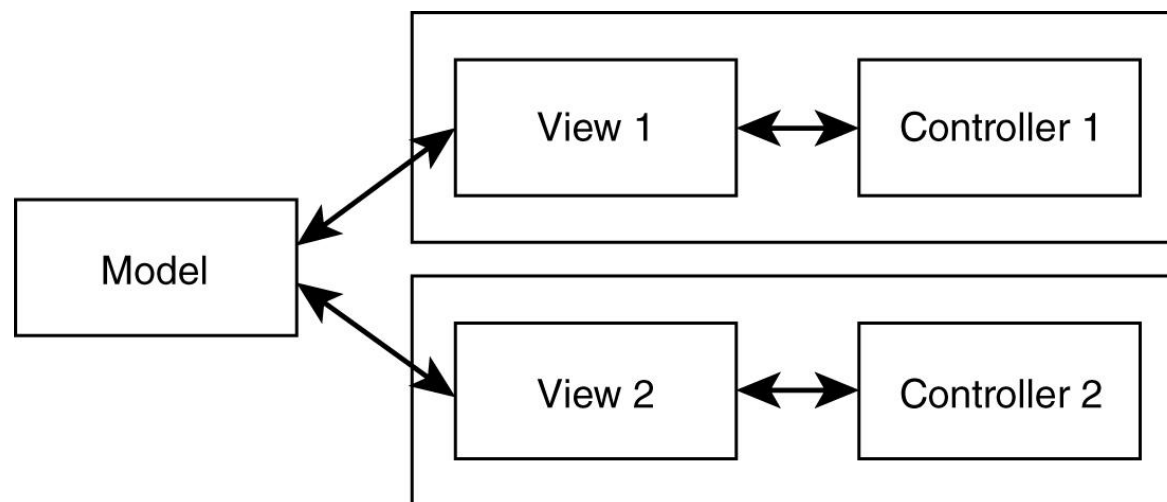
- Klienti mund të lidhet me më shumë se një server (serverët zakonisht janë të pavarur)



(b)

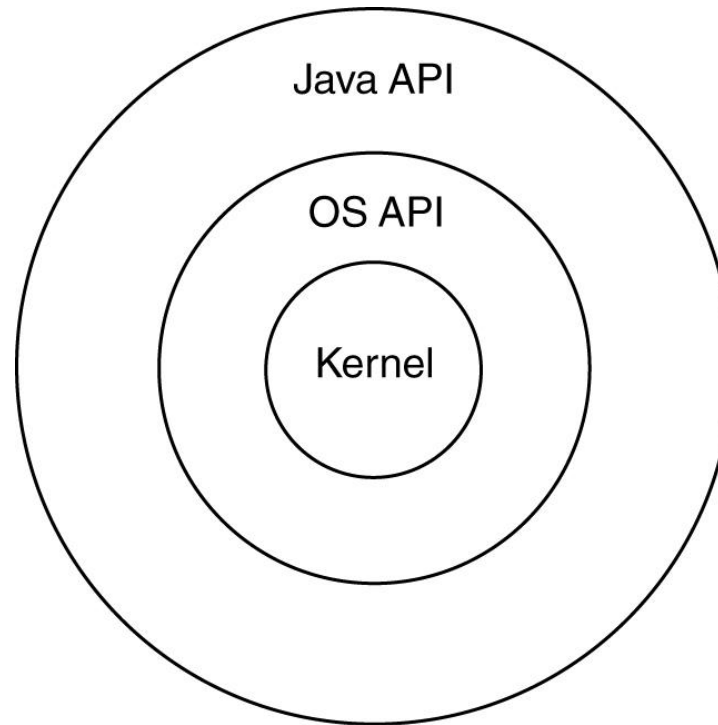
Stili Model-View-Control (MVC)

- Ndan modelin prej pamjes.
- Kontrolleri shpesh integrohet me pamjen.



Shumica e web aplikacioneve bie nën këtë stil.

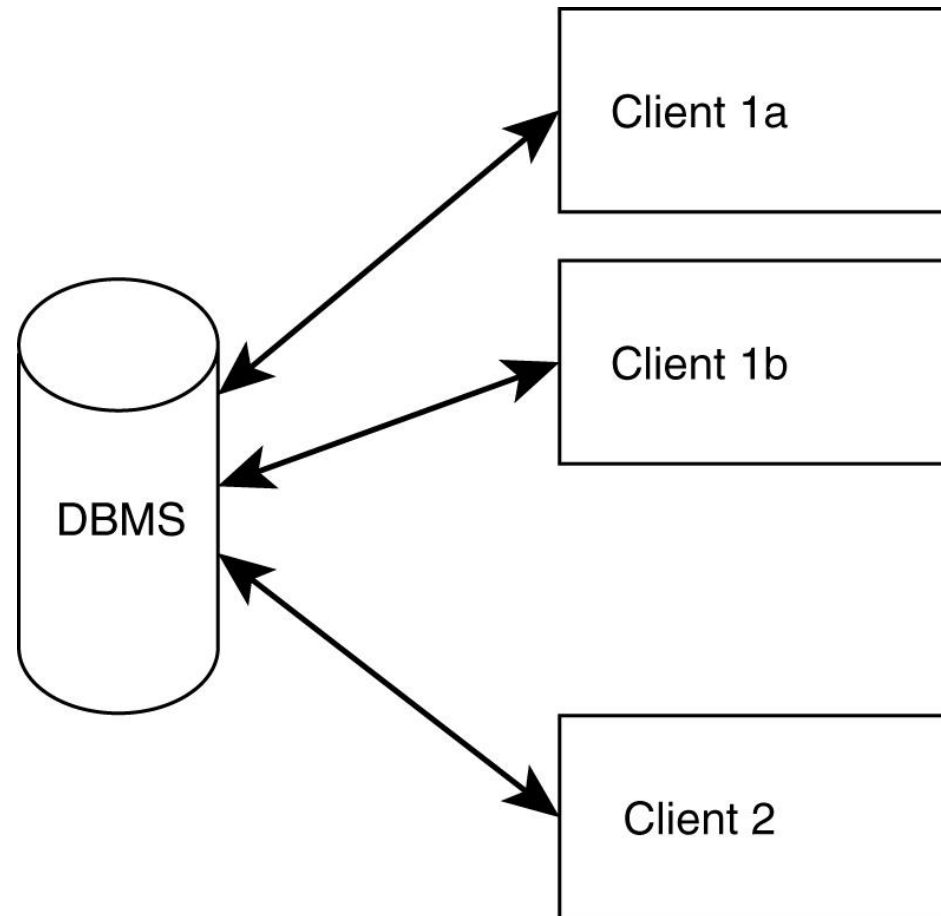
Stili Shtresor



Shtresa e jashtme mundet të kërkojë shërbim vetëm prej shtresës së brendshme.

- Shtresim strikt: vetëm shtresat direkt nën të
- Shtresim i relaksuar: cilindë prej shtresave nën të

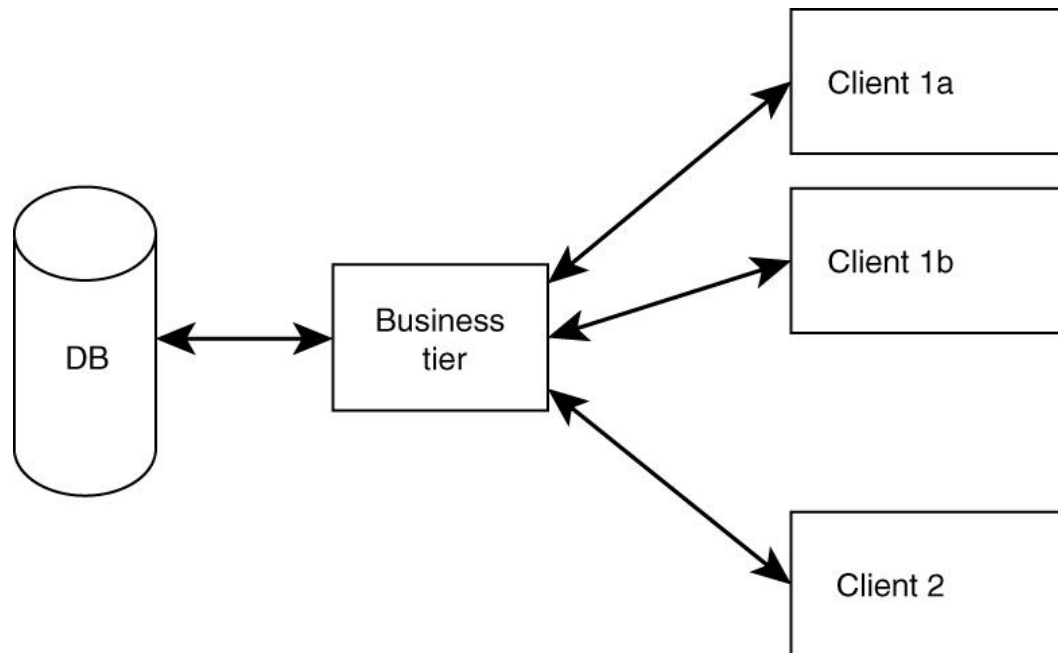
Stili (DB)-Centrik



**Shumë e
popularizuar me
aplikacionet
biznesore**

Stili me tre nivele

- Klientët nuk i qasen DB direkt.
- Fleksibilitet, integritet më i mirë (Përse?)

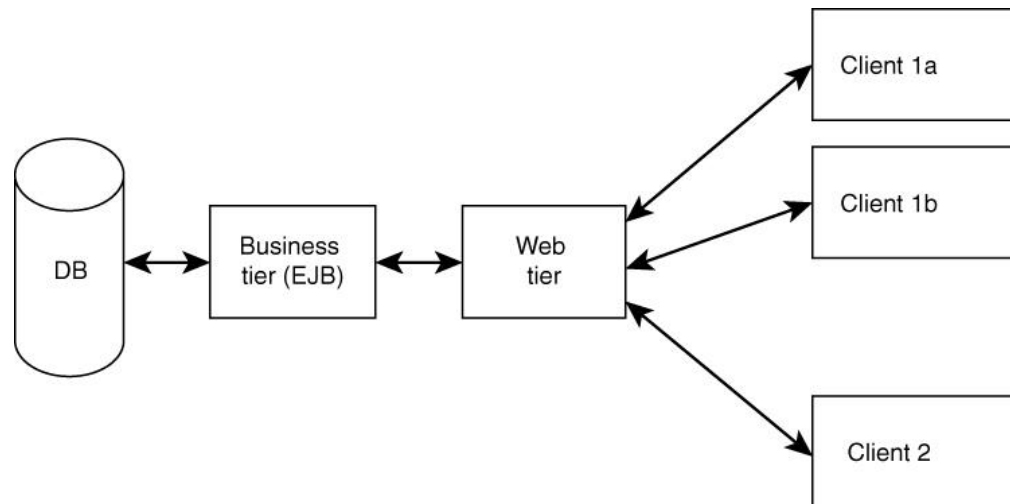


Taktikat Arkitekturnale

- Taktikat (me dallim nga stili arkitektural) janë për zgjidhjen e problemeve “më të vogla dhe specifike”
- Nuk ndikojnë në stukturën e përgjithshme të sistemit
- Shembull: shtojmë funksionalitete apo komponente specifike (p.sh., për rritjen e mbështetshmërisë) në dizajn për detektimin e gabimeve — sidomos për sistemet e shpërndara:
 - Heartbeat
 - Ping/echo

Arkitekturat referenciale

- Arkitektura të zhvilluara plotësisht.
- Shërbejnë si “templates” ose si “reference” për një klasë të sistemeve.
- Shembull: arkitektura referenciale J2EE (MVC2).



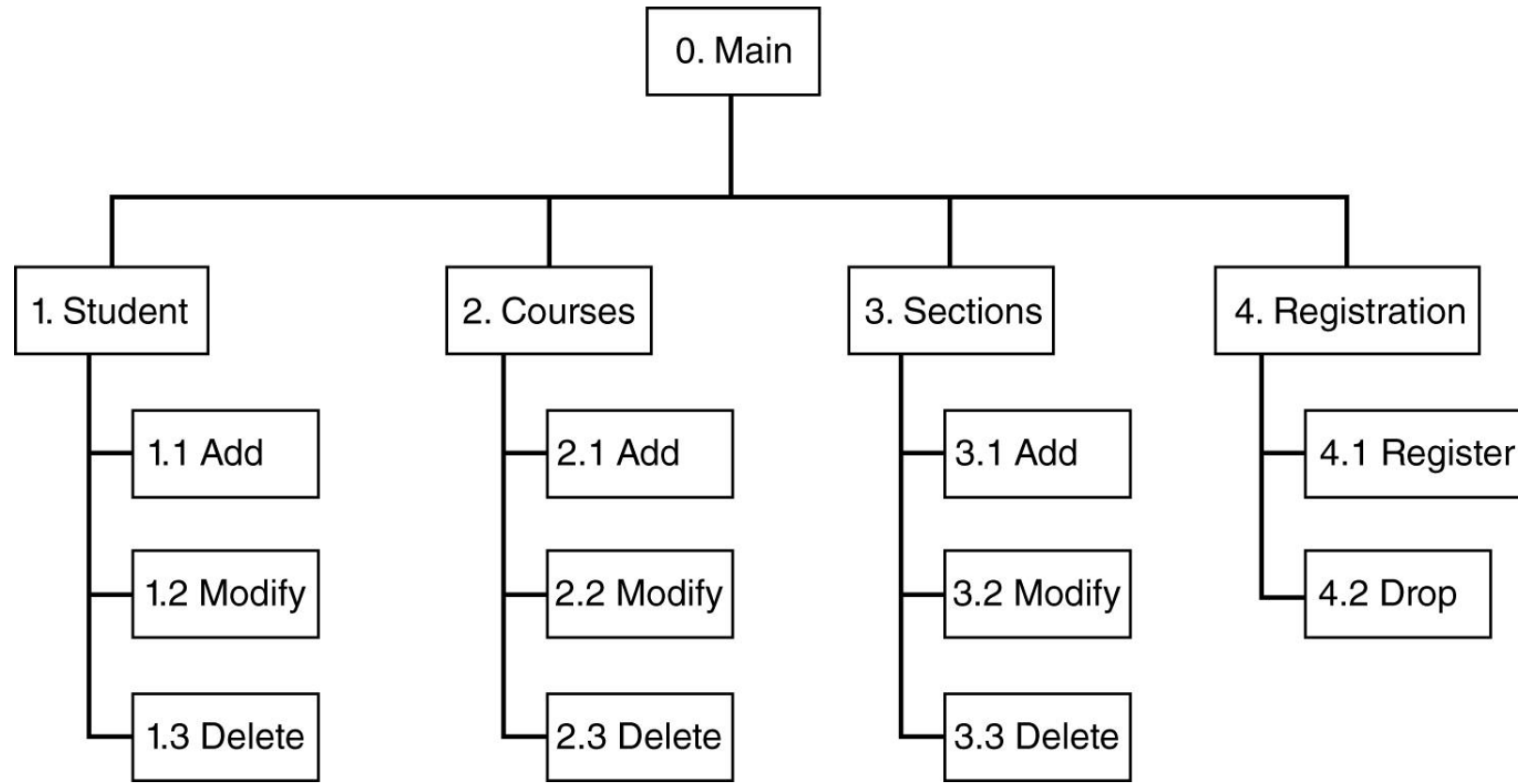
Dizanji i Detajuar

- Rafinimi i mëtutjeshëm i arkitekturës dhe përshtatja me kërkesat.
- Sa e detalizuar?
- Sa formale?
- Ndoshta nivele të ndryshme të detajeve për pamje të ndryshme.

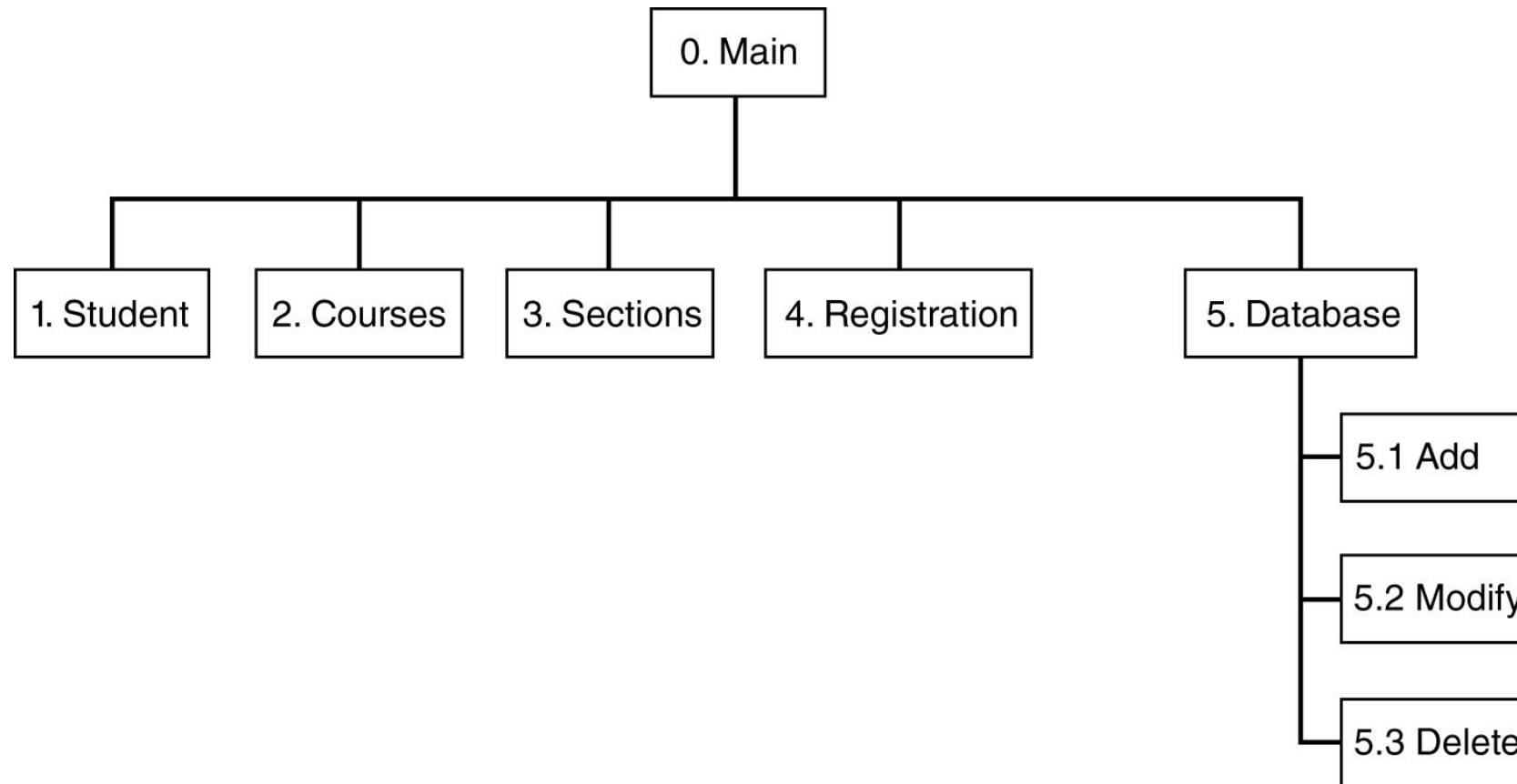
Teknika e dekompozimit funksional

- Daton prej kohës së “programimit të strukturuar” [tash Web apps me PHP].
- Starton me: Modulin kryesor (main)
- Rafinohet në nën-module.
- Ka alternativa të dekompozimit.

Dekompozimi i (aplikacionit menaxhues student-lëndë)



“Alternativa” Dekompozim/Kompozim



Dizajni i Databazave Relacionale

- Shumica e databazave përdorin teknologjinë relacionale.
- Relacionet (tabelat):
 - Bashkësi dy-dimensionale
 - Rreshtat (tuples), kolonat (tiparet)
 - Çelësi primar (identifikues unikat)
 - Çelësat e huaj (lidhin tabelat)

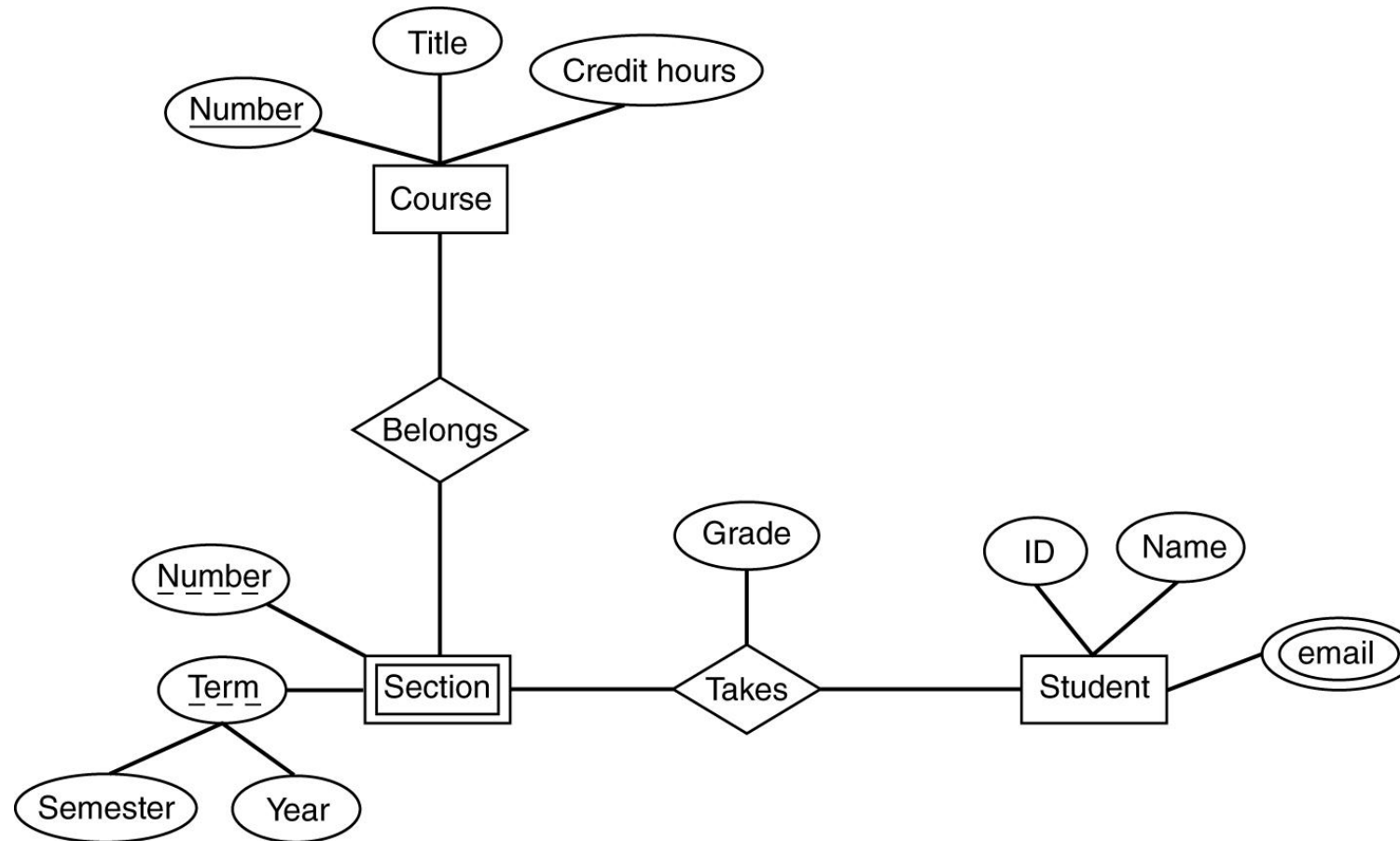
Dizajni i databazave

- Modelimi konceptual (bëhet gjatë fazës së analizës së kërkesave) prodhon diagramin ER.
- Dizajni logjik (në atë tabelor).
- Dizajni fizik (përcaktimi i data tipeve, etj.).
- Lëshimi / mirëmbajtja:
 - Niveli fizik (cili hdd, etj.)
 - Përshtatja e indeksëve

Diagramet Entity-Relationship

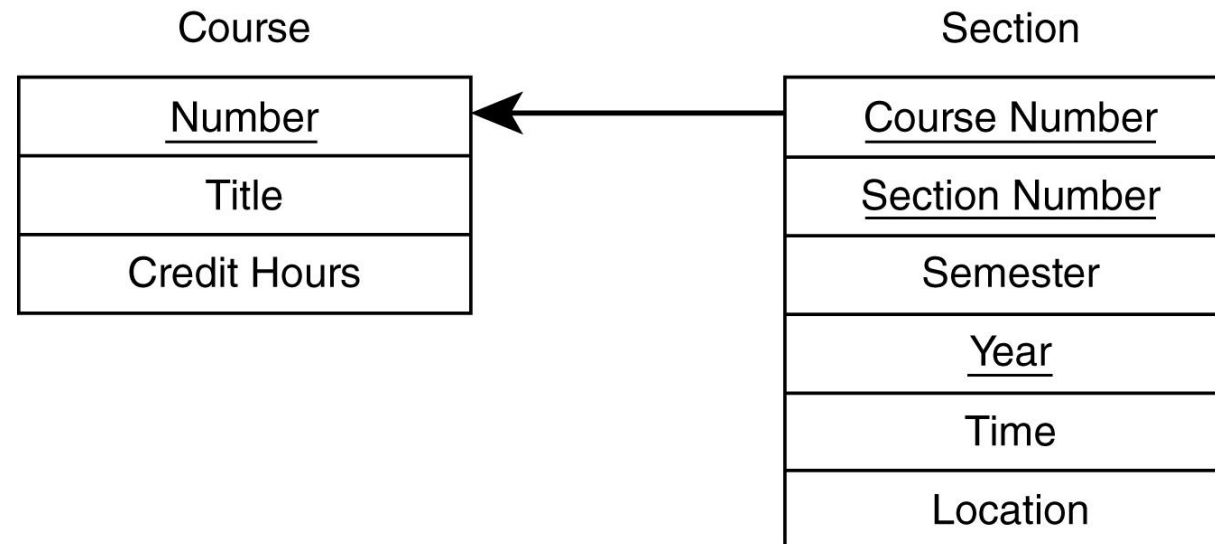
- Entitetet (drejtkëndësh)
 - Të dobët: linja dyfishe
- Marrëdhëniet (rombe)
- Atributet (elipsa)
 - Me shumë vlera: linja dyfishe
 - Identifikues: të nënvizuara

ER Diagram - Shembull



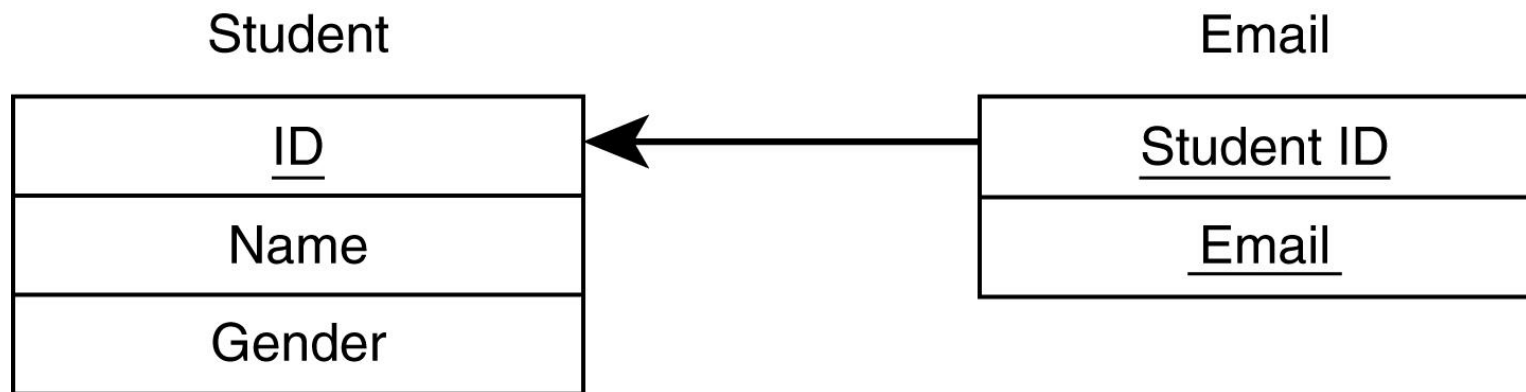
Dizajni logjik – Entitetet

- Entitetet në tabela.
- Atributet e përbëra rrafshohen.
- Te entitetet e dobëta, shtohet çelësi primar i entitetit të fortë.



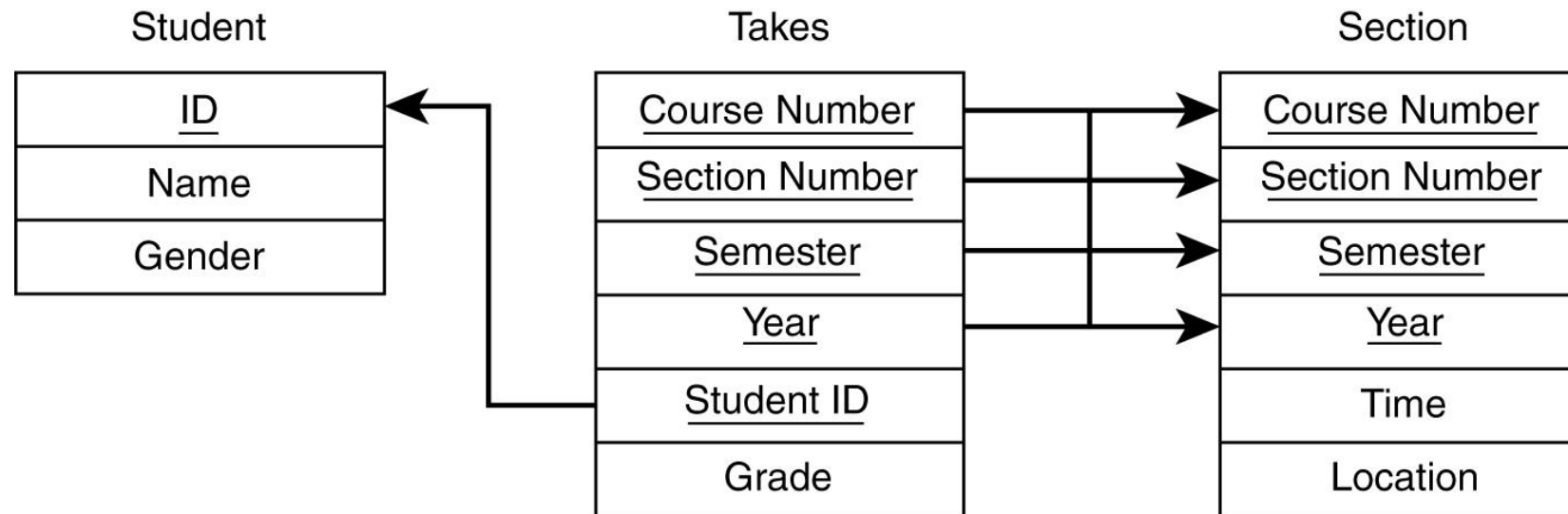
Dizajni logjik – Multi-Valued

- Përdorë një tabelë e re për atributet me shumë vlera.



Dizajni logjik – Marrëdhëniet

- Te 1-M, vendoset çelësi i huaj në tabelen e anës M.
- Te M-M, nevojitet tabelë e re.



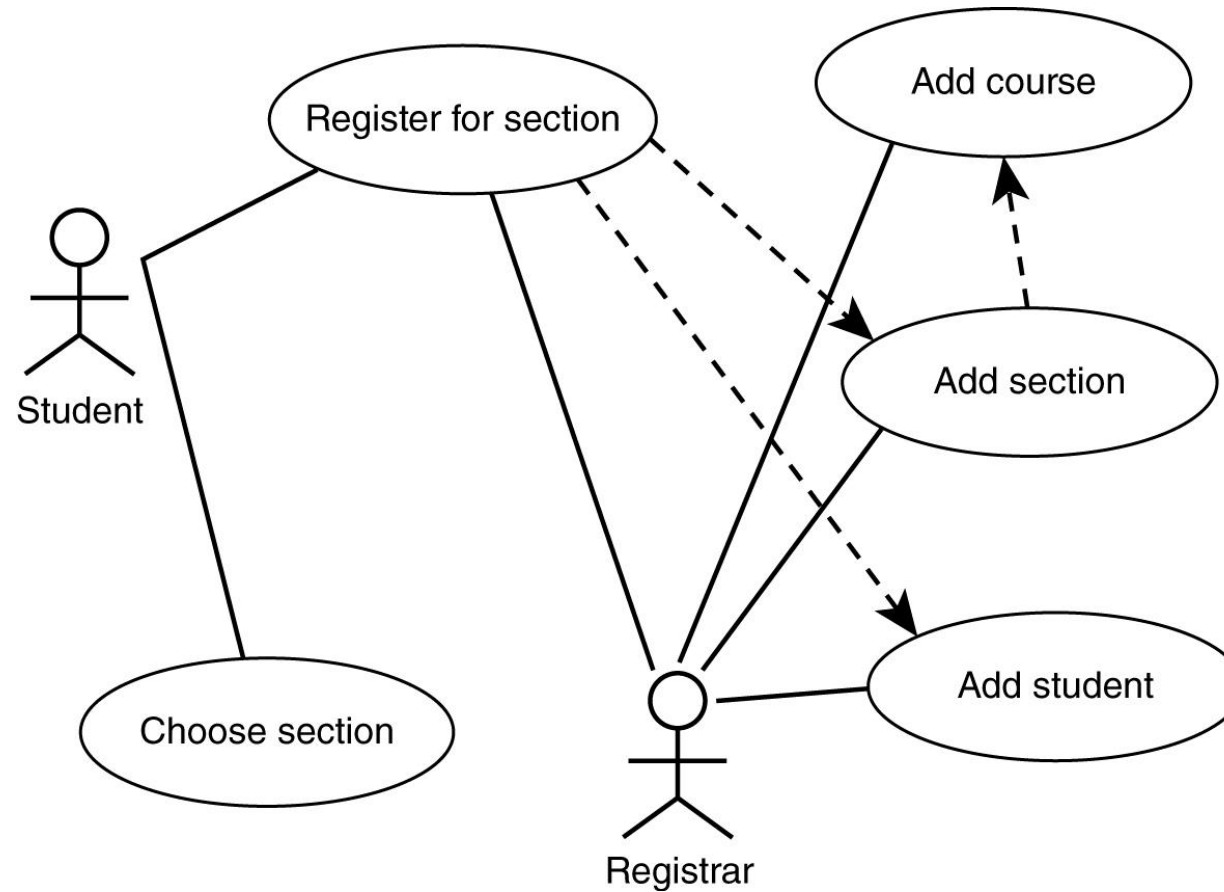
Dizajni fizik në DB

- Data typet për secilin atribut.
 - Mvaret edhe se cilat i mbështet DBMS në përdorim.
- Vendosja për indekset.
 - Kërkimet janë më të shpejta, azhurimet më të ngadalta.
 - Indeksët konsumojnë hapsirë.
 - Mund të përshtaten edhe gjatë lëshimit .
- Ndonjëherë bëhet edhe denormalizimi (shmanguni).

Dizajni OO

- Hapi i parë: rishiko dhe rafino “use cases”.
- Vendoset:
 - Cilat klasa të krijohen
 - Si ndërlidhen klasat
- Përdoret UML si gjuhë dizajnuese.

Diagrami “Use Case”

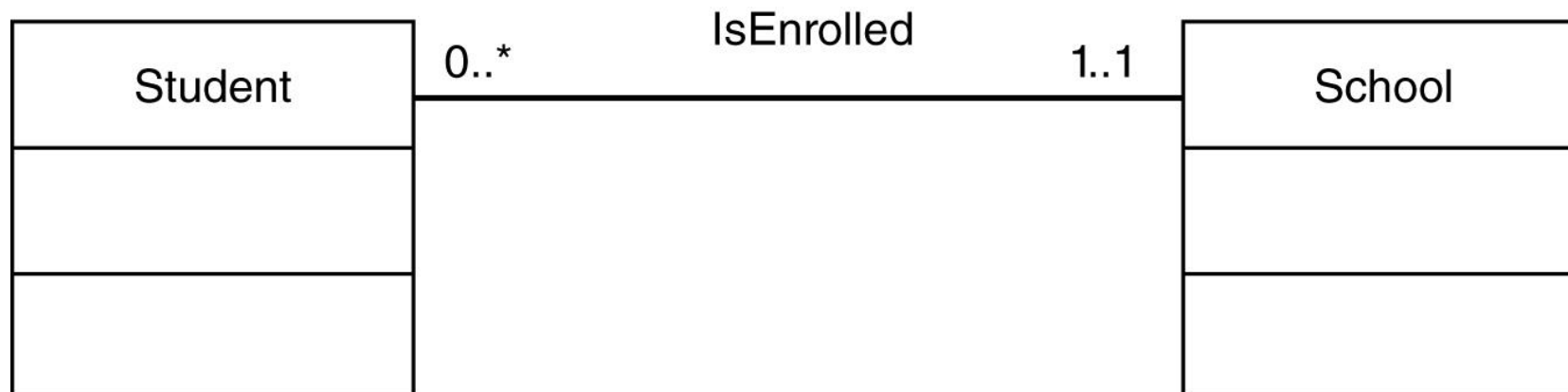


Dizajni i Klasave

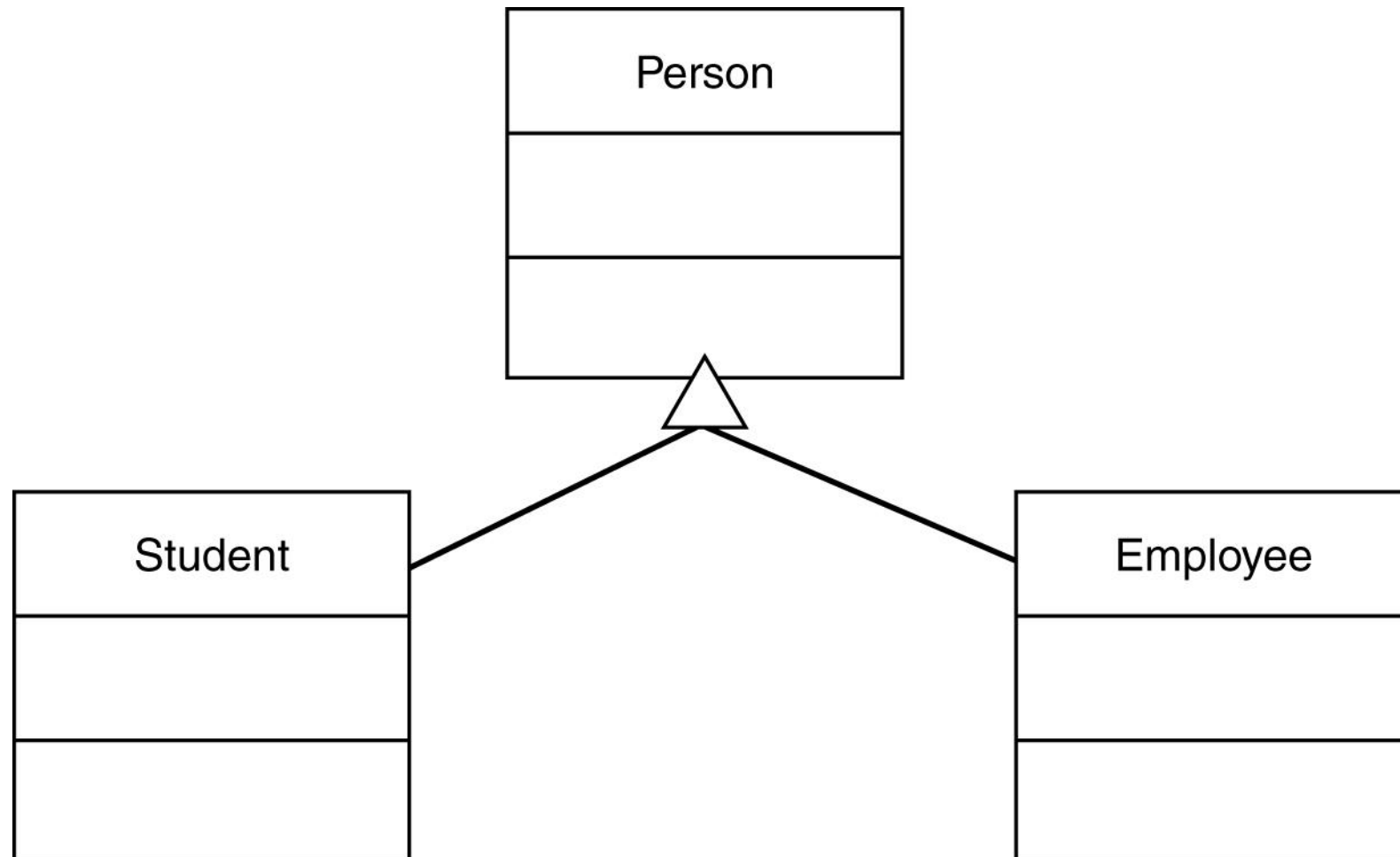
- Klasat përfaqsojnë entitetet e botës reale ose koncepte të sistemit.
- Organizimi në klasa: objektet në një klasë kanë tipare të ngjajshme.
- Klasat kanë tiparet (të dhënat).
- Klasat kanë edhe metodat (funksionet që kryhen).

Student
date of Birth : Date name : String
get Agein Years () : int get Agein Days () : int

Diagrami i klasave në UML

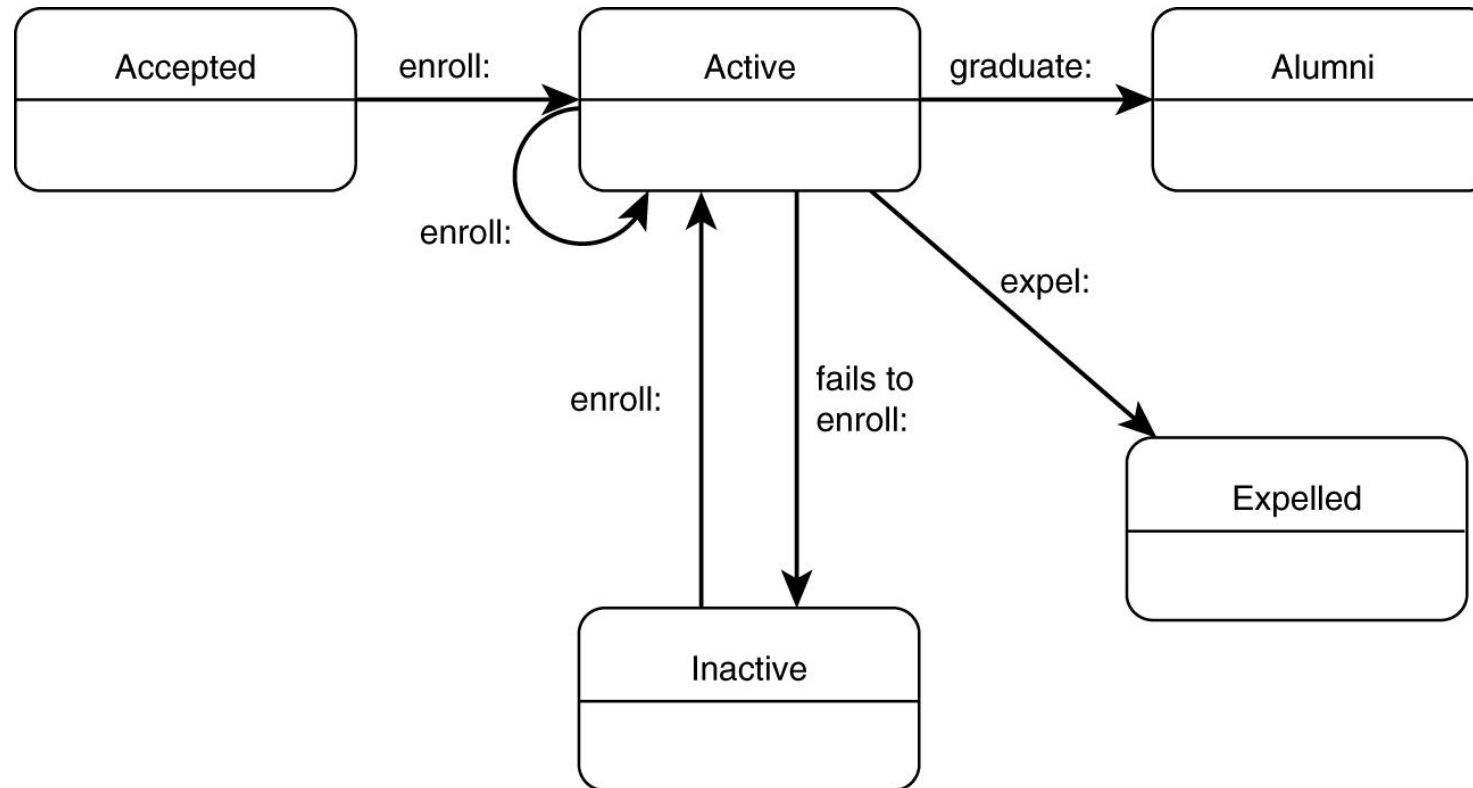


Diagrami i klasave në UML – Trashëgimia



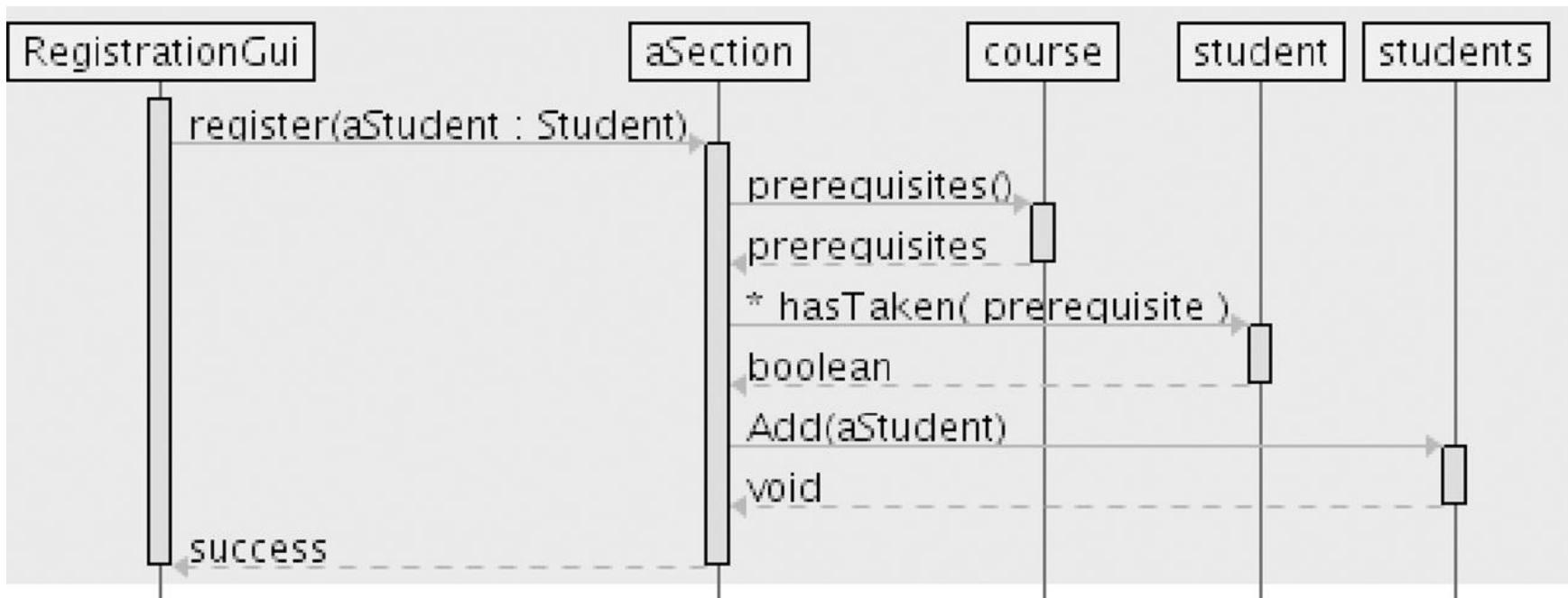
Diagrami i gjendjeve në UML

Tregon “statusin” e studentit në universitet



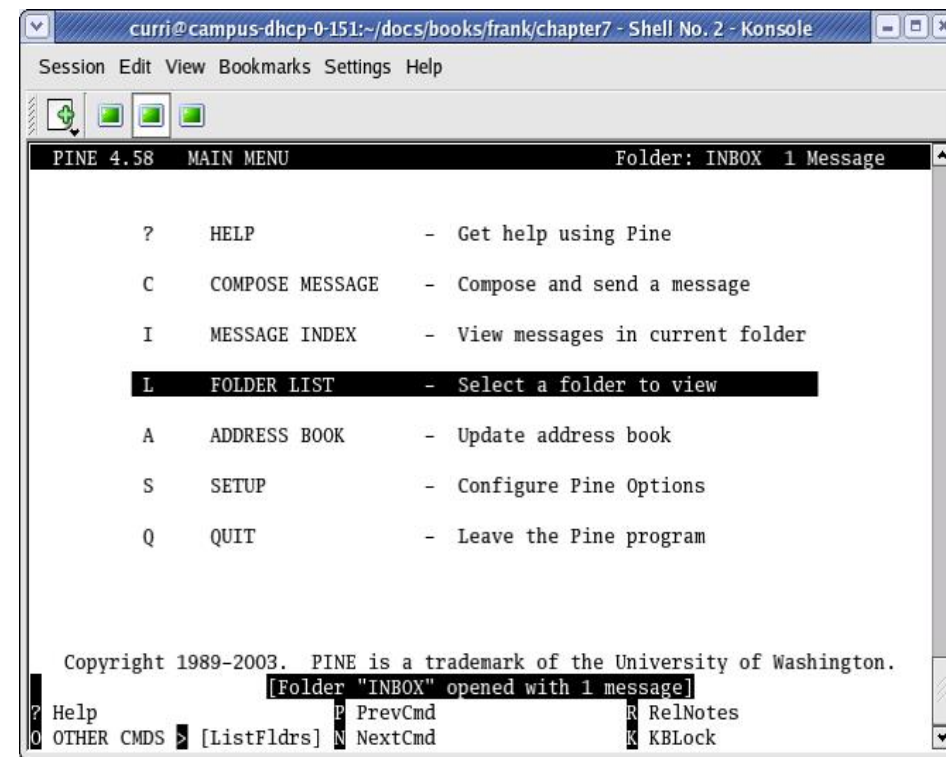
UML “Sequence Diagram”

Përdoret për të treguar rrjedhjen e interaksionit



Dizajni i “User Interface”

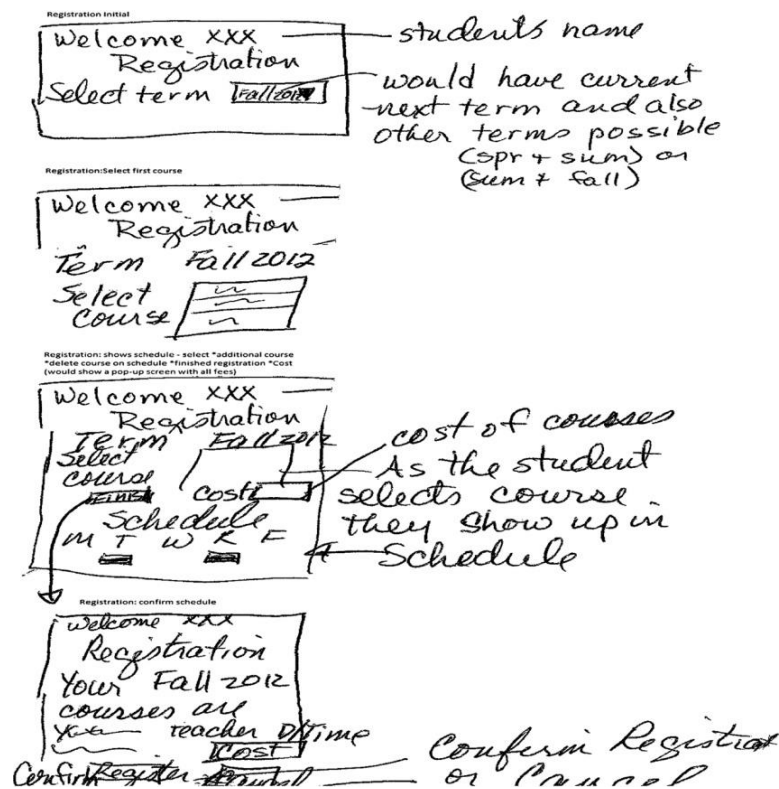
- Më i dukshmi për shfrytëzuesin
- Dy qështje kryesore:
 - Rrjedhja e interaksionit
 - Dukja
- Llojet e interfejsëve
 - Command-line
 - Text menus
 - Graphical (GUI)



Rrjedha e interaksioneve

Ekranet e prototipit

- Regjistrimi:
 - Zgjedhet semestri.
- Regjistrimi: paraqet semestrin.
 - Zgjedhet lënda e parë.
- Regjistrimi: tregon semestrin, lëndët me orar dhe koston.
 - Zgjedhet *Lëndët shitesë; *Fshirja e lëndëve; *përfundo regjistrimin.
- Regjistrimi: tregon orarin final.
 - Zgjedhet Konfirmo ose Anulo.



Prototipi “High Fidelity”

Welcome UserName

Registration

School term to register

Welcome aStudent

Registration

Desired School term to register - Spring 2012

Select course to add

Welcome aStudent

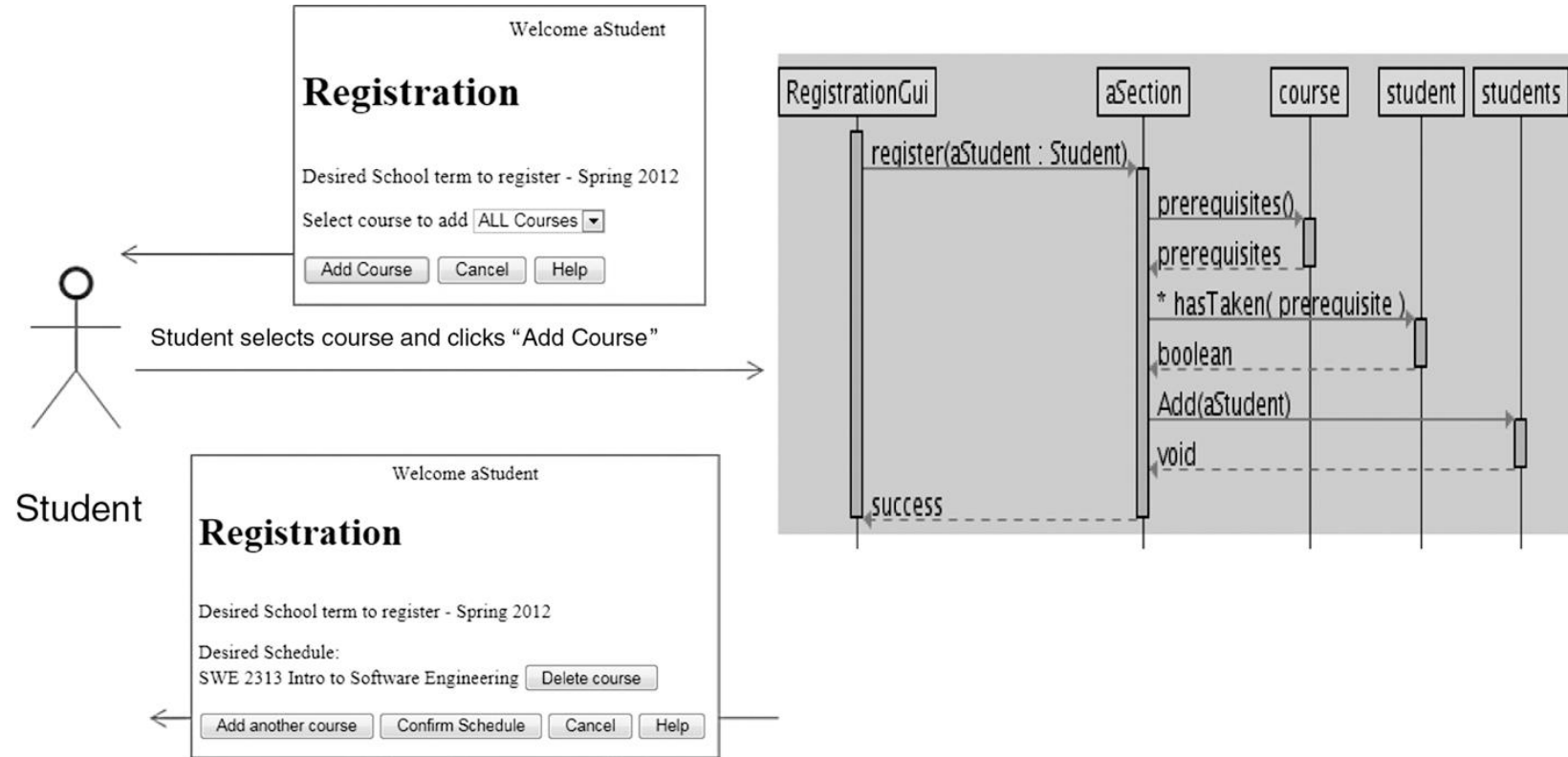
Registration

Desired School term to register - Spring 2012

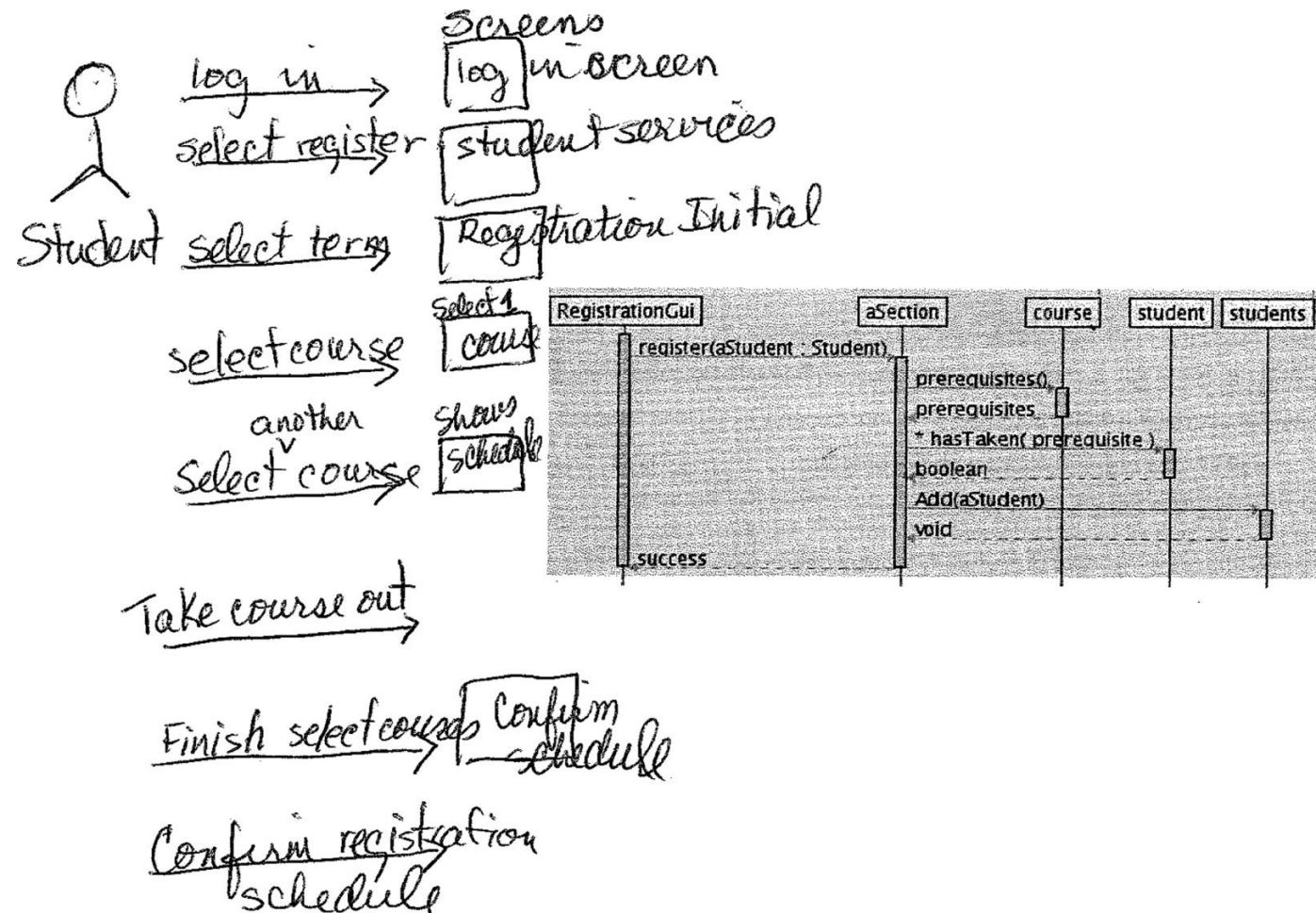
Desired Schedule:

SWE 2313 Intro to Software Engineering

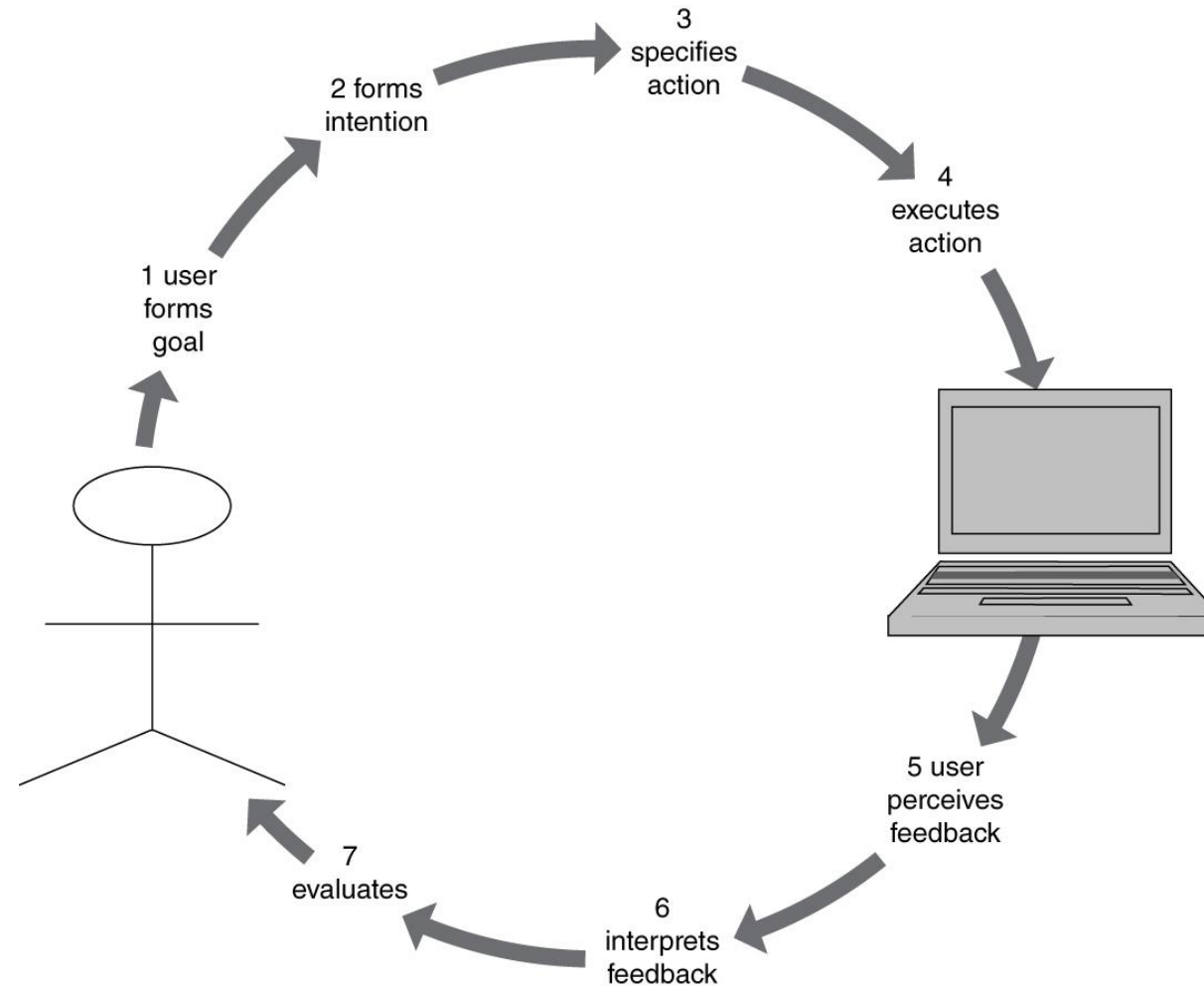
User: Screens: Process:



Interaksioni i shfrytëzuesit mbi diagramin e sekuencës



Modeli i Normanit me 7 faza



Modeli GOMS

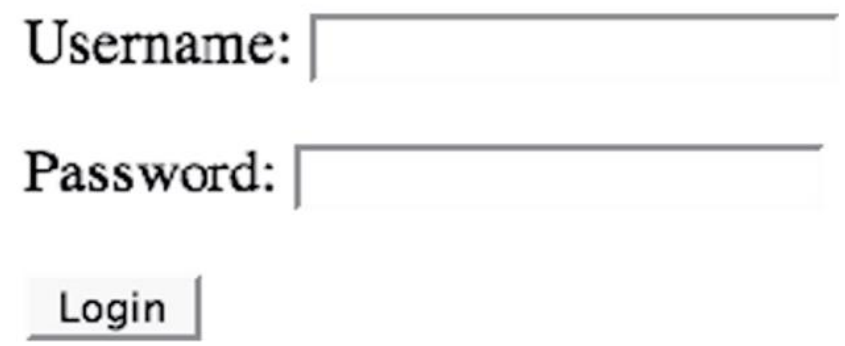
(temë e avansuar për UI)

- Konsiderohen lloje të ndryshme të shfrytëzuesëve.
- Katër faktorë (për llojin e shfrytëzuesit)
 - Goals of the user
 - Operations provided by the system
 - Methods or the sequence of operations
 - Selection rules for the methods

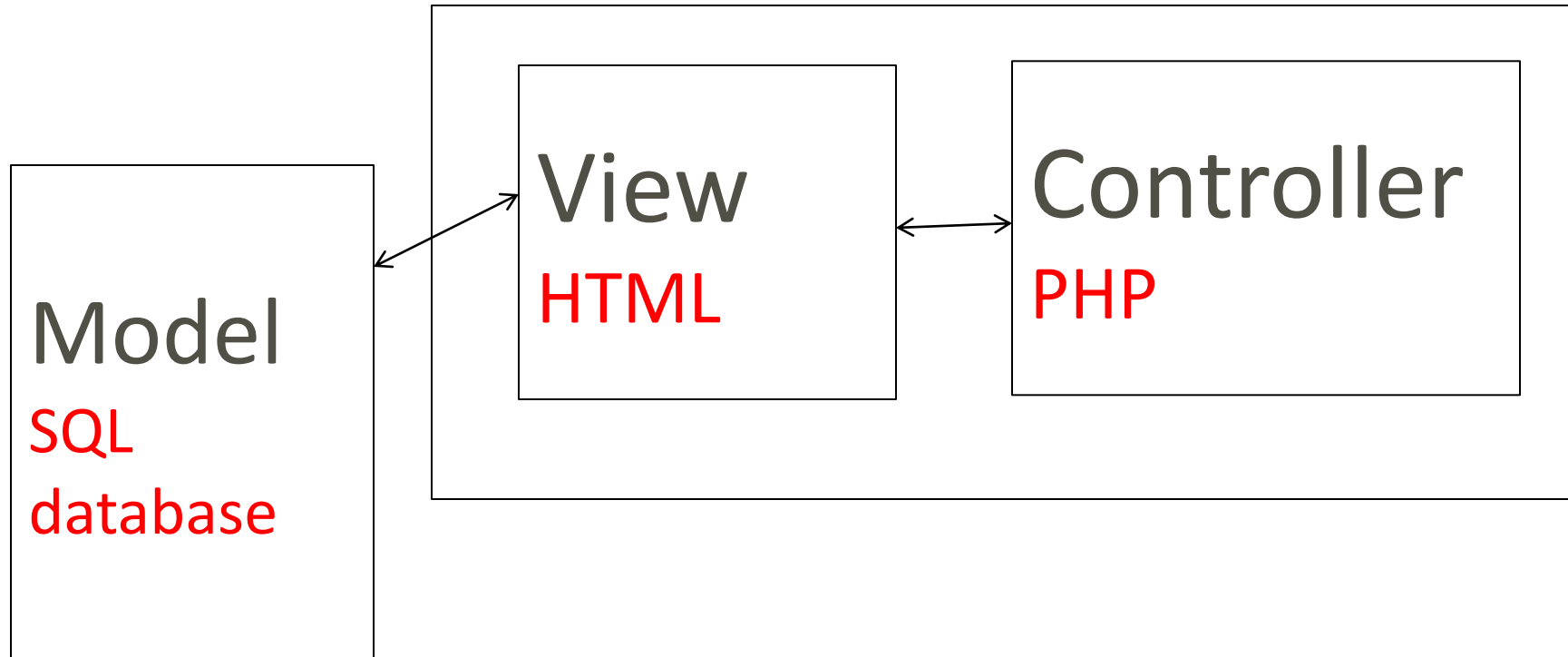
Qështje të tjera të “UI”

- Llojet e shfrytëzuesëve
- Heuristikat
- Udhëzuesit e “UI”
- Çështjet multikulturale
- Metaforat
- Softveri Multiplatform
- Akcesibiliteti
- Interfejsët multimedial

Shembull i thjesht i skriptës HTML

Sample HTML	Visual result (possible)
<pre> <form method="GET" action="something.php"> <p> Username: <input type="text" name="username"> </p> <p> Password: <input type="password" name="password"> </p> <input type="submit" value="Login"> </form> </pre>	

Model-View-Controller (MVC) Software Project



Pyetje???