

University for Business and Technology in Kosovo

## UBT Knowledge Center

---

Theses and Dissertations

Student Work

---

Spring 3-2021

### DOKUMENTIMI I ARKITEKTURËS SOFTUERIKE

Genita Abazi

*University for Business and Technology - UBT*

Follow this and additional works at: <https://knowledgecenter.ubt-uni.net/etd>



Part of the [Computer Sciences Commons](#)

---

#### Recommended Citation

Abazi, Genita, "DOKUMENTIMI I ARKITEKTURËS SOFTUERIKE" (2021). *Theses and Dissertations*. 2578.  
<https://knowledgecenter.ubt-uni.net/etd/2578>

This Thesis is brought to you for free and open access by the Student Work at UBT Knowledge Center. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of UBT Knowledge Center. For more information, please contact [knowledge.center@ubt-uni.net](mailto:knowledge.center@ubt-uni.net).



Programi për Shkenca Kompjuterike dhe Inxhinierise

**DOKUMENTIMI I ARKITEKTURËS SOFTUERIKE**  
Bachelor

Genita Abazi

Mars / 2021  
Prishtinë



Programi për Shkenca Kompjuterike dhe Inxhinierise

Punim Diplome  
Viti akademik 2014 – 2015

Genita Abazi

**DOKUMENTIMI I ARKITEKTURËS SOFTUERIKE**

Mentori: PhD. Blerton Abazi

Mars / 2021

Ky punim është përpiluar dhe dorëzuar në përmbushjen e kërkesave të  
pjeshme për Shkallën Bachelor

## **ABSTRAKT**

Dokumentimi ka luajtur një rol kyç në të ndihmuarit e kuptimit të softuerit për një kohë të gjatë. Dokumentimi softuerik mund të shprehet në forma të ndryshme, varësisht nga audienca dhe natyra e softuerit. Mirëpo jo gjithmonë dokumentimi e ilustron operimin e softuerit përkatës në mënyrë të saktë, si rezultat i së cilës softueri ka potencial të dështimit brenda një kohe shumë të shkurtër.

Ky punim diplome adreson çështjet lidhur me rëndësinë e posedimit të dokumentimit të softuerit, jep një prezantim të shkurtër të standardeve të dokumentimit, të cilat ofrojnë një hyrje dhe udhëzime të përgjithshme për dokumentim të mirëfilltë të softuerit përkatës. Tema fokus ka shtjellimin e dokumentimit të arkitekturës softuerike në veçanti, ku përveç që diskuton dhe ofron udhëzime për dokumentim adekuat të një arkitekture, do të analizojë edhe një shembull përkatës të një arkitekture softuerike në përputhshmëri me udhëzimet dhe njohuritë e fituara nga materiali i ofruar.

Është e vështirë të neglizhohet fakti që dokumentimi i dobët softuerik përbën një problem të madh në industrinë ekzistuese, por ballafaqimi me problemin në mundësinë më të hershme dhe njohuria adekuate për dokumentim të dobishëm do të mundësonin një zgjidhje graduale dhe efektive në këtë drejtim.

## **MIRËNJOHJE/FALENDERIME**

Dëshiroj që këtë seksion t'ja kushtoj profesorëve, kolegëve, miqve, e familjes, të cilëve i kam borxh kontributin, mbështetjen, e dashurinë e pamatshme. Falenderoj posaçërisht profesorin Prof. Dr. Blerton Abazi për udhëzimin e mbështetjen jo vetëm gjatë procesit të finalizimit të këtij punimi, por edhe gjatë studimeve. Falenderoj gjithashtu partnerin për këshillat dhe prezencën e tij gjatë kësaj kohe. Së fundmi, falenderimi më i ëmbël i dedikohet familjes sime e cila nuk hezitoi dhe nuk u kursya në asnjë nevojë timen në çdo hap të rrugës.

# PËRMBAJTJA

ABSTRAKT .....	I
MIRËNJOHJE/FALENDERIME .....	II
LISTA E FIGURAVE.....	IV
FJALORI I TERMAVE.....	V
1. HYRJE.....	1
2. SHQYRTIMI I LITERATURËS.....	2
2.1. STANDARDET E DOKUMENTIMIT .....	3
2.1.1. STANDARDET E PROCESIT .....	4
2.1.2. STANDARDET E PRODUKTIT.....	4
2.1.3. STANDARDET E SHKËMBIMIT .....	5
2.1.4. STANDARDI IEEE PËR DOKUMENTIMIN E PËRDORUESIT .....	6
2.2. DOKUMENTIMI I ARKITEKTURËS SOFTUERIKE.....	7
2.2.1. PËRDORIMI DHE AUDIENCAT E DOKUMENTIMIT TË ARKITEKTURËS..	8
2.2.2. SHËNIMET PËR DOKUMENTIMIN E ARKITEKTURËS .....	9
2.2.3. PAMJET .....	10
2.2.4. ZGJEDHJA E PAMJEVE.....	22
2.2.5. KOMBINIMI I PAMJEVE.....	24
2.2.6. NDËRTIMI I PAKETËS SË DOKUMENTIMIT .....	26
2.2.7. SJELLJA E DOKUMENTIMIT .....	32
3. DEKLARIMI I PROBLEMIT.....	35
4. METODOLOGJIA .....	38
5. REZULTATET.....	51
6. DISKUTIME DHE PËRFUNDIME .....	52
7. REFERENCAT .....	53

## LISTA E FIGURAVE

Figura 1. Përmbledhje e pamjeve të moduleve [12] .....	12
Figura 2. Përmbledhje e pamjeve C&C [12].....	17
Figura 3. Një përfaqësim UML i porteve në një tip përbërës C&C (majtas) dhe instance përbërësi (djathtas). Lloji i përbërësit të Bazës së të Dhënave të Llogarisë ka dy lloje portesh, Server dhe Admin (shënuar nga kutitë në kufirin e përbërësit) Porta e serverit përcaktohet me një shumëllojshmëri, që do të thotë se instancat e shumëfishta të portit lejohen në cilëndo instancë të komponentit përkatës [12] .....	18
Figura 4. Përmbledhje e karakteristikave të pamjeve të alokimit [12].....	20
Figura 5. Një pamje dekompozimi e mbivendosur me informacionin "përdor", që të krijojë një mbivendosje zërthimi/përdorimi [12] .....	25
Figura 6. Shabllon i pamjes.....	26
Figura 7. Përmbledhje e dokumentimit përtej pamjeve [12].....	29
Figura 8. Një shembull i thjeshtë i një UML diagrami sekuece.....	33
Figura 9. Rastet e rëndësishme të përdorimit nga ana Arkitekturale .....	40
Figura 10. Pasqyra Arkitekturale .....	442
Figura 11. Diagrami i procesit .....	464
Figura 12. Elementet nga procesi në dizajn .....	486
Figura 13. Modeli i procesit te dependencies e modelit të dizajnit.....	47
Figura 14. Proceset deri te implementimi .....	498
Figura 15. Pamja e vendosjes.....	49

## **FJALORI I TERMAVE**

ADL – Architecture Description Language

C&C – Component and Connector

DBMS – Database Management System

IEEE – Institute of Electrical and Electronics Engineers

IT – Information Technology

OODBMS – Object-Oriented Database Management System

PDF – Portable Document Format

SOA – Service-Oriented Architecture

UML – Unified Modeling Language



## 1. HYRJE

Dokumentimi i softuerit paraqet një shkrim formal në formë të shtypur dhe elektronike, i cili për bazë ka përdorimin efektiv dhe efikas të atij softueri në mjedisin që ka synuar [1]. Dokumentimi i softuerit konsiderohet si një pjesë integrale e procesit të zhvillimit softuerik [2] dhe ka një numër përdorimesh në ciklin e jetës së softuerit (si p.sh., mund të shërbejë si një medium komunikimi në mes të palëve të interesit, si depo e informacioneve për mirëmbajtësit, dhe si udhëzues për përdoruesit e softuerit) [3]. Esenca e procesit të zhvillimit të softuerit është koordinimi dhe komunikimi i individëve drejt arritjes së objektivave të përbashkëta me qëllim të prodhimit të një softueri funksional [4]. Dokumentimi softuerik në anën tjetër, siguron në mënyrë asinkrone komunikimin në mes të palëve të interesit, një faktor ky i cili ka potencialin e reduktimit të kufizimeve gjeografike dhe kohore gjatë procesit të zhvillimit të softuerit. Përmirësimi i cilësisë së dokumentimit të softuerit do të rezultojë në përputhshmëri me përmirësimin e cilësisë së softuerit [5].

“Dokumentimi është një nga pjesët më kryesore të një projekti softuerik. Megjithatë, më shumë se gjysma e këtyre projekteve kanë pak ose nuk kanë fare dokumentim për t’ju dalë në ndihmë përdoruesve potencial në mënyrë që të përdorin softuerin” thotë Eric Holscher, nënthemeluesi i Write the Docs [6], një komunitet global i individëve që gjatë procesit të zhvillimit softuerik të tyre, vendosin të mos t’a lënë dokumentimin anash.

Në vijim, ndër të tjera, fillimisht do të diskutohen standardet e dokumentimit të softuerit, ku do të familjarizohemi me hapat e parë të këtij procesi, dhe më pas do të shtjellohen metodologjitë e aplikuara në dokumentim të arkitekturës softuerike.

## 2. SHQYRTIMI I LITERATURËS

Të gjitha projektet e mëdha të zhvillimit softuerik, pavarësisht nga aplikimi, gjenerojnë një sasi të madhe të dokumentimit shoqërues. Për sistemet me madhësi mesatare, dokumentimi me shumë gjasa do të mbushë disa dollapë regjistrimi, ndërsa për sisteme të mëdha, mund të shkojë aq larg sa të mbushen dhoma të tëra. Për më tepër, gabimet dhe lëshimet e bëra në dokumentim mund të çojnë në gabime nga vetë përdoruesit dhe si pasojë i gjithë sistemi mund të ketë potencial dështimi nga kostot dhe përçarjet që zhvillohen gjatë dokumentimit. Prandaj, menagjerët dhe zhvilluesit e sistemit duhet t'i kushtojnë rëndësi të njëjtë dokumentimit dhe kostos përcjellëse, ashtu sikur zhvillimit të sistemit [7].

Dokumentimet shoqëruese të një projekti softuerik dhe sistemi në zhvillim e sipër kanë disa kërkesa shoqëruese:

1. Ato duhet të veprojnë si një medium komunikimi në mes të anëtarëve të ekipit të zhvillimit
2. Ato duhet të shërbejnë si depo të informacionit të sistemit që do të përdoren nga inxhinierët e mirëmbajtjes
3. Ato duhet t'i ofrojnë informacion menagjmentit në mënyrë që t'jau lehtësojnë planifikimin, buxhetin, dhe orarin e procesit të zhvillimit të softuerit
4. Disa nga dokumentimet duhet të udhëzojnë përdoruesit se si të përdorin dhe administrojnë sistemin

Plotësimi i këtyre kërkesave kërkon lloje të ndryshme të dokumenteve, nga lloji i dokumenteve jozyrtare deri te lloji i manualeve profesionale. Zhvilluesit e sistemit janë ata të cilët zakonisht janë përgjegjës për prodhimin e shumicës së këtij dokumentimi, edhe pse profesionistët teknik mund të ndihmojnë në finalizimin e deri te lëshimi i informacionit [8].

Qëllimet e këtij shkrimi janë të përshkruhet dokumentimi gjatë procesit të zhvillimit softuerik, të jipen disa sugjerime në atë se si të shkruhet dokumentim efektiv, dhe të përshkruhen proceset e përfshira në prodhimin e këtyre dokumentimeve. Filloj së pari me diskutimin e llojeve të ndryshme të dokumentimit që mund të shfletohen për një projekt softuerik. Pastaj mbuloj temën e rëndësishme të standardeve të dokumentimit.

## 2.1. STANDARDET E DOKUMENTIMIT

Standardet e dokumentimit veprojnë si bazë për sigurimin e cilësisë së dokumenteve [3]. Anëtarë të komunitetit softuerik dhe inxhinierik të IT-së mbajnë shumë keqkuptime në lidhje me standardet, veçanërisht me standardet e procesit. Më të zakonshmet janë ato që thonë se standardet janë të ngurta, të vjetëruara dhe të mërzitshme. Standardet perceptohen gjithashtu si antitezë e shkathhtësisë. Inxhinieria përfshin zgjedhjen e mjetit të duhur, që nënkupton një kuptim të dy mjeteve dhe problemit. Një sjellje e tillë është sjellja agile. Duke përdorur përcaktimin e Organizatës Ndërkombëtare të Standardizimit (ISO) si bazë, standardet janë "dokumentim udhëzues që pasqyron marrëveshje mbi produktet, praktikat ose operacionet nga shoqatat industriale, profesionale, tregtare ose organet qeveritare të njohura në shkallë kombëtare ose ndërkombëtare". Standardet referohen si dokumente udhëzuese sepse ato nuk janë të detyrueshme përveç nëse mandatohen kështu nga një individ, një organizatë ose treg. Ato janë marrëveshje sepse ato shpesh pasqyrojnë një nivel specifik të konsensusit [9]. Dokumentet e prodhuara sipas standardeve të duhura kanë një pamje, strukturë dhe cilësi të qëndrueshme. Megjithatë, nuk janë relevante vetëm standardet që fokus kanë dokumentimin. Ekzistojnë standarde të tjera të cilat poashtu mund të përdoren gjatë procesit të dokumentimit:

1. *Standardet e procesit* Këto standarde definojnë procesin i cili duhet të përcillet për prodhim të dokumentimit të cilësisë së lartë.
2. *Standardet e produktit* Këto janë standarde të cilat vetë rregullojnë dokumentet.
3. *Standardet e shkëmbimit* Është gjithnjë e më i rëndësishëm shkëmbimi i kopjeve të dokumenteve përmes postës elektronike dhe ruajtja e këtyre dokumenteve në bazat e të dhënave. Standardet e shkëmbimit bëjnë të mundur kompatibilitetin e të gjitha kopjeve elektronike të dokumenteve.

Standardet, nga natyra, janë të dizajnuara të mbulojnë të gjitha rastet dhe, si rezultat i kësaj, nganjëherë mund të duken kufizuese pa nevojë. Prandaj është e rëndësishme që, për çdo projekt, zgjidhen dhe modifikohen standardet e duhura për t'ju përshtatur projektit në fjalë. Projektet e vogla që zhvillojnë sisteme me jetëgjatësi të pritshme relativisht të vogël kanë nevojë për standarde të ndryshme prej projekteve të mëdha softuerike ku softueri pritet të mirëmbahet për 10 ose më shumë vite.

### **2.1.1. STANDARDET E PROCESIT**

Standardet e procesit përcaktojnë qasjen e përdorur gjatë prodhimit të dokumenteve. Kjo zakonisht nënkupton përcaktimin e mjeteve softuerike të cilat duhet të përdoren për prodhimin e dokumentimit dhe përcaktimin e procedurave të sigurimit të cilësisë që sigurojnë prodhimin e dokumenteve me cilësi të lartë.

Standardet e sigurimit të cilësisë së procesit të dokumentimit duhet të jenë të qëndrueshme dhe të jenë në gjendje të përballojnë të gjitha llojet e dokumenteve. Në rastet kur dokumentet janë thjesht dokumente pune ose memorandume, kontrollimi eksplicit i cilësisë nuk është i nevojshëm. Megjithatë, në rastet kur dokumentet janë dokumente zyrtare, më specifikisht, kur evolucioni i tyre duhet të kontrollohet nga procedurat e konfigurimit menagjerial, duhet të miratohet një proces zyrtar i cilësisë.

Hartimi, kontrollimi, rishikimi dhe ri-planifikimi është një proces përsëritës i cili duhet të vazhdojë derisa të prodhohet një dokument me cilësi të pranueshme. Niveli i cilësisë së pranueshme do të varet nga lloji i dokumentit dhe lexuesit potencialë të dokumentit.

### **2.1.2. STANDARDET E PRODUKTIT**

Standardet e produktit aplikohen në të gjitha dokumentet e prodhuara gjatë zhvillimit të softuerit. Dokumentet duhet të kenë një pamje të qëndrueshme dhe, dokumentet e klasës së njëjtë duhet të kenë një strukturë të qëndrueshme. Standardet e dokumentimit janë specifike për projekt por duhet të bazohen në standarde më të përgjithshme organizative.

Shembuj të standardeve të produktit të cilat duhet të zhvillohen janë:

1. *Standardet e identifikimit të dokumentit* Meqenëse projektet e mëdha zakonisht prodhojnë me mijëra dokumente, secili dokument duhet të identifikohet në mënyrë unike. Për dokumentet zyrtare, ky identifikues mund të jetë identifikuesi zyrtar i përcaktuar nga menagjeri i konfigurimit. Për dokumentet jo-zyrtare, stili i identifikuesit të dokumentit duhet të përcaktohet nga menagjeri i projektit.
2. *Standardet e strukturës së dokumentit* Ekziston një strukturë e përshtatshme për secilën klasë të dokumentit të prodhuar gjatë një projekti softuerik. Janë standardet e strukturës të cilat duhet të definojnë këtë organizim. Ato gjithashtu duhet të specifikojnë konventat e

përdorura për numërimin e faqeve, informacionin për header dhe footer të faqeve, dhe numërimin e seksioneve dhe nënseksioneve.

3. *Standardet e prezantimit të dokumentit* Standardet e prezantimit të dokumentit përcaktojnë një “stil shtëpie” për dokumentet dhe kontribuojnë në mënyrë të konsiderueshme në qëndrueshmërinë e dokumentit. Ato përfshijnë definimin e fonteve dhe stileve të përdorura në dokument, përdorimin e logove dhe emrave të kompanive, aplikimin e ngjyrave që theksojnë strukturën e dokumentit, etj.
4. *Standardet e përditësimit të dokumentit* Gjersa dokumenti ndryshon për të reflektuar ndryshimet në sistem, duhet të përdoret një mënyrë e qëndrueshme për të treguar këto ndryshime. Kjo mund të realizohet nga përdorimi i ngjyrave të ndryshme të kopertinës në mënyrë që të theksohet versioni i ri i dokumentit dhe përdorimi i shiritave të ndryshimit për të theksuar paragrafët e përditësuar ose të fshirë.

Standardet e dokumentimit duhet të zbatohen për të gjitha dokumentet e projektit dhe draftet fillestare të dokumentimit të përdoruesit. Megjithatë, në shumë raste, dokumentimi i përdoruesit duhet të paraqitet në një formë të përshtatshme të përdoruesit e jo të projekti dhe duhet të rindërtohet në atë formë gjatë procesit të prodhimit.

### **2.1.3. STANDARDET E SHKËMBIMIT**

Standardet e shkëmbimit të dokumentimit janë të rëndësishme pasi gjithnjë e më shumë dokumente po prodhohen në format elektronik sidhe ose në vend të letrës. Për dokumentimin që dorëzohet me sistem softuerik shumicën e kohës sot përdoret Adobe PDF. Sidoqoftë, kur dokumentet shkëmbehen nga ekipi i zhvillimit dhe gjejmë drafte që qarkullojnë brenda një organizate këto shihen shpesh në formatin e çfarëdo përpunuesi të tekstit (shpesh të atij Microsoft Word).

Duke supozuar se përdorimi i një përpunuesi standard të tekstit dhe sistemi të redaktimit grafik është i mandatuar në standardet e procesit, standardet e shkëmbimit përcaktojnë konventat për përdorimin e këtyre mjeteve. Përdorimi i standardeve të shkëmbimit lejon transferimin e dokumenteve në mënyrë elektronike dhe ri-krijimin në formën e tyre origjinale.

Standardet e shkëmbimit janë më shumë sesa thjesht një marrëveshje për përdorimin e një versioni të sistemit të përbashkët për prodhimin e dokumentit. Shembuj të standardeve të

shkëmbimit përfshijnë përdorimin e një seti standard macro të dakorduar në rastin kur një sistem i formatimit të tekstit është përdorur për prodhimin e dokumentit ose përdorimin e një fletë stili standard për një përpunues teksti. Standardet e shkëmbimit mund gjithashtu të kufizojnë fontet dhe stilet e tekstit të përdorura për shkak të aftësive të ndryshme të printerit dhe ekranit [3].

#### **2.1.4. STANDARDI IEEE PËR DOKUMENTIMIN E PËRDORUESIT**

Standardi i parë IEEE për dokumentimin e përdoruesit është prodhuar në vitin 1987. Ashtu si të gjitha standardet, edhe ky standard e përmbledh zgjuarsinë dhe përvojën rreth dokumentimit të softuerit dhe propozon një strukturë për dokumentimin e përdoruesit. Duke përdorur këtë strukturë si bazë, standardi diskuton përmbajtjen e dokumentimit të përdoruesit të softuerit dhe propozon standarde formatimi për këto dokumente.

*Dokumentimi duhet të sigurohet në media dhe formate që lejojnë përdorimin e tij nga individë me shikim, dëgjim ose kufizim tjetër fizik.*

*Një përshkrim se si të printohet dokumentimi elektronik duhet të jetë i përfshirë në të dy dokumentimet, atë elektronik dhe atë të printuar.*

*Për shkak se disa përdorues nuk mund të bëjnë dallimin në mes të ngjyrave, dokumentimi duhet të ofrojë sugjerime teksti në vend të ngjyrave si e kuqja dhe jeshilja si mënyra të vetme për të përcjellur kuptimin.*

*Paralajmërimet, njoftimet dhe shënimet duhet të paraqiten në një format të qëndrueshëm që dallohet lehtë nga teksti i zakonshëm ose hapat udhëzues.*

*Ilustrimet që shoqërojnë tekstin duhet të paraqiten afër referencës së tyre të parë në tekst ashtu që teksti dhe ilustrimi shoqëruet të mund të shihen në të njëjtën kohë.*

Ashtu si të gjitha standardet, edhe ky standard i dokumentimit duhet të përshtatet në situatën në të cilën është përdorur [10].

## 2.2. DOKUMENTIMI I ARKITEKTURËS SOFTUERIKE

Dokumentimi efektiv i njohurive arkitekturale është një nga faktorët kryesorë në shfrytëzimin e ndryshimit të paradigmës drejt ndarjes dhe ripërdorimit të tyre. Sidoqoftë, qasjet aktuale të dokumentimit kanë mangësi të mëdha në kapjen e njohurive të sistemeve të mëdha dhe komplekse dhe lehtësimin e përdorimit të tyre. [11] Edhe arkitektura më e mirë, më e përshtatshme për punën, do të jetë në thelb e padobishme nëse njerëzit që kanë nevojë t'a përdorin nuk e dinë se çfarë është; nuk mund t'a kuptojnë mjaftueshëm për t'a përdorur, ndërtuar ose modifikuar atë; ose (rasti më i keq i mundshëm) e keqkuptojnë dhe e aplikojnë gabimisht. Dhe i gjithë munda, analiza, dizajni dhe puna e palodhur nga ekipi i arkitekturës do të jenë tretur kot.

Krijimi i një arkitekture nuk mjafton. Duhet të komunikohet në atë mënyrë që t'i lejojë palët e interesit t'a përdorin atë siç duhet për të bërë punën e tyre. Nëse dëshironi të krijoni një arkitekturë të fortë, që t'i rezistojë kohës, atëherë duhet t'a përshkruani atë në detaje të mjaftueshme, pa paqartësi dhe t'a organizoni atë në mënyrë që të tjerët të gjejnë dhe përditësojnë informacionin shpejt.

E vërteta e hidhur sot është se dokumentimi arkitektural shpesh trajtohet si opsion i mëvonshëm, diçka që njerëzit e bëjnë sepse duhet bërë. Ndoshta e kërkon një kontratë. Ndoshta e kërkon një klient. Ndoshta e kërkon procesi standard i një kompanie. Të gjitha këto mund të jenë arsye legjitime. Por asnjë prej këtyre arsyeve nuk është aq bindëse sa të prodhojë dokumentim me cilësi të lartë. Pse duhet që arkitekti të shpenzojë kohë dhe energji të vlefshme vetëm që një menagjer të mund të kontrollojë një dërgesë? [12]

Arkitektët më të mirë prodhojnë dokumentim të mirë jo sepse është "i kërkuar" por sepse ata e shohin thelbësore çështjen në fjalë dhe prodhojnë një produkt me cilësi të lartë, në mënyrë të parashikueshme dhe me sa më pak ripunim të jetë e mundur. Ata i shohin njerëzit e përfshirë ngushtë në këtë ndërmarrje si palët e tyre të interesit: zhvilluesit, implementuesit, testerët dhe analistët.

Një skemë e mirë-menduar e dokumentimit mund të bëjë që procesi i dizajnit të shkojë shumë më normalisht dhe sistematikisht. Dokumentimi ndihmon arkitektin(ët) të arsyetojnë rreth modelit të arkitekturës dhe t'a komunikojnë atë derisa arkitektura është në proces, qoftë në një fazë gjashtëmujore të dizajnit apo në një sprint gjashtë-ditor Agile [13].

## 2.2.1. PËRDORIMI DHE AUDIENCAT E DOKUMENTIMIT TË ARKITEKTURËS

Dokumentimi i arkitekturës duhet të shërbejë për qëllime të ndryshme. Duhet të jetë transparent dhe i arritshëm për t'u kuptuar shpejt nga punonjësit e rinj. Duhet të jetë konkret që të shërbejë si bazë për ndërtim. Duhet të ketë informacion të mjaftueshëm për të shërbyer si bazë për analizë.

Dokumentimi i arkitekturës është përshkruar. Për disa audiencë, ajo përshkruan atë që duhet të jetë e vërtetë, duke vendosur kufizime në vendimet që duhet të merren. Për audiencat e tjera, ajo përshkruan atë që është e vërtetë, duke rrëfyer vendimet e marra në lidhje me modelin e një sistemi.

Dokumentimi më i mirë i arkitekturës sa i përket, të themi, analizës së performancës mund të jetë i ndryshëm nga dokumentimi më i mirë i arkitekturës që do të dëshironim t'i dorëzonim një implementuesi. Kur dizajnoni dhe rishikoni dokumentimin, duhet të siguroni mbështetje për të gjitha nevojat përkatëse.

Lloje të ndryshme të njerëzve do të kenë një interes të dhënë në një dokument arkitekture. Ata shpresojnë dhe presin që dokumenti i arkitekturës do t'i ndihmojë ata të bëjnë punën e tyre përkatëse. Të kuptuarit e qasjes së tyre në dokumentim të arkitekturës është thelbësor, pasi këto përdorime përcaktojnë informacionin e rëndësishëm për t'u kapur.

Në thelb, dokumentimi i arkitekturës ka tre përdorime:

1. *Dokumentimi i arkitekturës shërben si një mjet edukimi.* Përdorimi arsimor konsiston në prezantimin e njerëzve në sistem. Njerëzit mund të jenë anëtarë të rinj të ekipit, analistë të jashtëm, apo edhe një arkitekt i ri. Në shumë raste, personi "i ri" është klienti të cilit po i tregoni zgjidhjen tuaj për herë të parë, një prezantim që shpresoni të rezultojë në financim ose miratim të duhur.

2. *Dokumentimi i arkitekturës shërben si mjet kryesor për komunikimin në mes të palëve të interesit.* Përdorimi i saktë i një arkitekture si mjet komunikimi varet nga cila palë e interesit po bën komunikimin.

3. *Dokumentimi i arkitekturës shërben si bazë për analizën dhe ndërtimin e sistemit.* Arkitektura u tregon implementuesve se çfarë të implementojnë. Secili modul ka ndërfaqe që duhet të sigurohet dhe të përdorë ndërfaqe nga modulet e tjera. Kjo jo vetëm që ofron udhëzime në lidhje me ndërfaqet e ofruara dhe të përdorura, por përcakton gjithashtu se me cilat skuadra të tjera duhet të komunikojë ekipi i zhvillimit për modulin.



Gjatë zhvillimit, një arkitekturë mund të jetë shumë komplekse, me shumë çështje që mbesin për t'u zgjidhur. Dokumentimi mund të shërbejë si një mjet për regjistrimin dhe komunikimin e këtyre çështjeve që përndryshe mund të lihen anash.

Për ata që janë të interesuar në aftësinë e dizajnit për të përmbushur objektivat e cilësisë së sistemit, dokumentimi i arkitekturës shërben si foragjere për vlerësim. Ai duhet të përmbajë informacionin e nevojshëm për të vlerësuar një larmi atributesh, të tilla si siguria, performanca, përdorshmëria, disponueshmëria, dhe modifikueshmëria.

Për ndërtuesit e sistemit që përdorin mjete automatike për gjenerim të kodit, dokumentimi mund të përfshijë modelet e përdorura për atë gjenerim. Këto modele ofrojnë udhëzime për ata që dëshirojnë të kuptojnë sjelljen e modulit në më shumë detaje se sa parashihet të dokumentohet normalisht, por në më pak detaje sesa do të ofronte ekzaminimi i kodit [12].

## 2.2.2. SHËNIMET PËR DOKUMENTIMIN E ARKITEKTURËS

Shënimet për dokumentimin e views ndryshojnë shumë në shkallën e tyre të formalitetit. Përafërsisht, ekzistojnë tre kategori kryesore të shënimit:

- *Shënimet jo-formale.* Views përshkruhen (shpesh grafikisht) duke përdorur diagramim me qëllim të përgjithshëm dhe mjete të redaktimit e konventa vizuale të zgjedhura për sistemin në fjalë. Semantikën e përshkrimit karakterizohen në gjuhë natyrore dhe ato nuk mund të analizohen zyrtarisht. Mjeti më i zakonshëm për shënimet informale është Power Point.

- *Shënimet gjysmë-formale.* Views shprehen në një shënim të standardizuar që përshkruan elementet grafike dhe rregullat e ndërtimit, por nuk ofron një trajtim të plotë semantik të kuptimit të atyre elementeve. Analiza rudimentare mund të zbatohet për të përcaktuar nëse një përshkrim plotëson vetitë sintaksore. UML është një shënim gjysmë-formal në këtë kuptim.

- *Shënimet formale.* Views përshkruhen në një shënim që ka një saktësi (zakonisht të bazuar në matematikë) semantike. Këtu është e mundur analiza formale e sintaksës dhe semantikës. Ka në dispozicion një larmi të shënimeve zyrtare për arkitekturën e softuerit. Në përgjithësi të referuara si gjuhët e përshkrimit të arkitekturës (ADL), ato zakonisht ofrojnë një fjalor grafik dhe një themel semantikësor për përfaqësimin e arkitekturës. Në disa raste këto shënime janë të specializuara për views të veçanta arkitekturale. Në të tjera ato lejojnë shumë views, ose madje sigurojnë aftësinë për të përcaktuar zyrtarisht views të reja. Dobësia e ADL qëndron në aftësinë e

tyre për të mbështetur automatizimin përmes mjeteve shoqëruese: automatizimi për të siguruar një analizë të dobishme të arkitekturës ose të ndihmojë në gjenerimin e kodit. Në praktikë, përdorimi i shënimeve të tilla është i rrallë.

Përcaktimi se cila formë e shënimit duhet të përdoret përfshin disa shkëmbime. Në mënyrë tipike, shënimet zyrtare kërkojnë më shumë kohë dhe përpjekje për t'u krijuar dhe kuptuar, por e shlyejnë këtë mund përmes krijimit të zvogëluar të paqartësisë dhe më shumë mundësive për analizë. Nga ana tjetër, shënimet jo-formale janë më të lehta për t'u krijuar, por ato ofrojnë më pak garanci.

Pavarësisht nivelit të formalitetit, gjithmonë mbani mend se shënimet e ndryshme janë më të mira (ose më të këqija) për shprehjen e llojeve të ndryshme të informacionit. Formaliteti mënjanë, asnjë UML klasë diagram nuk do t'ju ndihmojë në arsyetim në lidhje me planifikimin, dhe asnjë tabelë sekuençash nuk do t'ju tregojë shumë për gjasat e dorëzimit të sistemit në kohë. Ju duhet të zgjedhni shënimet dhe gjuhët tuaja të përfaqësimit gjithmonë duke pasur parasysh çështjet e rëndësishme që duhet t'i ngërtheni dhe arsyetoni [13].

### **2.2.3. PAMJET**

Ndoshta koncepti më i rëndësishëm në lidhje me dokumentimin e arkitekturës së softuerit është ai i views [12]. Një arkitekturë softuerike është një entitet kompleks që nuk mund të përshkruhet në një formë të thjeshtë një-dimensionale. Një pamje është një përfaqësim i një sërë elementesh të sistemit dhe marrëdhënieve midis tyre – jo i të gjitha elementeve të sistemit – por ato të një lloji të veçantë. Për shembull, një pamje e shtresuar e një sistemi do të tregonte elemente të tipit "layer", domethënë, do të tregonte zbërthimin e sistemit në shtresa dhe marrëdhëniet midis atyre shtresave. Një pamje e pastër e shtresuar nuk do të tregonte shërbimet e sistemit, ose klientët dhe serverat, ose modelin e të dhënave, ose ndonjë element tjetër.

Kështu, pamjet na mundësojnë të ndajmë entitetin shumë-dimENSIONAL që është një arkitekturë softuerike në një numër interesant dhe të menagjueshëm të përfaqësimeve të sistemit. Koncepti i pamjeve na jep parimin më thelbësor të dokumentimit të arkitekturës:

Dokumentimi i një arkitekture është çështje e dokumentimit të pamjeve relevante dhe pastaj shtimin e dokumentimit që aplikohet për më shumë se një pamje.

Cilat janë pamjet përkatëse? Kjo varet tërësisht nga qëllimet tuaja. Siç e pamë më parë, dokumentimi i arkitekturës mund të shërbejë për shumë qëllime: deklaratë misioni për implementuesit, bazë për analiza, specifikim për gjenerim automatik të kodit, pika fillestare për të kuptuar sistemin dhe rikuperimin e asetëve, ose model për planifikim të projektit.

Pamje të ndryshme ekspozojnë attribute të ndryshme të cilësisë në shkallë të ndryshme. Prandaj, atributet e cilësisë që ju shqetësojnë më së shumti juve dhe palëve të tjera të interesit në zhvillimin e sistemit ndikojnë në zgjedhjen e pamjeve për të dokumentuar. Për shembull, një pamje e shtresuar do t'ju lejojë të arsyetoni transportueshmërinë e sistemit, një pamje e vendosjes do t'ju lejojë të arsyetoni për besueshmërinë dhe performancën e sistemit tuaj, e kështu me radhë.

Pamje të ndryshme mbështesin qëllime dhe përdorime të ndryshme. Pamjet që duhet të dokumentoni varen nga përdorimet që prisni të ofrojë dokumentimi. Pamje të ndryshme do të nxjerrin në pah elemente dhe marrëdhënie të ndryshme të sistemit. Çdo pamje ka një kosto dhe një përfitim, dhe duhet të siguroheni që përfitimet nga mbajtja e një pamjeje të veçantë janë më të mëdha se kostot e saj. Pamjet mund të nxiten nga nevoja për të dokumentuar një stil të veçantë në modelin tuaj. Disa modele janë të përbëra nga module, të tjerat nga components dhe connectors.

## **PAMJET E MODULIT**

Një modul është një njësi e plotë, që ofron një grup koherent të përgjegjësive. Një modul mund të marrë formën e një klase, një koleksioni të klasave, një shtrese, një aspekti, ose të ndonjë dekompozimi të njësisë së zbatimit. Shembuj të modulit janë dekompozimi, përdorimet dhe shtresat. Çdo modul paraqet një grupim të vetive që i caktohen. Këto veti kanë për qëllim të shprehin informacion të rëndësishëm që i shoqërohet modulit, sidhe kufizimet e modulit. Karakteristikat mostër janë përgjegjësitë, informacioni i dukshmërisë dhe historiku i rishikimeve. Marrëdhëniet që kanë modulet me njëri tjetrin përfshijnë *është pjesë e*, *varet*, dhe *është një*.

Mënyra në të cilën softueri i një sistemi zërthehet në njësi të menagjueshme mbetet njëra nga format e rëndësishme të strukturës së sistemit. Kjo përcakton se si kodi burimor i një sistemi zërthehet në njësi, çfarë lloj supozimesh mund të bëjë secila njësi në lidhje me shërbimet e ofruara nga një tjetër, dhe se si ato njësi grumbullohen në ansamble më të mëdha. Kjo gjithashtu përfshin strukturat globale të të dhënave që ndikojnë dhe ndikohen nga njësi të shumta. Strukturat e moduleve shpesh përcaktojnë se si ndryshimet e një pjese të sistemit mund të ndikojnë në pjesë të

tjera dhe kështu, kjo shpie te aftësia e një sistemi për të mbështetur modifikimin, transportueshmërinë, dhe ripërdorimin.

Nuk ka gjasa që dokumentimi i një arkitekture të softuerit të jetë i plotë pa të paktën një pamje të modulit.

Tabela në vazhdim përmbledh elementet, marrëdhëniet, kufizimet dhe qëllimin e pamjeve të modulit në përgjithësi:

Elements	Modules, which are implementation units of software that provide a coherent set of responsibilities.
Relations	<ul style="list-style-type: none"> <li>▪ <i>Is part of</i>, which defines a part/whole relationship between the submodule—the part—and the aggregate module—the whole.</li> <li>▪ <i>Depends on</i>, which defines a dependency relationship between two modules. Specific module views elaborate what dependency is meant.</li> <li>▪ <i>Is a</i>, which defines a generalization/specialization relationship between a more specific module—the child—and a more general module—the parent.</li> </ul>
Constraints	Different module views may impose specific topological constraints, such as limitations on the visibility between modules.
Usage	<ul style="list-style-type: none"> <li>▪ Blueprint for construction of the code</li> <li>▪ Change-impact analysis</li> <li>▪ Planning incremental development</li> <li>▪ Requirements traceability analysis</li> <li>▪ Communicating the functionality of a system and the structure of its code base</li> <li>▪ Supporting the definition of work assignments, implementation schedules, and budget information</li> <li>▪ Showing the structure of information that the system needs to manage</li> </ul>

Figura 1. Përmbledhje e pamjeve të moduleve [12]

Karakteristikat e moduleve që ndihmojnë në udhëzimin e implementimit ose kontribuojnë në analiza duhet të regjistrohen si pjesë të dokumentimit mbështetës për një pamje të modulit. Lista e karakteristikave mund të ndryshojë, por mund të përfshijë:

- *Emri*. Emri i një moduli është, sigurisht, mjete kryesor për t'iu referuar atij. Emri i një moduli shpesh sugjeron diçka në lidhje me rolin e tij në sistem. Përveç kësaj, emri i një moduli

mund të pasqyrojë pozicionin e tij në hierarkinë e dekompozimit; emri A.B.C, për shembull, i referohet një moduli C që është një nën-modul i një moduli B, vetë një nën-modul i A.

- *Përgjegjësitë*. Prona e përgjegjësisë për një modul është një mënyrë për të identifikuar rolin e tij në sistemin e përgjithshëm dhe krijon një identitet për të përtej emrit. Përderisa emri i një moduli mund të sugjerojë rolin e tij, një deklaratë e përgjegjësisë e përcakton atë me shumë më shumë siguri. Përgjegjësitë duhet të përshkruhen në detaje të mjaftueshme për t'ju bërë të qartë lexuesve se çfarë bën secili modul.

- *Dukshmëria e ndërfaqes(eve)*. Kur një modul ka nën module, disa ndërfaqe të nënmoduleve janë publike dhe disa mund të jenë private; domethënë ndërfaqet përdoren vetëm nga nënmodulet brenda modulit prind. Ndërfaqet private nuk janë të dukshme jashtë këtij konteksti.

- *Informacioni për implementimin*. Modulet janë njësi të implementimit. Prandaj është e dobishme të regjistroni informacionin në lidhje me zbatimin e tij nga këndvështrimi i menagjimit të zhvillimit të tij dhe ndërtimin e sistemit që i përmban ato. Kjo mund të përfshijë:

- *Hartimi në njësi të kodit burimor*. Kjo identifikon skedarët që përbëjnë zbatimin e një moduli. Për shembull, një modul Account, nëse implementohet në Java, mund të ketë disa skedarë që përbëjnë zbatimin e tij: IAccount.java (një ndërfaqe), Accountimpl.java (zbatimi i funksionalitetit të llogarisë), AccountBean.java (një klasë për të mbajtur gjendjen e një llogarie në memorie), AccountOrmMapping.xml (një skedar që përcakton hartën ndërmjet AccountBean dhe një table të bazës së të dhënave—mapimi relacional i objektit), dhe ndoshta edhe një test i njësisë AccountTest.java.

- *Informacioni i provës*. Plani i provës së modulit, rastet e provës, skelat e provës dhe të dhënat e provës janë të rëndësishme për t'u dokumentuar. Ky informacion mund të jetë thjesht një tregues për vendndodhjen e këtyre objekteve.

- *Informacioni i menagjimit*. Një menagjer mund të ketë nevojë për informacion në lidhje me parashikimin e orarit dhe buxhetit të modulit. Ky informacion mund të jetë thjesht një tregues për vendndodhjen e këtyre objekteve.

- *Kufizimet e implementimit*. Në shumë raste, arkitekti do të ketë në mendje një strategji implementimi për një modul ose mund të dijë për kufizimet që ai implementim duhet të ndjekë.

- *Historiku i rishikimeve*. Njohja e historisë së një moduli duke përfshirë autorët dhe ndryshimet specifike mund të ndihmojë kur kryeni aktivitete mirëmbajtjeje.

Për shkak se modulet e ndajnë sistemin, duhet të jetë e mundur të përcaktohet se si kërkesat funksionale të një sistemi mbështeten nga përgjegjësitë e modulit. Pamjet e moduleve që tregojnë varësitë midis moduleve ose shtresave (të cilat janë grupe modulesh që kanë një model specifik përdorimi të lejuar) sigurojnë një bazë të mirë për analizën e ndikimit në ndryshim. Modulet modifikohen në mënyrë tipike si rezultat i raporteve të problemeve ose kërkesave të ndryshimit. Analiza e ndikimit kërkon një shkallë të caktuar të kompletimit të dizajnit dhe integritetin e përshkrimit të modulit. Në veçanti, informacioni i varësisë duhet të jetë në dispozicion dhe korrekt për të qenë në gjendje të krijojë rezultate të dobishme.

Një pamje e modulit mund të përdoret për t'i shpjeguar funksionalitetin e sistemit dikujt që nuk e njeh atë. Nivelet e ndryshme të grincimit të dekompozimit të modulit ofrojnë një prezantim nga lart-poshtë të përgjegjësisë të sistemit dhe për këtë arsye mund të drejtojnë procesin e të nxënimit. Për një sistem, implementimi i të cilit është tashmë i vendosur, pamjet e moduleve, nëse janë të azhuruara, janë të dobishme, sepse i shpjegojnë strukturën e bazës së kodit një zhvilluesi të ri në ekip. Kështu, pamjet e azhuruara të modulit mund të thjeshtësojnë dhe rregullojnë mirëmbajtjen e sistemit. Nga ana tjetër, është e vështirë të përdorësh pamjet e modulit për të bërë përfundime në lidhje me sjelljen e kohës së ekzekutimit, sepse këto pamje janë vetëm një ndarje statike e funksioneve të softuerit. Kështu, një pamje e modulit nuk përdoret zakonisht për të analizuar performancën, besueshmërinë dhe shumë cilësi të tjera të kohës së ekzekutimit. Për këto, ne mbështetemi në pamjet përbërëse sidhe lidhëse dhe të shpërndarjes.

Pamjet e moduleve zakonisht shënohen në pamjet përbërëse dhe lidhëse. Njësitë e zbatimit të paraqitura në pamjet e modulit kanë një hartëzim të përbërësve që ekzekutohen gjatë kohës së ekzekutimit. Për më tepër, një modul i vetëm do të replikohet si pjesë e shumë komponentëve të kohës së ekzekutimit, dhe një komponent i caktuar mund të hartojë në disa module. Pamjet e modulit ofrojnë gjithashtu elementet e softuerit që janë hartuar në shumëllojshmëri elementesh jo-softuerike të mjedisit të sistemit në pamje të ndryshme të alokimit [13].

## **PAMJET E PËRBËRËSIT DHE TË LIDHËSIT**

Pamjet e përbërësit dhe të lidhësit tregojnë elemente që kanë një prezencë të kohës së ekzekutimit, të tilla si proceset, objekte, klientë, serverë dhe dyqane të dhënash. Këto elemente janë të emërtuar si përbërës. Për më tepër, pamjet përbërëse dhe lidhëse përfshijnë si element rrugët e ndërveprimit, të tilla si lidhjet dhe protokollet e komunikimit, rrjedhat e informacionit dhe qasjen

në hapësirën e përbashkët. Pamjet e mostrave C&C janë arkitektura e orientuar drejt shërbimit (SOA), arkitektura server-klient ose pamjet e procesit të komunikimit.

Komponentët kanë ndërfaqe të quajtura porte. Një port përcakton një pikë të bashkëveprimit të mundshëm të një përbërësi me mjedisin e tij. Një port zakonisht ka një tip të qartë, i cili përcakton llojin e sjelljes që mund të ndodhë në atë pikë të ndërveprimit. Një komponent mund të ketë shumë porte të të njëjtit lloj, secili duke formuar një kanal të ndryshëm hyrës ose dalës gjatë kohës së ekzekutimit. Në këtë drejtim, portet ndryshojnë nga ndërfaqet e moduleve, ndërfaqet e të cilave nuk përsëriten kurrë. Ju mund të shënoni një port me një numër ose me një rang numrash për të treguar replikimin; për shembull, "1. .4" mund të nënkuptojë që një ndërfaqe mund të kopjohet deri në katër herë. Portet e një përbërësi duhet të dokumentohen në mënyrë të qartë, duke i treguar ato dhe përcaktimin e tyre në diagram në dokumentimin mbështetës. Një komponent në këndvështrim të C&C mund të përfaqësojë një nënsistem kompleks, i cili vetë mund të përshkruhet si një nënarkitekturë e C&C. Kjo nënarkitekturë mund të përshkruhet në mënyrë grafike *in situ* kur nënstruktura është jo shumë komplekse, duke e treguar atë të vendosur brenda përbërësit që rafinon. Sidoqoftë, shpesh dokumentohet veçmas. Nënarkitektura e një komponenti mund të përdorë një model tjetër nga ai në të cilin shfaqet komponenti.

Lidhësit janë lloji tjetër i elementit në një këndvështrim C&C. Lidhësit kanë *role*, të cilat janë ndërfaqet e tij, duke përcaktuar mënyrat në të cilat lidhësi mund të përdoret nga komponentë për të kryer ndërveprimin. Ashtu si portet përbërëse, rolet e lidhësit ndryshojnë nga ndërfaqet e modulit në atë që ato mund të përsëriten, duke treguar sa komponentë mund të përfshihen në ndërveprimin e tij.

Ashtu si përbërësit, lidhësit kompleks nga ana tjetër mund të zërthehen në koleksione përbërësish dhe lidhësit që përshkruajnë nënshtrukturën arkitekturale të atyre lidhësve. Lidhësit nuk ka pse të jenë binarë. Kjo sepse, ata nuk kanë nevojë të kenë saktësisht dy role. Lidhësit mishërojnë një protokoll ndërveprimi. Kur dy ose më shumë komponentë ndërveprojnë, ata duhet t'i binden konventave për rendin e ndërveprimeve, vendndodhjen e kontrollit dhe trajtimin e kushteve të gabimit dhe afateve kohore. Protokollin e ndërveprimit duhet të dokumentohet.

Relacioni kryesor brenda një këndvështrimi C&C është *lidhja*. *Lidhjet* tregojnë se cilët lidhës i janë bashkangjitur cilit komponent, duke përcaktuar kështu një sistem si një grafik të përbërësve dhe lidhjeve. Në mënyrë të veçantë, një shtojcë shënohet duke shoqëruar (bashkangjitur) portin e një përbërësi në një lidhës. Një shtojcë e vlefshme është ajo në të cilën

portet dhe rolet janë në përputhje me njëri-tjetrin, nën kufizimet semantike të përcaktuara nga pamja. Përputhshmëria shpesh përcaktohet në lidhje me llojin e informacionit dhe protokollin. Një element (përbërës ose lidhës) i një pamje C&C do të ketë të lidhura veti të ndryshme. Çdo element duhet të ketë një emër dhe tip. Karakteristikat shtesë varen nga lloji i përbërësit ose lidhësit. Përcaktoni vlera për vetitë që mbështesin analizat e synuara për pamje C&C të veçanta. Për shembull, nëse pamja do të përdoret për analizën e performancës, vonesat, kapacitetet e radhës dhe prioritetet e threadave mund të jenë të nevojshme. Më poshtë tregohen ca shembuj të disa vetive tipike dhe përdorimit të tyre:

- *Besueshmëria*. Cila janë gjasat e dështimit për një përbërës ose lidhës të caktuar? Kjo pronë mund të përdoret për të ndihmuar në përcaktimin e disponueshmërisë së përgjithshme të sistemit.

- *Performanca*. Çfarë lloje të kohës së përgjigjes do të sigurojë komponenti nën cilat ngarkesa? Çfarë lloji i gjerësisë së benduidhit, latentes, jitter, vëllimi të transaksionit, ose xhiros mund të pritët për një lidhës të dhënë? Kjo pronë mund të përdoret me të tjerat për të përcaktuar vetitë në të gjithë sistemin si p.sh. kohën e reagimit, xhiros dhe nevojat e tamponimit.

- *Kërkesat burimore*. Cilat janë nevojat e përpunimit dhe magazinimit të një përbërësi ose një lidhësi? Kjo pronë mund të përdoret për të përcaktuar nëse një konfigurim i propozuar i pajisjes do të jetë adekuate.

- *Funksionaliteti*. Çfarë funksionesh kryen një element? Kjo pronë mund të përdoret për të arsyetuar rreth llogaritjes së përgjithshme të kryer nga një sistem.

- *Siguria*. A zbaton ose ofron një komponent ose një lidhës tipare të sigurisë, të tilla si kriptimi, gjurmët e auditimit, apo vërtetimi? Kjo pronë mund të përdoret për të përcaktuar dobësitë e sistemit të sigurisë.

- *Konkurrenca*. A ekzekutohet ky komponent si një proces ose thread e veçantë? Kjo pronë mund të ndihmojë për të analizuar ose simuluar performancën e përbërësve të njëkohshëm dhe për të identifikuar ngërçet e mundshëm.

- *Modifikueshmëria*. A e mbështet struktura e mesazheve një strukturë për t'u kujdesur për evoluimin e shkëmbimit të të dhënave? A mund të përshtaten përbërësit për të përpunuar ato mesazhe të reja? Kjo pronë mund të përcaktohet për të zgjeruar funksionalitetin e një përbërësi.



• *Niveli*. Në një topologji të niveleve, në çfarë niveli banon përbërësi? Kjo pronë ndihmon në përcaktimin e procedurave të ndërtimit dhe vendosjes, sidhe kërkesave të platformës për secilën shtresë.

Pamjet C&C përdoren zakonisht për t'u treguar zhvilluesve dhe palëve të tjera të interesit se si funksionon sistemi. Pamjet C&C përdoren gjithashtu për të arsyetuar atributet e cilësisë së sistemit të ekzekutimit, të tilla si performanca dhe disponueshmëria. Në veçanti, një pamje e mirë-dokumentuar lejon arkitektët të parashikojnë vetitë e përgjithshme të sistemit siç janë vonesa ose besueshmëria, vlerësimet e dhëna ose matjet e vetive të elementeve individuale dhe bashkëveprimet e tyre.

Tabela në vazhdim përmbledh elementet, marrëdhëniet dhe vetitë që mund të shfaqen në pamjet C&C. Kjo tabelë vazhdon me një diskutim më të hollësishëm të këtyre koncepteve, së bashku me udhëzimet në lidhje me dokumentimin e tyre.

Elements	<ul style="list-style-type: none"> <li>▪ <i>Components</i>. Principal processing units and data stores. A component has a set of <i>ports</i> through which it interacts with other components (via connectors).</li> <li>▪ <i>Connectors</i>. Pathways of interaction between components. Connectors have a set of roles (interfaces) that indicate how components may use a connector in interactions.</li> </ul>
Relations	<ul style="list-style-type: none"> <li>▪ <i>Attachments</i>. Component ports are associated with connector roles to yield a graph of components and connectors.</li> <li>▪ <i>Interface delegation</i>. In some situations component ports are associated with one or more ports in an "internal" subarchitecture. The case is similar for the roles of a connector.</li> </ul>
Constraints	<ul style="list-style-type: none"> <li>▪ Components can only be attached to connectors, not directly to other components.</li> <li>▪ Connectors can only be attached to components, not directly to other connectors.</li> <li>▪ Attachments can only be made between compatible ports and roles.</li> <li>▪ Interface delegation can only be defined between two compatible ports (or two compatible roles).</li> <li>▪ Connectors cannot appear in isolation; a connector must be attached to a component.</li> </ul>
Usage	<ul style="list-style-type: none"> <li>▪ Show how the system works.</li> <li>▪ Guide development by specifying structure and behavior of runtime elements.</li> <li>▪ Help reason about runtime system quality attributes, such as performance and availability.</li> </ul>

Figura 2. Përmbledhje e pamjeve C&C [12]

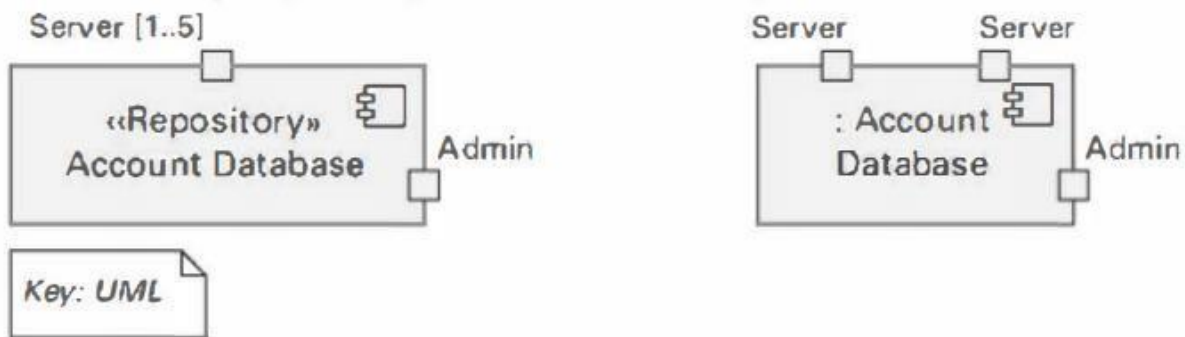
## SHËNIMET PËR PAMJET C&C

Vizatimet me kuti dhe me vijë janë gjithmonë në dispozicion për të përfaqësuar pamjet C&C. Edhe pse shënimet informale janë të kufizuara në semantikën që mund të përcillet, duke ndjekur disa udhëzime të thjeshta mund t'i jepni thelb përshkrimeve të tyre.

Udhëzimi kryesor është i thjeshtë: caktoni secilit lloj të përbërësit dhe secilit lloj të lidhësit një formë të veçantë vizuale (simbol) dhe renditni secilin prej llojeve në një çelës.

Komponentët e UML janë një përputhje e mirë kuptimore me përbërësit C&C sepse lejojnë dokumentim intuitiv të dokumentimit të informacionit të rëndësishëm si ndërfaqet, vetitë dhe përshkrimet e sjelljes. UML komponentët gjithashtu bëjnë dallimin midis llojeve të përbërësve dhe instancave të përbërësve, gjë që është e dobishme kur përcaktojmë llojet e komponentëve specifik të pamjes.

Portet UML janë një ndeshje e mirë kuptimore me portet C&C. Një portë UML mund të zbkurohet me një shumëfishim, megjithëse kjo zakonisht bëhet vetëm në llojet e përbërësve. Numri i porteve në instancat përbërëse, zakonisht lidhet me një numër specifik. Komponentët që krijojnë dhe menajojnë në mënyrë dinamike një grup të porteve duhet të mbajnë një përshkrues të shumëllojshmërisë në përshkrimet e shembujve.



*Figura 3. Një përfaqësim UML i porteve në një tip përbërës C&C (majtas) dhe instance përbërësi (djathtas). Lloji i përbërësit të Bazës së të Dhënave të Llogarisë ka dy lloje portesh, Server dhe Admin (shënuar nga kutitë në kufirin e përbërësit) Porta e serverit përcaktohet me një shumëllojshmëri, që do të thotë se instancat e shumëfishta të portit lejohen në cilëndo instancë të komponentit përkatës [12]*

Ndërsa lidhësit C&C janë po aq semantiksht të pasur sa përbërësit C&C, e njëjta gjë nuk është e vërtetë për konektorët e UML. Konektorët e UML nuk mund të kenë nënstrukturë, attribute ose përshkrime të sjelljes. Kjo e bën më të vështirë zgjedhjen se si të përfaqësosh lidhësit C&C, pasi lidhësit e UML nuk janë gjithmonë mjaftueshëm të pasur.

Ju duhet të përfaqësoni një lidhës "të thjeshtë" C&C duke përdorur një lidhës UML – një linjë. Shumë lidhës të përdorur C&C kanë semantikë dhe implementime të njohura, të pavarura nga aplikacioni, të tilla si thirrje funksioni ose operacione të leximit të të dhënave. Nëse informacioni i vetëm që duhet të furnizoni është lloji lidhës, atëherë një lidhës UML është adekuat. Lidhësit e kthimit të thirrjes mund të përfaqësohen nga një UML lidhës i montimit, i cili lidh ndërfaqen e kërkuar të një përbërësi (priza) me atë të ndërfaqes së ofruar të komponentit tjetër. Ju mund të përdorni një stereotip për të treguar llojin e lidhësit. Shtojca tregohet duke lidhur pikat përfundimtare të lidhësit në portet e përbërësve. Rolet lidhëse nuk mund të përfaqësohen shprehimisht me një lidhës UML sepse elementi lidhës UML nuk lejon përfshirjen e ndërfaqeve (ndryshe nga porta UML, e cila lejon ndërfaqet). Përafrimi më i mirë është të etiketosh skajet e lidhësit dhe të përdorni këto etiketa për të identifikuar përshkrimet e roleve që duhet të dokumentohen diku tjetër.

Ju duhet të përfaqësoni një lidhës "të pasur" C&C duke përdorur një përbërës UML, ose duke shënuar një linjë të një lidhësi UML me një etiketë ose dokumentim tjetër ndihmës që shpjegon kuptimin e lidhësit kompleks.

## **PAMJET E ALOKIMIT**

Pamjet e alokimit përshkruajnë hartëzimin e njësive të softuerit në elementet e një mjedisi në të cilin zhvillohet softueri ose në të cilin ekzekutohet ai. Ambienti mund të jetë hardueri, sistemi operativ në të cilin ekzekutohet softueri, sistemet e skedarëve që mbështesin zhvillimin ose vendosjen, ose organizatën(at) e zhvillimit.

Tabela në vazhdim përmbledh karakteristikat e pamjeve të alokimit. Pamjet e alokimit përbëhen nga elemente softuerike dhe elemente mjedisore. Shembuj të elementeve të mjedisit janë një procesor, një farm disk, një skedar ose dosje, ose një grup zhvilluesish. Elementet e softuerit vijnë nga një modul ose pamje C&C.

Elements	<ul style="list-style-type: none"> <li>▪ <i>Software element.</i> A software element has properties that are <i>required</i> of the environment.</li> <li>▪ <i>Environmental element.</i> An environmental element has properties that are <i>provided</i> to the software.</li> </ul>
Relations	<i>Allocated to.</i> A software element is mapped (allocated to) an environmental element. Properties are dependent on the particular view.
Constraints	Varies by view
Usage	<ul style="list-style-type: none"> <li>▪ For reasoning about performance, availability, security, and safety.</li> <li>▪ For reasoning about distributed development and allocation of work to teams.</li> <li>▪ For reasoning about concurrent access to software versions.</li> <li>▪ For reasoning about the form and mechanisms of system installation.</li> </ul>

Figura 4. Përmbledhje e karakteristikave të pamjeve të alokimit [12]

Relacioni në një pamje të alokimit është *i alokuar në*. Një element i vetëm i softuerit mund të alokohet në shumë elemente mjedisore dhe shumë elemente të softuerit mund të alokohen në një element të vetëm mjedisor. Nëse këto alokime ndryshojnë me kalimin e kohës, ose gjatë zhvillimit ose ekzekutimit të sistemit, atëherë arkitektura thuhet të jetë dinamike në përputhshmëri me atë alokim. Për shembull, proceset mund të migrojnë nga një procesor ose makinë virtuale në tjetrën. Në mënyrë të ngjashme modulet mund të migrojnë nga një ekip zhvillimi në tjetrin.

Elementet e softuerit dhe elementet e mjedisit kanë veti në pamjet e alokimit. Qëllimi i zakonshëm i një pamje alokimi është krahasimi i vetive *të kërkuara* nga elementi i softuerit me vetitë *e ofruara* nga elementet mjedisore për të përcaktuar nëse alokimi do të jetë i suksesshëm.

Pamjet e alokimit mund të përshkruajnë pamje statike ose dinamike. Një pamje statike përshkruan një alokim fiks të burimeve në një mjedis. Një pamje dinamike përshkruan kushtet dhe shkaktarët për të cilët alokimi i burimeve ndryshon sipas ngarkimit. Disa sisteme rekrutojnë dhe shfrytëzojnë burime të reja ndërsa ngarkesa e tyre rritet.

## PAMJET E KUALITETIT

Pamjet e modulit C&C dhe të shpërndarjes janë të gjitha pamje strukturore: Ato kryesisht tregojnë strukturat që arkitekti i ka hartuar në arkitekturë për të përmbushur kërkesat funksionale dhe cilësore të attributeve.

Këto pamje janë të shkëlqyera për të drejtuar dhe detyruar zhvilluesit, puna kryesore e të cilëve është të zbatojnë ato struktura. Sidoqoftë, në sistemet në të cilat atributet e caktuara të cilësisë janë veçanërisht të rëndësishme dhe të përhapura, pamjet strukturore mund të mos jenë mënyra më e mirë për të paraqitur zgjidhjen arkitekturale të atyre nevojave. Arsyeja është se zgjidhja mund të përhapet nëpër struktura të shumta që janë të papërshtatshme për t'u kombinuar (sepse llojet e elementeve të paraqitura në secilën janë të ndryshme).

Këto pamje cilësore formohen duke nxjerrë pjesët përkatëse të pamjeve të strukturës dhe paketimin e tyre së bashku. Këtu janë pesë shembuj:

- *Një pamje e sigurisë* mund të tregojë të gjitha masat e marra arkitekturale për të ofruar siguri. Kjo do të tregonte përbërësit që kanë ndonjë rol ose përgjegjësi të sigurisë, si komunikojnë këta përbërës, çdo depo të dhënash për informacionin e sigurisë dhe depot që janë me interes të sigurisë.

- *Një pamje e komunikimit* mund të jetë veçanërisht e dobishme për sistemet që janë heterogjene dhe të shpërndara globalisht. Kjo pamje do të tregonte të gjitha kanalet përbërëse për komponent, kanalet e ndryshme të rrjetit, vlerat e parametrave të cilësisë së shërbimit dhe zonat e përputhjes. Kjo pamje mund të përdoret për të analizuar lloje të caktuara të performancës dhe besueshmërisë.

- *Një pamje e përjashtimeve ose pamje e trajtimit të gabimeve* mund të ndihmojë në tërheqjen e vëmendjes ndaj raportimit të gabimeve dhe mekanizmave të zgjidhjes. Një pamje e tillë do të tregonte se si komponentët zbulojnë, raportojnë dhe zgjidhin defektet ose gabimet. Kjo do të ndihmonte në identifikimin e burimeve të gabimeve dhe veprimet e duhura korrigjuese për secilin. Analiza e shkakut rrënjësor në ato raste mund të lehtësohet nga një këndvështrim i tillë.

- *Një pamje e besueshmërisë* do të ishte ajo në të cilën modelohen mekanizmat e besueshmërisë të tilla si replikimi dhe kalimi. Do të përshkruante gjithashtu çështjet e kohës dhe integritetin e transaksioneve.

- *Një pamje e performancës* do të përfshinte ato aspekte të arkitekturës të dobishme për të nxjerrë në pah performancën e sistemit. Një pamje e tillë mund të tregojë modele të trafikut të rrjetit, vonesën maksimale për operacione, etj.

Këto dhe pamje të tjera cilësore pasqyrojnë filozofinë e dokumentimit të standardit ISO/IEC/IEEE 4201 0:2011, i cili përshkruan krijimin e pamjeve të drejtuara nga shqetësimet e palëve të interesuara në lidhje me arkitekturën [12].

## 2.2.4. ZGJEDHJA E PAMJEVE

Dokumentimi i vendimeve gjatë procesit të projektimit prodhon pamje, të cilat janë zemra e një dokumenti arkitekture. Ka shumë të ngjarë që këto pamje të jenë më shumë skica sesa produkte të gatshme për publikim; kjo do t'ju japë lirinë për të mbështetur dhe rimenduar vendimet e projektimit që mund të jenë problematike.

Atëherë kur jeni gati të publikoni një dokument të arkitekturës, ka të ngjarë të keni një koleksion mjaft të përpunuar të pamjeve të arkitekturës. Dikur do t'ju duhet të vendosni se cilën dëshironi t'a çoni deri në fund, me sa detaje, dhe cilën t'a përfshini në një njoftim të caktuar. Gjithashtu do t'ju duhet të vendosni se cilat pamje mund të kombinohen në mënyrë të dobishme me të tjerat, në mënyrë që të zvogëloni numrin e përgjithshëm të shikimeve në dokument dhe të zbuloni marrëdhënie të rëndësishme midis pamjeve.

Ju mund të përcaktoni se cilat pamje kërkohen, kur t'i krijoni ato dhe se sa detaje të përfshini nëse dini ato që vijojnë:

- Cilët njerëz janë në dispozicion, dhe me çfarë aftësish
- Cilat standarde duhet të respektoni
- Çfarë buxheti është në dispozicion
- Cili është orari
- Cilat janë nevojat për informacion të palëve të rëndësishme të interesit
- Cilat janë kërkesat e atributit të cilësisë së ngasjes
- Sa është i madh sistemi

Prisni të keni të paktën një pamje të modulit, të paktën një pamje C&C, dhe për sisteme më të mëdha, të paktën një pamje të alokimit në dokumentin tuaj të arkitekturës. Përtej këtij rregulli themelor, megjithatë, ekziston një metodë me tre hapa për zgjedhjen e pamjeve:

• **Hapi 1. Ndërtimi i një tablele të palëve të interesit/pamjes.** Numëroni palët e interesuara për dokumentimin e arkitekturës softuerike të projektit tuaj poshtë rreshtave. Jini sa më gjithëpërfshirës. Për kolonat, renditni pamjet që vlejné për sistemin tuaj.

Disa pamje (të tilla si zberthimi, përdorimet dhe caktimi i punës) zbatohen për çdo sistem, ndërsa të tjerat (pamje të ndryshme C&C, pamja e shtresuar) zbatohen vetëm për disa sisteme. Për kolonat, sigurohuni që të përfshini pamjet ose skicat e pamjeve që keni tashmë si rezultat i punës tuaj të projektimit deri më tani.

Pasi të keni mbushur rreshtat dhe kolonat, plotësoni secilën fushë për të përshkruar se sa informacion kërkon pala e interesit nga pamja: asnjë, vetëm përmbledhje, detaje të moderuara ose detaje të larta.

- **Hapi 2. Kombino pamjet.** Lista e shikimeve të kandidatëve nga hapi 1 ka të ngjarë të japë një numër të madh të shikimeve. Kërkoni pamje marginale në tabelë: ato që kërkojnë vetëm një pasqyrë të përgjithshme, ose që u shërbejnë shumë pak palëve të interesit. Kombinoni çdo pamje marginale me një pamje tjetër që ka zonë elektorale më të fortë.

- **Hapi 3. Përcaktoni përparësitë dhe bëjini gati.** Pas hapi 2 ju duhet të keni grupin minimal të pamjeve të nevojshme për t'i shërbyer komunitetit tuaj të palëve të interesit. Në këtë pikë duhet të vendosni se çfarë të bëni më përpara. Ajo që bëni së pari varet nga projekti juaj, por këtu janë disa gjëra për t'u marrë parasysh:

- Pamja e zberthimit (një nga pamjet e modulit) është një pamje veçanërisht e dobishme për t'u lëshuar herët. Zberthimet e nivelit të lartë (domethënë të gjera dhe të cekëta) shpesh janë të lehta për t'u hartuar dhe me këto informacione menagjeri i projektit mund të fillojë me ekipet e zhvillimit të stafit, të vendosë trajnimin, të përcaktojë se cilat pjesë të transferohen dhe të fillojë të prodhojë buxhete dhe orare.

- Jini të vetëdijshëm se nuk keni për të përmbushur të gjitha nevojat për informacion që të gjitha palëve të interesit në nivelin më të plotë. Sigurimi i 80 përqind të informacionit shkon shumë, dhe kjo mund të jetë mjaftueshëm në mënyrë që palët e interesuara të bëjnë punën e tyre. Kontrolloni me palët e interesit për të parë nëse një nëngrup i informacionit do të ishte i mjaftueshëm. Ata zakonisht preferojnë një produkt që dorëzohet në kohë dhe brenda buxhetit sesa marrjen e dokumentimit të përsosur.

- Ju nuk duhet të plotësoni një pamje para se të filloni një tjetër. Njerëzit mund të bëjnë progres me informacionin e nivelit të përgjithshëm, kështuqë qasja e përgjithshmërisë është shpesh më e mira [14].

## 2.2.5. KOMBINIMI I PAMJEVE

Parimi themelor i dokumentimit të një arkitekture si një grup pamjesh të veçanta sjell një avantazh ndarjeje dhe pushtimi në detyrën e dokumentimit, por nëse pamjet do të ishin të ndryshme në mënyrë të pakthyeshme, pa asnjë shoqërim me njëra-tjetrën, askush nuk do të jetë në gjendje të kuptojë sistemin në tërësi.

Për shkak se të gjitha pamjet në një arkitekturë janë pjesë e së njëjtës arkitekture dhe ekzistojnë për të arritur një qëllim të përbashkët, shumë prej tyre kanë lidhje të forta me njëra-tjetrën. Menagjimi i lidhjes së strukturave arkitekturnale është një pjesë e rëndësishme e punës së arkitektit, pavarësisht nëse ekziston ndonjë dokumentim i atyre strukturave.

Ndonjëherë mënyra më e përshtatshme për të treguar një shoqërim të fortë midis dy pamjeve është shembja e tyre në një *pamje të vetme të kombinuar*, siç diktohet nga hapi 2 i metodës me tre hapa të paraqitur për të zgjedhur pamjet. Një pamje e kombinuar është një pamje që përmban elemente dhe marrëdhënie që vijnë nga dy ose më shumë pamje të tjera. Pamjet e kombinuara mund të jenë shumë të dobishme për sa kohë që nuk përpiqeni t'i mbingarkoni ato me shumë hartime.

Mënyra më e lehtë për të bashkuar pamjet është të krijoni një *mbivendosje* që kombinon informacionin që përndryshe do të kishte qenë në dy pamje të ndara. Kjo funksionon mirë nëse bashkimi midis dy pamjeve është i ngushtë; domethënë ekzistojnë shoqata të forta midis elementeve në një pamje dhe elementeve në pamjen tjetër.

Po qe kështu, struktura e përshkruar nga pamja e kombinuar do të jetë më e lehtë për t'u kuptuar sesa dy pamjet që shihen veçmas.



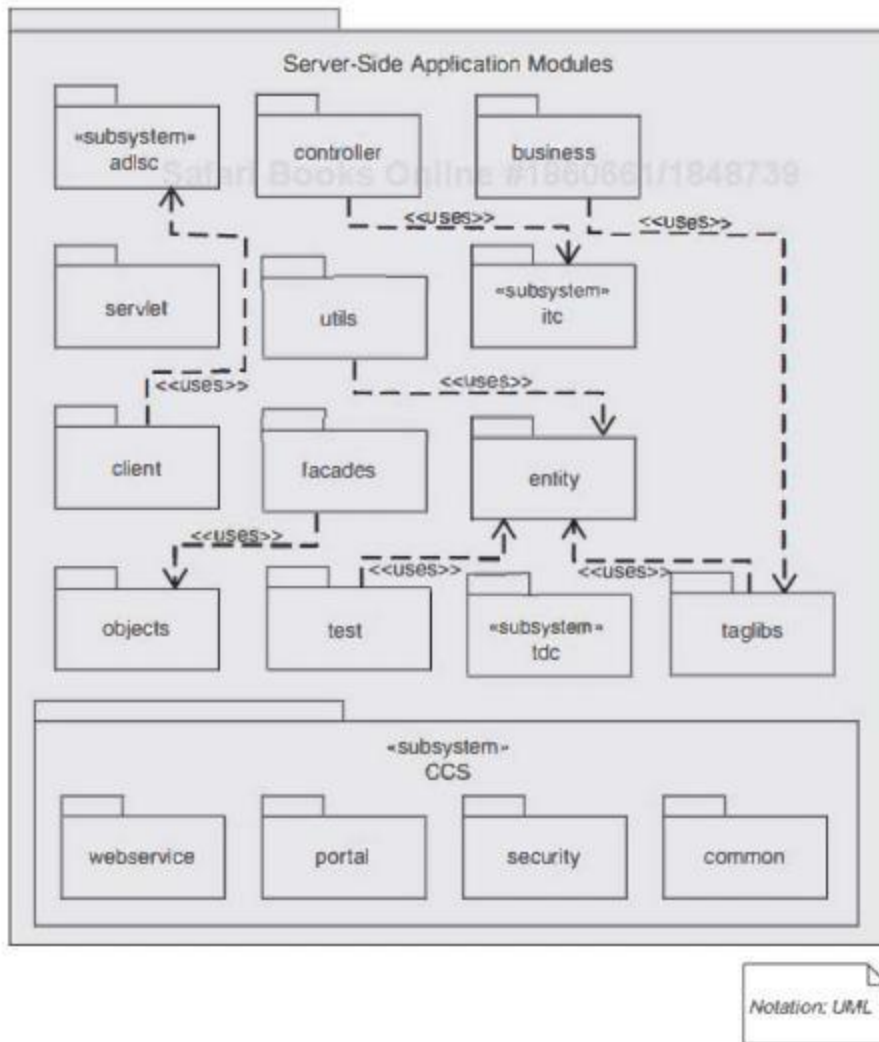


Figura 5. Një pamje dekompozimi e mbivendosur me informacionin "përdor", që të krijojë një mbivendosje zbërthimi/përdorimi [12]

Pamjet më poshtë shpesh kombinohen natyrshëm:

- *Pamjet e ndryshme C&C.* Meqenëse pamjet C&C tregojnë të gjitha marrëdhëniet e kohës së funksionimit midis përbërësve dhe lidhjeve të llojeve të ndryshme, ato kanë tendencë të kombinohen mirë. Pamje të ndryshme (të ndara) C&C priren të tregojnë pjesë të ndryshme të sistemit, ose kanë tendencë të tregojnë përsosjet e zbërthimit të përbërësve në pamjet e tjera. Rezultati shpesh është një grup pamjesh që mund të kombinohen lehtësisht.

- *Pamja e vendosjes ose SOA ose pamjet e proceseve komunikuese.* Një pamje SOA tregon shërbimet, dhe një pamje e proceseve komunikuese tregon proceset. Në të dy rastet, këta janë përbërës që vendosen në procesorë. Kështuqë ekziston një shoqatë e fortë midis elementeve në këto pamje.

• Pamja e zbërthimit dhe cilado nga detyrat, zbatimi, përdorimet ose pamjet e shtresave të punës. Modulet e zbërthyer kërkojnë njësitë e punës, zhvillimin dhe përdorimin. Përveç kësaj, këto module mbushin shtresat [12].

## 2.2.6. NDËRTIMI I PAKETËS SË DOKUMENTIMIT

Parimi i dokumentimit të arkitekturës na tregon se detyra jonë është të dokumentojmë pamjet përkatëse dhe të dokumentojmë informacionin që zbatohet për më shumë se një pamje [13].

## DOKUMENTIMI I NJË PAMJEJE

Figura 6 tregon një model për dokumentimin e një pamjeje.

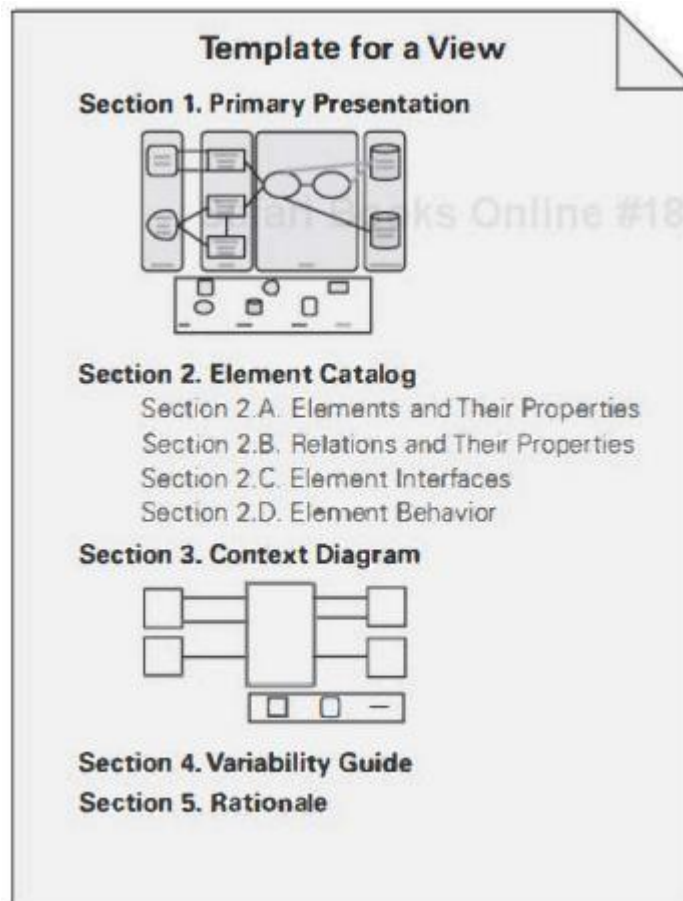


Figura 6. Shabllon i pamjes

Pavarësisht nga pamja, dokumentimi për një pamje mund të vendoset në një organizim standard të përbërë nga këto pjesë:

- **Seksioni 1: Prezantimi Primar.** *Prezantimi primar* tregon elementet dhe marrëdhëniet e pamjes. Prezantimi primar duhet të përmbajë informacionin që dëshironi të përçoni në lidhje me sistemin – në fjalorin e asaj pamjeje. Sigurisht që duhet të përfshijë elementet dhe marrëdhëniet kryesore, por në disa rrethana mund të mos i përfshijë të gjithë.

Prezantimi primar është më shpesh grafik. Mund të jetë një diagram që keni vizatuar në një shënim informal duke përdorur një mjet të thjeshtë vizatimi, ose mund të jetë një diagram në një shënim gjysmëformal ose zyrtar të importuar nga një mjet i modelit ose modelimit që po përdorni.

Nëse prezantimi juaj kryesor është grafik, sigurohuni që të përfshini një çelës që shpjegon shënimin. Mungesa e një çelësi është gabimi më i zakonshëm që shihet në dokumentim në praktikë.

Herë pas here, prezantimi kryesor do të jetë me tekst, i tillë si një tabelë ose një listë. Nëse ai tekst paraqitet sipas rregullave të caktuara stilistike, këto rregulla duhet të deklarohen ose përfshihen me referencë, si analoge me çelësin e shënimit grafik. Pavarësisht nëse prezantimi primar është tekstual në vend të grafikut, roli i tij është të paraqesë një përmbledhje të shkurtër të informacionit më të rëndësishëm në pamje.

- **Seksioni 2: Katalogu i Elementeve.** *Katalogu i elementeve* detajizon elementet e përshkruara në prezantimin primar. Për shembull, nëse një diagram tregon elementët A, B dhe C, atëherë katalogu i elementeve duhet të shpjegojë se çfarë janë A, B dhe C. Përveç kësaj, nëse elementet ose marrëdhëniet përkatëse për këtë pamje u hoqën nga prezantimi primar, ato duhet të prezantohen dhe shpjegohen në katalog. Pjesët specifike të katalogut përfshijnë si vijon:

- *Elementet dhe vetitë e tyre.* Kjo pjesë emëron secilin element në pamje dhe rendit vetitë e këtij elementi. Për shembull, elementet në një pamje dekompozimi mund të kenë vetinë e "përgjegjësisë" – një shpjegim i rolit të secilit modul në sistem – dhe elementet në një pamje të proceseve komunikuese mund të kenë si veti parametra të kohës, ndër të tjera. Pavarësisht nëse pronat janë të përgjithshme për pamjen e zgjedhur ose arkitekti ka prezantuar të reja, këtu dokumentohen dhe u jepen vlera.

- *Marrëdhëniet dhe vetitë e tyre.* Çdo pamje ka lloje specifike të relacionit që ajo përshkruan midis elementeve në atë pamje. Kryesisht, këto marrëdhënie tregohen në prezantimin primar. Sidoqoftë, nëse prezantimi primar nuk tregon të gjitha marrëdhëniet ose nëse ka

përrjashtime nga ajo që përshkruhet në prezantimin primar, ky është vendi për të regjistruar atë informacion.

- *Ndërfaqet e elementeve.* Ky seksion dokumenton ndërfaqet e elementeve.

- *Sjellja e elementit.* Ky seksion dokumenton sjelljen e elementit që nuk është e qartë nga prezantimi primar.

- **Seksioni 3: Diagrami i kontekstit.** Një *diagram konteksti* tregon se si sistemi ose pjesa e sistemit të përshkruar në këtë pamje ka të bëjë me mjedisin e tij. Qëllimi i një **diagrami kontekstual** është të përshkruajë fushën e një pamjeje. Këtu "kontekst" do të thotë një mjedis me të cilin ndërvepron pjesa e sistemit. Subjektet në mjedis mund të jenë njerëzit, sisteme të tjera kompjuterike, ose objekte fizike, të tilla si sensorë ose pajisje të kontrolluara.

- **Seksioni 4: Udhëzuesi i variabilitetit.** Një *udhëzues i variabilitetit* tregon se si të ushtroni çdo pikë ndryshimi që është pjesë e arkitekturës së treguar në këtë pamje.

- **Seksioni 5: Arsyetimi.** *Arsyetimi* shpjegon pse u krijua modeli i pasqyruar në pamje. Qëllimi i kësaj pjese është të shpjegojë pse dizajni është ashtu siç është dhe të sigurojë një argument bindës se ai është valid. Zgjedhja e një modeli në këtë pamje duhet të justifikohet këtu duke përshkruar problemet arkitekturale që zgjidh modeli i zgjedhur dhe arsyetimin për zgjedhjen e tij mbi një tjetër.

## DOKUMENTIMI I INFORMACIONIT PËRTEJ PAMJEVE

Siç tregohet në figurën 6, dokumentimi përtej pamjeve mund të ndahet në dy pjesë:

1. *Pasqyra e dokumentimit të arkitekturës.* Kjo tregon se si paraqitet dhe organizohet dokumentimi në atë mënyrë që një palë e interesuar e arkitekturës të gjejë informacionin që i nevojitet në mënyrë efikase dhe të besueshme.

2. *Informacioni në lidhje me arkitekturën.* Këtu, informacioni që mbetet për t'u kapur përtej vetë pamjeve është një përmbledhje e shkurtër e sistemit për të mbështetur çdo lexues në lidhje me qëllimin e sistemit dhe mënyra sesi pamjet lidhen me njëra-tjetrën, një përmbledhje dhe arsyetim i qasjeve të projektimit në të gjithë sistemin, një listë e elementeve dhe ku shfaqen, dhe një fjalor e një listë e shkurtuar për tërë arkitekturën.

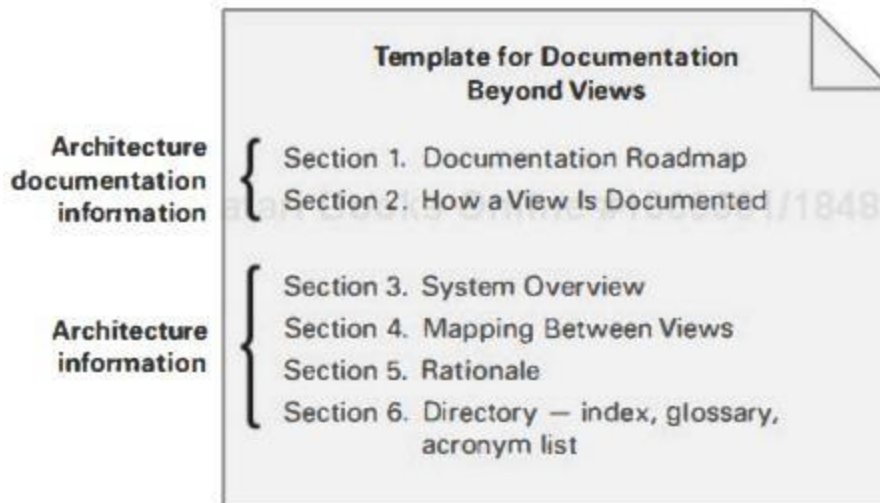


Figura 7. Përmbledhje e dokumentimit përtej pamjeve [12]

Figura 7 përmbledh modelin për dokumentimin përtej pamjeve. Dokumentimi përtej pamjeve përbëhet nga pjesët e mëposhtme:

- **Dokumenti i kontrollimit të informacionit.** Renditni organizatën lëshuese, numrin aktual të versionit, datën e lëshimit dhe statusin, një historik të ndryshimeve dhe procedurën për paraqitjen e kërkesave për ndryshim në dokument. Mjetet e kontrollimit të ndryshimit mund të sigurojnë pjesën më të madhe të këtij informacioni.

- **Seksioni 1: Udhërrëfyesi i Dokumentimit.** Udhërrëfyesi i dokumentimit i tregon lexuesit se çfarë informacioni ka në dokumentim dhe ku t'a gjejë atë. Një hartë e dokumentimit përbëhet nga katër seksione:

- *Shtrirja dhe përmbledhja.* Shpjegoni qëllimin e dokumentit dhe përmbledhni shkurtimisht atë që është mbuluar dhe (nëse mendoni se do t'ju ndihmojë) atë që nuk është mbuluar. Shpjegoni lidhjen me dokumentet tjera (siç janë dokumentet e projektimit në rrjedhën e poshtme ose dokumentet inxhinierike të sistemit në rrjedhën e sipërme).

- *Si organizohet dokumentimi.* Për secilën pjesë të dokumentimit, jepni një përmbledhje të shkurtër të informacionit që mund të gjendet aty. Një alternativë për këtë është përdorimi i një tabele të shënuar përmbajtjeje. Kjo është një tabelë që nuk rendit vetëm titujt dhe numrat e faqeve të seksioneve, por jep gjithashtu një përmbledhje me secilën hyrje. Ajo siguron pazar me një ndalesë për një lexues që përpiqet të kërkojë një lloj të veçantë informacioni.

- *Përmbledhja e Pamjes.* Pjesa kryesore e hartës përshkruan pamjet që arkitekti ka përfshirë në pako. Për secilën pamje, harta jep informacionin e mëposhtëm:

- Emrin e pamjes dhe çfarë modeli paraqet, nëse ka të tillë.
- Një përshkrim të llojeve të elementeve të pamjes, llojeve të relacioneve dhe llojeve të pronave. Kjo lejon që një lexues të fillojë të kuptojë llojin e informacionit që paraqitet në pamje.
- Një përshkrim i gjuhës, teknikave të modelimit ose metodave analitike të përdorura në ndërtimin e pamjes.

• *Si mund t'a përdorin palët e interesit dokumentimin.* Harta vijon me një pjesë që përshkruan cilat palë të interesit dhe shqetësime adresohen nga secila pamje; kjo kapet në mënyrë të përshtatshme në formë të një table. Poashtu tregon se si palët e ndryshme të interesit mund të përdorin dokumentimin në ndihmesë të adresimit të shqetësimeve të tyre. Përfshini skenarë të shkurtër, të tillë si "Një mirëmbajtës dëshiron të njohë njësitë e softuerit që ka të ngjarë të ndryshohen nga një modifikim i propozuar. Mirëmbajtësi konsulton pamjen e zbërthimit për të kuptuar përgjegjësitë e secilit modul në mënyrë që të identifikojë modulet që mund të ndryshojnë. Mirëmbajtësi pastaj konsultohet me pamjen e përdorimeve për të parë se cilat module përdorin modulet e prekura (dhe kështu gjithashtu mund të duhet të ndryshojnë). "Për të qenë në përputhje me ISO/IEC 4201 0-2007, duhet të merrni parasysh shqetësimet e së paku përdoruesve, blerësve, zhvilluesve dhe mirëmbajtësve.

• **Seksioni 2: Si dokumentohet një pamje.** Këtu shpjegoni organizimin standard që po përdorni për të dokumentuar pamjet. Ai u tregon lexuesve tuaj se si të gjejnë informacione në një pamje. Nëse organizata juaj është standardizuar në një model për një pamje, siç duhet, atëherë ju thjesht mund t'i referoheni këtij standardi.

• **Seksioni 3: Përmbledhja e sistemit.** Ky është një përshkrim i shkurtër në prozë të funksionit të sistemit, përdoruesve të tij, dhe çdo sfondi ose kufizimi të rëndësishëm. Kjo pjesë u ofron lexuesve tuaj një model të qëndrueshëm mendor të sistemit dhe qëllimit të tij. Ky mund të jetë thjesht një tregues për një dokument të konceptit të operacioneve.

• **Seksioni 4: Mapimi ndërmjet pamjeve.** Për shkak se të gjitha pamjet e një arkitekture përshkruajnë të njëjtin sistem, duhet të arsyetojmë se çdo dy pamje do të kenë shumë të përbashkëta. Ndihmesa e lexuesit për të kuptuar lidhjen midis pamjeve do t'a ndihmojë lexuesin të fitojë një pasqyrë të fuqishme se si funksionon arkitektura si një tërësi konceptuale e unifikuar. Asociacionet midis elementeve nëpër pamje në një arkitekturë janë, në përgjithësi, shumë- shumë. Marrëdhëniet view-to-view mund të kapen lehtësisht si tabela. Renditni elementet e pamjes së parë në një mënyrë të përshtatshme për kërkim. Tabela në vetvete duhet të shënohet ose

prezantohet me një shpjegim të lidhjes që ajo përshkruan; domethënë, çfarë është korrespondenca midis elementeve në të dy pamjet. Shembujt përfshijnë "është zbatuar nga" për hartëzimin nga një pamje e përbërësit dhe lidhësit në një pamje të modulit, "zbaton" për hartëzimin nga një pamje e modulit në një pamje të përbërësit dhe lidhësit, "përfshirë në" për hartëzimin nga një pamje e zbërthimit në një pamje të shtresuar, e shumë të tjerë.

- **Seksioni 5: Arsyetimi.** Kjo pjesë dokumenton vendimet arkitekturale që vlejné për më shumë se një pamje. Kandidatët kryesorë përfshijnë dokumentimin e sfondit ose kufizimeve organizative ose kërkesat kryesore që çuan në vendimet e importit në të gjithë sistemin. Shpesh, vendimet për cilat modele themelore të arkitekturës të përdoren përshkruhen këtu.

- **Seksioni 6: Drejtoria.** Drejtoria është një grup i materialeve referuese që i ndihmon lexuesit të gjejné shumë informacion, shpejt. Ai përfshin një indeks të termave, një fjalor dhe një listë të shkurtimeve [12].

## MODELET E DOKUMENTIMIT

Arkitektët munden dhe zakonisht përdorin modelet si një pikënisje për hartimin e tyre. Këto modele mund të botohen në katalogët ekzistues ose në depon e pronarëve të një organizate të modeleve standarde, ose të krijuara posaçërisht për problemin në dorë nga arkitekti. Në secilin nga këto raste, ato ofrojnë një qasje të përgjithshme (domethënë jo të plotë) të zgjidhjes që arkitektit do t'i duhet të përsosë dhe çastëzojë.

Së pari, regjistroni faktin që modeli i dhënë po përdoret. Pastaj tregoni pse u zgjodh kjo qasje e zgjidhjes – pse është një përshtatje e mirë për problemin në fjalë. Nëse qasja e zgjedhur vjen nga një model, kjo do të thotë që do të konsistojë në thelb tregimin se problemi në fjalë i përshtatet problemit dhe kontekstit të modelit.

Përdorimi i një modeli nënkupton marrjen e vendimeve të njëpasnjëshme të dizajnit që përfundimisht rezultojné në një arkitekturë. Këto vendime të dizajnit shfaqen si elemente dhe marrëdhënie të sapo instancuara midis tyre. Arkitekti mund të dokumentojë një fotografi të arkitekturës në secilën fazë. Sa faza ekzistojné varet nga shumë gjëra, jo më e rëndësishmja prej të cilave është aftësia e lexuesve për të ndjekur procesin e dizajnit në rast se do të duhet t'a rishikojné atë në të ardhmen [14].

## 2.2.7. SJELLJA E DOKUMENTIMIT

Dokumentimi i një arkitekture kërkon dokumentimin e sjelljes që plotëson pamjet strukturore duke përshkruar se si elementet e arkitekturës ndërveprojnë me njëra-tjetrën. Arsyeimi rreth karakteristikave të tilla si potenciali i një sistemi për bllokim, aftësia e një sistemi për të përfunduar një detyrë në kohën e dëshiruar ose konsumimi maksimal i memories kërkon që përshkrimi i arkitekturës të përmbajë informacione në lidhje me karakteristikat e elementeve individuale, sidhe modelet e ndërveprimit ndërmjet atyre që janë – si sillen me njëri-tjetrin. Në këtë pjesë, ofrohen udhëzime se cilat lloje të gjërave do të dëshironi të dokumentoni në mënyrë që të merrni këto përfitime.

Ekzistojnë dy lloje të shënimeve në dispozicion për dokumentimin e sjelljes. Lloji i parë i shënimit quhet gjuhë e orientuar drejt gjurmës; e dyta quhet gjuhë gjithëpërfshirëse.

*Gjurmët* janë sekuenca të aktiviteteve ose ndërveprimeve që përshkruajnë përgjigjen e sistemit ndaj një stimulusi specifik kur sistemi është në një gjendje specifike. Një gjurmë përshkruan një sekuençë aktivitësh ose ndërveprimësh midis elementeve strukturore të sistemit. Megjithëse është e mundshme të përshkruhen të gjitha gjurmët e mundshme për të gjeneruar ekuivalentin e një modeli gjithëpërfshirës të sjelljes, nuk është qëllimi i dokumentimit të orientuar drejt gjurmës për t'a bërë këtë. Më poshtë përshkruhen katër shënime për dokumentimin e gjurmëve: raste përdorimi, diagrame sekuencash, diagrame komunikimi dhe diagrame aktiviteti.

- *Rastet e përdorimit* përshkruajnë se si aktorët mund të përdorin një sistem për të përmbushur qëllimet e tyre. Rastet e përdorimit përdoren shpesh për të kapur kërkesat funksionale për një sistem. UML ofron një shënim grafik për skemat e rasteve të përdorimit, por nuk tregon se si duhet të shkruhet teksti i një rasti përdorimi. Diagrami i rastit të përdorimit të UML mund të përdoret në mënyrë efektive si një përmbledhje e aktorëve dhe sjelljes së një sistemi. Përshkrimi i rastit të përdorimit është tekstual dhe duhet të përmbajë emrin e rastit të përdorimit dhe përshkrimin e shkurtër, aktorin ose aktorët që fillojnë rastin e përdorimit (aktorët kryesorë), aktorët e tjerë që marrin pjesë në rastin e përdorimit (aktorët dytësorë), rrjedhën e ngjarjeve, rrjedhat alternative, dhe rastet e dështimit.

- Një *diagram i sekuençës* UML tregon një sekuençë ndërveprimësh midis shembujve të elementeve të tërhequr nga dokumentimi strukturor. Ajo tregon vetëm rastet që marrin pjesë në skenarin që dokumentohen. Një diagram i sekuençës ka dy dimensione: vertikale, që përfaqëson



kohën dhe horizontale, që përfaqëson instancat e ndryshme. Ndërveprimet janë rregulluar në sekuencën kohore nga lart poshtë. Figura 8 është një shembull i një diagrami sekuence që ilustron shënimin bazë UML.

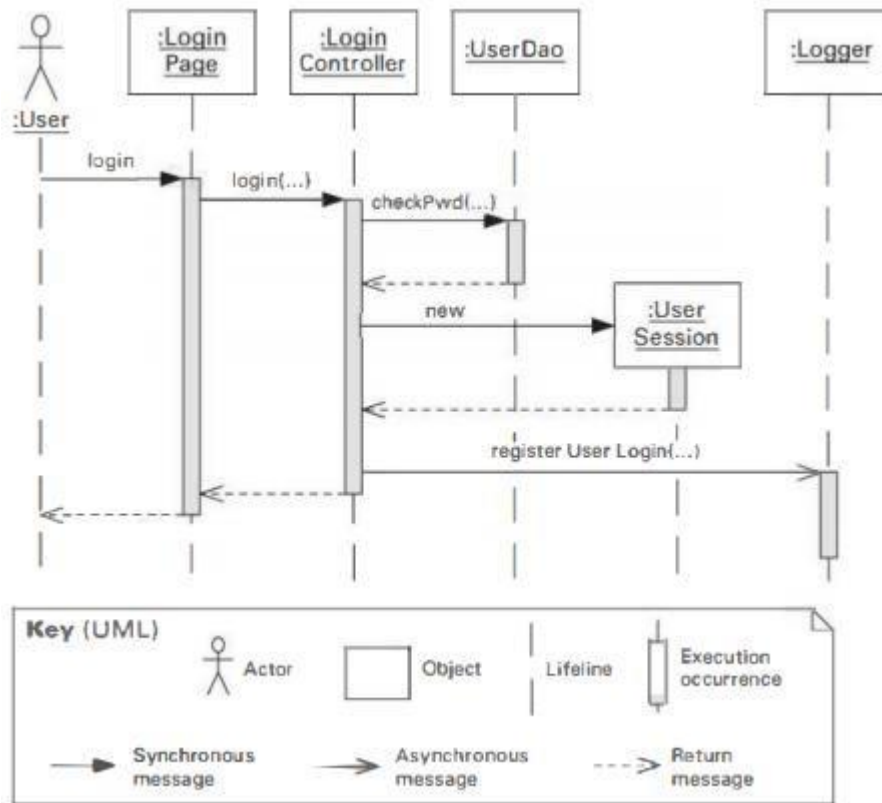


Figura 8. Një shembull i thjeshtë i një UML diagrami sekuence

Objektet (d.m.th., shembujt e elementeve) kanë një vijë shpëtimi, të vizatuar si një vijë vertikale e ndërprerë përgjatë boshtit kohor. Sekuenca zakonisht fillohet nga një aktor në ekstremin e majtë. Instancat bashkëveprojnë duke dërguar mesazhe, të cilat shfaqen si shigjeta horizontale. Një mesazh mund të jetë një metodë ose thirrje funksioni, një ngjarje e dërguar përmes një radhe, ose diçka tjetër. Mesazhi zakonisht harton në një burim (operacion) në ndërfaqen e shembullit të marrësit. Një majë shigjete e mbushur në një vijë të fortë përfaqëson një mesazh sinkron, ndërsa maja e shigjetës së hapur përfaqëson një mesazh asinkron. Shigjeta e ndërprerë është një mesazh kthimi. Shiritat e ndodhjes së ekzekutimit përgjatë vijës së shpëtimin tregojnë se instanca po përpunohet ose bllokohet duke pritur për një kthim.

- Një *diagram komunikimi* UML tregon një grafik të elementeve bashkëvepruese dhe shënon secilin bashkëveprim me një numër që tregon rendin. Ngjashëm me diagramet e sekuencës, shembujt e treguar në një diagram komunikimi janë elementet e përshkruara në dokumentimin

strukturor shoqërues. Diagramet e komunikimit janë të dobishme kur detyra është të verifikoni që një arkitekturë mund të përmbushë kërkesat funksionale.

- *Diagramet e aktiviteteve* UML janë të ngjashme me tabelat e rrjedhës. Ato tregojnë një proces biznesi si një sekuencë hapash (të quajtura veprime) dhe përfshijnë shënime për të shprehur degëzimin e kushtëzuar dhe përputhjen, sidhe për të treguar ngjarjet e dërgimit dhe marrjes. Shigjetat midis veprimeve tregojnë rrjedhën e kontrollit. Në mënyrë opsionale, diagramet e aktivitetit mund të tregojnë elementin e arkitekturës ose aktorin që kryen veprimet. Diagramet e veprimtarisë mund të shprehin përputhjen. Një nyje piruni (përshkruar si një shirit i trashë ortogonal i shigjetave të rrjedhës) ndan rrjedhën në dy ose më shumë rrjedha të njëkohshme të veprimeve. Rrjedhat e njëkohshme mund të sinkronizohen më vonë në një rrjedhë të vetme përmes një nyjeje bashkimi (përshkruar gjithashtu si një shirit ortogonal). Nyja e bashkimit pret që të gjitha rrjedhat hyrëse të përfundojnë para se të vazhdojnë. Ndryshe nga diagramet e sekuencës dhe komunikimit, diagramet e aktivitetit nuk tregojnë operacionet aktuale që kryhen në objekte të veçanta. Diagramet e veprimtarisë janë të dobishme për të përshkruar gjerësisht hapat në një rrjedhë specifike të punës. Degëzimi i kushtëzuar (simboli i diamantit) lejon që një diagram i vetëm të përfaqësojë gjurmë të shumëfishta, megjithëse zakonisht qëllimi i një diagrami aktiviteti nuk është të tregojë të gjitha gjurmët e mundshme ose sjelljen e plotë të sistemit ose një pjese të tij.

Në kontrast me gjurmët e shënimeve, *modelet gjithëpërfshirëse* tregojnë sjelljen e plotë të elementeve strukturore. Duke pasur parasysh këtë lloj dokumentimi, është e mundur të nxirren të gjitha rrugët e mundshme nga gjendja fillestare në gjendjen përfundimtare. Formalizmi i makinës së gjendjes paraqet sjelljen e elementeve të arkitekturës sepse secila gjendje është një abstraksion i të gjitha historive të mundshme që mund të çojnë në atë gjendje. Gjuhët e makinerive të gjendjes ju lejojnë të plotësoni një përshkrim strukturor të elementeve të sistemit me kufizime në bashkëveprime dhe reagime me kohë si ndaj stimujve të brendshëm ashtu edhe ndaj mjedisit.

Një diagram i makinës së gjendjes UML tregon gjendjet e përfaqësuara si kuti dhe tranzicionet midis gjendjeve të përfaqësuara si shigjeta. Diagramet e makinerive të gjendjes ndihmojnë në modelimin e elementeve të arkitekturës dhe ndihmojnë në ilustrimin e ndërveprimeve të tyre të ekzekutimit [12].

### 3. DEKLARIMI I PROBLEMIT

Duke pasur parasysh gjerësinë e fushës së dokumentimit, tutje do të fokusohemi vetëm në dokumentimin e sistemit dhe atë të përdoruesit. Thamë pra që dokumentimi i sistemit përfaqëson dokumentet që përshkruajnë sistemin në tërësi dhe pjesët e tij. Përfshin dokumentet e kërkesave, vendimet për dizajnin, përshkrimet arkitekturale, kodin burimor, dhe udhëzuesit për ndihmë. Dokumentimi i përdoruesit në anën tjetër, mbulon manualët të cilat kryesisht targetojnë përdoruesit e produktit dhe administratorët e sistemit. Ky i fundit përfshin tutoriale, udhëzues për përdorues, manuale për zgjidhje të problemeve, instalime, dhe manuale referuese. Këto dy lloje të dokumentimit gjejnë një pikë të përbashkët dhe si të tilla i ofrojnë shumë benefite palës pranuese. Por në mungesë të tyre, posedojnë potencialin e afektimit negativ të projektit ashtu edhe të raportit me klientin. Këtu edhe shfaqet zanafilla e problemit.

Epoka e kompleksitetit filloi prejse softueri u bë realitet. Ishte e gjitha e lehtë në fillim, por problemi i vërtetë filloi kur përdoruesit filluan të përdorin softuerin (edhe pse ai ishte menduar për ta). Anëtarët e rinj të ekipit të cilët kanë nevojë të kuptojnë produktin gjithashtu shtuan nevojën për dokumentimin. Për anëtarët e rinj që i bashkohen ekipit në role të ndryshme të zhvillimit të softuerit, dokumentimi dhe prezantimi i mirëfilltë i kodit është mënyra e vetme për të vazhduar tutje. Të shkruarit, në anën tjetër, është një aftësi e injoruar rëndë në të gjithë botën. Shumë studentë e profesionistë e kanë të vështirë të konstruktojnë fjali e paragrafe të mirë-strukturuara. Sidoqoftë, në botën e Silicon Valley ose qendrave të tjera të IT, dokumentimi ndoshta është aspekti më i urryer i zhvillimit softuerik. Është kaq i urryer sa që ndonjëherë zhvilluesit e softuerit gjejnë arsye nga më të ndryshmet për të mos dokumentuar projektin e tyre.

Më poshtë diskutoj arsyet e shumta për këtë sjellje të çuditshme.

**Zhvilluesit janë shumë të afruar me projektin.** Kjo mund të tingëllojë pak qesharake, por e vërteta është që shumica e zhvilluesve kanë krijuar një lidhje me projektin e tyre për të shkruar dokumentim për të. Ndikimi i këtij fenomeni shihet nga këndvështrimi i anëtarëve të tjerë të ekipit, përdoruesve të softuerit, ose kompanisë me të cilën kanë të bëjnë – zhvilluesit nuk mendojnë jashtë vetvetes, dhe kështu nuk ua shpjegojnë lexuesve qartë gjërat. Marrëdhëniet e ndërlikuara që kanë zhvilluesit me kodin e tyre ndonjëherë mund të shkaktojnë dokumentim të

dobët nga ana e zhvilluesve sepse këta të fundit janë në një pozitë ku e kuptojnë se çfarë pondodh dhe konkludojnë që nuk është e nevojshme të shpjegohen gjërat sa më qartë që të jetë e mundur.

**Mungon kultura për dokumentim.** Shumë zhvillues nuk e ndiejnë nevojën për të shkruar dokumentim. Përkundrazi, ekziston gjithmonë një mënyrë për t'ju qasur një problemi, qoftë ai me shkrim apo programim, por jo në këtë rast. Kur e kanë të detyrueshme, këta zhvillues shkruajnë dokumentim të tmerrshëm pasi thjesht nuk shfaqin aspak interes për detyrën në fjalë. Rezultati përfundimtar? Zhvillues të pakënaqur me dokumentim të dobët që vetëm ata mund t'a kuptojnë.

**Zhvilluesit nuk kanë nevojë për dokumentim.** Zhvilluesit e kuptojnë se për çfarë po punojnë dhe kjo është arsyeja pse ata zgjedhin të evitojnë dokumentimin. Edhe pse duket të jetë e arsyeshme, po të njejtit zhvillues mund të harrojnë ndërlëkimet e punës së tyre kur e vizitojnë atë pas disa javësh, muajsh ose vitesh. Për më tepër, pa dokumentim të mirëfilltë ose komentim të kodit, si komunikimi ashtu edhe mirëmbajtja mbesin mbrapa.

**Dokumentimi kërkon kohë dhe përpjekje.** Shkrimi nuk është një detyrë e lehtë; kërkon mendim kritik, durim dhe përpjekje. Për zhvilluesit, dokumentimi mund të duket si punë imponuese. Sipas shumë njerëzve, kjo është thjesht një humbje kohe dhe përpjekjeje pasi ata tashmë e dinë që kërkohet kohë e konsiderueshme për prodhimin e një dokumentimi të mirë. Zhvilluesit preferojnë që këtë kohë dhe përpjekje t'a përdorin në implementimin e veçorive të reja, zgjidhjen e problemeve, ose eliminimin e gabimeve.

**Presioni për të dorëzuar projektin në kohë.** Gjithmonë ekziston një datë e caktuar e varur në qoshe. Zhvilluesi duhet të dorëzojë projektin në kohë, dhe askush nuk kujdeset për dokumentimin. Kompania dëshiron të dorëzojë projektin sa më shpejt që të jetë e mundur për shkak të kërkesave të klientit, dhe gjëja e parë që bie në kullon është dokumentimi. I vetmi përjashtim ndodh në rastin kur klienti në mënyrë eksplicite përmend nevojën për dokumentim.

**Programerët janë dembelë.** Edhe me aftësi të mjaftueshme për të shkruar dokumentim të mirë, shumë programerë vendosin të mos e shkruajnë atë. Nganjëherë arsyeja pas kësaj është thjesht dembelia. Në këtë rast, programerët që motivohen vetëm nga çequet e parave janë fajtorët më të mëdhenj për të mos u përfshirë në aktivitete të cilat nuk u japin atyre më shumë para.

**Zhvillimi dhe shkrimi janë dy aftësi të ndryshme.** Imagjinoni sikur jeni një zhvillues dhe ju pëlqen të programoni gjëra. Programimi kërkon arsyetim logjik dhe hapësinor – tipare që shpesh janë më shumë matematikore se diçka tjetër. Shkrimi, nga ana tjetër, kërkon disa tipare të

ngjajshme, por ato shfaqen krejt ndryshe. Shkurtimisht, shkrimi dhe zhvillimi janë dy aftësi të ndryshme, që kërkojnë mjeshhtëri dhe procese të ndryshme të të menduarit.

**Ndryshime të shpeshta të kodit burimor.** Zhvillimi agile është mënyra më e re e zhvillimit. Është zhvillim i shpejtë, i pafalshëm, dhe nuk inkurajon dokumentim. Arsyeja është e thjeshtë; kodi ndryshon shumë shpesh që të dokumentohet dhe kështu çon në dokumentim të dobët ose në dokumentim jo-ekzistues.

**Vetëbesimi i tepruar.** Besimi i tepruar në vetvete mund të bëjë që shumë zhvillues të besojnë në mungesë të komentimit ose dokumentimit të punës së tyre. Edhe pse mund të posedojnë një kapacitet të madh për të shkruar dokumentim të mirë, shumë programerë zgjedhin të mos shkruajnë dokumentim fare në mënyrë që të tregojnë aftësitë e tyre dhe të mohojnë përgjegjësitë e tyre që të tjerët të kuptojnë kodin e tyre.

**Mosbesimi në vetvete.** Kjo pikë kundërshton fuqimisht atë të fundit. Shumë zhvillues që janë të aftë të shkruajnë dokumentim nuk e bëjnë sepse mendojnë që nuk janë të mirë në atë fushë. Ndërkaq faktori që ndikon më së shumti negativisht në fushën e dokumentimit të softuerit është ai ku zhvilluesit shumë herë humbasin në krijimin e dokumentimit dhe si rezultat krijojnë dokumentim për hir të krijimit të dokumentimit. Pra, bëjnë dokumentim vetëm që dokumentimi të ekzistojë. Mirëpo çdo dokumentim i projektit duhet të projektohet rreth një qëllimi të caktuar, ashtu që t'i ofrojë vlerë dikujt. Por për t'a arritur këtë gjithmonë duhet të pyesim veten – a ekziston ndonjë qasje më efektive dhe më efikase, pa krijuar dokumentim të panevojshëm. [11]

## **4. METODOLOGJIA**

Tutje do të bëhet analiza e një rasti studimor për një sistem ku ofrohet regjistrimi i studentëve në grupe apo kurse. Ky dokument ofron një pasqyrë gjithëpërfshirëse arkitekturale të sistemit, duke përdorur një numër të ndryshëm të pamjeve arkitekturale për të përshkruar aspekte të ndryshme të sistemit. Ky reprezentim ka për qëllim të kapë dhe të përcjellë vendimet e rëndësishme arkitekturale të cilat janë marrë në sistem.

### **Përfaqësimi Arkitektural**

Ky dokument paraqet arkitekturën si një seri pamjesh; pamjen use case, pamjen logjike, pamjen e procesit dhe pamjen e implementimit. Në këtë dokument, nuk ka të përshkruar ndonjë pamje të veçantë të zbatimit. Këto janë pamje të modelit themelor të Gjuhës së Unifikuar të Modelimit (UML).

### **Qëllimet dhe kufizimet arkitekturale**

Ekzistojnë disa kërkesa dhe kufizime kryesore të sistemit që kanë një ndikim të rëndësishëm në arkitekturë. Ato janë:

1. Sistemi ekzistues i katalogut të kurseve në kolegjin UBT duhet të qaset për të marrë të gjitha informacionet e kursit për semestrin aktual. Sistemi i regjistrimit duhet të mbështesë formatet e të dhënave dhe DBMS të sistemit të trashëguar të katalogut të kurseve.
2. Sistemi ekzistues i faturimit i trashëguar në kolegjin UBT duhet të ndërlidhet për të mbështetur faturimin e studentëve. Kjo ndërfaqe përcaktohet në Specifikimin e Ndërfaqes së Faturimit të Kursit.
3. Të gjitha funksionet e studentëve, profesorëve dhe regjistruarëve duhet të jenë të disponueshme si nga PC-të lokale të kampusit ashtu edhe nga PC-të e largëta me lidhje telefonike në internet.
4. Sistemi i regjistrimit duhet të sigurojë mbrojtje të plotë të të dhënave nga hyrja e paautorizuar. Të gjitha qasjet në distancë i nënshtrohen identifikimit të përdoruesit dhe kontrollit të fjalëkalimit.
5. Sistemi i Regjistrimit do të implementohet si një sistem klient-server.
6. Të gjitha kërkesat e performancës dhe ngarkimit duhet të merren parasysh përderisa arkitektura zhvillohet.

## **Pamja Use Case**

Pamja use case është një kontribut i rëndësishëm për zgjedhjen e grupit të skenarëve dhe/ose përdorimin e rasteve që janë në qendër të një përsëritjeje. Kjo pamje përshkruan grupin e skenarëve dhe/ose rastet e përdorimit që përfaqësojnë disa funksionalitete të rëndësishme qendrore. Gjithashtu, përshkruan skenarët dhe/ose rastet e përdorimit që kanë një mbulim të konsiderueshëm arkitekturale (që ushtrojnë shumë elemente arkitekturale) ose që theksojnë ose ilustrojnë një pikë specifike, delikate të arkitekturës [14].

Rastet e përdorimit të Regjistrimit janë:

- Hyni
- Regjistrohuni në kurse
- Mirëmbani Informacionin e Studentit
- Mirëmbani informacionin e profesorit
- Zgjidhni Kurse për të Mësuar
- Dorëzoni notat
- Shikoni Transkripten e Notave
- Mbyll Regjistrimin

Këto raste përdorimi iniciohen nga studenti, profesori ose aktorët e sistemit të regjistrimit. Përveç kësaj, ndodh edhe bashkëveprimi me aktorët e jashtëm; Katalogu i Kursit dhe Sistemi i Faturimit.

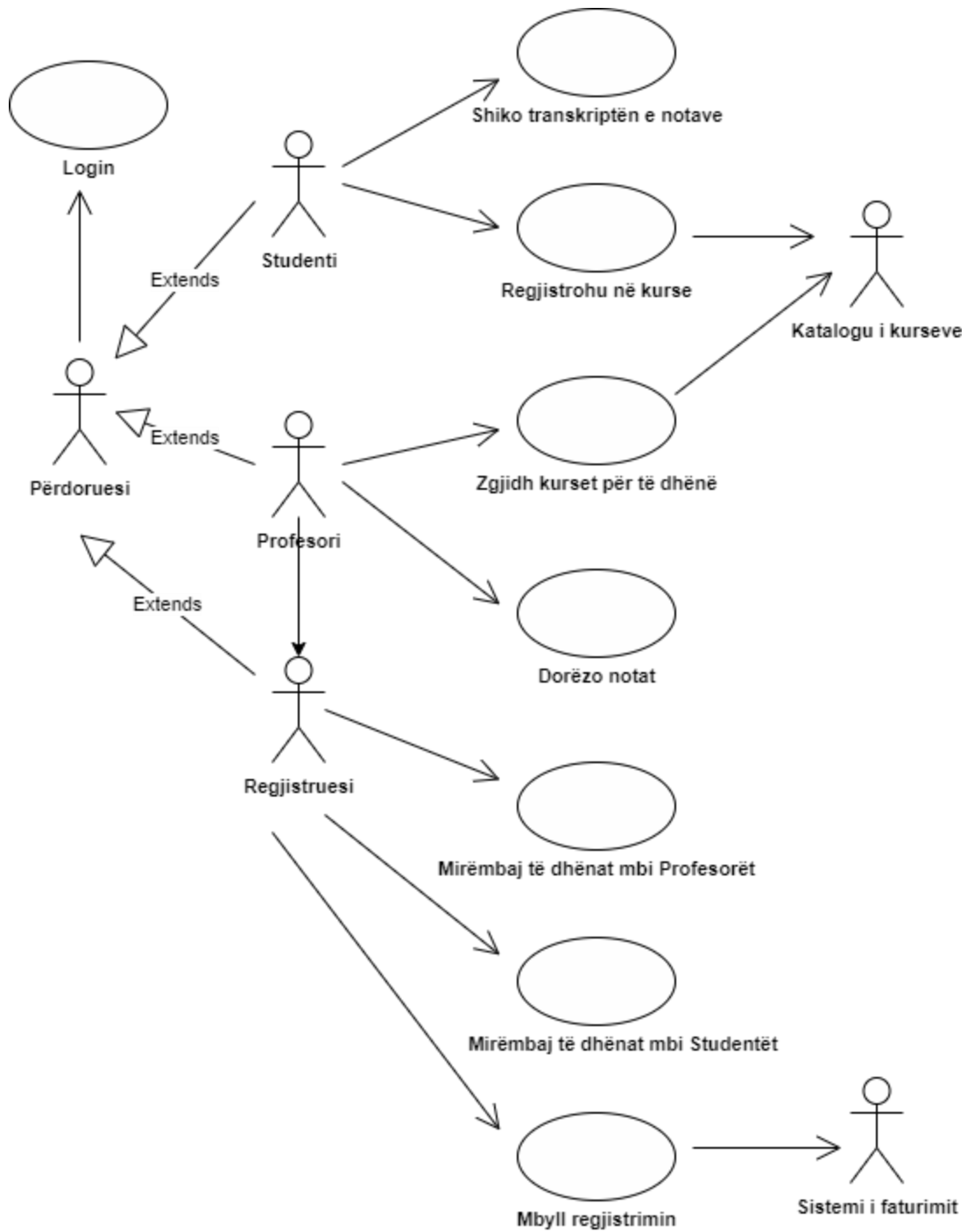


Figura 9. Rastet e rëndësishme të përdorimit nga ana Arkitekturale

91 *Mbyll regjistrimin:* Ky rast përdorimi lejon që një regjistrues të mbyllë procesin e regjistrimit. Ofertat e kurseve që nuk kanë mjaft studentë anulohen. Ofertat e kurseve duhet të kenë së paku tre studentë. Sistemi i Faturimit njoftohet për secilin student në secilën ofertë kursi që nuk është anuluar, kështu që studenti mund të faturohet për ofertën e kursit. Aktori kryesor i kësaj



çështjeje përdorimi është sekretari. Sistemi i Faturimit është një aktor i përfshirë në këtë rast përdorimi.

92 *Login:* Ky rast përdorimi përshkruan se si një përdorues hyn në Sistemin e Regjistrimit të Kursit. Aktorët që fillojnë këtë rast përdorimi janë Studenti, Profesori dhe Regjistruesi.

93 *Mirëmbaj të dhënat mbi Profesorët:* Ky rast përdorimi lejon që regjistruesi të mbajë informacionin e profesorit në sistemin e regjistrimit. Kjo përfshin shtimin, modifikimin dhe fshirjen e profesorëve nga sistemi. Aktori i këtij rasti përdorimi është Regjistruesi.

94 *Zgjidh kurset për të dhënë:* Ky rast përdorimi lejon një profesor të zgjedhë ofertat e kurseve (do të jepen kurse specifike për datën dhe kohën) nga katalogu i kursit për lëndët për të cilat ai/ajo është i kualifikuar dhe dëshiron të japë mësim në semestrin e ardhshëm. Aktori që fillon këtë rast përdorimi është Profesori. Sistemi i Katalogut të Kurseve është një aktor në rastin e përdorimit.

95 *Regjistrohu në kurse:* Ky rast përdorimi lejon një student të regjistrohet për kurse në semestrin aktual. Studenti gjithashtu mund të modifikojë ose fshijë zgjedhjet e kurseve nëse ndryshimet bëhen brenda periudhës së shtimit/rënies në fillim të semestrit. Sistemi i Faturimit njoftohet për të gjitha azhurnimet e regjistrimit. Katalogu i Kurseve ofron një listë të të gjitha ofertave të kurseve për semestrin aktual. Aktori kryesor i këtij rasti përdorimi është studenti. Sistemi i Katalogut të Kurseve është një aktor në rastin e përdorimit.

96 *Shiko transkriptën e notave:* Ky rast përdorimi lejon që një student të shikojë raportin e tij/saj për semestrin e përfunduar. Studenti është aktori i këtij rasti të përdorimit.

97 *Dorëzo notat:* Ky rast përdorimi lejon një profesor të paraqesë notat e studentëve për një ose më shumë klasa të përfunduara në semestrin e mëparshëm. Aktori në këtë rast përdorimi është Profesori.

98 *Mirëmbaj të dhënat mbi Studentët:* Ky rast përdorimi lejon që regjistruesi të mirëmbajë informacionin e studentit në sistemin e regjistrimit. Kjo përfshin shtimin, modifikimin dhe fshirjen e studentëve nga sistemi. Aktori për këtë rast përdorimi është Regjistruesi.

## **Pamja Logjike**

Përshkruan klasat më të rëndësishme, organizimin e tyre në paketat e shërbimeve dhe nënsistemet, dhe organizimin e këtyre nënsistemeve në shtresa. Gjithashtu përshkruan realizimet më të rëndësishme të rasteve të përdorimit, për shembull, aspektet dinamike të arkitekturës.

Diagramet e klasave mund të përfshihen për të ilustruar marrëdhëniet midis klasave, nënsistemeve, paketave dhe shtresave të rëndësishme arkitekturale [14].

Pamja logjike e sistemit të regjistrimit të kursit përbëhet nga 3 paketat kryesore: Ndërfaqja e përdoruesit, Shërbimet e biznesit dhe Objektet e biznesit.

Paketa e Ndërfaqes së Përdoruesit përmban klasa për secilën nga format që përdorin aktorët për të komunikuar me Sistemin. Klasat kufitare ekzistojnë për të mbështetur identifikimin, mirëmbajtjen e orareve, mirëmbajtjen e informacionit të profesorit, zgjedhjen e kurseve, dorëzimin e notave, mirëmbajtjen e informacionit të studentit, mbylljen e regjistrimit dhe shikimin e transkriptave.

Paketa e Shërbimeve të Biznesit përmban klasa kontrolli për ndërveprimin me sistemin e faturimit, kontrollimin e regjistrimit të studentit dhe menagjimin e vlerësimit të studentit.

Paketa e Objekteve të Biznesit përfshin klasat e njëjse për objektet e universitetit (d.m.th. ofrimi i kursit, orari) dhe klasat kufizuese për ndërfaqen me Sistemin e Katalogut të Kurseve.

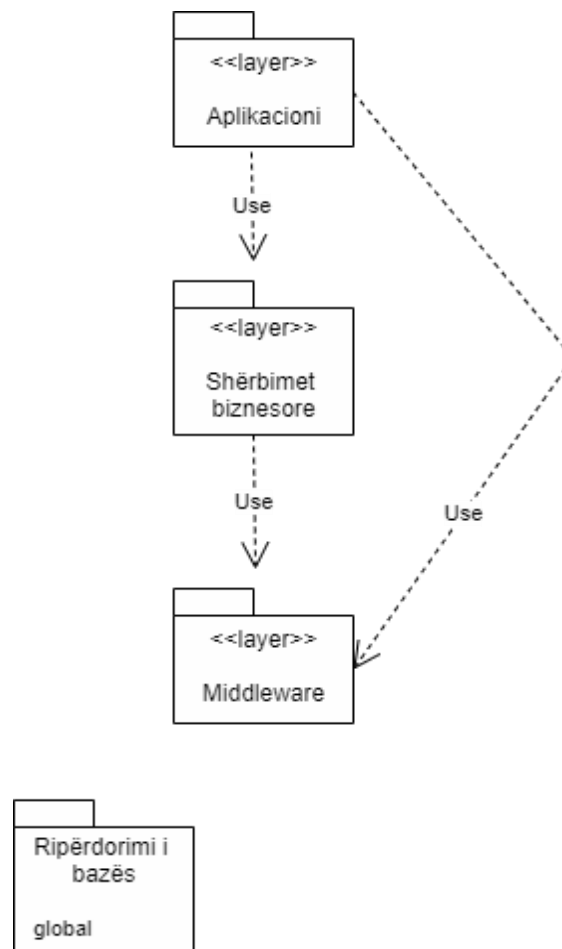


Figura 10. Pasqyra Arkitekturale

*10.1 Aplikacioni:* shtresë; Kjo shtresë e aplikacionit ka të gjitha klasat kufizuese që përfaqësojnë ekranet e aplikacioneve që përdoruesi i sheh. Kjo shtresë varet nga shtresa e Objekteve të Procesit; që ndan ndarjen e klientit nga niveli i mesëm.

*10.2 : Shërbimet biznesore:* shtresë; Shtresa e procesit të Shërbimeve të Biznesit ka të gjitha klasat e kontrolluesit që përfaqësojnë menagjerët e rasteve të përdorimit që drejtojnë sjelljen e aplikacionit. Kjo shtresë përfaqëson kufirin e klientit në mes të nivelit. Shtresa e Shërbimeve të Biznesit varet nga shtresa e Objekteve të Procesit; që ndan ndarjen e klientit nga niveli i mesëm.

*10.3 Middleware:* shtresë; Shtresa e Middleware mbështet qasjen në DBMS Relacionale dhe OODBMS.

*10.4 Ripërdorimi i bazes:* Kjo paketë përfshin klasa për të mbështetur funksionet dhe modelet e listës.

### **Pamja e Procesit**

Përshkruan detyrat (proceset dhe temat) të përfshira në ekzekutimin e sistemit, ndërveprimet dhe konfigurimet e tyre. Gjithashtu përshkruan alokimin e objekteve dhe klasave në detyra [14].

Modeli i Procesit ilustron klasat e regjistrimit të kurseve të organizuara si procese të ekzekutueshme. Ekzistojnë procese për të mbështetur regjistrimin e studentëve, funksionet e profesorëve, mbylljen e regjistrimit dhe qasjen në sistemin e jashtëm të faturimit dhe sistemin e katalogut të kurseve.

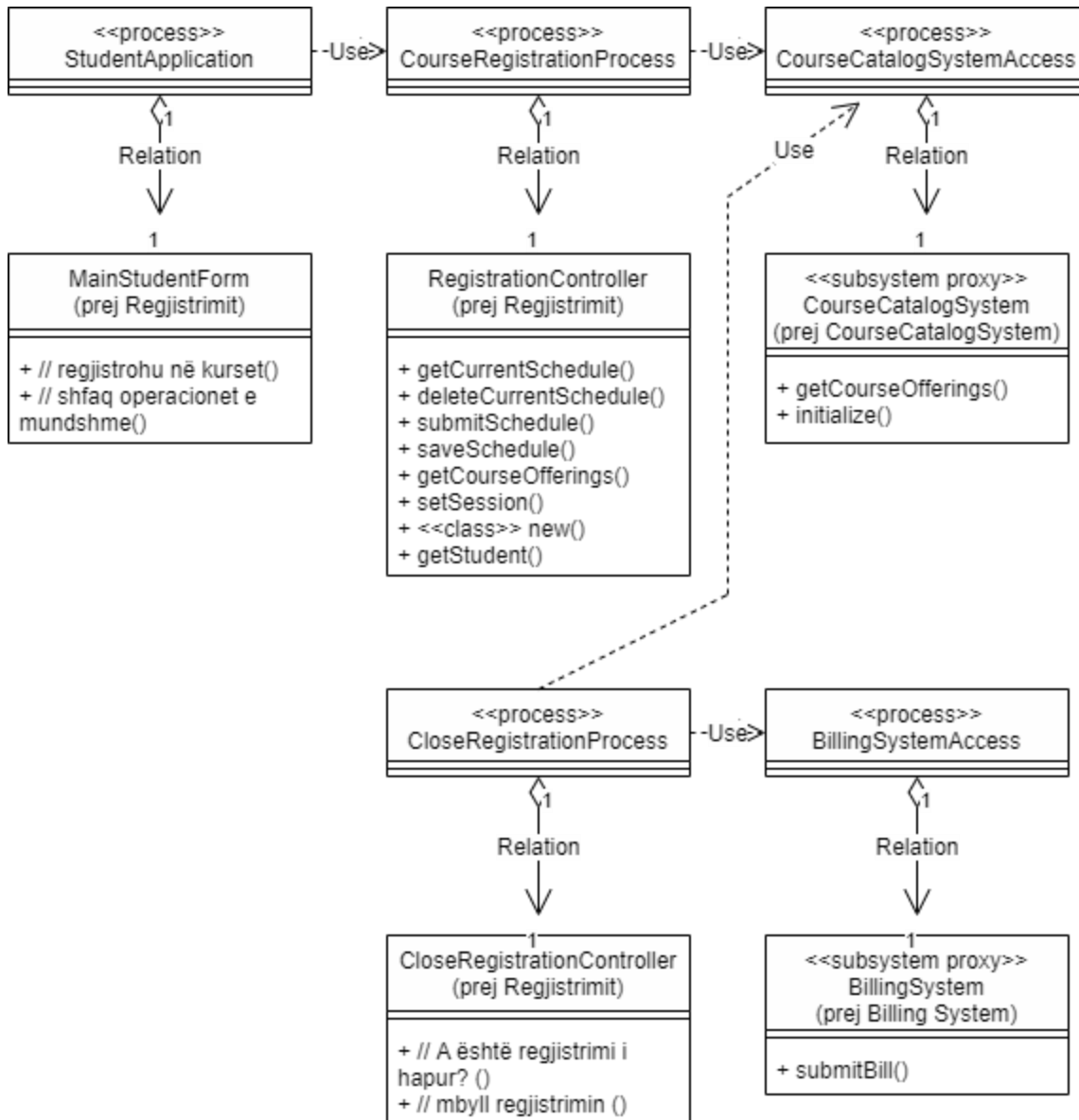


Figura 11. Diagrami i procesit

11.1 *CourseCatalogSystemAccess*: Ky proces menagjon qasjen në sistemin e katalogut të kurseve. Mund të ndahet nga përdorues të shumtë që regjistrohen në kurse. Kjo lejon një cache të kurseve dhe ofertave të marra së fundmi për të përmirësuar performancën.

Thread-at e ndara brenda procesit *CourseCatalog*, *CourseCache* dhe *OfferingCache* përdoren për të marrë itemat nga sistemi në mënyrë asinkrone.

Mekanizmat e analizës:

- Ndërfaqja e trashëgimisë

Kërkesat e Gjurmueshmërisë:

- Kufizimet e Dizajnit: Sistemi do të integrohet me sistemin ekzistues të trashëgimisë (baza e të dhënave të katalogut të kurseve).

*11.2 CourseCatalog:* Katalogu i papërmbledhur i të gjitha kurseve dhe ofertave të kurseve të ofruara nga universiteti, përfshirë ato nga semestrat e mëparshëm.

Kjo klasë vepron si një përshtatës. Funkcionon për t'u siguruar që sistemi CourseCatalogSystem mund të arrihet përmes ndërfaqes ICourseCatalog në nënsistem.

*11.3 CourseRegistrationProcess:* Ekziston një rast i këtij procesi për secilin student që regjistrohet aktualisht në kurse.

*11.4 RegistrationController:* Kjo mbështet rastin e përdorimit që lejon një student të regjistrohet në kurse në semestrin aktual. Studenti mund gjithashtu të modifikojë ose fshijë zgjedhjet e kurseve nëse ndryshimet bëhen brenda periudhës së shtimit/rënies në fillim të semestrit.

Mekanizmat e analizës:

- Shpërndarja

*11.5 StudentApplication:* Menagjon funksionalitetin e studentit, përfshirë përpunimin dhe koordinimin e ndërfaqes së përdoruesit me proceset e biznesit.

Ekziston një rast i këtij procesi për secilin student që regjistrohet aktualisht në kurse.

*11.6 MainStudentForm:* Kontrollon ndërfaqen e aplikacionit Student. Kontrollon familjen e formave që përdor Nxënësi.

*11.7 BillingSystemAccess:* Ky proces komunikon me Sistemin e jashtëm të Faturimit për të filluar faturimin e studentëve.

*11.8 CloseRegistrationProcess:* Procesi i Regjistrimit të Mbyllur fillon në fund të periudhës së regjistrimit. Ky proces komunikon me procesin që kontrollon qasjen në sistemin e faturimit.

*11.9 BillingSystem:* Sistemi i Faturimit mbështet dorëzimin e faturave të studentëve për lëndët e regjistruara nga studentit për semestrin aktual.

Mekanizmat e analizës:

- Ndërfaqja e trashëgimisë

*11.10 CloseRegistrationController:* Kontrolluesi i Mbylljes së Regjistrimit kontrollon hyrjen në sistemin e faturimit.

Mekanizmat e analizës:

- Shpërndarja

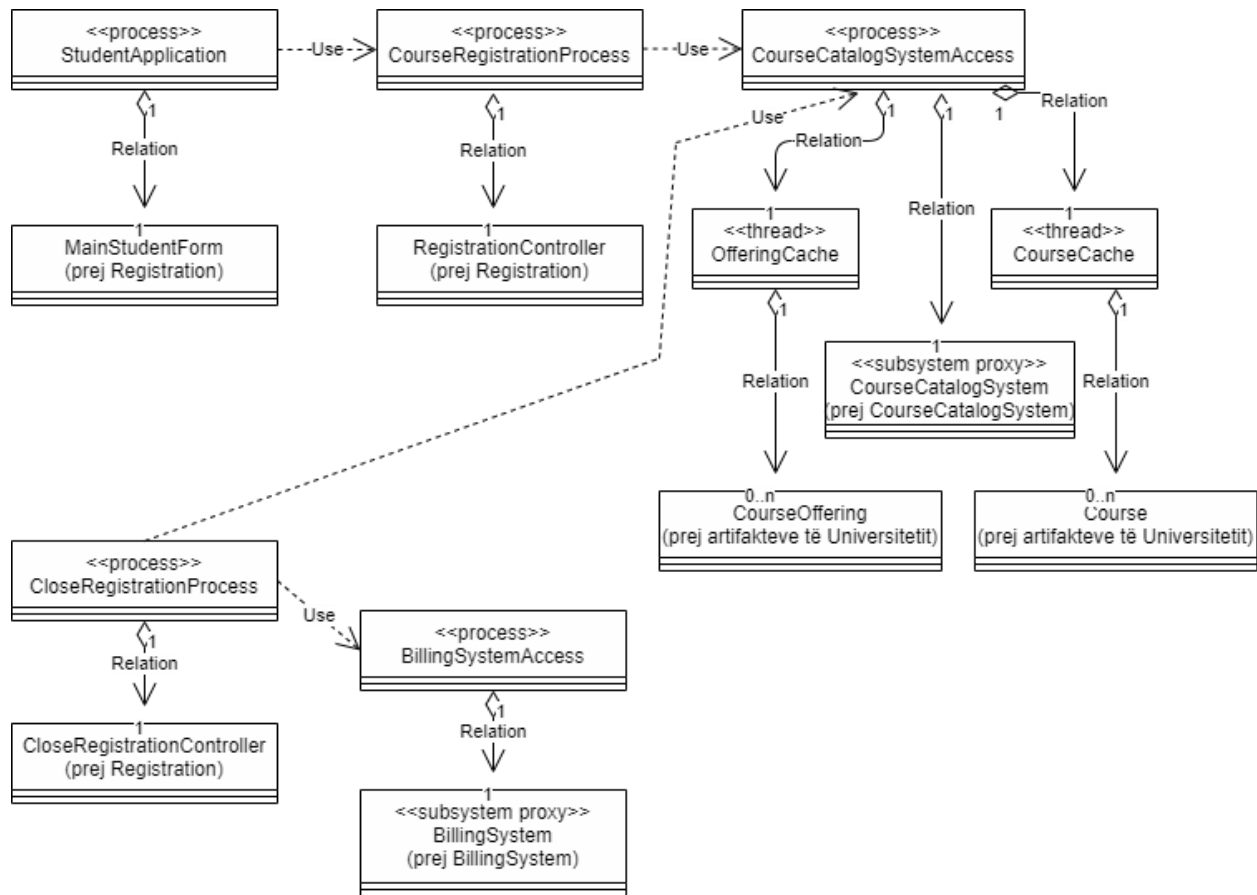


Figura 12. Elementet nga procesi në dizajn

121 *CourseCache*: Thread-i i Course Cache përdoret për të marrë itemat nga sistemi i katalogut të kurseve në mënyrë asinkrone.

122 *OfferingCache*: Thread-i OfferingCashe përdoret për të marrë itemat nga sistemi i Katalogut të Kurseve në mënyrë asinkrone.

123 *Course*: Një klasë e ofruar nga universiteti.

Mekanizmat e analizës:

- Qëndrueshmëria
- Ndërfaqja e trashëgimisë

124 *CourseOffering*: Një ofertë specifike për një kurs, përfshirë ditët dhe orët e javës.

Mekanizmat e analizës:

- Qëndrueshmëria
- Ndërfaqja e trashëgimisë

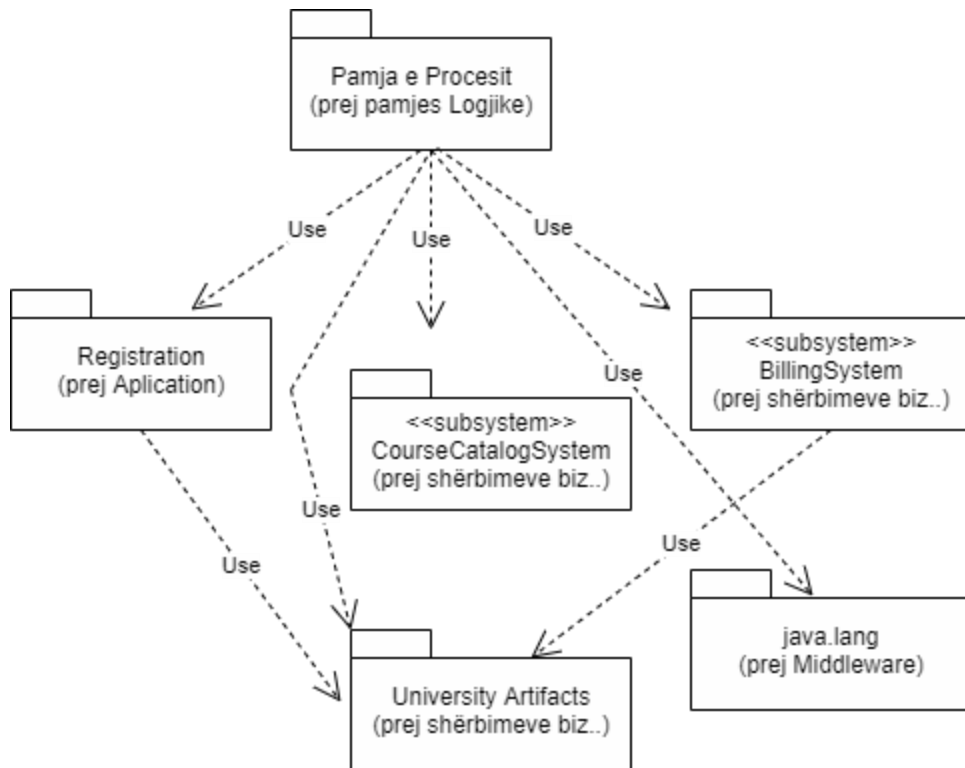


Figura 13. Modeli i procesit te dependencies e modelit të dizajnit

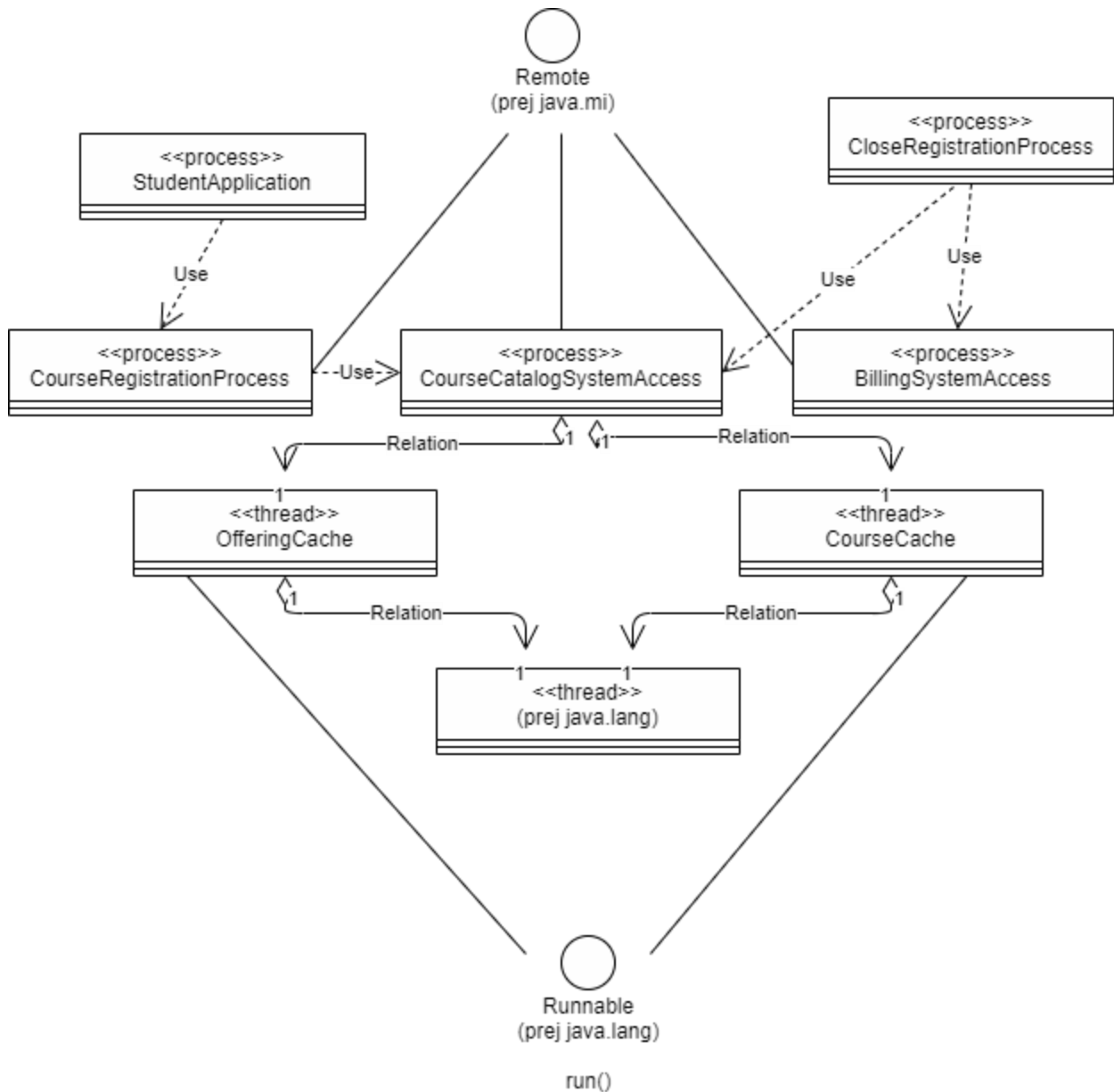


Figura 104. Proceset deri te implementimi

#### 14.1 Remote:

Ndërfaqja Remote shërben për të identifikuar të gjitha objektet e largëta. Çdo objekt që është një objekt i largët duhet të zbatojë direkt ose indirekt këtë ndërfaqe. Vetëm ato metoda të specifikuara në një ndërfaqe të largët janë në dispozicion nga distanca.

Klasat e implementimit mund të implementojnë çdo numër të ndërfaqeve në distancë dhe mund të zgjerojnë klasat e tjera të implementimit në distancë.

#### 14.2 Runnable:



Ndërfaqja Runnable duhet të implementohet nga çdo klasë, instancat e së cilës synojnë të ekzekutohen nga një thread. Klasa duhet të përcaktojë një metodë pa argumente të quajtur run.

Kjo ndërfaqe është krijuar për të siguruar një protokoll të përbashkët për objektet që dëshirojnë të ekzekutojnë kodin përdherisa janë aktiv. Për shembull, Runnable zbatohet nga klasa Thread.

Të jesh aktiv do të thotë që një thread është filluar dhe nuk është ndalur ende.

### 14.3 Thread:

Një thread është një fije ekzekutimi në një program. Java Virtual Machine lejon që një aplikacion të ketë thread-a të shumta ekzekutimi që ekzekutohen njëkohësisht.

Çdo thread ka një përparësi. Thread-at me përparësi më të lartë ekzekutohen më parë sesa thread-at me përparësi më të ulët.

## Pamja e Vendorsjes

Përshkruan nyjet e ndryshme fizike për konfigurimet më tipike të platformës. Gjithashtu përshkruan alokimin e detyrave (nga Pamja e Procesit) në nyjet fizike [14].

Ky seksion është i organizuar nga konfigurimi i rrjetit fizik; secili konfigurim i tillë ilustron nga një diagram i vendosjes, i ndjekur nga një hartëzim i proceseve për secilin procesor.

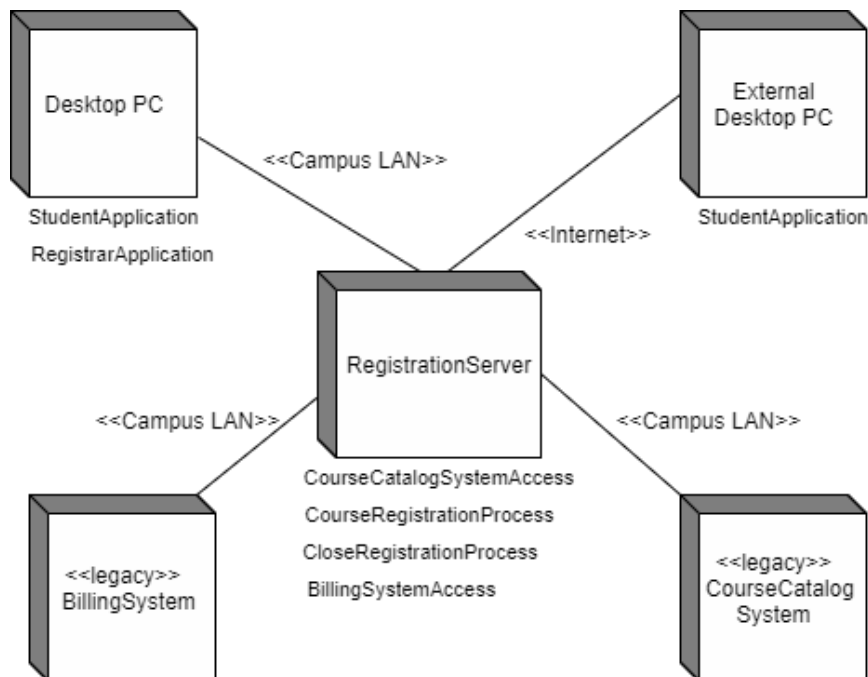


Figura 15. Pamja e vendorsjes

*15.1 External Desktop PC:* Studentët regjistrohen në kurse duke përdorur PC të jashtëm të cilët janë të lidhur me serverin e kolegjit përmes qasjes në internet.

*15.2 Desktop PC:* Studentët regjistrohen në kurse përmes PC-ve lokale që janë të lidhur drejtpërdrejt me serverin e kolegjit përmes LAN. Këta PC lokale përdoren gjithashtu nga profesorët për të zgjedhur kursin dhe për të paraqitur notat e studentëve. Regjistruesi përdor këta PC lokale për të ruajtur informacionin e studentëve dhe profesorëve.

*15.3 Registration Server:* Serveri i Regjistrimit është serveri kryesor UNIX i kampusit. Të gjithë pedagogët dhe studentët kanë qasje në server përmes LAN-it të kampusit.

*15.4 Course Catalog:* Sistemi i Katalogut të Kurseve është një sistem i trashëguar që përmban katalogun e plotë të kurseve. Aksesin në të është i disponueshëm përmes kolegjit server dhe LAN.

*15.5 Billing System:* Sistemi i Faturimit (i quajtur edhe Sistemi i Financave) është një sistem i trashëguar që gjeneron faturat e studentit çdo semestër.

## 5. REZULTATET

### Madhësia dhe Performanca

Arkitektura e zgjedhur e softuerit mbështet kërkesat kryesore të madhësisë dhe kërkesave:

- Sistemi do të mbështesë deri në 2000 përdorues të njëkohsishëm kundrejt bazës së të dhënave në çdo kohë, dhe deri në 500 përdorues të njëkohsishëm kundrejt serverave lokalë në çdo kohë.
- Sistemi do të sigurojë qasje në bazën e të dhënave të katalogut të kurseve të trashëguara me jo më shumë se 10 sekonda vonesë.
- Sistemi duhet të jetë në gjendje të përfundojë 80% të të gjitha transaksioneve brenda 2 minutash.
- Pjesa e klientit do të kërkojë më pak se 20 MB hapësirë në disk dhe 32 MB RAM.
- Arkitektura e zgjedhur mbështet kërkesat për përmasat dhe kohën përmes zbatimit të një arkitekture klient-server. Pjesa e klientit zbatohet në PC lokalë të kampusit ose në PC me distancë. Komponentët janë krijuar për të siguruar që kërkesat minimale të diskut dhe memories janë të nevojshme në pjesën e klientit me PC.

### Kualiteti

Arkitektura e softuerit mbështet kërkesat e cilësisë:

- Ndërfaqja e përdoruesit në desktop do të jetë në përputhje me sistemin operativ Windows.
- Ndërfaqja e përdoruesit të Sistemit të Regjistrimit do të projektohet për lehtësi në përdorim dhe do të jetë e përshtatshme për një komunitet përdoruesish të familjarizuar me kompjuter, pa ndonjë trajnim shtesë mbi Sistemin.
- Sistemi i Regjistrimit do të jetë i disponueshëm 24 orë në ditë, 7 ditë në javë.
- Koha mesatare ndërmjet dështimeve do të kalojë 300 orë.
- Përmirësimet në pjesën e klientit të PC-së për Regjistrim do të shkarkohen nga serveri UNIX përmes internetit. Kjo veçori u mundëson studentëve që të kenë qasje të lehtë në azhurnimet e sistemit.

## 6. DISKUTIME DHE PËRFUNDIME

Shkrimi i dokumentimit arkitektural ngjan shumë me llojet e tjera të shkrimit. Duhet kuptuar përdorimet për të cilat do të vendoset shkrimi dhe audiencia për to. Dokumentimi arkitektural shërben si mjet për komunikim midis aktorëve të ndryshëm, jo vetëm për zinxhirin e menagjimit dhe për zhvilluesit, por edhe për kolegët.

Një arkitekturë është një objekt i komplikuar, i shprehur më së miri duke u përqëndruar në perspektiva të veçanta në varësi të mesazhit që do të komunikohet. Këto perspektiva quhen pamje, dhe ju duhet të zgjidhni pamjet për të dokumentuar, duhet të zgjidhni shënimin për t'i dokumentuar ato dhe duhet të zgjidhni një grup pamjesh që janë minimale dhe adekuate. Kjo mund të përfshijë kombinimin e pamjeve të ndryshme që kanë një mbivendosje të madhe. Duhet dokumentuar jo vetëm strukturën e arkitekturës por edhe sjelljen.

Pasi të vendosen pamjet, duhet vendosur se si të pakëtohet dokumentimi. Paketimi do të varet nga media e përdorur për shprehjen e dokumentimit. Printimi ka karakteristika të ndryshme për të kuptuar dhe grupuar sesa media të ndryshme në internet. Media të ndryshme në internet do të kenë gjithashtu karakteristika të ndryshme.

Në dokumentim do të ndikojë gjithashtu konteksti i projektit. Disa nga faktorët kontekstualë janë atributet e rëndësishme të cilësisë së sistemit, shkalla e ndryshimit të sistemit dhe strategjia e menagjimit të projektit.

## 7. REFERENCAT

- [1] T. Barker, *Writing Software Documentation: A Task-Oriented Approach*, 2nd ed, Allyn and Bacon, 2003.
- [2] D. Parnas, *Document based rational software development*, Knowledge-Based Systems, 2009.
- [3] I. Sommerville, *Software Documentation*, in: *Software Engineering, Vol 2: The Supporting*, New York, USA: Wiley-IEEE Press, 2002.
- [4] L. S. R.E. Kraut, "Coordination in software development," *ACM Digital Library*, vol. 38, no. 3, pp. 69-81, 1995.
- [5] D. Parnas, *Precise Documentation: the Key to Better Software*, in: *The Future of Software*, Zürich, Switzerland: Springer, 2011.
- [6] D. Oragui, "Helpjuice," Helpjuice Inc., 2020.
- [7] I. Sommerville, *Software Engineering*, 4th edition, Addison Wesley, 1992.
- [8] I. Sommerville, *Software Engineering*, 6th Edition, Harlow, UK: London: Pearson Education Ltd, 2001.
- [9] F. Coallier, "Standards, Agility, and Engineering," 9 September 2007. [Online]. Available: <https://www.ieee.org/>. [Accessed 24 February 2021].
- [10] IEEE, "IEEE Standard for Software User Documentation," Institute of Electrical and Electronics Engineers, New York, 1987.
- [11] A. Jansen, "Journal of Systems and Software: Enriching software architecture documentation," 8 May 2009. [Online]. Available: <https://www.sciencedirect.com/>. [Accessed 27 February 2021].
- [12] P. C. R. K. Len Bass, *Software Architecture in Practice*, Pearson Education, 1997.

[13] F. B. L. B. D. G. J. I. R. L. P. M. R. N. a. J. S. Paul Clements, Documenting Software Architectures: Views and Beyond, Second Edition, Addison-Wesley, 2010.

[14] I. Gorton, Essential Software Architecture, Springer Science & Business Media, 2011, 2006.