UNIVERZITETI I EVROPËS JUGLINDORE



LËNDA: Inxhinieri softuerike

TEMA:”Sistem I menaxhimit të palestrës”

MENTOR: STUDENTËT:

Besnik Selimi Lejla Toska

Blend Rexhepi

Ridon Ramadani

Tetovë,2023

Contents

[1.OBJEKTIVI I PROEKTIT 4](#_Toc126683833)

[2.Analiza e sistemit 6](#_Toc126683834)

[2.1 Identifikimi i nevojës së klientit 7](#_Toc126683835)

[2.2 Hetimi paraprak 8](#_Toc126683836)

[2.3 Studimi i Fizibilitetit 9](#_Toc126683837)

[4. Inxhinieri Software 12](#_Toc126683838)

[Analiza e Kërkesave 13](#_Toc126683839)

[Dizajnimi i softuerit 13](#_Toc126683840)

[Kodimi 14](#_Toc126683841)

[Testimi 14](#_Toc126683842)

[Zbatimi 14](#_Toc126683843)

[**5.** **Metodologjia e zhvillimit të projekteve softuerike** 15](#_Toc126683844)

[6.DIZAJNIMI 18](#_Toc126683845)

[6.1 DIAGRAMI I RRJEDHJES SË TË DHËNAVE (DFD) 18](#_Toc126683846)

[6.2 DIZAJNI I BAZAVE TË TË DHËNAVE 18](#_Toc126683847)

[6.3 ENTITY-RELATIONSHIP DIAGRAM 21](#_Toc126683848)

[7.KODI I BURIMIT 21](#_Toc126683849)

[8.TESTIMI 25](#_Toc126683851)

[9.IMPLEMENTIMI DHE MIRËMBAJTJA 29](#_Toc126683852)

[9.1 Kërkesat e harduerit 30](#_Toc126683853)

[9.2 Kërkesat për softuer 30](#_Toc126683854)

[10.KONKLUZIONI 36](#_Toc126683855)

HYRJE

Çdo biznes duhet të ketë një faqe interneti. Arsyeja kryesore është se këto ditë shumica e njerëzve janë më shumë online dhe hulumtojnë produkte dhe kompani, përpara se të bëjnë një blerje. Për këtë arsye ky project është dizajnuar për të lehtësuar punën e palestrës për të automatizuar operacionet e mbajtjes së të dhënave dhe ruatjen e tyre ne formën e një baze të dhënash të madhe për përdoruesit, duhe lehtësuar më tej aksesin e lehtë të personelit.

Zakonisht, klienti përdor MS Excel ose letër, dhe ruan të dhënat e tyre, megjithatë nuk është e mundur që ata të ndajnë të dhënat nga një sistem i shumëfishtë në mjedisin e shumë përdoruesve, ka shumë punë të dyfishta dhe mundësi gabimi. Kur të dhënat ndryshohen, ata duhet të përditësojnë çdo skedar excel. Sistemi i Menaxhimit Smart Gym eliminon shumicën e kufizimeve të softuerit ekzistues. Rritja e efikasitetit dhe efektivitetit, automatizimi, saktësia, ndërfaqja miqësore për përdoruesit, disponueshmëria e informacionit, kapaciteti i komunikimit, mirëmbajtja, ulja e kostos e bën sistemin tonë më të zgjuar se sistemi ekzistues. Integrojmë disa veçori të reja dhe të spikatura së bashku me të gjitha veçoritë e nevojshme.

# 1.OBJEKTIVI I PROEKTIT

Cilat janë problemet ?

* Sistemi egzistues ishte manual dhe kosumonte shumë kohë për të shkrujatur të dhënat.
* Kërkimi i të dhënave të veçanta specifike për kërkesa të veçanta është gjithashtu shumë i mundimshëm në një sistem të tillë. Për të tërhequr të dhënat, personi përgjegjës duhet të gjejë manualisht regjistrin e duhur dhe të gjejë vendosjen e duhur të atij regjistrimi të veçantë i cili mund të marrë shumë kohë.
* Kërcënimi për sigurinë-është e vështirë të dihet nëse një dokument është kopjuar apo keqtrajtuar.
* E vështirë për të ndarë dokumentet-kur një kompani ka zyra në nivel kombëtar ose ndërkombëtar, aftësia për të dërguar dhe ndarë shpejt dokumentet është thelbësore.
* Mungesa e hapësirës së ruajtjes.
* Probleme me modifikimin.
* Kostoja e mbajtjes së të dhënave dhe evidencave të ndodhjes së transaksioneve është shumë e lartë.
* Teprica e të dhënave është gjithashtu një problem i madh në një sistem të tillë."Redundency" do të thotë përsëritje; Kështu të dhënat e modifikuara ose përditësuara në një vend të caktuar nuk mund të modifikohen ose përditësohen të dhënat në një vend tjetër të lidhur që mund të krijojë mospërputhje në trajtimin e të dhënave, Shkatërron Integriteti i të dhënave dhe krijon konfuzion për pronarin.

Zgjidhja e problemit:

* Sistemi i propozuar do të jetë më i shpejtë se sistemi ekzistues.
* Projekti është bërë me qëllim që të plotësojë në mënyrë efektive dhe efikase kërkesat e qendrës së fitnesit. Shumë shpesh personi që përgjithësisht mban detyrat për të menaxhuar qendrën duhet të mbajë shënime për të gjitha transaksionet si dhe të dhënat manualisht. Në përgjithësi, për të strukturuar këto detyra mbahen regjistra të veçantë. Kështu, i gjithë ky proces bëhet mjaft i rëndë për ta kontrolluar me dorë. Për më tepër, çdo e dhënë e gabuar e futur gabimisht mund të sjellë rezultate serioze.
* Softueri është mjaftueshëm i aftë që t'i lejojë personit të interesuar të ruajë dhe të marrë çdo lloj regjistrimi vetëm me një klik të vetëm të mausit. Softueri lejon një mjedis Interaktiv, Vetëpërshkrues Grafik të Ndërfaqes së Përdoruesit ku edhe përdoruesit e pavarur mund të punojnë shumë rehat dhe lehtë.
* Të gjitha të dhënat që kanë të bëjnë me transaksionet apo subjektet e tjera të rëndësishme mbahen në bazën qendrore të të dhënave nga ku mund të kontrollohen lehtësisht atributet e saj. Por, detaje të tilla teknike fshihen nga Përdoruesi i pavarur. Ai thjesht duhet të shkruajë detajet e sakta të entitetit të dhënë dhe pastaj të klikojë butonin ruaj me ndihmën e miut. Megjithatë, ajo depo qendrore e të dhënave mund të aksesohet lehtësisht nëse kërkohet.
* Teprica e të dhënave nuk është më problem tani. Të dhënat e modifikuara nga një formë e veçantë e futjes së të dhënave do të pasqyrojnë modifikimet edhe në formularët e tjerë të lidhur. Kjo ka reduktuar kështu shanset e mospërputhjes së të dhënave në ruajtjen e të dhënave tona.
* Nuk ka nevojë për të menaxhuar regjistrat e mëdhenj tani pasi të dhënat e ruajtura në bazën e të dhënave prapa mund të merren në mënyrë radikale ose nga vetë formulari frontend ose direkt nga baza e të dhënave.
* Kërkon një investim një herë për konfigurimin e Hardëare dhe Softëare-it të kërkuar pas të cilit nuk kërkohet më dhimbje koke nga menaxherët. Për më tepër, ai gjithashtu redukton varësinë nga fuqia njerëzore.
* Masat efektive të Kërkimit janë të pranishme në çdo formë të transaksionit të të dhënave nga ku, duke futur vetëm një fjalë kyçe unike për ato të dhëna, të gjitha të dhënat e tij mund të shihen lehtësisht brenda mikrosekondave. Për më tepër, Lehtësia e Përditësimit dhe Fshirjes së të dhënave përmes kërkimit është gjithashtu e disponueshme.

# 2.Analiza e sistemit

Analiza e sistemit i referohet procesit të shqyrtimit të një situate me synimin për ta përmirësuar atë përmes procedurave dhe metodave më të mira. Analiza e Sistemit është procesi i planifikimit të një sistemi të ri për të zëvendësuar ose plotësuar një sistem ekzistues. Por, përpara se të bëhet ndonjë planifikim, sistemi i vjetër duhet të kuptohet plotësisht dhe të përcaktohen kërkesat. Analiza e sistemit, pra, është procesi i mbledhjes dhe interpretimit të fakteve, diagnostikimit të problemeve dhe përdorimit të informacionit për të rikomentuar përmirësimet në Sistem. Ose me fjalë të tjera, Analiza e Sistemit nënkupton një shpjegim ose përshkrim të hollësishëm. Para se të kompjuterizohet një sistem në shqyrtim, ai duhet të analizohet. Ne duhet të studiojmë se si funksionon aktualisht, cilat janë problemet dhe cilat janë kërkesat që duhet të plotësojë sistemi i propozuar.

*Analiza e sistemit kryhet duke pasur parasysh objektivat e mëposhtëm:*

1. Identifikoni nevojën e klientit.

2. Vlerësoni konceptin e sistemit për fizibilitet.

3. Kryen analiza ekonomike dhe teknike.

4. Shpërndani funksionet për harduerin, njerëzit e softuerit, bazën e të dhënave dhe elementët e tjerë të sistemit.

5. Vendosni kufizime të kostos dhe planit.

6. Krijo një përkufizim të sistemit që formon themelin për të gjithë punën inxhinierike të mëvonshme.

## 2.1 Identifikimi i nevojës së klientit

Përpara se të vazhdohet më tej, është shumë e nevojshme të grumbullohen kërkesat e vlefshme dhe bindëse të projektit dhe të njëjtat t'u komunikohen palëve të ndryshme të interesit të projektit.

Ky hap është fillimi i Analizës së Sistemit. Është bërë një pasqyrë e kërkesës së klientit. Analizohet nevoja themelore e klientit për të zgjedhur një lloj projekti të tillë. Sipas skenarit aktual të marketingut, kërkohej një sistem i tërë për të gjurmuar transaksionet e përditshme.

Klienti po ndiqte një proces manual, i cili nuk përputhet aspak me kushtet aktuale të punës. Jo vetëm që kërkonte kohë, por i mungonte edhe saktësia. Nga pikëpamja e sigurisë, sistemi manual nuk arriti të fshihte informacionin nga ndonjë personel i paautentikuar ose ndonjë person i jashtëm. Prandaj, ekzistonte një kërkesë urgjente për një sistem të tillë kompjuterik i cili mund të plotësojë të gjitha kërkesat e tij aktuale dhe të ardhshme. Për më tepër, trajtimi i të dhënave ishte gjithashtu një problem serioz për ta.

## 2.2 Hetimi paraprak

Kompleti i klientit është thjesht një punëtor(a), i cili rregullisht kënaqet me transaksionet e mirëmbajtjes manuale, duke mbajtur shënime të rregullta, duke mbajtur të dhënat e detajeve të imta të anëtarëve.

Regjistrat e mëposhtëm manual mbahen:

1. Regjistri i detajeve të anëtarëve:

Ky regjistër mbahet në liu për të ruajtur të dhënat e anëtarëve të ndryshëm të qendrës së palestrës. Dokumenti përmban informacione përkatëse për anëtarët e ndryshëm, si ID-ja e anëtarëve, Emri, Adresa, Numri i telefonit.

1. Regjistri i të dhënave të punonjësve:

Ky regjistër mbahet në liu për të ruajtur të dhënat e punonjësve të ndryshëm të qendrës së palestrës. Dokumenti përmban informacione përkatëse për punonjësit e ndryshëm si ID e punonjësit, Emri, Adresa, Numri i telefonit.

1. Regjistri i inventarit:

Regjistri përdoret për të regjistruar detajet e produkteve (suplementet, pijet dhe veshjet e furnizuara dhe të kërkuara) si ID e tij, Përshkrimi, Sasia, Çmimi, data e shërbimit, periudha kohore për të cilën është nën mirëmbajtje etj.

## 2.3 Studimi i Fizibilitetit

Në varësi të rezultateve të hetimit fillestar, sondazhi zgjerohet në një studim fizibiliteti më të detajuar. Studimi i fizibilitetit është një test i propozimit të sistemit sipas funksionueshmërisë së tij, ndikimit në organizatë, aftësisë për të përmbushur nevojat e përdoruesve dhe përdorimit efektiv të burimeve. Objektivi i studimit të fizibilitetit nuk është të zgjidhë problemin, por të kuptojë qëllimin e tij. Gjatë studimit kristalizohet përkufizimi i problemit dhe përcaktohen aspektet e problemit që do të përfshihen në sistem. Rrjedhimisht, kostot dhe përfitimet përshkruhen me saktësi më të madhe në këtë fazë.

Ai përbëhet nga sa vijon:

1. Deklarata e problemit:

Një deklaratë e formuluar me kujdes e problemit që çoi në analizë.

2. Përmbledhje e gjetjeve dhe rekomandimeve:

Një listë e gjetjeve dhe rekomandimeve kryesore të studimit. Është ideal për përdoruesit që kërkon qasje të shpejtë në rezultatet e analizës së sistemit në studim. Janë deklaruar konkluzioni, pasuar nga një listë e rekomandimeve dhe një arsyetim për to.

3. Detajet e gjetjeve:

Një përmbledhje e metodave dhe procedurave të ndërmarra nga sistemi ekzistues, i ndjekur nga mbulimi i objektivave dhe procedurave të sistemit kandidat. Të përfshira janë gjithashtu diskutimet e raporteve të prodhimit, strukturat e skedarëve dhe kostot dhe përfitimet e sistemit kandidat.

4. Rekomandime dhe përfundime:

Rekomandime specifike në lidhje me sistemin e kandidatëve, duke përfshirë detyrat e personelit, kostot, oraret e projektit dhe datat e synuara.

**2.3.1 Fizibiliteti Teknik**

Fizibiliteti teknik përqendrohet rreth sistemit ekzistues kompjuterik (Hardëare dhe Softëare etj) dhe deri në çfarë mase ai mbështet shtimin e propozuar. Për shembull, nëse kompjuteri aktual funksionon me kapacitet 80 përqind - atëherë ekzekutimi i një aplikacioni tjetër mund të mbingarkojë sistemin ose të kërkojë pajisje shtesë. Kjo përfshin konsiderata financiare për të akomoduar përmirësimet teknike. Nëse buxhetet janë një kufizim serioz, atëherë projekti gjykohet i pazbatueshëm. Në këtë projekt janë bërë të gjitha kujdeset e nevojshme për ta bërë atë teknikisht të realizueshëm. Duke përdorur një çelës, shfaqja e tekstit/objektit është shumë e shpejtë. Gjithashtu, mjetet, sistemi operativ dhe gjuha programuese e përdorur në këtë proces lokalizimi është në përputhje me atë ekzistuese.

**2.3.2 Fizibiliteti ekonomik**

Analiza ekonomike është metoda më e përdorur për vlerësimin e efektivitetit të sistemit kandidat. E njohur më shpesh si analiza kosto/përfitim, procedura është të përcaktojë përfitimet dhe kursimet që priten nga një kandidat dhe t'i krahasojë ato me kostot. Nëse përfitimet peshojnë më shumë se kostot, atëherë merret vendimi për të hartuar dhe zbatuar sistemin.

Një përfitim financiar i sistemit duhet të tejkalojë koston e zhvillimit të atij sistemi. dmth një sistem i ri që po zhvillohet duhet të jetë një investim i mirë për organizatën. Fizibiliteti ekonomik merr parasysh sa vijon:

1. Kostoja për të kryer një hetim të plotë të sistemit.
2. Kostoja e harduerit dhe softuerit për klasën e aplikacionit.
3. Përfitimet në formën e kostos së reduktuar ose gabimeve më pak të kushtueshme.
4. Kostoja nëse asgjë nuk ndryshon (d.m.th. sistemi i propozuar nuk është zhvilluar).

SISTEMI i propozuar është ekonomikisht i realizueshëm sepse

1. Sistemi kërkon shumë më pak faktorë kohe.
2. Sistemi do të sigurojë një mjedis të shpejtë dhe efikas të automatizuar në vend të sistemit manual të ngadaltë dhe të prirur për gabime, duke reduktuar kështu kohën dhe fuqinë e njeriut të shpenzuar në funksionimin e sistemit.
3. Sistemi do të ketë ndërfaqe GUI dhe kërkohet shumë më pak trajnim i përdoruesit për ta mësuar atë.
4. Sistemi do të ofrojë shërbimin për të parë informacione të ndryshme nëse kërkohet për një vendimmarrje.

**2.3.3 Fizibiliteti operacional**

Ky aplikacion është shumë i lehtë për t'u përdorur pasi është bërë miqësor me përdoruesit me ndihmën e mjeteve shumë efektive GUI. Konsiderata kryesore është qasja e lehtë e përdoruesit në të gjithë funksionalitetin e Aplikacionit. Një tjetër konsideratë kryesore këtu është nëse organizata e përdoruesit është e trajnuar mjaftueshëm për të përdorur aplikacionin më të ri. Këtu çdo funksionalitet është sipas strategjisë së mëparshme operative e cila nuk pritet të jetë e rëndë për klientët potencial.

**2.3.4 Fizibiliteti i sjelljes**

Njerëzit janë në thelb rezistent ndaj ndryshimit dhe kompjuterët dihet se lehtësojnë ndryshimin. Duhet bërë një vlerësim se sa i fortë ka të ngjarë të ketë reagimi i stafit përdorues ndaj zhvillimit të një sistemi të kompjuterizuar. Prandaj është e kuptueshme që futja e një sistemi kandidat kërkon përpjekje të veçanta për edukimin dhe trajnimin e stafit. Softueri që po zhvillohet është miqësor për përdoruesit dhe i lehtë për t'u mësuar. Në këtë mënyrë, softueri i zhvilluar është vërtet efikas dhe mund të funksionojë në çdo rrethanat, traditat, vendet.

Studimi i sjelljes përpiqet të sigurojë që ekuilibri i organizatës dhe status quo-ja në organizatë nuk janë të shqetësuara dhe ndryshimet pranohen lehtësisht nga përdoruesit.

**2.4 Funksionaliteti i propozuar i sistemit**

Sistemi i propozuar do të projektohet për të mbështetur karakteristikat e mëposhtme:

• Sistemi i propozuar ka një ndërfaqe miqësore për përdoruesit për transferimin e të dhënave në server.

• Sistemi i propozuar ofron lehtësinë për të tërhequr të dhënat nga serveri duke përdorur një çelës (siç është id) dhe për të marrë raportin e dëshiruar.

• Sistemi i propozuar nuk siguron replikim të të dhënave

# Inxhinieri Software

Objektivi themelor i inxhinierisë softuerike është: të zhvillojë metoda dhe procedura për zhvillimin e softuerit që mund të zgjerohet për sisteme të mëdha dhe që mund të përdoren për të prodhuar vazhdimisht softuer me cilësi të lartë me kosto të ulët dhe me një kohë të vogël cikli. Kjo do të thotë, objektivat kryesore janë qëndrueshmëria, kostoja e ulët, cilësia e lartë, koha e ciklit të vogël dhe shkallëzueshmëria.

Qasja bazë që merr inxhinieria e softuerit është të ndajë procesin e zhvillimit nga softueri. Premisa është se procesi i zhvillimit kontrollon cilësinë, shkallëzueshmërinë, qëndrueshmërinë dhe produktivitetin. Prandaj, për të përmbushur objektivat, duhet fokusuar në procesin e zhvillimit. Projektimi i procesit të duhur të zhvillimit dhe kontrolli i tyre është qëllimi kryesor i inxhinierisë softuerike. Është ky fokus në proces që e dallon atë nga shumica e disiplinave të tjera të informatikës. Shumica e disiplinave të tjera kompjuterike fokusohen në disa lloje të algoritmeve të produkteve, sistemeve operative, bazave të të dhënave etj., ndërsa inxhinieria e softuerit fokusohet në procesin e prodhimit të produkteve. Për të menaxhuar më mirë procesin e zhvillimit dhe për të arritur qëndrueshmëri, është thelbësore që zhvillimi i softuerit të bëhet në faza.

Fazat e ndryshme të procesit të zhvillimit:

## Analiza e Kërkesave

Analiza e kërkesave bëhet për të kuptuar problemin që duhet të zgjidhë sistemi softuer. Problemi mund të jetë automatizimi i një procesi manual ekzistues, zhvillimi i një sistemi të ri të automatizuar ose një kombinim i të dyjave. Theksi në analizën e kërkesave është në identifikimin e asaj që nevojitet nga sistemi, jo se si sistemi do t'i arrijë qëllimet e tij. Ka të paktën dy palë të përfshira në zhvillimin e softuerit - një klient dhe një zhvillues. Zhvilluesi duhet të zhvillojë sistemin për të kënaqur nevojat e klientit. Zhvilluesi nuk e kupton domenin e problemit të klientit dhe klienti nuk i kupton çështjet e përfshira në sistemet e softuerit. Kjo shkakton një hendek komunikimi, i cili duhet të kapërcehet në mënyrë adekuate gjatë analizës së kërkesave.

## Dizajnimi i softuerit

Qëllimi i fazës së projektimit është të planifikojë një zgjidhje të problemit të specifikuar nga dokumentet e kërkesave. Kjo fazë është hapi i parë për të kaluar nga fusha e problemit në domenin e zgjidhjes. Duke filluar me atë që nevojitet, dizajni na çon drejt mënyrës se si të kënaqim nevojat. Dizajni i një sistemi është ndoshta faktori më kritik që ndikon në cilësinë e softuerit. Ai ka një ndikim të madh në fazat e mëvonshme, veçanërisht në testimin dhe mirëmbajtjen.

Aktiviteti i projektimit ndahet në dy faza: Projektimi i Sistemit dhe Projektimi i Detajuar. Në dizajnimin e sistemit fokusi është në identifikimin e moduleve, ndërsa gjatë dizajnit të detajuar fokusi është në dizajnimin e logjikës për secilin prej moduleve.

## Kodimi

Qëllimi i fazës së kodimit është përkthimi i dizajnit të sistemit në kod në një gjuhë të caktuar programimi. Prandaj gjatë kodimit, fokusi duhet të jetë në zhvillimin e programeve që janë të lehta për t'u lexuar dhe kuptuar, dhe jo thjesht në zhvillimin e programeve që janë të lehta për t'u shkruar.

## Testimi

Testimi është masa kryesore e kontrollit të cilësisë që përdoret gjatë zhvillimit të softuerit. Funksioni i tij themelor është të zbulojë gabimet në softuer. Testimi jo vetëm që zbulon gabimet e paraqitura gjatë kodimit, por edhe gabimet e paraqitura gjatë fazave të mëparshme. Kështu, qëllimi i testimit është të zbulojë kërkesat, gabimet e projektimit dhe kodimit në programe. Prandaj, përdoren nivele të ndryshme testimi. Testimi është një aktivitet jashtëzakonisht kritik dhe kërkon kohë. Kërkon planifikim të duhur të procesit të përgjithshëm të testimit. Rezultati i fazës së testimit është raporti i testit dhe raporti i gabimit. Raporti i testit përmban grupin e rasteve të testimit dhe rezultatin e ekzekutimit të kodit me këto raste testimi. Raporti i gabimit përshkruan gabimet e hasura dhe veprimet e ndërmarra për të hequr gabimet.

## Zbatimi

Është procesi i kontrollit të personelit të sistemeve dhe vënies në përdorim të pajisjeve të reja, trajnimit të përdoruesve, instalimit të aplikacionit të ri dhe ndërtimit të çdo skedari të të dhënave të nevojshme për ta përdorur atë. Kjo fazë është më pak kreative sesa dizajni i sistemit. Në varësi të madhësisë së organizatës që do të përfshihet në përdorimin e aplikacionit dhe rrezikut të përfshirë në përdorimin e tij, zhvilluesit e sistemeve mund të zgjedhin të testojnë funksionimin vetëm në një zonë të firmës me vetëm një ose dy persona. Ndonjëherë, ata do të drejtojnë sistemin e vjetër dhe të ri në mënyrë paralele për të krahasuar rezultatet.

# Metodologjia e zhvillimit të projekteve softuerike

Metodologjia:

Më caktuan detyrën për zhvillimin e një sistemi të kompjuterizuar për një palestër. Koha dhe burimet e projektit ishin shumë të kufizuara. Përdorimi optimal i kohës praktike kërkon që çdo seancë dhe çdo aktivitet të planifikohet. Për planifikim efektiv, kërkohen aftësi efikase menaxheriale të projektit, aftësi efikase, më pas gjurmoni metodologjinë më të mirë për t'u përdorur.

Metodologjia e përdorur nga unë përfshin gjërat e mëposhtme:

**Ndarja modulare – Ndarja e Sistemit:**

• Identifikoni modulet e ndryshme në sistem.

• Rendisni ato në hierarkinë e duhur.

• Identifikoni prioritetin e tyre të zhvillimit.

Module të ndryshme në sistemin e kompjuterizuar

Modulet – Ky projekt përbëhet nga ndërfaqe të ndryshme të cilat do të aksesohen përmes një dritareje MDI (Multiple Document Interface). Module të ndryshme që përbëjnë këtë sistem janë përshkruar tashmë në analizën e sistemit.

• Moduli 1: Moduli i Anëtarit

Në këtë modul, Pronari mund të shohë detajet e anëtarësimit që përfshijnë:

DETAJET E ANËTARIT

* 1. ID-ja e anëtarit
  2. Emri
  3. Gjinia

DETAJET E ANËTARËSISË

* 1. Lloji i anëtarësimit
  2. Data e skadencës
  3. Adresë
  4. Data e lindjes
  5. Numri i telefonit

INFORMACION I LLOGARISË

* 1. Data e duhur
  2. Sasia e detyruar për pagesë
  3. Shuma e paguar
  4. Ditë me vonesë
  5. Bilanci

Pas përpunimit, të gjitha shënimet e informacionit ruhen dhe në vetë këtë modul ka një dispozitë për gjenerimin e raporteve për shikimin e detajeve të të gjithë anëtarëve.

Ekziston gjithashtu një dispozitë për gjenerimin e një faturë për një anëtar individual, duke dhënë kështu përsëri disa nga detajet e tij përkatëse.

• Moduli 2: Moduli i punonjësve

Në këtë modul, pronari mund të shohë detajet e punonjësit duke përfshirë fushat:

1. ID
2. Emri
3. Gjinia
4. Paga
5. Adresa
6. Data e lindjes
7. Numri i telefonit
8. Llogaritja e listës së pagave
   1. Data e pagesës
   2. ID
   3. Emri
   4. Orët e punës
   5. Paga bruto
   6. Taksa
   7. Paga neto

Së bashku me këtë ekziston një dispozitë për caktimin e orarit të punonjësve me informacione si ora dhe dita e ofruar dhe pronari mund të shikojë orarin apo edhe ta ndryshojë atë.

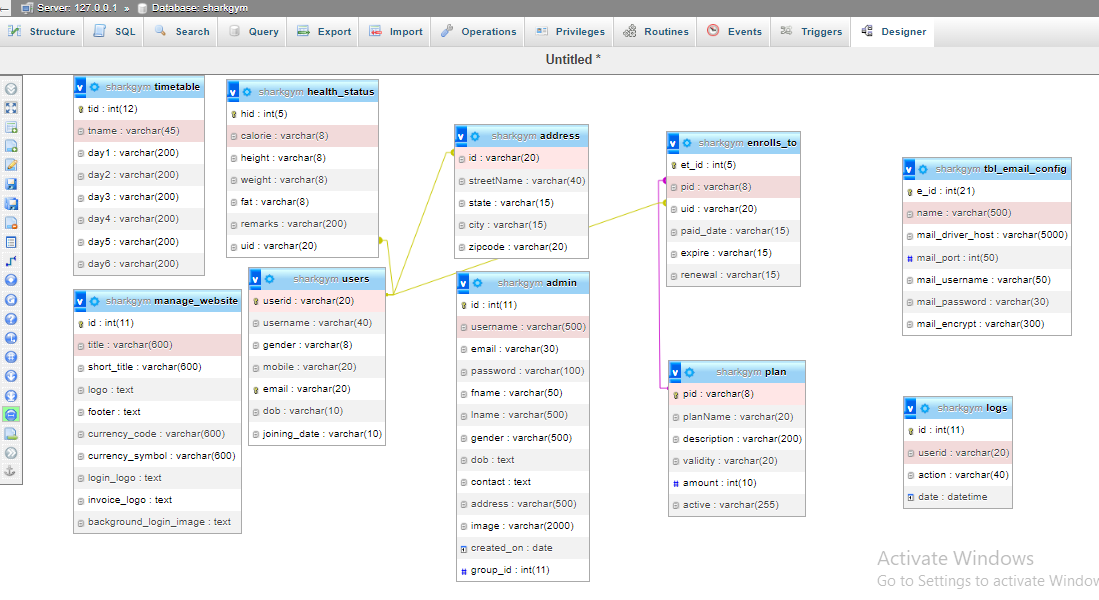
• Moduli 3: Moduli i inventarit

Më tej ndahet në tre komponentë:

1. INVENTORI

Jep përshkrim rreth një produkti bazuar në kategorinë e tij dhe përfshin:

1. Kategoria
2. Id
3. Përshkrimi
4. Marka
5. Sasia
6. Data



# 6.DIZAJNIMI

## 6.1 DIAGRAMI I RRJEDHJES SË TË DHËNAVE (DFD)

DFD është një model, i cili jep një pasqyrë të fushës së informacionit dhe fushës funksionale në të njëjtën kohë. DFD është rafinuar në nivele të ndryshme. Sa më i rafinuar të jetë DFD, aq më shumë detaje të sistemit janë përfshirë. Në procesin e krijimit të një DFD, ne e zbërthejmë sistemin në nënsisteme të ndryshme funksionale. Përsosja e DFD rezulton në një përsosje përkatëse të të dhënave.

Më poshtë është DFD e “Sistemit të propozuar”. Ne e kemi rafinuar sistemin deri në dy nivele. Çdo ndarje është numëruar sipas rregullit të DFD. Jemi përpjekur të inkorporojmë të gjitha detajet e sistemit, por ka disa mundësi për improvizim të mëtejshëm për shkak të studimit që është ende në vazhdim për zhvillimin e projektit.

Niveli i kontekstit ose DFD i nivelit zero

Ky nivel tregon kontekstin e përgjithshëm të sistemit dhe mjedisin e tij operativ dhe tregon të gjithë sistemin si vetëm një proces.

**Diagrami i kontekstit ose DFD e nivelit zero.**

## 6.2 DIZAJNI I BAZAVE TË TË DHËNAVE

Prezentimi

• Baza e të dhënave

Një bazë të dhënash është mbledhja sistematike e të dhënave të lidhura logjikisht me një qëllim të natyrshëm për të aksesuar dhe për të funksionuar në të dhëna të ndryshme të ruajtura në formën e regjistrave në mënyrë që të rritet lehtësia dhe efikasiteti në trajtimin e të dhënave.

• Baza e të dhënave relacionale

Është ai që punon mbi specifikimin e modelit relacional, i cili është një model bazë të dhënash i bazuar në logjikën e kallëzuesit të rendit të parë, i formuluar dhe propozuar për herë të parë në 1969 nga Edgar Codd.

Ideja e saj thelbësore është të përshkruajë një bazë të dhënash si një koleksion kallëzuesish mbi një grup të kufizuar variablash kallëzues, duke përshkruar kufizimet në vlerat e mundshme dhe kombinimet e vlerave. Përmbajtja e bazës së të dhënave në çdo kohë të caktuar është një model (logjik) i fundëm i bazës së të dhënave, d.m.th. një grup relacionesh, një për variabël kallëzues, të tillë që të plotësohen të gjithë kallëzuesit. Një kërkesë për informacion nga baza e të dhënave (një pyetje e bazës së të dhënave) është gjithashtu një kallëzues.

Qëllimi i modelit relacional është të ofrojë një metodë deklarative për specifikimin e të dhënave dhe pyetjeve: ne deklarojmë drejtpërdrejt se çfarë informacioni përmban baza e të dhënave dhe çfarë informacioni duam prej saj, dhe lejojmë që softueri i sistemit të menaxhimit të bazës së të dhënave të kujdeset për përshkrimin e strukturave të të dhënave për ruajtjen e të dhënat dhe procedurat e rikthimit për marrjen e përgjigjeve të pyetjeve.

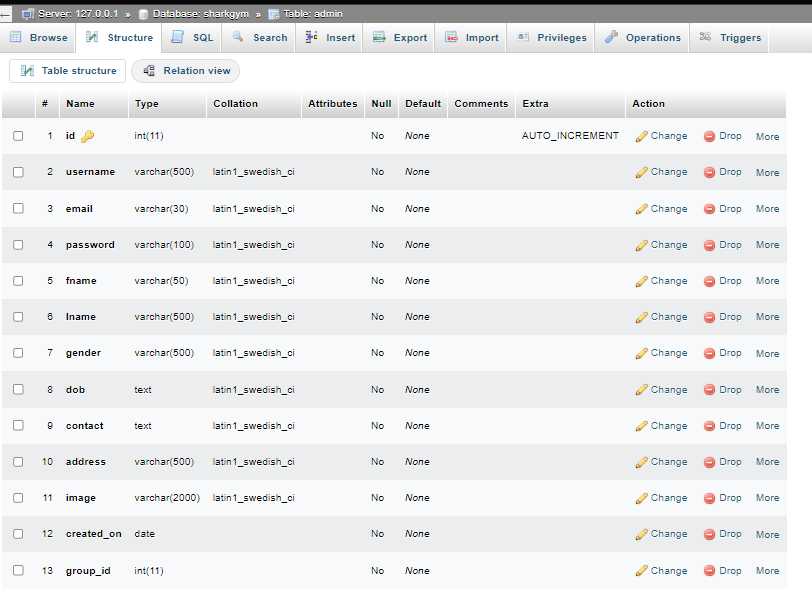
• Tabelat e bazës së të dhënave

Një tabelë është një grup elementesh të dhënash (vlerash) që organizohet duke përdorur një model të kolonave vertikale (të cilat identifikohen me emrin e tyre) dhe rreshtave horizontale. Një tabelë ka një numër të caktuar kolonash, por mund të ketë çdo numër rreshtash.

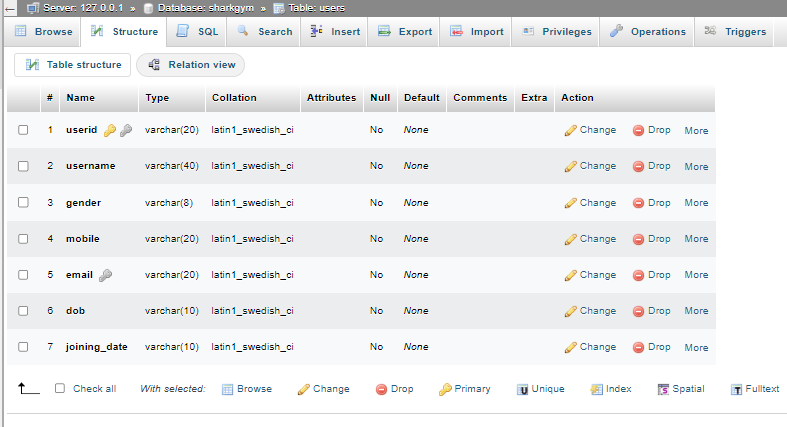
Lista e tabelave në bazën e të dhënave:

Emri i bazës së të dhënave është "sharkgym.mdb"

Tabela 1: Admin



**Table 2: User**



## ENTITY-RELATIONSHIP DIAGRAM

# 7.KODI I BURIMIT

*INDEX.PHP*

<?php

session\_start();

error\_reporting(1);

include('head.php');

include('./constant/connect.php');

if(isset($\_POST['btn\_login']))

{

$unm = $\_POST['email'];

$passw = hash('sha256', $\_POST['password']);

function createSalt()

{

return '2123293dsj2hu2nikhiljdsd';

}

$salt = createSalt();

$pass = hash('sha256', $salt . $passw);

$sql = "SELECT \* FROM admin WHERE email='" .$unm . "' and password = '". $pass."'";

$result = mysqli\_query($con,$sql);

$row = mysqli\_fetch\_array($result);

$\_SESSION["id"] = $row['id'];

$\_SESSION["username"] = $row['username'];

$\_SESSION["password"] = $row['password'];

$\_SESSION["email"] = $row['email'];

$\_SESSION["fname"] = $row['fname'];

$\_SESSION["lname"] = $row['lname'];

$\_SESSION["image"] = $row['image'];

$count=mysqli\_num\_rows($result);

if(($count)==1 && isset($\_SESSION["email"]) && isset($\_SESSION["password"])) {

{

?>

<div class="popup popup--icon -success js\_success-popup popup--visible">

<div class="popup\_\_background"></div>

<div class="popup\_\_content">

<h3 class="popup\_\_content\_\_title">

Success

</h1>

<p>Login Successfully</p>

<p>

<?php echo "<script>setTimeout(\"location.href = './admin/dashboard.php';\",1500);</script>"; ?>

</p>

</div>

</div>

*LOGIN PHP*

*<?php*

*include './constant/connect.php';*

*$user\_id\_auth = ltrim($\_POST['user\_id\_auth']);*

*$user\_id\_auth = rtrim($user\_id\_auth);*

*$pass\_key = ltrim($\_POST['pass\_key']);*

*$pass\_key = rtrim($\_POST['pass\_key']);*

*$user\_id\_auth = stripslashes($user\_id\_auth);*

*$pass\_key = stripslashes($pass\_key);*

*if($pass\_key=="" && $user\_id\_auth==""){*

*echo "<head><script>alert('Username and Password cannot be empty');</script></head></html>";*

*echo "<meta http-equiv='refresh' content='0; url=index.php'>";*

*}*

*else if($pass\_key=="" ){*

*echo "<head><script>alert('Password cannot be empty');</script></head></html>";*

*echo "<meta http-equiv='refresh' content='0; url=index.php'>";*

*}*

*else if($user\_id\_auth=="" ){*

*echo "<head><script>alert('Username cannot be empty');</script></head></html>";*

*echo "<meta http-equiv='refresh' content='0; url=index.php'>";*

*}*

*else{*

*$user\_id\_auth = mysqli\_real\_escape\_string($con, $user\_id\_auth);*

*$pass\_key = mysqli\_real\_escape\_string($con, $pass\_key);*

*$sql = "SELECT \* FROM admin WHERE username='$user\_id\_auth' and pass\_key='$pass\_key'";*

*$result = mysqli\_query($con, $sql);*

*$count = mysqli\_num\_rows($result);*

*if ($count == 1) {*

*$row = mysqli\_fetch\_assoc($result);*

*session\_start();*

*$\_SESSION['user\_data'] = $user\_id\_auth;*

*$\_SESSION['logged'] = "start";*

*$\_SESSION['full\_name'] = $user\_id\_auth;*

*$\_SESSION['username']=$row['Full\_name'];?>*

*<div class="popup popup--icon -success js\_success-popup popup--visible">*

*<div class="popup\_\_background"></div>*

*<div class="popup\_\_content">*

*<h3 class="popup\_\_content\_\_title">*

*Success*

*</h1>*

*<p>Login Successfully</p>*

*<p>*

*<?php echo "<script>setTimeout(\"location.href = './admin/dashboard.php';\",1500);</script>"; ?>*

*</p>*

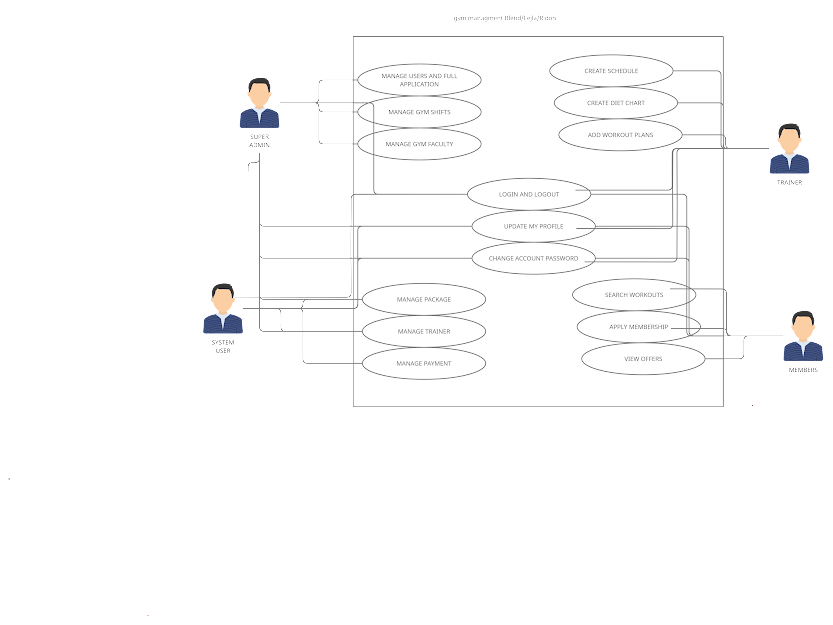
*</div>*

*</div>*

*<?php }*

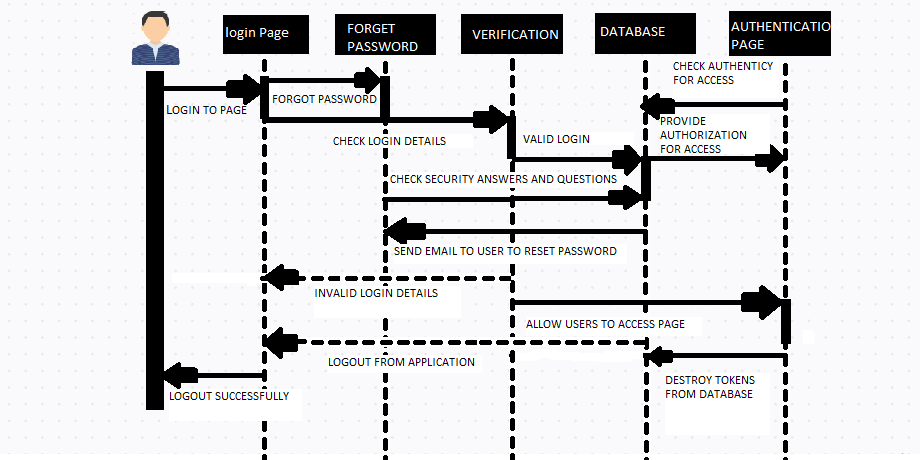
Use Case Diagram

Ky diagram i Rastit të Përdorimit është një përshkrim grafik i ndërveprimeve midis elementeve të sistemit të menaxhimit të palestres. Ai përfaqëson metodologjinë e përdorur në analizën e sistemit, qartësimin dhe organizimin e kërkesave të sistemit të sistemit të menaxhimit të palestrës. Aktorët kryesorë nëse sistemi i menaxhimit të palestrës në këtë diagram rasti përdorimi janë super administratori, përdoruesi i sistemit, trajneri, anëtarët që kryejnë llojet e ndryshme të rasteve të përdorimit si menaxhimin e palestrës, menaxhimin e ndërrimit të palestrës, menaxhimin e paketës, menaxhimin e stërvitjes etj.



Sequence Diagram:

Ky është diagrami i sekuencës së hyrjes së sistemit të menaxhimit të palestrës, ku administratori do të jetë në gjendje të identifikohet në llogarinë e tij duke përdorur kredencialet e tij pasi përdoruesi i hyrjes mund të menaxhojë të gjitha operacionet në ndërrimin e palestrës, trajnerin, degën, paketën, diagrami më poshtë ndihmon për të demonstruar se si identifikimi faqja punon në një sistem të menaxhimit të palestër. ata ndërveprojnë gjatë rrjedhës së sekuencës dhe përdoruesi nuk do të jetë në gjendje të hyjë në këtë faqe pa verifikuar identitetin e tyre



# 8.TESTIMI

Nivelet e testimit:

Sistemet nuk janë të dizajnuara si sisteme të tëra dhe as nuk testohen si sisteme të vetme. Analisti duhet të kryejë testimin e njësisë dhe të sistemit.

Testimi i njësisë:

Në testimin e njësive, analisti teston programet që përbëjnë një sistem. Për këtë arsye, testimi i njësisë nganjëherë quhet testimi i programit. Testimi i njësisë i jep stres moduleve në mënyrë të pavarur nga njëri-tjetri, për të gjetur gabime. Kjo ndihmon testuesin në zbulimin e gabimeve në kodim dhe logjikë që gjenden vetëm brenda atij moduli. Gabimet që rezultojnë nga ndërveprimi ndërmjet moduleve fillimisht shmangen. Rastet e testimit të nevojshme për testimin e njësisë duhet të ushtrojnë çdo kusht dhe opsion.

Testimi i njësisë mund të kryhet nga poshtë lart, duke filluar me modulet më të vogla dhe të nivelit më të ulët dhe duke vazhduar një nga një. Për çdo modul në testimin nga poshtë-lart, përdoret një program i shkurtër për të ekzekutuar modulin dhe siguron të dhënat e nevojshme, në mënyrë që moduli të kërkohet të performojë ashtu siç do të jetë kur të futet brenda sistemit më të madh.

Testimi i sistemit:

Pjesa e rëndësishme dhe thelbësore e fazës së zhvillimit të sistemit, pas dizajnimit dhe zhvillimit të softuerit është testimi i sistemit. Nuk mund të themi se çdo program ose dizajn i sistemit është i përsosur dhe për shkak të mungesës së komunikimit midis përdoruesit dhe projektuesit, ka ndonjë gabim në zhvillimin e softuerit. Numri dhe natyra e gabimeve në një sistem të ri të projektuar varet nga disa faktorë të zakonshëm si komunikimi ndërmjet përdoruesit dhe projektuesit; aftësia e programuesit për të gjeneruar një kod që pasqyron saktësisht specifikimet e sistemeve dhe kornizën kohore për dizajnin.

Teorikisht, një sistem i ri i projektuar duhet të ketë të gjitha pjesët ose nënsistemet të jenë në gjendje pune, por në realitet, çdo nënsistem funksionon në mënyrë të pavarur. Kjo është koha për të mbledhur të gjithë nënsistemin në një grup dhe për të testuar të gjithë sistemin për të përcaktuar nëse i plotëson kërkesat e përdoruesit. Ky është ndryshimi i fundit për të zbuluar dhe korrigjuar gabimet përpara se sistemi të instalohet për testimin e pranimit të përdoruesit. Qëllimi i testimit të sistemit është të marrë në konsideratë të gjitha variacionet e mundshme ndaj të cilave do t'i nënshtrohet dhe më pas ta shtyjë sistemin në kufijtë e tij.

Testimi është një funksion i rëndësishëm për suksesin e sistemit. Testimi i sistemit bën një supozim logjik që nëse të gjitha pjesët e sistemit janë të sakta, qëllimi do të aktivizohet me sukses. Një arsye tjetër për testimin e sistemit është përdorimi i tij si mjet i orientuar nga përdoruesi përpara zbatimit.

Testimi i sistemit përbëhet nga pesë hapat e mëposhtëm:

1) Testimi i programit

Një program përfaqëson elementet logjike të një sistemi. Që një program të funksionojë në mënyrë të kënaqshme, ai duhet të përpilojë dhe testojë saktë të dhënat dhe të lidhet siç duhet me programet e tjera. është përgjegjësi e një programuesi që të ketë një program pa gabime. Në kohën e testimit të sistemit, ekzistojnë dy lloje gabimesh që duhet të kontrollohen. Këto gabime janë sintaksore dhe logjike. Një gabim sintaksor është një deklaratë programi që shkel një ose më shumë rregulla të gjuhës në të cilën është shkruar. Një dimension i përcaktuar gabimisht i fushës ose fjalë kyçe të hequra janë gabime të zakonshme sintaksore. Këto gabime shfaqen përmes mesazheve të gabimit të krijuara nga kompjuteri. Një gabim logjik, nga ana tjetër, ka të bëjë me fusha të pasakta të të dhënave jashtë gamës së artikujve dhe kombinime të pavlefshme. Meqenëse gabimet logjike nuk zbulohen nga përpiluesi, programuesi duhet të ekzaminojë me kujdes daljen për t'i zbuluar ato.

Kur testohet një program, prodhimi aktual krahasohet me daljen e pritur. Kur ka një mospërputhje, sekuenca e udhëzimeve, duhet të gjurmohet për të përcaktuar problemin. Procesi lehtësohet duke e zbërthyer programin në pjesë të pavarura, secila prej të cilave mund të kontrollohet në pika të caktuara kyçe.

2) Testimi i vargut

Programet janë të lidhura pa ndryshim me njëri-tjetrin dhe ndërveprojnë në një sistem total. Çdo program testohet për të parë nëse përputhet me programet përkatëse në sistem. Secila pjesë e sistemit testohet kundrejt të gjithë modulit me të dhëna testuese dhe të drejtpërdrejta përpara se i gjithë sistemi të jetë gati për t'u testuar.

3) Testimi i Sistemit

Testimi i sistemit është krijuar për të zbuluar dobësitë që nuk janë gjetur në testet e mëparshme. Kjo përfshin dështimin e detyruar të sistemit dhe vlefshmërinë e sistemit total pasi ai do të zbatohet nga përdoruesi i tij në mjedisin operacional. Sipas këtij testimi, përgjithësisht ne marrim vëllime të ulëta transaksionesh bazuar në të dhëna të drejtpërdrejta. Ky vëllim rritet derisa të arrihet niveli maksimal për çdo lloj transaksioni. Sistemi total testohet gjithashtu për rikuperim dhe rikthim pas dështimeve të ndryshme të mëdha për të siguruar që të mos humbasin të dhëna gjatë urgjencës. E gjithë kjo bëhet me sistemin e vjetër ende në funksion. Kur shohim që sistemi i propozuar është i suksesshëm në provë, sistemi i vjetër ndërpritet.

4) Dokumentacioni i Sistemit

I gjithë dokumentacioni i projektimit dhe testimit duhet të përgatitet mirë dhe të ruhet në bibliotekë për referencë në të ardhmen. Biblioteka është lokacioni qendror për mirëmbajtjen e sistemit të ri.

5) Testimi i pranimit të përdoruesit

Një test pranimi ka për qëllim shitjen e përdoruesit mbi vlefshmërinë dhe besueshmërinë e sistemit. Ai verifikon që procedurat e sistemit funksionojnë sipas specifikimeve të sistemit dhe se ruhet integriteti i të dhënave të rëndësishme. Kryerja e një testi pranimi është në të vërtetë shfaqja e përdoruesit. Motivimi i përdoruesit është shumë i rëndësishëm për performancën e suksesshme të sistemit. Pas kësaj përgatitet një raport gjithëpërfshirës i testit. Ky raport tregon tolerancën e sistemit, diapazonin e performancës, shkallën e gabimit dhe saktësinë.

Testet e sistemeve speciale:

Ka gjashtë teste të tjera të cilat i përkasin kategorisë së veçantë. Ato përshkruhen më poshtë:

1. Testi i ngarkesës maksimale: Përcakton nëse sistemi do të trajtojë vëllimin e aktiviteteve që ndodhin kur sistemi është në kulmin e kërkesës së tij për përpunim. Për shembull, testoni sistemin duke aktivizuar të gjithë terminalet në të njëjtën kohë.
2. Testimi i ruajtjes: Përcakton kapacitetin e sistemit për të ruajtur të dhënat e transaksionit në një disk ose në skedarë të tjerë. Për shembull, verifikoni deklaratat e dokumentacionit që sistemi do të ruajë 10,000 regjistrime me gjatësi 400 bajt në një disk të vetëm fleksibël.
3. Testimi i kohës së performancës: përcakton kohëzgjatjen e sistemit të përdorur nga sistemi për të përpunuar të dhënat e transaksionit. Ky test kryhet përpara zbatimit për të përcaktuar se sa kohë duhet për të marrë një përgjigje ndaj një hetimi, për të bërë një kopje rezervë të një skedari ose për të dërguar një transmetim dhe për të marrë një përgjigje.
4. Testimi i Rimëkëmbjes: Ky testim përcakton aftësinë e përdoruesit për të rikuperuar të dhënat ose për të rifilluar sistemin pas dështimit. Për shembull, ngarkoni kopjen rezervë të të dhënave dhe rifilloni përpunimin pa humbje të të dhënave ose integritetit.
5. Testimi i procedurës: Përcakton qartësinë e dokumentacionit mbi funksionimin dhe përdorimin e sistemit duke i detyruar përdoruesit të bëjnë pikërisht atë që kërkojnë manualet. Për shembull, fikja e sistemit në fund të javës ose reagimi ndaj dritës së shkarkimit të letrës në printer.
6. Testimi i Faktorëve Njerëzor: Përcakton se si përdoruesit do ta përdorin sistemin kur përpunojnë të dhënat ose përgatitin raporte.

# 9.IMPLEMENTIMI DHE MIRËMBAJTJA

Pas përfundimit të procesit të paketimit dhe medias së prodhuar të shpërndarjes për aplikacionin, aplikacioni kërkon funksionimin e përsosur të Microsoft Visual Studio 6.0 të instaluar në sistemin e klientit së bashku me Ms Office Access. Mund të funksionojë në të gjitha sistemet operative të aplikueshme.

## 9.1 Kërkesat e harduerit

Hardëare është termi që i jepet vetë makinerive dhe pjesëve të ndryshme individuale të pajisjeve. I referohet pajisjeve fizike të një sistemi kompjuterik. Kështu, pajisjet hyrëse, ruajtëse, të kontrollit të përpunimit dhe të daljes janë harduer.

Kërkesa minimale e harduerit nga ana e klientit:

Procesor: Çdo makinë Pentium ose ekuivalente

RAM: 256 MB

HDD: 1.2 GB

CD-ROM: 32X

Monitor me ngjyra 14 inç

Tastiera 104 taste

Printeri: DeskJet 670 C

## 9.2 Kërkesat për softuer

Softëare nënkupton një koleksion programesh ku objektivi është të përmirësojë aftësitë e makinës harduerike.

Kërkesa minimale e softuerit nga ana e klientit:

Sistemi operativ: Ëindoës 9x/NT/2000/Xp

Baza e të dhënave: Microsoft Office Access

Pjesa e përparme: Visual Basic 6.0

**ZBATIMI**

Pasi sistemi u testua, filloi faza e zbatimit. Një fazë vendimtare në ciklin jetësor të zhvillimit të sistemit është zbatimi i suksesshëm i dizajnit të sistemit të ri. Zbatimet thjesht nënkuptojnë shndërrimin e dizajnit të ri të sistemit në funksionim. Ky është momenti i së vërtetës pyetja e parë që bie në mendjen e secilit nëse sistemi do të jetë në gjendje t'i japë të gjitha dëshirat rezultate siç priten nga sistemi. Faza e zbatimit ka të bëjë me trajnimin e përdoruesve dhe konvertimin e skedarëve.

Termi implementim ka kuptime të ndryshme, duke filluar nga konvertimi i një aplikacioni bazë deri te një zëvendësim i plotë i sistemit kompjuterik. Implementimi përdoret këtu për të nënkuptuar procesin e konvertimit të një dizajni të ri ose të rishikuar të sistemit në një dizajn operacional. Konvertimi është një aspekt i zbatimit. Aspektet e tjera janë rishikimi pas zbatimit dhe mirëmbajtja e softuerit. Ekzistojnë tre lloje të zbatimit:

Zbatimi i një sistemi kompjuterik për të zëvendësuar një sistem manual

Zbatimi i një sistemi të ri kompjuterik për të zëvendësuar një ekzistues.

Zbatimi i një aplikacioni të modifikuar për të zëvendësuar një ekzistues.

**Konvertimi**

Konvertimi nënkupton ndryshimin nga një sistem në tjetrin. Objektivi është të vihet në funksion sistemi i testuar duke mbajtur në minimum kostot, rreziqet dhe acarimin e personelit. Ai përfshin krijimin e skedarëve të përputhshëm me kompjuterin; trajnimin e stafit operativ; instalimi i terminaleve dhe pajisjeve. Një aspekt kritik i konvertimit nuk është prishja e funksionimit të organizatës.

**Zbatimi i drejtpërdrejtë**

Në zbatim të drejtpërdrejtë; sistemi i mëparshëm ndalet dhe sistemi i ri niset rastësisht. Këtu ka një ndryshim të drejtpërdrejtë nga sistemi manual në sistem të bazuar në kompjuter. Në ndryshim të drejtpërdrejtë mbi zbatimin; punëmarrësit mund të përballen me problemet. Supozoni se softueri ynë nuk funksionon me shumë efikasitet si manual, atëherë ne nuk mund të gjejmë defekte në softuerin tonë. Nuk do të jetë e dobishme në gjetjen e gabimeve.

Sistemi i propozuar zbatohet plotësisht duke përdorur Implementimin Direkt.

**MIRËMBAJTJA**

Pasi softueri dorëzohet dhe zhvillohet, ai hyn në fazën e mirëmbajtjes. Të gjitha sistemet kanë nevojë për mirëmbajtje. Softueri duhet të mirëmbahet sepse shpesh ka disa gabime ose gabime të mbetura në sistem që duhet të hiqen kur zbulohen. Shumë nga këto sipërfaqe vetëm pasi sistemi ka qenë në funksion ndonjëherë për një kohë të gjatë. Këto gabime sapo të zbulohen duhet të hiqen, duke çuar në ndryshimin e softuerit. Megjithëse mirëmbajtja nuk është pjesë e zhvillimit të softuerit, është një aktivitet jashtëzakonisht i rëndësishëm në jetën e një produkti softuer.

Mirëmbajtja përfshin të kuptuarit e softuerit ekzistues (kodi dhe dokumentet përkatëse), të kuptuarit e efekteve të ndryshimit, duke bërë ndryshime - si në kod ashtu edhe në dokumente - testimin e pjesëve të reja dhe ritestimin e pjesës së vjetër.

# 10.KONKLUZIONI

Objektivi i këtij projekti ishte ndërtimi i një programi për ruajtjen e detajeve të të gjithë anëtarëve, punonjësve dhe inventarit. Sistemi i zhvilluar është në gjendje të plotësojë të gjitha kërkesat bazë. Menaxhimi i të dhënave (si anëtarët ashtu edhe punonjësit) do të përfitohet gjithashtu nga sistemi i propozuar, pasi do të automatizojë të gjithë procedurën, gjë që do të reduktojë ngarkesën e punës. Siguria e sistemit është gjithashtu një nga shqetësimet kryesore.

Gjithmonë ka një vend për përmirësim në çdo softuer, sado efikas të jetë sistemi. Gjëja e rëndësishme është që sistemi duhet të jetë mjaft fleksibël për modifikimet e ardhshme. Sistemi është faktorizuar në module të ndryshme për ta bërë sistemin të përshtatet me ndryshimet e mëtejshme. Është bërë çdo përpjekje për të mbuluar të gjitha kërkesat e përdoruesit dhe për ta bërë atë miqësore.

Qëllimi i arritur: Sistemi është në gjendje t'i sigurojë pronarit ndërfaqen në mënyrë që ai të mund të përsërisë të dhënat e tij të dëshiruara. .

Miqësia ndaj përdoruesit: Megjithëse pjesa më e madhe e sistemit supozohet të veprojë në sfond, janë bërë përpjekje për të bërë sa më të qetë ndërveprimin në plan të parë me përdoruesin (pronarin). Gjithashtu integrimi i sistemit ekzistues me projektin është mbajtur parasysh gjatë gjithë fazës së zhvillimit.