****

**Fakulteti i Shkencave dhe Teknologjive Bashkëkohore**

**Tetovë**

**Software Engeenering**

**(Sem. 5, 2022/2023)**

**Projekti: TaxiTaxi**

**Studentët**: **Mentor**:

1. 129237, Elham Tairovski Prof. dr. Besnik Selimi
2. 129403, Berat Sherifi
3. 129502, Ardian Beqiri
4. 129444, Semin Amiti

Shkurt 2023, Tetovë

Table of Contents

[1. Hyrje 4](#_Toc126946517)

[1.1 Qellimi 4](#_Toc126946518)

[1.2 Audienca e synuar 4](#_Toc126946519)

[1.3 Përdorimi i caktuar 4](#_Toc126946520)

[1.4 Fusheveprimi (Scope) 4](#_Toc126946521)

[1.5 Nevojat e përdoruesit 5](#_Toc126946522)

[1.6 Supozimet dhe varësitë 5](#_Toc126946523)

[2. Karakteristikat dhe kërkesat e sistemit 5](#_Toc126946524)

[2.1 Kërkesat Funksionale 5](#_Toc126946525)

[2.2 Karakteristikat e sistemit 5](#_Toc126946526)

[3. Objektivat e Projektit 6](#_Toc126946527)

[4. Organizimi 6](#_Toc126946528)

[1. 4.1 Organizimi 6](#_Toc126946529)

[1. 4.2 Dokumentimi dhe Prezantimi 7](#_Toc126946530)

[1. 4.3 Zbatimi 7](#_Toc126946531)

[5. Procesi i zhvillimit 7](#_Toc126946532)

[6. Komunikimi 9](#_Toc126946533)

[6.1 Takimet 9](#_Toc126946534)

[6.1.1 Takimet ne Discord 9](#_Toc126946535)

[6.1.2 Takime me mbikëqyrësin e projektit 9](#_Toc126946536)

[7. Analiza e sistemit 10](#_Toc126946537)

[7.1 Feasibility Study: 10](#_Toc126946538)

[7.2 Economic Feasibility: 10](#_Toc126946539)

[7.3 Technical Feasibility: 10](#_Toc126946540)

[7.4 Behavioral Feasibility: 10](#_Toc126946541)

[8. Dizajnimi I aplikacionit Taxi 11](#_Toc126946542)

[8.1 Sequence Diagrams 13](#_Toc126946543)

[9. Modeli konceptual i të dhënave: 14](#_Toc126946544)

[10. Zbatimi I Aplikacionit Mobile dhe Web Aplication 15](#_Toc126946545)

[10.1 Moduli i Regjistrimit 15](#_Toc126946546)

[10.2Moduli i Porosisë së Vendit 16](#_Toc126946547)

[10.3 Taxi Locator 16](#_Toc126946548)

[10.4 Moduli i Kontaktit 17](#_Toc126946549)

[11. Testimi i projektit: 18](#_Toc126946550)

[11.1 Llojet e testeve të softuerit: 18](#_Toc126946551)

[11.2 Pse testimi I sistemit? 19](#_Toc126946552)

[11.3 Testimi I sistemeve 20](#_Toc126946553)

[11.4 Testimi I pranimit 20](#_Toc126946554)

[12. Verifikimi dhe vleresimi (V&V) 21](#_Toc126946555)

[13. Mirembajtja pas shqyrtimit dhe implementimit 22](#_Toc126946556)

[13.1 Faktorët e Vlerësimit të Hardware 22](#_Toc126946557)

[13.2 Faktorët e Vlerësimit të Softuerit 22](#_Toc126946558)

[13.3 Konvertimi dhe Trajnimi 23](#_Toc126946559)

[14 Kontrollet e sigurise dhe vlefshmerise 24](#_Toc126946560)

[15. Konklusioni 25](#_Toc126946561)

# Hyrje

Ky dokument paraqet një plan të detajuar të projektit të një projekti të shërbimit taksi. Ky dokument përcaktohet në fillim të projektit dhe do të rishikohet sa herë që ndodh një ndryshim në organizimin e projektit. Çdo ndryshim në afatet, dokumentet ose pjesët e programit që supozohet të dorëzohen do të dokumentohet gjithashtu në një nga rishikimet. Çdo dokument tjetër i dorëzuar duhet të bazohet në përmbajtjen e këtij dokumenti.

## 1.1 Qellimi

TaxiTaxi është një kompani udhëtimi që ofron nje Mobile applikacion dhe nje web-faqe , të cilin mund ta përdorin për të paraqitur një kërkesë udhëtimi që i dërgohet automatikisht Applikacionit tone dhe ne do ta drejtojme drejte shoferit , duke njoftuar shoferin për vendndodhjen tuaj. Shoferi i Taksisë pranuese do të vijë më pas, do t'ju marrë dhe do t'ju çojë në destinacionin e kërkuar.

## 1.2 Audienca e synuar

Për klientët që kanë nevojë për taksi, sa më shpejt të jetë e mundur duke kontrolluar se cila taksi është më afër,Aplikacioni do te mundsoje shoferët te cilet jane me afer tu dergoje klienteve per ti permbushur kerkesen e udhetimit.

## 1.3 Përdorimi i caktuar

Softueri dhe Web-faqja që mendojmë të ndërtojmë është për Aplikacionin Taxi, për të marrë një udhëtim, duhej të ishe pranë një linje autobusi ose metroje në kohën e duhur ose të ishe në një zonë ku ekzistonte një taksi. Kjo mund t'i bëjë gjërat sfiduese për ata që jetojnë në zona pa shumë transport publik. Me Aplikacionin Taxi tani mund të bëni një udhëtim kudo, në çdo kohë.

## 1.4 Fusheveprimi (Scope)

Ky dokument jep informacion të përgjithshëm për projektin e shërbimit Taxi. Ai përshkruan organizimin e ekipit dhe procesin e zhvillimit në përgjithësi. Në këtë dokument përcaktohen të gjitha pikat kryesore dhe rezultatet e rëndësishme për projektin si dhe rreziqet e projektit. Megjithatë, informacione të detajuara për disa pjesë të projektit (siç janë detajet e zbatimit dhe detajet e takimit) do të jepen në dokumente të tjera.

Objektivat e taksive*:*

* Fokusi kryesor i TaxiTaxi është të ofrojë shërbim transporti të besueshëm dhe lehtësisht të disponueshëm.
* Qëllimi i TaxiTaxi është të zgjerohet vazhdimisht globalisht dhe të sjellë shërbimet e tij në qytete të ndryshme.
* Të eliminohet pronësia e makinave private.

Misioni i TaxiTaxi është të riimagjinojë mënyrën se si bota lëviz për mirë. Në thelb, TaxiTaxi synon t'u sigurojë njerëzve akses më të madh në transport. Misioni në TaxiTaxi përfshin krijimin e mundësive duke e bërë transportin më të aksesueshëm.

## 1.5 Nevojat e përdoruesit

Në qytetet ku ofrohet TaxiTaxi, mund të përdoret aplikacionin TaxiTaxi dhe Web-faqja për të kërkuar një udhëtim, Pasiqe behet rezervimi per Taxi, Admini I Web-faqes degron nje shofer per destinacionin e caktuar.

## 1.6 Supozimet dhe varësitë

problemet që supozohet të ndodhin mund të jenë:

* Të dhëna të rreme të shoferit ose klientit
* Problemet e pageses
* Problemet me veturat

# 2. Karakteristikat dhe kërkesat e sistemit

## 2.1 Kërkesat Funksionale

**a) Kërkesat funksionale nga këndvështrimi i shoferit:**

* shoferi duhet të jetë në gjendje të përputhet me një klient aty pranë me vendndodhjen e tij aktuale.
* shoferët duhet të jenë në gjendje të ndajnë kohën e pritur të mbërritjes me klientin.
* shoferi duhet të jetë në gjendje të përfundojë dhe të fillojë udhëtimin.

**b) Kërkesat funksionale nga këndvështrimi i klientit:**

* klientët duhet të jenë në gjendje të shohin makinat e disponueshme aty pranë.
* klientët duhet të jenë në gjendje të ndajnë vendndodhjen gjeografike.
* klientët duhet të jenë në gjendje të shkruajnë komente për shoferin.

## 2.2 Karakteristikat e sistemit

* Softueri duhet të ketë drejtues dhe klientë.
* Shoferët duhet të kenë patentë shoferi.
* Shoferët duhet të kenë orë të kufizuara pune.

# 3. Objektivat e Projektit

Sistemi i caktimit të taksisë në internet është ndërtuar për të ofruar lehtësi për ata që janë të interesuar të dinë rreth taksisë dhe rrugëve të tyre. Ky sistem ofron informacion në lidhje me të gjitha rrugët e disponueshme dhe orarin e tyre.

Punonjësit gjithashtu lehtësohen duke u ofruar atyre raportin e statusit të punës dhe raportin e tabelës së tyre të punës. Në këtë sistem të gjitha informacionet në lidhje me punonjësit e mbajtur nga menaxheri i projektit dhe alokimi i punës bëhen nga ky menaxher i sistemit.

Sistemi i caktimit të taksisë në internet është puna midis pasagjerëve dhe punonjësve që punojnë, ky sistem përdoret për të përfituar udhëtarët e taksisë dhe për të ofruar informacion të detajuar në lidhje me itineraret e taksisë dhe disponueshmërinë e vendeve.

# Organizimi

Megjithëse të gjithë anëtarët e ekipit janë të regjistruar në një universitetet (SEEU), ekipi në fakt është i ndarë gjeografikisht në tre vende:

* Gostivar (2 anëtarët e ekipit)
* Tetove (1 anëtar i ekipit)
* Veles(1 anëtar i ekipit)

Puna në projekt ndahet në tri kategori: Organizim, Dokumentim, Prezantim dhe Zbatim. Është vendosur që të gjithë anëtarët e ekipit të marrin pjesë në mënyrë të barabartë në çdo pjesë të projektit.

## 4.1 Organizimi

* **Drejtuesi i projektit**

Udhëheqësi i projektit është përgjegjës për ekipin në përgjithësi. Përgjegjësia e tij është të jetë gjithmonë i informuar për çdo çështje të rëndësishme. Përgjegjësia e tij është gjithashtu të informojë të tjerët për ato çështje.

* **Lideri i ekipit**

Përgjegjësia e udhëheqësit të ekipit është të monitorojë studentët informojë drejtuesin e ekipit për çështje të rëndësishme që po ndodhin në anën e bashkpunuesve.

* **Të tjerët**

Të gjithë anëtarët e ekipit ndajnë përgjegjësinë për organizimin e takimeve të brendshme, takimet me mbikëqyrësin e projektit, ndarjen e detyrave të projektit dhe dorëzimin e dokumenteve në kohë.

## 4.2 Dokumentimi dhe Prezantimi

Dokumentimi dhe Prezantimi janë të dyja përgjegjësi e çdo anëtari të ekipit. Çdo dokument që kërkohet të dorëzohet u besohet disa anëtarëve të ekipit (numri varet nga dokumenti). Pasi të shkruajnë dokumentin, anëtarët e tjerë të ekipit duhet ta kontrollojnë atë dhe të bëjnë korrigjime nëse është e nevojshme. Përmbajtja e dokumenteve diskutohet në takimet javore.

Prezantimet duhet të bëhen nga anëtarët e ekipit që do t'i paraqesin ato dhe të kontrollohen dhe korrigjohen nga anëtarët e tjerë të ekipit. Është rënë dakord që dy ose më shumë anëtarë të ekipit të prezantojnë.

## 4.3 Zbatimi

Meqenëse projekti është i ndarë në tre pjesë kryesore, rolet e projektit përcaktohen në mënyrë të ngjashme:

* **Zhvilluesi i aplikacionit Taxi Mobile (2 anëtarë të ekipit)**

Përgjegjësia: zhvillimi i aplikacionit celular që do të përdoret në Taxis. Komunikimi: me zhvilluesit nga ana e serverit

* **Zhvillues nga ana e serverit (3 anëtarë të ekipit)**

Përgjegjësia: zhvillimi i një shërbimi ueb i cili do të jetë komunikimi me aplikacionet celulare, zhvillimi i aplikacionit në internet për klientët që duan të porosisin një komunikim Taksi: me zhvilluesit e aplikacioneve celulare

# Procesi i zhvillimit

Ne ndjekim një metodologji të modifikuar SCRUM për Shërbimin Taksi. Ne bëjmë planifikimin e projektit në nivel historik dhe e dorëzojmë projektin në bazë të veçorive për veçori.

**Planifikimi dhe Orari i Dorëzimit**

Planifikimi bëhet duke përcaktuar piketat dhe duke llogaritur kohën e dorëzimit të secilit.

Për të ndërtuar një moment historik, fillimisht ne e zbërthejmë problemin në veçori vertikale, që do të thotë gjëra që kanë kuptim për përdoruesit e sistemit. Kështu, për shembull, duke qenë se zbatimi i një pjese të bazës së të dhënave në server nuk është diçka që do të ishte e dukshme për asnjë nga palët e interesuara të projektit, ai nuk konsiderohet një veçori. Një tipar shembull është "Një taksi raporton vazhdimisht vendndodhjen e tij aktuale në serverin qendror", duke nënkuptuar se ndërfaqja e përdoruesit dhe puna e fundit duhet të përfundojnë si në pajisjen e taksive ashtu edhe në server dhe të integrohen përpara se të konsiderohet e përfunduar.

Pasi të kemi një grup karakteristikash që mbulojnë funksionalitetin që duam të mbulojmë në momentin e ardhshëm historik, ne vlerësojmë kompleksitetin e çdo veçori në pikat e kompleksitetit, duke caktuar një vlerë të plotë midis një dhe tre. Masa e kompleksitetit është vetëm relativ, kështu që një rezultat më i ulët për veçorinë A krahasuar me tiparin B do të thotë që veçoria A është relativisht më e thjeshtë për t'u zbatuar sesa veçoria B.

Kur vlerësohen të gjitha tiparet e momentit historik, ne përmbledhim vlerat e tyre të kompleksitetit për të marrë kompleksitetin total të momentit historik. Ne përdorim më pas shpejtësinë tonë aktuale (e matur në pikë kompleksiteti në javë) për të vlerësuar se sa kohë do të duhet për të përfunduar momentin historik (kompleksiteti total i ndarë me shpejtësinë).\* Rezultati është një plan lëshimi.

**Procesi I zhvillimit**

Zhvillimi bëhet në bazë të veçorive për veçori.

Pasi të kemi planifikuar një moment milestone, ne fillojmë punën për të në përsëritje javore (ajo që zakonisht njihet brenda SCRUM si "sprint"). Shpejtësia jonë aktuale na ofron një mënyrë të thjeshtë për të llogaritur kapacitetin e ekipit për çdo javë të caktuar (shpejtësia e shumëzuar me gjatësinë e përsëritjes në javë). Në fillim të javës ne zgjedhim mjaft veçori për të punuar për të mbushur atë kapacitet dhe për të filluar zhvillimin. Ne implementojmë, testojmë dhe integrojmë çdo komponent gjatë gjithë javës në mënyrë ad-hoc. Në fund të javës, ne kontrollojmë se sa nga veçoritë që kemi punuar janë përfunduar, ia tregojmë pronarit të produktit tonë dhe përmbledhim kompleksitetin e tyre për të na dhënë shpejtësinë tonë të re aktuale. Më pas kontrollojmë për të parë nëse kjo shpejtësi na mban në rrugën e duhur për të përfunduar momentin historik në kohë dhe për të bërë rregullimet e duhura (thjeshtimi i veçorive, shtimi i më shumë veçorive, shtyrja e momentit historik data e dorëzimit, etj.). Më pas vazhdojmë të planifikojmë përsëritjen e mëposhtme, duke përsëritur të njëjtin proces derisa të plotësohen të gjitha tiparet e momentit historik.

Pasi të përfundojë një moment historik, ne fillojmë përsëri, duke ndërtuar një listë karakteristikash që do të përbëjnë momentin historik të mëposhtëm. Ky proces vazhdon deri në afatin përfundimtar të dorëzimit të kursit.

**Rolet e projektit**

1. Pronari i produktit: Samedin Ekremi

Ekipi i zhvillimit takohet me të për të përcaktuar kërkesat dhe veçoritë. Ai gjithashtu shqyrton dhe nënshkruan çdo veçori që ofron ekipi i zhvillimit.

2. Scrum Master: Muhadin Skenderi

Ekipi i zhvillimit komunikon pengesat dhe vështirësitë që po përjetojnë që i pengojnë ata të kryejnë punën e kërkuar për të kryer punën në kohë dhe ai përpiqet të heqë pengesat e përmendura.

3. Ekipi i zhvillimit: Elham Tairovski, Berat Sherifi, Ardian Beqiri, Semin Amiti.

Anëtarët e ekipit të zhvillimit mbledhin kërkesat dhe dizajnojnë, zbatojnë, testojnë dhe integrojnë veçori.

# Komunikimi

## 6.1 Takimet

### 6.1.1 Takimet ne Discord

Çdo të enjte në orën 19.00 ekipi diskuton ecurinë e projektit dhe çështje të rëndësishme në një takim të rregullt javor. Përveç takimit të rregullt, mund të organizohen më shumë takime gjatë javës nëse lind nevoja për këtë.

Çdo takim i besohet një anëtari të ekipit që drejton takimin. Ky anëtar i ekipit duhet të përgatitet për takimin. Ai duhet të vendosë për temat që do të diskutohen dhe të shkruajë procesverbalin e dokumentit të mbledhjes pas përfundimit të takimit. Personi që drejton takimin është shumë i rëndësishëm pasi takimet janë zakonisht takime në Discord, të cilat do të ishte shumë e vështirë të koordinoheshin pa drejtuesin.

### 6.1.2 Takime me mbikëqyrësin e projektit

Takimet me mbikëqyrësin e projektit zakonisht organizohen përmes postës elektronike(E-mail). Ato organizohen kur është e nevojshme të diskutohen disa çështje të rëndësishme me mbikëqyrësin. I gjithë ekipi diskuton temën e takimit dhe pyetjet që do të bëhen në takimet e Discord.Konsiderohet gjithashtu që ato takime të bëhen të rregullta një herë në javë. Anëtarët e ekipit nga qytetet e ndryshme gjithashtu duhet të jenë në gjendje të marrin pjesë në ato takime duke përdorur takimet ne Discord

# Analiza e sistemit

## 7.1 Feasibility Study:

Studimi i fizibilitetit është faza në të cilën analisti kontrollon nëse sistemi kandidat është i realizueshëm për organizatën apo jo. Kjo përfshin identifikimin, përshkrimin dhe vlerësimin e sistemit. Studimi i fizibilitetit është bërë për të zgjedhur sistemin më të mirë që plotëson kërkesat e performancës. Nëse studimi i fizibilitetit do të shërbejë si një dokument vendimi, ai duhet t'u përgjigjet pyetjeve kryesore.

* + A ka ndonjë mënyrë të re dhe më të mirë për të bërë punën që do të përfitojë nga përdoruesi?
  + Cilat janë kostot dhe kursimet e alternativave?
  + Çfarë rekomandohet?

Projektet më të suksesshme të sistemit nuk janë domosdoshmërisht më të mëdhenjtë ose më të dukshëm në biznes, por ato tcilet përmbushin vërtet pritshmëritë e përdoruesve.

## 7.2 Economic Feasibility:

Analiza ekonomike është metoda më e përdorur për vlerësimin e efektivitetit të sistemit kandidat.

Ne analizojmë se sistemi kandidat (sistemi i kompjuterizuar) është i realizueshëm sesa sistemi manual sepse kursen para, kohë dhe fuqi punëtore. Gjithashtu i realizueshëm sipas analizës së përfitimeve të kostos.

## 7.3 Technical Feasibility:

Përqendrohet fizibiliteti teknik rreth teknologjisë së përdorur. Do të thotë se sistemi kandidat është teknikisht i realizueshëm, pra nuk ka ndonjë defekt teknik dhe funksionon siç duhet në mjedisin e caktuar. Sistemi ynë është teknikisht i realizueshëm; po na siguron produktin e kërkuar.

## 7.4 Behavioral Feasibility:

Fizibiliteti i sjelljes është analiza e sjelljes së sistemit kandidat. Në këtë analizojmë se sistemi kandidat po funksionon siç duhet apo jo. Nëse funksionon, atëherë komunikon siç duhet me mjedisin apo jo. Të gjitha këto çështje janë analizuar dhe është përgatitur një sistem i mirë kandidatësh. Për shkak të ndryshimit të sistemit cili është ndryshimi në sjelljen e përdoruesve, analizohen edhe këta faktorë.

# Dizajnimi I aplikacionit Taxi

Bazuar në ndërveprimin me palët përkatëse të interesit (klientët, personelin në kompani) dhe një studim të aplikacioneve ekzistuese të ngjashme, ne identifikuam kërkesat e mëposhtme si thelbësore për aplikacionin:

Një modul për hyrje në platformë

Një modul për kontrollin e rezervimeve në platformë

Një modul për pranimin e rezervimeve në platformë

Një modul për fillimin e udhëtimit në platformë

Një model rasti përdorimi është përdorur për të kapur këto funksione bazë dhe është paraqitur në Fig. 1. Tabela 1 është tregimi i rastit të përdorimit që përshkruan:

Diagram

Description automatically generated

|  |  |
| --- | --- |
|  | Taxi app |
| Përshkrim i shkurtër | Ky rast përdorimi përshkruan se si një përdorues i bën një porosi një shoferi taksie dhe se si përdoruesi monitoron taksinë deri në mbërritjen e tij |
| Aktori | Menaxheri I Projektit (admini) |
| Rrjedha e ngjarjeve(flow of events) | Rrjedha bazë  Rasti i përdorimit fillon kur aktori fut të dhënat e tij në një faqe regjistrimi e cila shfaqet në instalim  1. Aktori mund të përditësojë të dhënat fillestare duke u loguar  2. Aktori bën kontrollin e rezervimeve pasi u kemi loguar  3. Shfaqet një faqe me fusha të ndryshme teksti që kërkon që admini të pranoje rezervimin.  4. Detajet e porosive të bëra shfaqen dhe mund ne historikun dhe mund shoferi I taksise te drejtohet ne destinacionin qe duhet |
| Kërkesa të veçanta | Asnje |
| Kushtet paraprake(pre-condition) | Aktorët duhet të kenë plotësuar plotësisht kërkesat per pranimin e rezervimit. |
| Kushtet pas(post-condition) | Një mesazh do të merret nga përdoruesi kundrejt drejtuesit te caktuar |
| Pikat e Zgjatjes(extension points) | Asnje |

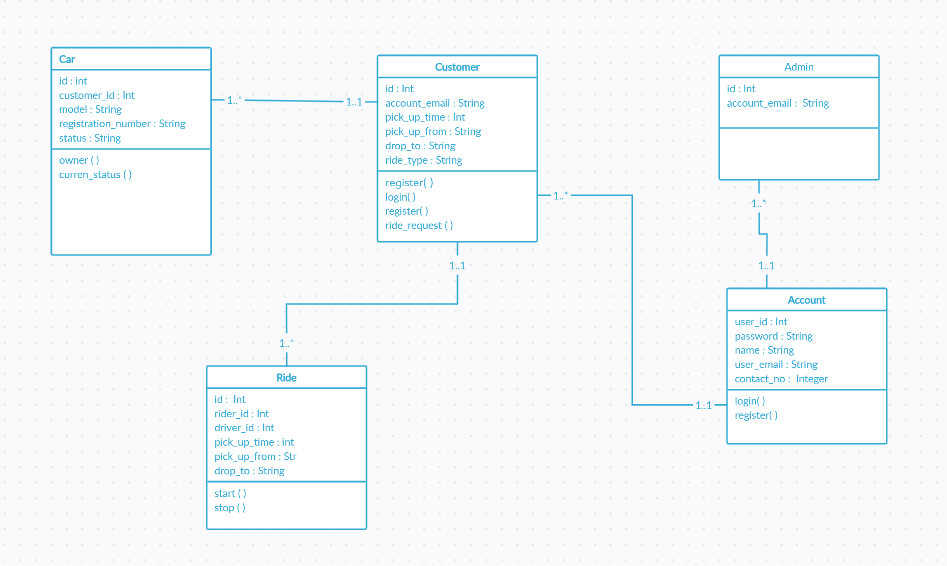
## 8.1 Sequence Diagrams

Diagrami i sekuencës përdoret për të treguar ndërveprimin e objekteve në një sekuencë të bazuar në kohë të asaj që ndodhi së pari, çfarë ndodhi më pas e kështu me radhë. Diagrami i mëposhtëm tregon se si një përdorues lidh me sistemin dhe një nga një çfarë ka ndodhur në sistem.

Chart

Description automatically generated

# Modeli konceptual i të dhënave:



* + Përshkrimi i subjekteve:
* Customer Entity: Ai përmban të gjitha të dhënat personale të klientit
* ID: ID-ja e klientit gjenerohet automatikisht sa herë që një përdorues regjistrohet si klient.
* Customer Email: duhet te regjistrohet me email per te konfirmuar adresen.

.

.

* Car Entity: Ai përbëhet nga informacioni i mjetit që shoferi do ket
* Car Id: Gjenerohet automatikisht
* Car Model: modeli I vetures
* Car status: a eshte vetura e lir apo ka klient

.

.

* Ride Entity: Ai permban te gjitha informacionet per rrugtimin
* Ride Id: gjenerohet sa here qe dikush udhton
* Driver Id: kur regjistrohet shoferi ateher gjenerohet id
* Pick up time: koha kur do te nsie udhtimi
* Pick up from: Vendi prej ku do niset udhetimi

.

.

* Account Entity: Permban te dhenat per gjitj accounted
* Name: Vendosja e emrit
* Password: Secili account ta kete passwordin e vet
* User\_Email: Vendosja e emailit per konfirmim

# Zbatimi I Aplikacionit Mobile dhe Web Aplication

Android është zyrtarisht sistemi operativ cellular shum I perdorur. Aplikacioni është në gjendje të ndërlidhet me hartat e Google për të ofruar informacion të detajuar rreth vendndodhjes së një taksie. Modulet e aplikacionit celular përfshijnë: një faqe menyje - që shfaqet menjëherë pas ngarkimit të aplikacionit. Nëse një përdorues po shkarkon aplikacionin për herë të parë, atëherë ai/ajo do të duhet të regjistrohet, por më pas ekrani i menysë shfaqet siç tregohet në Figuren me poshte :

Graphical user interface, application

Description automatically generated

## 10.1 Moduli i Regjistrimit

Table

Description automatically generated with medium confidenceKy është ekrani i parë interaktiv që shfaqet pas ekranit spërkatës nëse përdoruesi po shkarkon aplikacionin për herë të parë. Ai ofron fusha që kapin emrin (mbiemrin) Email dhe fjalëkalimin për hyrje në aplikacion. Kjo është paraqitur në figuren me poshte :

## 10.2Moduli i Porosisë së Vendit

Ky modul është aktiviteti kryesor i aplikacionit cellular dhe web faqes. Ai lejon një përdorues të futë vendndodhjen e tij/saj aktuale me detaje specifike. Ky informacion më pas dërgohet tek admini dhe admini do e lajmroje shoferin Kjo është paraqitur në Figurat me poshte :

A picture containing shape

Description automatically generatedMap

Description automatically generated

## 10.3 Taxi Locator

Ky modul është i integruar me Google Maps që tregon cmimin se sa kushton udhetimi nga ana e nje taksie sipas lokacionit qe paraqitet ne harte . Përdoret për të monitoruar ne menyre me te leht per perdoruesit cmimin dhe kohen siç tregohet në Figurat :

Map

Description automatically generated

## 10.4 Moduli i Kontaktit

Web faqja mund t'i lejojë një përdoruesi të regjistrohen me te dhenat e tyre dhe numrin e telefontit me pas duhet te shtypin butonin Submit dhe I marrim te dhenat qe na nevojiten. Është paraqitur në Figuren :

Graphical user interface, application, website, map

Description automatically generated

# Testimi i projektit:

Faza e testimit vjen pas zhvillimit dhe para vendosjes në procesin e softuerit dhe përbëhet nga një grup llojesh dhe metodash të përdorura për të vlerësuar dhe kontrolluar një produkt softuerësh për gabime dhe boshllëqe të performancës. Me fjalë të tjera, testimi është një hap i rëndësishëm pasi vërteton nëse softueri i plotëson kërkesat e përdoruesit dhe performancën e synuar apo jo.

## 11.1 Llojet e testeve të softuerit:

* Testimi i njësisë: konsiston në testimin e një komponenti të vetëm të aplikacionit të softuerit, siç është një funksion i vetëm, në të njëjtën kohë. Qëllimi i tij është të kontrollojë, zakonisht nëpërmjet funksioneve, sjelljen e njësisë në skenarë të ndryshëm. Për shembull, në aplikacionin tonë të Rezervimit të Taksive, ne mund të kryejmë një testim njësie në funksionin e vërtetimit për të kontrolluar nëse funksionon saktë në të gjitha rastet e mundshme, si p.sh. futja e një fjalëkalimi të gabuar ose një adrese emaili që nuk ekziston.
* Testimi i integrimit: konsiston në bashkimin e dy ose më shumë njësive të lidhura logjikisht të aplikacionit softuer dhe testimin e tyre së bashku si një njësi e madhe. Qëllimi i tij është të testojë ndërveprimin dhe rrjedhën e të dhënave midis funksionaliteteve të ndryshme. Në aplikacionin tonë të Rezervimit Taxi, testimi i integrimit mund të bëhet për shembull duke integruar funksionet e regjistrimit dhe hyrjes për të kontrolluar nëse një përdorues i sapo regjistruar mund të identifikohet apo jo.
* Testimi i Sistemit: në këtë nivel, të gjitha funksionalitetet e aplikacionit softuer integrohen plotësisht në një njësi të vetme dhe kontrollohen kundrejt kërkesave të specifikuara. Ai përbëhet nga disa kategori, ato që përdoren zakonisht janë testimi i kutisë së zezë, i cili fokusohet kryesisht në hyrje dhe dalje në vend që të fokusohet në dizajnin e brendshëm dhe strukturën e kodit, dhe testimi nga fundi në fund, i cili përbëhet nga duke kontrolluar nëse aplikacioni funksionon saktë në situata të tilla si një ndërveprim me një bazë të dhënash ose një sistem komunikimi në rrjet. Në produktin tonë softuerik, testimi nga fundi në fund mund të bëhet për të kontrolluar nëse një ndryshim në të dhënat e kalorësve është përditësuar në bazën e të dhënave apo jo.
* Testimi i pranimit: synon të kontrollojë nëse i gjithë sistemi i softuerit plotëson kërkesat dhe kriteret e biznesit. Ai përbëhet nga disa lloje:
* Testimi i pranimit të përdoruesit: kontrollon nëse aplikacioni funksionon plotësisht për përdoruesin.
* Testimi i pranimit të biznesit: kontrollon nëse aplikacioni plotëson biznesin e synuar performanca dhe qëllimet.
* Testimi i pranimit të kontratës: bazohet në një kontratë që specifikon një periudhë të caktuar për të cilën testi i pranimit të produktit softuer duhet të kryhet saktë për të gjithë skenarët e specifikuar.

## Pse testimi I sistemit?

Testimi është jetik për suksesin e sistemit. Testimi i sistemit bën një supozim logjik që nëse të gjitha pjesët e sistemit janë të sakta, qëllimi do të arrihet me sukses. Rezultatet joadekuate të testimit në dy lloje problemesh:

1. Vonesa kohore midis shkakut dhe shfaqjes së problemit.
2. Efekti i gabimeve të sistemit në skedarët dhe regjistrimet brenda sistemit.

**Pergaditja e testimit:**

Një plan testimi i zbatueshëm duhet të përgatitet në përputhje me specifikimet e përcaktuara të projektimit. Ai përfshin artikujt e mëposhtëm:

* Rezultatet e pritura nga sistemi.
* Kriteret për vlerësimin e rezultateve.
* Një vëllim të dhënash testimi.
* Procedura e përdorimit të të dhënave të testit.

**Specifikimi i kushteve per testimin e pranimit te perdoruesve**

Planifikimi për testimin e pranimit të përdoruesit kërkon që analisti dhe përdoruesi të bien dakord për kushtet e testit.

**Pergaditja e te dhenave te testimit per testim te programit**

Ndërsa çdo program është i koduar, të dhënat e testimit përgatiten dhe dokumentohen për të siguruar që të gjitha aspektet e programit janë testuar siç duhet.

**Pergaditja e te dhenave te testimit per testimin e rruges se transaksionit**

Ky aktivitet zhvillon të dhënat e nevojshme për testimin e çdo kushti dhe transaksioni që do të futet në sistem. Rruga e çdo transaksioni nga origjina në destinacion është testuar me kujdes rezultate të besueshme.

**Planifikimi I Trajnimit te perdoruesit**

Trajnimi i përdoruesit është krijuar për të përgatitur përdoruesin për testimin dhe konvertimin e sistemit. Përfshirja dhe trajnimi i përdoruesve zhvillohen paralelisht me programimin për tre arsye:

* Grupi i sistemit ka kohë në dispozicion për të shpenzuar në trajnime ndërsa programet janë duke u shkruar.
* Fillimi i një programi trajnimi për përdoruesit i jep grupit të sistemeve një imazh më të qartë të interesit të përdoruesit për sistemin e ri.
* Një përdorues i trajnuar merr pjesë në mënyrë më efektive në testimin e sistemit.

Plani i trajnimit pasohet nga përgatitja e manualit të trajnimit të përdoruesit dhe materialeve të tjera tekstuale.

## 11.3 Testimi I sistemeve

Departamenti i kompjuterit për të siguruar që sistemi funksionon siç është specifikuar e bën këtë testim. Ky testim është i rëndësishëm për të siguruar që një sistem pune t'i dorëzohet përdoruesit për testimin e pranimit.

## 11.4 Testimi I pranimit

Përdoruesi për të siguruar që sistemi funksionon, ashtu siç dëshironte përdoruesi, kryen këtë testim. Me teknikat e prototipit, kjo fazë bëhet shumë një formalitet për të kontrolluar saktësinë dhe plotësinë e përpunimit. Paraqitjet e ekranit dhe dalja duhet të jenë testuar tashmë gjatë fazës së prototipit.

Një gabim në kodin e programit mund të mbetet i pazbuluar për një kohë të pacaktuar. Për të parandaluar që kjo të ndodhë, kodi u testua në nivele të ndryshme. Për të testuar me sukses një sistem, çdo kusht dhe kombinime të kushteve duhej të testoheshin. Çdo program u testua dhe u lidh me programe të tjera. Kjo njësi programi testohet dhe lidhet me njësi të tjera dhe kështu me radhë derisa të testohet sistemi i plotë.

Qëllimi i testimit është të sigurohet që çdo program të jetë testuar plotësisht. Për ta bërë këtë duhej të krijohej një plan testimi. Plani i provës përbëhet nga një numër testesh të tilla si shtigjet e vlefshme përmes kodit dhe shtigjet e trajtimit të përjashtimeve dhe gabimeve. Për çdo ekzekutim provë ka një listë të kushteve të testuara, të dhënat e testit të përdorura dhe rezultatin e pritur. Plani i provës u rishikua më pas për të kontrolluar nëse çdo shteg përmes kodit është testuar saktë. Është përgjegjësi e programuesit të mbledhë të dhënat që do të prodhojnë kushtin e kërkuar të testit.

# Verifikimi dhe vleresimi (V&V)

Objektivat e verifikimit, aktiviteteve të vlefshmërisë janë vlerësimi dhe përmirësimi i cilësisë së produkteve të punës të krijuara gjatë zhvillimit dhe modifikimit të softuerit. Cilësia varet nga atributet e ndryshme si korrektësia, plotësia, qëndrueshmëria, besueshmëria, dobia, përdorshmëria, efikasiteti dhe përputhshmëria me standardet.

Termat verifikim dhe vërtetim përdoren në mënyrë sinonime. Këtu janë përcaktuar si më poshtë:

**Verifikimi:** "A po e ndërtojmë produktin si duhet?"

**Vlersimi:** "A po ndërtojmë produktin e duhur?"

Aktivitetet e verifikimit përfshijnë vërtetimin, testimin dhe rishikimet. Validimi është procesi i vlerësimit të softuerit në fund të zhvillimit të softuerit për të siguruar përputhjen me kërkesat e softuerit. Testimi është një metodë e zakonshme e vërtetimit. Është e qartë se për besueshmëri të lartë duhet të kryejmë të dyja aktivitetet. Së bashku, ato shpesh quhen aktivitete V&V.

Aktivitetet kryesore të V&V për zhvillimin e softuerit janë inspektimi, rishikimet dhe testimi (si statik ashtu edhe dinamik). Plani V&V identifikon detyrat e ndryshme V&V për faza të ndryshme dhe specifikon se si këto detyra kontribuojnë në qëllimet e projektit V&V. Përcaktohen gjithashtu metodat që do të përdoren për kryerjen e këtyre aktiviteteve V&V, përgjegjësitë dhe etapat për secilin prej këtyre aktiviteteve, inputet dhe outputet për secilën detyrë V&V dhe kriteret për vlerësimin e rezultateve.

Dy qasjet kryesore të V&V janë testimi dhe inspektimet. Testimi është një aktivitet që në përgjithësi mund të kryhet vetëm në kod. Është një aktivitet i rëndësishëm dhe diskutohet në detaje në një kapitull të mëvonshëm. Inspektimi është një aktivitet më i përgjithshëm që mund të zbatohet për çdo produkt pune, duke përfshirë kodin. Shumë nga detyrat V&V janë të tilla që për ta, një lloj aktiviteti inspektimi është mënyra e vetme e mundshme për të kryer detyrat (p.sh. aftësia e gjurmimit dhe vlerësimi i dokumenteve). Për shkak të kësaj, inspektimet luajnë një rol të rëndësishëm në verifikim.

# Mirembajtja pas shqyrtimit dhe implementimit

Siç e dimë, krijimi i softuerit është një gjë dhe zbatimi i softuerit të krijuar është një gjë tjetër. Procesi i zbatimit të softuerit është shumë i vështirë në krahasim me detyrën e krijimit të projektit. Fillimisht duhet të implementojmë softuerin në një shkallë të vogël për heqjen e gabimeve dhe gabimeve të tjera në projekt dhe pas heqjes së tyre mund ta implementojmë softuerin në shkallë të gjerë.

Përpara se të mendojmë për zbatimin e Softuerit në një bazë të gjerë, duhet të marrim parasysh kërkesat e harduerit.

Kurdoherë që ne zhvillojmë softuer ose projektim, një harduer dhe softuer i caktuar përdoret nga programuesi për zhvillimin e projektit. Hardware dhe softuer që do të përdoret nga programuesi për zhvillimin e projektit duhet të jetë i tillë që të rezultojë në zhvillimin e një projekti, i cili do të plotësonte të gjitha nevojat themelore për të cilat është krijuar projekti nga programuesi. Hardueri duhet të jetë i tillë që kufizimet e kostos së Klientit gjithashtu duhet të merren parasysh pa ndikuar në performancën.

## 13.1 Faktorët e Vlerësimit të Hardware

*Kur vlerësojmë harduerin e kompjuterit, fillimisht duhet të hetojmë karakteristikat specifike fizike dhe të performancës për secilin komponent harduer që do të blihet. Këto pyetje specifike duhet t'u përgjigjen në lidhje me shumë faktorë të rëndësishëm. Këto pyetje të faktorëve të vlerësimit të harduerit janë përmbledhur në figurën e mëposhtme.*

Na tregon se ka shumë më tepër për të vlerësuar harduerin sesa për të përcaktuar pajisjen informatike më të shpejtë dhe më të lirë. Për p.sh. çështja e vjetërsimit të mundshëm duhet të trajtohet duke bërë një vlerësim teknologjik. Faktori i ergonomisë është gjithashtu shumë i rëndësishëm. Ergonomia është shkenca dhe teknologjia që përpiqet të sigurojë që kompjuterët dhe teknologjitë e tjera të jenë "miqësore për përdoruesit", që janë të sigurta, komode dhe të lehta për t'u përdorur.

**Faktorët e Vlerësimit të Hardware**

1. Performanca
2. Kostoja
3. Besueshmëria
4. Disponueshmëria
5. Pajtueshmëria
6. Teknologjia
7. Lidhshmëria
8. Kërkesat mjedisore
9. Software
10. Mbështetje

## Faktorët e Vlerësimit të Softuerit

Softueri mund të vlerësohet sipas shumë faktorëve të ngjashëm me vlerësimin e harduerit. Kështu, faktorët e *performancës, kostos, besueshmërisë, përputhshmërisë, teknologjisë, dhe mbështetjes* duhet të përdoren për të vlerësuar blerjet e softuerit të propozuar. Përveç kësaj, megjithatë, faktorët e vlerësimit të softuerit janë përmbledhur në figurën më poshtë. Për p.sh. disa paketa softuerike kërkojnë shumë kapacitet memorie dhe janë jashtëzakonisht të ngadalta, të vështira për t'u përdorur ose të dokumentuara dobët. Ata nuk janë një përzgjedhje e mirë për shumicën e përdoruesve fundorë, edhe nëse ofrohen me çmime tërheqëse.

**Faktorët e vlerësimit të softuerit:**

1. **Efikasiteti:** a është softueri një sistem i mirëshkruar udhëzimesh kompjuterike që nuk përdor shumë kapacitet memorie ose kohë të procesorit?
2. **Fleksibiliteti:** a mund të trajtojë lehtësisht detyrat e tij të përpunimit pa modifikime të mëdha?
3. **Siguria**: a ofron procedura kontrolli për gabime, keqfunksionime dhe përdorim jo të duhur?
4. **Gjuha:** a e shkruajnë programuesit dhe përdoruesit tanë të kompjuterave në një gjuhë programimi që përdoret?
5. **Dokumentimi:** a është e dokumentuar mirë? A përfshin udhëzime të dobishme për përdoruesit?
6. **Hardware:**A ka pajisjet ekzistuese funksionet e nevojshme për të përdorur më së miri këtë softuer?
7. Karakteristika të tjera të pajisjeve të tilla si performanca, çfarë për koston, sa është e besueshme etj.

## 13.3 Konvertimi dhe Trajnimi

Një aspekt i rëndësishëm është të siguroheni që dizajni i ri të zbatohet për të vendosur standarde. Termi implementim ka kuptime të ndryshme, duke filluar nga shndërrimi i një aplikacioni bazë në një zëvendësim të plotë të një sistemi kompjuterik. Zbatimi përdoret këtu për të nënkuptuar procesin e konvertimit të një sistemi të ri ose të rishikuar në një sistem operacional. Konvertimi është një aspekt i zbatimit. 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 dhe instalimin e terminalit dhe harduerit. Një aspekt kritik i konvertimit nuk është ndërprerja e funksionimit të organizatës.

Kur një sistem i ri përdoret pa pushim, ekzistues dhe në funksionim, gjithmonë ka gabime të përputhshmërisë. Këto gabime janë shkaktuar për shkak të mungesës së pajisjeve apo personelit për të punuar sistemin e ri. Drejtimi i ndonjë sistemi të specifikuar në një organizatë kërkon një pjesë ose një tjetër harduer ose, në këtë rast, gjithashtu kërkesa softuerike.

**Konvertimi është një aspekt i rishikimit të zbatimit dhe mirëmbajtjes së softuerit.**

Ekzistojnë tre lloje të zbatimit:

1. Zbatimi i një sistemi kompjuterik për të zëvendësuar një sistem manual. Problemet e hasura janë konvertimi i skedarëve, trajnimi i përdoruesve, krijimi i skedarëve të saktë dhe verifikimi i printimeve për integritet.
2. Zbatimi i një sistemi të ri kompjuterik në vend që ekziston. Ky është zakonisht një konvertim i vështirë. Nëse nuk planifikohet siç duhet, mund të ketë shumë probleme. Disa sisteme të mëdha kompjuterike u është dashur një vit për t'u konvertuar.
3. Zbatimi i një aplikacioni të modifikuar për të zëvendësuar një ekzistues, duke përdorur të njëjtin kompjuter. Ky lloj konvertimi është relativisht i lehtë për t'u trajtuar, me kusht që të mos ketë ndryshime të mëdha në skedarë.

# Kontrollet e sigurise dhe vlefshmerise

***Dobësia e softuerit ndaj sulmit***

Zhvillimi i softuerit nuk është ende një shkencë apo një disiplinë dhe procesi i zhvillimit në përgjithësi nuk kontrollohet për të minimizuar dobësitë që sulmuesit shfrytëzojnë.

Siguria e softuerit kërcënohet në pika të ndryshme gjatë ciklit të tij jetësor, si nga zgjedhjet dhe veprimet e paqëllimshme dhe të qëllimshme të ndërmarra nga "të brendshëm" - individë të lidhur ngushtë me organizatën që prodhon, vendos, operon ose mirëmban softuerin, dhe kështu të besuar nga ajo organizatë - dhe nga "të huajt" që nuk kanë lidhje me organizata. Siguria e softuerit mund të kërcënohet

* **gjatë zhvillimit të tij**: Një zhvillues mund të korruptojë softuerin – qëllimisht ose pa dashje – në mënyra që do të rrezikojnë besueshmërinë dhe besueshmërinë e softuerit kur ai është funksional.
* **gjatë vendosjes së tij (shpërndarja dhe instalimi**): Nëse ata që janë përgjegjës për shpërndarjen e softuerit dështojnë të dëmtojnë softuerin përpara dërgimit ose ngarkimit, ose ta transmetojnë atë përmes kanaleve të komunikimit që përgjohen lehtësisht, ata e lënë softuerin të prekshëm ndaj korrupsionit të qëllimshëm ose të paqëllimshëm. Në mënyrë të ngjashme, nëse instaluesi i softuerit nuk arrin të "bllokojë" platformën pritës, ose konfiguron softuerin në mënyrë të pasigurt, softueri lihet i prekshëm ndaj aksesit nga sulmuesit.
* **gjatë funksionimit të tij:** softueri me burim të hapur është funksionalizuar, dobësitë mund të zbulohen dhe publikohen; nëse nuk aplikohen arnime dhe përditësime të sigurisë dhe nuk miratohen versione më të reja të mbështetura (nga të cilat janë eliminuar shkaqet rrënjësore të dobësive), një softuer i tillë do të bëhet gjithnjë e më i prekshëm. Softueri jokomercial dhe softueri me burim të hapur mund të jenë gjithashtu të cenueshëm, veçanërisht pasi mund të shfaqin sjellje jo të besueshme me kalimin e kohës për shkak të ndryshimeve në mjedisin e tij që e stresojnë softuerin në mënyra që nuk ishin parashikuar dhe simuluar gjatë testimit të tij.
* **gjatë mbajtjes së tij:** Nëse ata që janë përgjegjës për adresimin e dobësive të zbuluara në softuerin e lëshuar nuk arrijnë të lëshojnë arna ose përditësime në kohën e duhur

ose nuk arrin të gjejë dhe eliminojë shkaqet rrënjësore të dobësive për të parandaluar përjetësimin e tyre në lëshimet e ardhshme të softuerit, softueri do të bëhet gjithnjë e më i prekshëm ndaj kërcënimeve me kalimin e kohës. Gjithashtu, mirëmbajtësi i softuerit mund të provojë të jetë një insajder me qëllim të keq dhe mund të ngulit kod keqdashës, të meta të shfrytëzueshme, etj., në versionet e përditësuara të kodit.

***Sfida e ndërtimit të softuerit të sigurt***

1. **Pavarshmeria:** Softueri i pavarshem ekzekuton në mënyrë të parashikueshme dhe funksionon saktë në të gjitha kushtet, duke përfshirë kushtet armiqësore, duke përfshirë kur softueri sulmohet ose funksionon në një host me qëllim të keq.
2. **Besueshmëria:** Softueri i besueshëm përmban pak ose aspak dobësi ose dobësi që mund të shfrytëzohen qëllimisht për të përmbysur ose sabotuar besueshmërinë e softuerit. Përveç kësaj, për t'u konsideruar i besueshëm, softueri nuk duhet të përmbajë asnjë logjikë keqdashëse që e bën atë të sillet në një mënyrë dashakeqe.
3. **Mbijetueshmëria (e përmendur edhe si "Elasticitet**"): I mbijetueshëm ose elastik softueri është softuer që është mjaft elastik për t'i rezistuar (d.m.th., për t'u mbrojtur kundër) ose për të toleruar (d.m.th., për të vazhduar funksionimin në mënyrë të besueshme pavarësisht) sulmeve më të njohura plus sa më shumë sulme të reja që të jetë e mundur, dhe rikuperohet sa më shpejt që të jetë e mundur, dhe me sa më pak dëme, nga ato sulme që as mund t'i rezistojë dhe as t'i tolerojë.

# Konklusioni

Projekti "Taxi Service" bazohet në menaxhimin e Udhëtimit me Taksi në një sistem. Ky sistem i lejon menaxherit të projektit të mbajë të gjitha detajet e nevojshme në lidhje me menaxhimin e kabinës, menaxhimin e rrugeve, menaxhimin e pasagjerëve dhe menaxhimin e anëtarëve të stafit. Sistemi ofron një përdorues grafik ndërfaqe, e cila i ndihmon të gjithë përdoruesit të njohin rruget e taksisë dhe tabelat e lidhura me to. Ky sistem ofron raportin e statusit të punës për anëtarët e stafit që punojnë me këtë sistem. Ky sistem jep përparësi duke ofruar të gjitha informacionet me një klikim të vetëm. Të gjithë përdoruesit që duan të dinë informacione për Udhëtimet me Taksi dhe ata që duan të rezervojnë bileta, lehtësohen nga ky sistem. Përmirësimet e ardhshme për këtë projekt mund të krijohen gjithashtu duke përdorur grafikët me shirita me anë të të cilave performanca e secilit projekt mund të analizohet më mirë dhe duke përdorur këtë alokim të burimeve mund të bëhet në mënyrë efikase.