

Softvérové inžinierstvo - semestrálne zadanie č.1

Peter Šebest

2017

Slovenská Technická univerzita v Bratislave - Elektrotechnická fakulta



10. Logistický informačný systém

Obsah

1	Používateľská špecifikácia	3
1.1	Stručný úvod do problematiky	3
1.1.1	Delenie spoločnosti	3
1.1.2	Delenie zásielok	4
1.2	Používateľské požiadavky	5
1.2.1	Zákazníkové ciele:	5
1.2.2	Merateľné požiadavky	5
1.2.3	Nemerateľné požiadavky	5
1.2.4	Funkcionálne požiadavky	5
1.2.5	Nefunkcionálne požiadavky	6
1.2.6	Doménové požiadavky	6
2	Systémová špecifikácia	7
2.1	Diagramy prípadov použitia	7
2.2	Stručne popíšte role jednotlivých hráčov	7
2.2.1	Skladník	8
2.2.2	Skladník - super-visor	8
2.2.3	Zákazník	8
2.2.4	Zákazník - registrovaný	8
2.2.5	špeditér	8
2.2.6	Autorizovaný špeditér - letecká doprava	8
2.2.7	triediacia linka	8
2.2.8	kuriér - vodič	8
2.3	Tabuľka - tri najzložitejšie prípady	8
2.4	Diagramy aktivít a sekvenčné diagramy	9
2.4.1	Diagramy aktivít	9
2.4.2	Sekvenčné diagramy	9
2.5	Stavový diagram	10
	Použitá Literatúra	10
	časť 2	11
2.6	Doplnenie dokumentácie	11
2.6.1	Zaznamenávanie zmien	11
3	Dátový model	11
4	Akceptačné testy	12
5	Projektové plánovanie	13
5.0.1	Môj tím	13
5.1	WBS-work breakdown schedule	13
5.2	Ganttov graf a kalendár/plán úloh pre jednotlivých členov tímu	13

1 Používateľská špecifikácia

1.1 Stručný úvod do problematiky

1.1.1 Delenie spoločnosti

V tomto zadaní bude vyhotovený návrh pre informačný systém logistickej spoločnosti. Ako brigádnik som už pracoval vo viacerých takýchto spoločnostiach pri spracovávaní zásielok v triediacich centrách. Postup práce bol všade veľmi podobný. Spracovávané boli zásielky v rámci krajiny (DOMESTIK) alebo zahraničné zásielky, Zahraničné zásielky z okolitých krajín spracovávala nejaká partnerská firma kvôli väčšiemu dopytu pri týchto zásielkach, ostatné boli posielané do najväčších centier po svete. Týchto centier môže byť niekoľko desiatok po celom svete a sú to hlavné uzly pre spracovávanie zásielok ktoré väčšinou obsahujú veľké letisko, prístav alebo železničný uzol. Tieto centrá sa prezývajú "Huby". V Huboch sa dokáže denne spracovať denne niekoľko stotisíc až miliónov zásielok. Spracovanie zásielok v huboch je veľmi zložitá, pracuje sa tam aj v nepretržitej prevádzke, lietadlá sem prilietajú z rôznych časových pásiem, a existuje tam veľa oddelení kde pracuje niekoľko tisíc zamestnancov, takže sa spracovaním zásielok v huboch už nebudem zaoberať. Ďalej existujú menšie centrá väčšinou v hlavných mestách alebo v najväčších mestách z menších štátnych celkov ktoré majú k dispozícii nákladnú časť letiska odkiaľ môže odletieť lietadlo do najbližšieho hubu a naspäť. Tieto strediská môžu byť označené napríklad AIR alebo podobne. Do týchto centier sa zhromažďujú zásielky pomocou kamionovej prepravy z okolitých okresných miest. Sem dovážajú zásielky priamo kuriéri po rannom rozvoze a poobednej nákladke, prípadne do väčších firiem je posielané nákladné auto ktoré môže pravážiť aj paletové zásielky. V posledných rokoch sa začali v miestach s veľkou frekvenciou obyvateľstva (nákupné centrá, autobusové a vlakové stanice) rozmietať aj automaty pre zber a odber zásielok.

Pribeh práce v týchto centrách nasledujúci:

- v skorých ranných hodinách sa začína ranný import - vyloženie prepravného dopravného prostriedku, spracovanie a roztriedenie zásielok na doručenie
- 1. – kuriéri si naložia zásielky do dodávky podľa trasy rajónu - pre dverách dodávky sú zásielky s dorúčením na začiatku trasy (First in last out)
 - zásielky doručujú priamo k zákazníkovi, ktorých vopred telefonicky kontaktujú
 - v prípade potreby zhromažďujú informácie pre poobedný zvoz balíkov, pre menšie množstvo balíkov ho môžu vykonať aj hneď
- 2. – Po ukončení rozvozu vyzdvihnú zásielky pre export do požadovaných destinácií
 - Zákazník môže kontaktovať do operačného strediska kde operátor (špeditér) hlási kuriérom informácie o množstve tovaru ktoré treba vyzdvihnúť
 - ak ide o pravidelného zákazníka ktorý má so spoločnosťou zmluvu, tak môže priamo kontaktovať kuriéra
- IT, Human resources, marketing, riadiace, špeditérské oddelenie počas dňa riešia im požadované problémy spoločnosti
 - 1. **IT oddelenie:** správa, vývoj a údržba systému na riadenie centra, vytváranie databáz, reportov komunikácia s vedením
 - 2. **Human resources:** vybavovanie požiadaviek zamestnancov, nábor nových zamestnancov, vybavovanie výplat, spolupráca s finančným oddelením
 - 3. **Marketing:** komunikácia so zákazníkmi, hľadanie nových zákazníkov sledovanie trhu, stretnutia - vedenie/zákazníci, spolupráca s finančným oddelením
 - 4. **vedenie:** Optimalizuje a riadi proces, navrhuje riešenie problémov, komunikuje s supervirom skladu a vedúcimi jednotlivých oddelení

5. **špeditéry:** komunikujú s kuriérmi, vodičmi a pilotmi zákazníkmi, navrhujú trasovanie dopravy, podieľajú sa na riadení nákladky pre leteckú dopravu, pracujú s geografickým informačným systémom
- Večerný export spracúva zásielky pre doručenie do požadovaných destinácií
 1. zásielky sú vyložené a skontrolované v operačnom centre
 2. kuriéry na finančnom oddelení odovzdajú hotovosť za doručené zásielky
 3. nedoručené zásielky uskladňujú v sklade pre doručenie na ďalší deň, ak zásielka prekročila limit pre uskladnenie, alebo bola odmietnutá je odoslaná naspäť k odosielateľovi
 4. zásielky pre export sú spracované pomocou triadiacej linky, umelej inteligencie a zamestnancov a sú naložené do dopravných prostriedkov (váženie a meranie zásielok, pridanie "labelov štítkov), uloženie stavu zásielky do systému, v prípade nákladky lietadla treba vykonať uloženie kontajnerov so zásielkami do lietadla správnym vyvážením a rozmiestnením nebezpečných zásielok
 5. doprava zásielky závisí od triedy produktu
 - počas noci sú prepravované zásielky v dopravných prostriedkoch do destinácií kde budú spracované ranným importom
 - * existuje aj výhodnejšia možnosť kde si dvaja šoféry vymenia náklad ak existuje niaka destinácia približne v strede ich cesty a nemusia odchádzať z destinácií naprázdno alebo čakať v destinácií na export z ďalšieho dňa

1.1.2 Delenie zásielok

I Krabicové zásielky - zásielky spracováva triediaca linka	triedy, ich spracovanie vyžaduje odborníka Podmienky prepravy podľa tried nebezpečia
II Listové zásielky - zásielky nemôže triediť triediaca linka	<ul style="list-style-type: none"> • nesmú sa prepravovať lietadlom
III Nadrozmerné zásielky - zásielky nemôže triediť triediaca linka	<ul style="list-style-type: none"> • zásielky musia byť prístupné pilotovi počas letu
IV Paletové zásielky - zásielky nemôže triediť triediaca linka	<ul style="list-style-type: none"> • zásielky nejakej triedy nebezpečia sa nesmú nachádzať v kombinácii so zásielkami z inej triedy nebezpečia
V Nebezpečné zásielky - zásielky nemôže triediť triediaca linka, treba ich oštiepkovať podľa	<ul style="list-style-type: none"> • zásielky sa nesmú nachádzať v lietadle s pasažiermi

Danger good classes - triedy nebezpečných zásielok

Trieda 1. Výbušniny - nitroglycerin, granáty, strelivo, airbasy, n.o.s., TNT, dymovnice...

Trieda 2. Horľavé plyny - , nehorľavé a netoxické, toxické

Trieda 3. Horľavé kvapaliny - kvapaliny ktoré pri teplote 60° emitujú horľavé plyny

Trieda 4. Horľavé pevné látky - fosfor, odpadový kaučuk, lítium...

Trieda 5. Oxidy a organické peroxidy - čistiace prášky

Trieda 6. Toxické a nákazlivé substancie - pesticídy, biologické substancie nakazená krv a orgány...

Trieda 7. Rádioaktívne substancie - Rádium, Urán, iod123...

Trieda 8. Korozívne materiály - kyseliny, autobatérie...

Trieda 9. Rozličné nebezpečné materiály - Azbest, magnetický materiál, suchý ľad...

Všetky tieto zásielky musia byť označené príslušným štítkom.

Zásielky môžu byť označené aj štítkami krehké, premokavé, neukladať na seba, vrchnou stranou hore, ranné doručenie, sobotné doručenie.

1.2 Používateľské požiadavky

1.2.1 Zákazníkové ciele:

Doručenie zásielok v stanovenom termíne, celosvetové pokrytie, spracovanie zásielok rôzneho typu, veľkosti a hmotnosti, nepoškodenie zásielok, spracovanie peňažnej hotovosti, cena prepravy

1.2.2 Merateľné požiadavky

- Dĺžka prepravy - čas
- Počet pokrytých štátov, hustota
- Počet druhov tovaru
- Kapacita a nosnosť prepravy / hustota
- Čas spracovania hotovosti, dostupnosť v štátoch
- Suma v peňažných menách

1.2.3 Nemerateľné požiadavky

- zásielky musia byť vo nepoškodenom stave
- systém musí byť pochopiteľný
- Pre Danger Goods spracovanie a vyvažovanie lietadla - bezpečnosť
- Optimálne využitie kapacity

1.2.4 Funkcionálne požiadavky

- Systém musí obsahovať GIS (Geografický Informačný systém)
- Systém musí mať databázy pre archivovanie údajov o zásielkach a pre ostatné podsystémy
- Systém musí zdialať informácie medzi centrami
- Pre každú zásielku sa musí vygenerovať unikátny čiarový kód
- Zásielky musia byť doručené podľa dohody so zákazníkom s čo najmenšou odchylkou
- Počet pokrytých štátov musí byť čo najrozsiahlejší s optimálnou hustotou
- Spoločnosť musí byť schopná prepraviť tovar ktorý zákazník požaduje prepraviť
- Spoločnosť musí byť schopná prepraviť množstvo tovaru aké zákazník požaduje prepraviť
- Spoločnosť musí vybaviť vyplatenie tovaru za zákazníka
- Cena musí byť prijateľná pre zákazníka
- Zákazník musí dostávať informácie o zmene stavu zásielky
- Systém musí mať záložný zdroj energie
- Pre spracovávanie zásielok budú potrebovať kuriéry a skladníci ručné skenery
- GIS (Geografický Informačný systém) bude musieť aktualizovaný raz do roka

- Údržba systému sa bude vykonávať každého polroka
- Databázový systém potrebuje vedieť uchovať informácie pre dva milióny zásielok ročne
- Informácie musia byť zdieľané medzi centrami do niekoľko minút až hodiny
- Systém musí mať možnosť nastaviť zásielke unikátnu kombináciu skratiek štátu s destináciou
- Systém musí byť preložený v hlavných svetových jazykoch
- Systém musí obsahovať program na výpočet vyvažovania lietadla a uloženie zásielok s extrémne vysokou presnosťou, viacnásobnou kontrolou a spoľahlivosťou
- Užívateľ musí byť schopný pri používaní systému robiť maximálne jednu chybu za hodinu
- Zákazník musí dostávať informácie o zmene stavu zásielky do niekoľkých minút po zmene
- Pri výpadku systému musí nabehnúť záložný zdroj do jednej minúty
- Systém pre váženie a meranie zásielok musí byť spustiteľný cez internetový prehliadač
- Systém musí obsahovať prostredie pre zákazníka
- Systém skenerov musí byť kompatibilný s informačným systémom
- Užívateľ musí vedieť zmeniť stav zásielky v priebehu jej spracovania
- Pri výluke dopravného uzla musí užívateľ mať schopnosť vložiť informácie o výluke do GIS
- zákaznícka zóna musí obsahovať vysvetlenie a označené povinné atribúty a výpočet ceny prepravného
- Skenery budú musieť vydržať osem - hodinú smenu a na každý skener bude treba jeden záložný zdroj, dvakrát za hodinu budú musieť odosielať informácie do databázy
- pre každú krajinu musí byť dostupná štátna mena

1.2.5 Nefunkcionálne požiadavky

- Geografický systém musí ukladať súradnice objektu s presnosťou do jedného metra každú uplynulú sekundu
- Systém bude používať na PC aktuálny operačný systém
- Spracovanie bude zásielok prebiehať cez internetový prehliadač kde bude použitá knižnica JQUERY
- pracovať sa bude s ORACLE databázami
- pre finančné operácie a účtovníctvo sa bude používať program SAP
- Systém pre vyvažovanie lietadla bude vyvíjaný pomocou C,C++ a musí spĺňať smernice podľa DO-178B, Software Considerations in Airborne Systems and Equipment Certification
- Systém musí spĺňať smernice CIS (Certified International Specialist)
- systémy budú vyvíjané a plánované inkrementálnym a V modelom
- Záložný zdroj - ELEKTRICKÝ ZDROJOVÝ AGREGÁT
- siete Wifi + lan, 1024 zariadení na sieť v centre

1.2.6 Doménové požiadavky

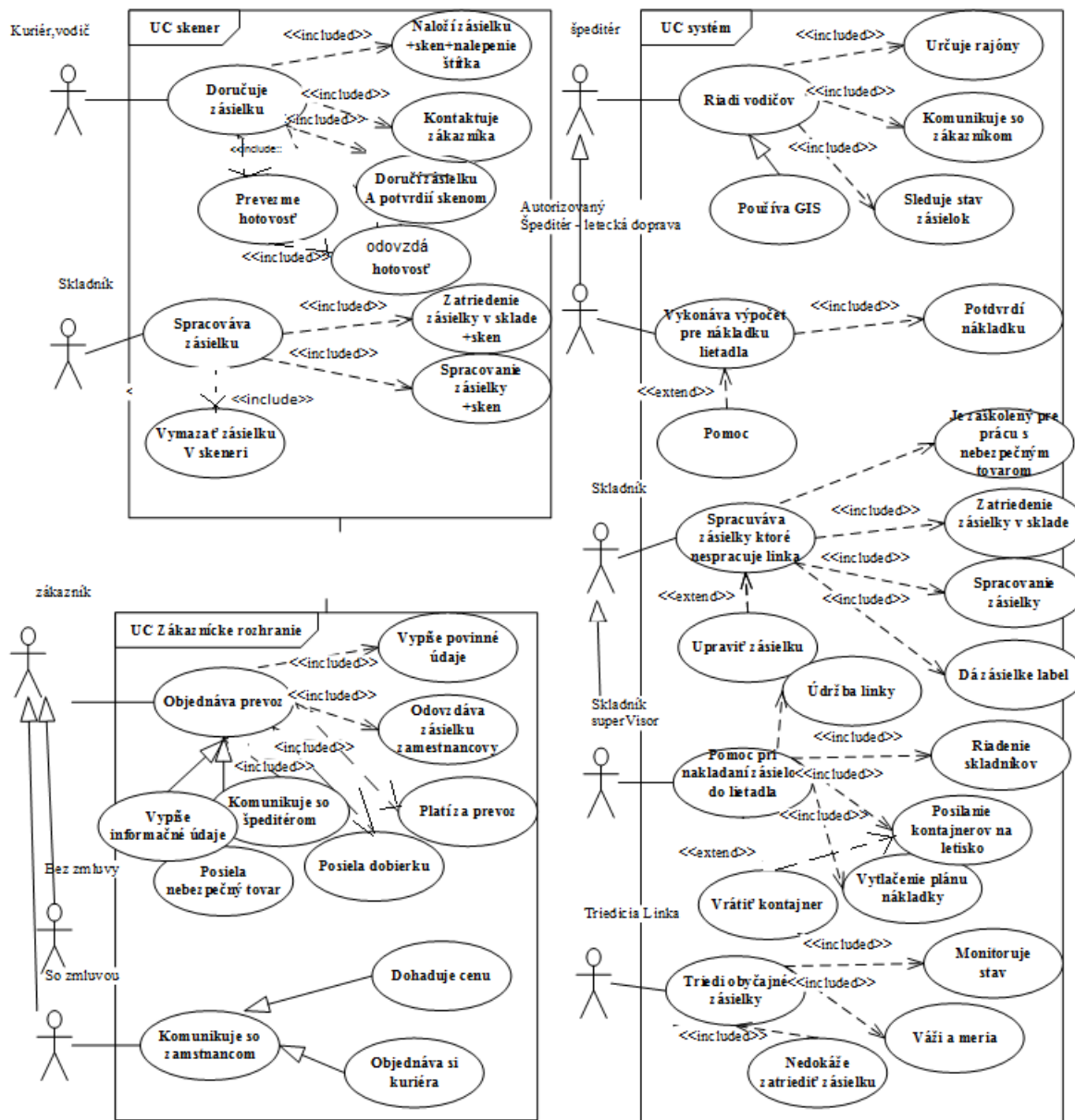
- Váha zásielok prepravovaného lietadlom nesmie prekročiť 90% užitočnej hmotnosti lietadla, pri každej operácii musí užívateľ potvrdiť operáciu v potvrdzovacom okne
- Váha zásielok musí byť meraná najmenej na celé čísla s max. povolenou odchýlkou 0,5kg, potvrdenie zásielky a skákkanie medzi atribútmi sa bude vykonávať pomocou klávesy

TAB

- tlač štítku pri spracovaní zásielky sa vykoná hneď po naskenovaní zásielky
- Pre položky Danger Goods musí byť vytvorené choiceMenu s triedami nebezpečia
- po zaškrtnutí buttonu dobierka musí byť vložená suma v požadovanej mene

2 Systémová špecifikácia

2.1 Diagramy prípadov použitia



2.2 Stručne popíšte role jednotlivých hráčov

2.2.1 Skladník

Manipuluje a spracováva zásielky v sklade. Je zaškolený pre prácu s nebezpečnými balíkmi. Nadrozmerné balíky, listové zásielky, a palety spracuje a vloží údaje do systému.

2.2.2 Skladník - super-visor

Riadi skladníkov a zúčastňuje sa priamo nákladky lietadla kontajnermi. Má na starosti aj údržbu triediacej linky.

2.2.3 Zákazník

Pomocou zákazníckeho rozhrania vkladá údaje o preprave tovaru. Vypĺňa povinné (adresa odosielaťa, príjemateľa), informačné údaje a ak chce poslať nebezpečný tovar musí vyplniť údaje o triede nebezpečného tovaru.

2.2.4 Zákazník - registrovaný

Ide o zákazníka ktorí posielajú tovar pravidelne spoločnosť mu ponúka prepravu za výhodnejšiu cenu. Pri nákladke môže aj priamo kontaktovať kuriéra pre rýchlejšie vybavenie prepravy.

2.2.5 špeditér

Riadi kuriérov v rajónoch, snaží sa rozdeliť prácu rovnomerne. Môže si pomáhať aj s geografickým informačným systémom ktorý v ktorom môže vkladať informácie o zákazníkoch a výlukách v doprave.

2.2.6 Autorizovaný špeditér - letecká doprava

Komunikuje so zamestnancami letiska, plánuje naloženie lietadla podľa vopred určených štandardov

2.2.7 triediacia linka

Pomocou skenovania, umelej inteligencie, neurónových sietí a strojového učenia triedi obyčajné zásielky. ak zásielku nedokáže spracovať tak ju pošle medzi nespracované.

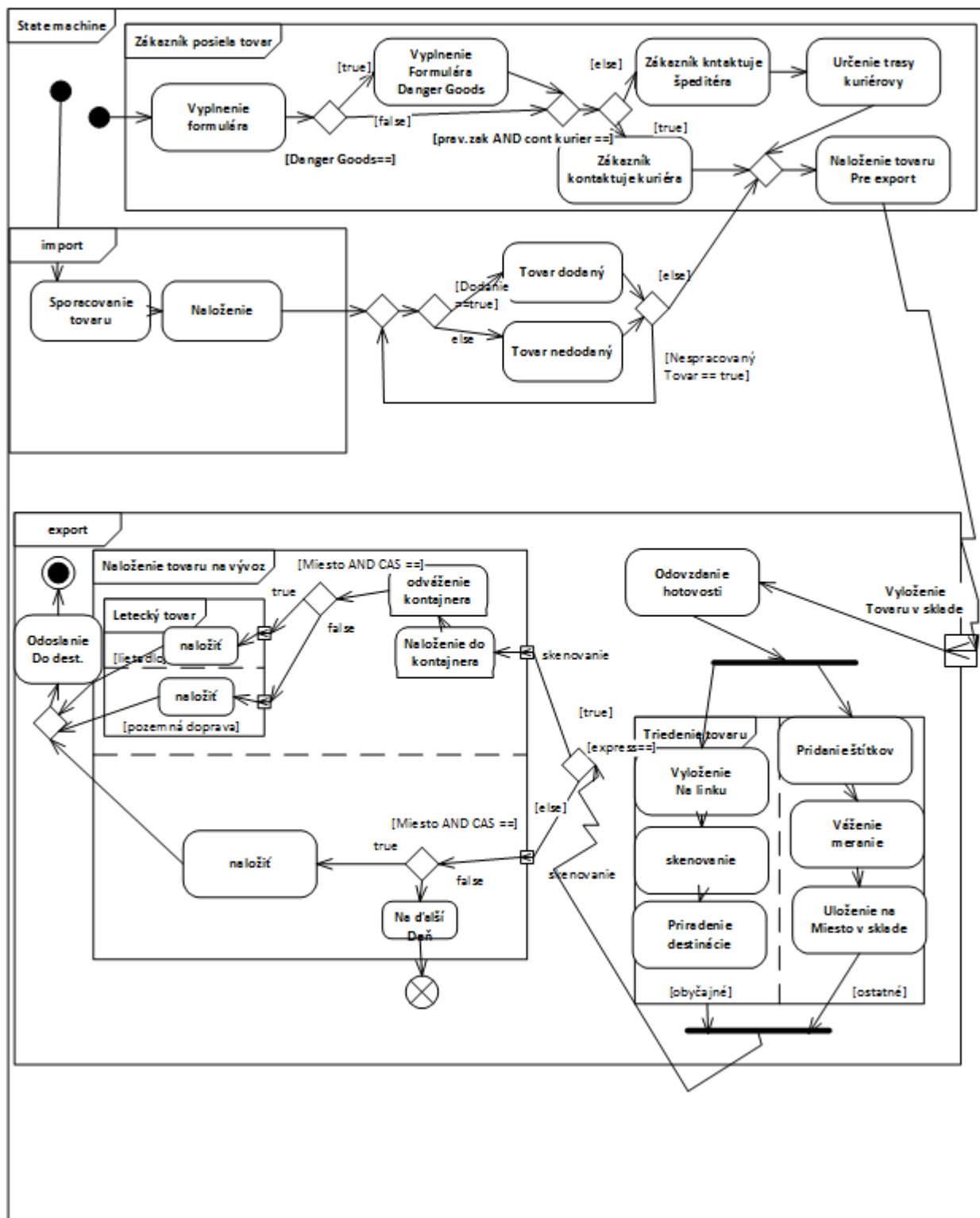
2.2.8 kuriér - vodič

Vykonáva rozvoz a nákladku zásielok po rajóne, pritom komunikuje so špeditériami pravidelnými zákazníkmi. Pri doručení zásielky pomocou skenovanie mení stav zásielky. Pre zásielky na dobierku vyberá hotovosť, Pri nákladke tovaru musí kuriér označiť zásielky vstupnými štítkami.

2.3 Tabuľka - tri najzložitejšie prípady

ID	názov	vstupné podmienky	hráči	postupnosť udalostí	alternatívnu postupnosť	poznámky
US1	výpočet pre nákladku lietadla	naplnená skupina kontajnerov a odvážaná	Skladník supervisor autorizovaný špeditér pre leteckú dopravu	čaká sa na naplnenie kontajnerov, ktoré sa označujú a vkladajú do systému ktorí vypočítajú ich rozloženie v lietadle	ak sa kontajnery nestihú naplniť pred odletom môžu byť poslané cez pozemnú dopravu do iného AIR strediska	nesmie byť prekročená úžitková kapacita lietadla
US2	Triedenie obyčajných zásielok	balík so štítkom a so vstupným čiarovým kódom	Skladník supervisor triediacia linka Skladník	Skladníci nakladajú zásielky na triediaciu linku ktorá hľadá optimálne riešenie pre zatriedenie	zásielky sa spracujú ručne ako nadrozmerné zásielky	
US3	objednanie prevozu	vyplnenie povinného formulára	zákazník (pravidelný) kuriér	zákazník kontaktuje zamestnancov po vyplnení formulára, kuriér si prídje vyzdvihnúť a potvrdiť zásielku	zákazník si priniesie tovar priamo do skladu na spracovanie	v prípade nebezpečného tovaru musí byť vypísaná trieda nebezpečia

2.5 Stavový diagram



PS názov spoločnosti: Jupí-Yes Hard Systems

Použitá Literatúra

Regional / Global Restricted Commodities Group, Danger Goods Training - ICAO/IATA Cat.8 - Handling Staff Loading / Unloading, 2015

Globálne riaditeľstvo spoločnosti DHL EXPRESS, Príručka pre účastníkov, DHL Global EXPRESS Fritz Erdel Strasse 5, 53113 Germany, www.dhl.com

2.6.1 Zaznamenávanie zmien

Tabul'ka 1: Zmeny

```

classDiagram
    class GIS {
        <<interface>>
        + názovRajónu : string
        + urobRajón(lokality : string)
    }
    class pomocnyPodsystemGIS {
        - helpData : string
        + makeHelp(input : string) : string
    }
    class zakaznik {
        + zavolaaj()
        + pouzivaZakRozhranie()
    }
    class zakaznikSoZmluvou {
        - DieloZák : int
        - telCislo : unsigned short
        + vybavCenu() : float
        + kontaktujKuriéra()
    }
    class zakaznickeRozhranie {
        <<interface>>
        + vytvorZasielku() : zasielka
    }
    class dodávka {
    }
    class vozidlo_nad_3_5 {
        <<interface>>
        + váhaMax : float
        + kapacita : int
    }
    class parametreProstriedku {
        - váhaMax : float
        - kapacita : int
    }
    class VendorParameters {
        - meno : string
        - priezvisko : string
        - kontaktná osoba : string
        - názov firmy : string
        - Telefón : unsigned short
        - address : string
        + setVendorPar(new_Vendor : VendorParameters)
        + getVendorPar() : VendorParameters
    }
    class address {
        - mesto : string
        - ulica : string
        - cislo_domu : string
        - PSC : unsigned short
        + setAddress(newAddr : address)
        + getAddr() : address
    }
    class origin {
        - origVen : VendorParameters
    }
    class destination {
        - destVen : VendorParameters
    }
    class triediaciaLinka {
        - ide : bool
        + triedZasielky()
    }
    class vozidlo_nad_3_5 {
        - seriové číslo : int
        - dataSoZasielkami : zasielka
    }
    class zasielka {
        - from : origin
        - to : destination
        - hodnota : float
        - dobierka : float
        - váha : float
        - rozmiery : 3Drozmer
        + getZasielka() : zasielka
        + setZasielka(new_zasielka : zasielka)
    }
    class dangerGoods {
        - classDGzasielka : classDG
        + setClassDG(inout actClassDG : classDG)
        + getClassDG() : classDG
    }
    class kontajner {
        - dataSoZasielkami : zasielka
        - váha : float
    }
    class skladnik {
        + odovzajZasielku() : float
        + odmerajZasielku() : 3Drozmer
        + skenujZasielku() : string
        + spracujNadRozmeryTovar(new_parameter : nadrozmerne)
        + uloZZasielku() : zasielka
    }
    class skladnikSupervisor {
        + ovladaajLinku()
        + odovzajKontajner() : kontajner
        + naloZPalety() : paleta
    }
    class nadrozmerne {
        - maxRozmiery : 3Drozmer
        - minVáha : int
        + setMaxRozmiery(newMaxRoz : 3Drozmer)
        + getMinVáha(newMinVáha : int)
        + getMinVáha() : int
    }
    class paleta {
        - maxVáha : float
        + setMaxVáha(newVáha : float)
        + getMinVáha() : float
    }
    class 3Drozmer {
        - x : float
        - y : float
        - z : float
    }
    class lietadlo {
        - odvazeneKontajnery : kontajner
        + getVáhaKontajnr() : float
    }
    class spediter {
        - zoznamRajónov : rajón
        + urciRajón()
        + komunikuj()
    }
    class rajón {
        - uliceArray : string
    }
    class vodič_kuriér {
        - zásielkyNaRozvoz : zasielka
        - aktRajón : rajón
    }
    class autorizovanySpediter {
        - zoznamKontajnerov : kontajner
        + vyvážLietadlo() : kontajner
        + pouzivaajGIS()
    }
    class GIS {
        <<interface>>
        + getSum() : float
        + setSum(newSum : float)
    }
    class dobierka {
        + getSum() : float
        + setSum(newSum : float)
    }
    class sklad {
        - zasielky : zasielka
    }
    class GIS {
        <<interface>>
        + getSum() : float
        + setSum(newSum : float)
    }
    class dobierka {
        + getSum() : float
        + setSum(newSum : float)
    }
    class sklad {
        - zasielky : zasielka
    }
    class nadrozmerne {
        - maxRozmiery : 3Drozmer
        - minVáha : int
        + setMaxRozmiery(newMaxRoz : 3Drozmer)
        + getMinVáha(newMinVáha : int)
        + getMinVáha() : int
    }
    class paleta {
        - maxVáha : float
        + setMaxVáha(newVáha : float)
        + getMinVáha() : float
    }
    class 3Drozmer {
        - x : float
        - y : float
        - z : float
    }
    class lietadlo {
        - odvazeneKontajnery : kontajner
        + getVáhaKontajnr() : float
    }
    class spediter {
        - zoznamRajónov : rajón
        + urciRajón()
        + komunikuj()
    }
    class rajón {
        - uliceArray : string
    }
    class vodič_kuriér {
        - zásielkyNaRozvoz : zasielka
        - aktRajón : rajón
    }
    class autorizovanySpediter {
        - zoznamKontajnerov : kontajner
        + vyvážLietadlo() : kontajner
        + pouzivaajGIS()
    }
    class GIS {
        <<interface>>
        + getSum() : float
        + setSum(newSum : float)
    }
    class dobierka {
        + getSum() : float
        + setSum(newSum : float)
    }
    class sklad {
        - zasielky : zasielka
    }
    class nadrozmerne {
        - maxRozmiery : 3Drozmer
        - minVáha : int
        + setMaxRozmiery(newMaxRoz : 3Drozmer)
        + getMinVáha(newMinVáha : int)
        + getMinVáha() : int
    }
    class paleta {
        - maxVáha : float
        + setMaxVáha(newVáha : float)
        + getMinVáha() : float
    }
    class 3Drozmer {
        - x : float
        - y : float
        - z : float
    }
    class lietadlo {
        - odvazeneKontajnery : kontajner
        + getVáhaKontajnr() : float
    }
    class spediter {
        - zoznamRajónov : rajón
        + urciRajón()
        + komunikuj()
    }
    class rajón {
        - uliceArray : string
    }
    class vodič_kuriér {
        - zásielkyNaRozvoz : zasielka
        - aktRajón : rajón
    }
    class autorizovanySpediter {
        - zoznamKontajnerov : kontajner
        + vyvážLietadlo() : kontajner
        + pouzivaajGIS()
    }
    class GIS {
        <<interface>>
        + getSum() : float
        + setSum(newSum : float)
    }
    class dobierka {
        + getSum() : float
        + setSum(newSum : float)
    }
    class sklad {
        - zasielky : zasielka
    }
    class nadrozmerne {
        - maxRozmiery : 3Drozmer
        - minVáha : int
        + setMaxRozmiery(newMaxRoz : 3Drozmer)
        + getMinVáha(newMinVáha : int)
        + getMinVáha() : int
    }
    class paleta {
        - maxVáha : float
        + setMaxVáha(newVáha : float)
        + getMinVáha() : float
    }
    class 3Drozmer {
        - x : float
        - y : float
        - z : float
    }
    class lietadlo {
        - odvazeneKontajnery : kontajner
        + getVáhaKontajnr() : float
    }
    class spediter {
        - zoznamRajónov : rajón
        + urciRajón()
        + komunikuj()
    }
    class rajón {
        - uliceArray : string
    }
    class vodič_kuriér {
        - zásielkyNaRozvoz : zasielka
        - aktRajón : rajón
    }
    class autorizovanySpediter {
        - zoznamKontajnerov : kontajner
        + vyvážLietadlo() : kontajner
        + pouzivaajGIS()
    }
    class GIS {
        <<interface>>
        + getSum() : float
        + setSum(newSum : float)
    }
    class dobierka {
        + getSum() : float
        + setSum(newSum : float)
    }
    class sklad {
        - zasielky : zasielka
    }
    class nadrozmerne {
        - maxRozmiery : 3Drozmer
        - minVáha : int
        + setMaxRozmiery(newMaxRoz : 3Drozmer)
        + getMinVáha(newMinVáha : int)
        + getMinVáha() : int
    }
    class paleta {
        - maxVáha : float
        + setMaxVáha(newVáha : float)
        + getMinVáha() : float
    }
    class 3Drozmer {
        - x : float
        - y : float
        - z : float
    }
    class lietadlo {
        - odvazeneKontajnery : kontajner
        + getVáhaKontajnr() : float
    }
    class spediter {
        - zoznamRajónov : rajón
        + urciRajón()
        + komunikuj()
    }
    class rajón {
        - uliceArray : string
    }
    class vodič_kuriér {
        - zásielkyNaRozvoz : zasielka
        - aktRajón : rajón
    }
    class autorizovanySpediter {
        - zoznamKontajnerov : kontajner
        + vyvážLietadlo() : kontajner
        + pouzivaajGIS()
    }
    class GIS {
        <<interface>>
        + getSum() : float
        + setSum(newSum : float)
    }
    class dobierka {
        + getSum() : float
        + setSum(newSum : float)
    }
    class sklad {
        - zasielky : zasielka
    }
    class nadrozmerne {
        - maxRozmiery : 3Drozmer
        - minVáha : int
        + setMaxRozmiery(newMaxRoz : 3Drozmer)
        + getMinVáha(newMinVáha : int)
        + getMinVáha() : int
    }
    class paleta {
        - maxVáha : float
        + setMaxVáha(newVáha : float)
        + getMinVáha() : float
    }
    class 3Drozmer {
        - x : float
        - y : float
        - z : float
    }
    class lietadlo {
        - odvazeneKontajnery : kontajner
        + getVáhaKontajnr() : float
    }
    class spediter {
        - zoznamRajónov : rajón
        + urciRajón()
        + komunikuj()
    }
    class rajón {
        - uliceArray : string
    }
    class vodič_kuriér {
        - zásielkyNaRozvoz : zasielka
        - aktRajón : rajón
    }
    class autorizovanySpediter {
        - zoznamKontajnerov : kontajner
        + vyvážLietadlo() : kontajner
        + pouzivaajGIS()
    }
    class GIS {
        <<interface>>
        + getSum() : float
        + setSum(newSum : float)
    }
    class dobierka {
        + getSum() : float
        + setSum(newSum : float)
    }
    class sklad {
        - zasielky : zasielka
    }
    class nadrozmerne {
        - maxRozmiery : 3Drozmer
        - minVáha : int
        + setMaxRozmiery(newMaxRoz : 3Drozmer)
        + getMinVáha(newMinVáha : int)
        + getMinVáha() : int
    }
    class paleta {
        - maxVáha : float
        + setMaxVáha(newVáha : float)
        + getMinVáha() : float
    }
    class 3Drozmer {
        - x : float
        - y : float
        - z : float
    }
    class lietadlo {
        - odvazeneKontajnery : kontajner
        + getVáhaKontajnr() : float
    }
    class spediter {
        - zoznamRajónov : rajón
        + urciRajón()
        + komunikuj()
    }
    class rajón {
        - uliceArray : string
    }
    class vodič_kuriér {
        - zásielkyNaRozvoz : zasielka
        - aktRajón : rajón
    }
    class autorizovanySpediter {
        - zoznamKontajnerov : kontajner
        + vyvážLietadlo() : kontajner
        + pouzivaajGIS()
    }
    class GIS {
        <<interface>>
        + getSum() : float
        + setSum(newSum : float)
    }
    class dobierka {
        + getSum() : float
        + setSum(newSum : float)
    }
    class sklad {
        - zasielky : zasielka
    }
    class nadrozmerne {
        - maxRozmiery : 3Drozmer
        - minVáha : int
        + setMaxRozmiery(newMaxRoz : 3Drozmer)
        + getMinVáha(newMinVáha : int)
        + getMinVáha() : int
    }
    class paleta {
        - maxVáha : float
        + setMaxVáha(newVáha : float)
        + getMinVáha() : float
    }
   
```

4 Akceptačné testy

Funkcionálne požiadavky

id	1	Názov	Vygenerovanie kódu zásielky
Pripad použitia	UC1	Úroveň splnenia testu	Musí —mal by— mohol by
rozhranie	zákazník užívateľské rozhranie		
Účel	Pre každú zásielku sa musí vygenerovať unikátny čiarový kód		
Vstupné podmienky	vyplnený formulár pre novú zásielku, skener		
Výstupné podmienky	vygenerovaný správny čiarový kód		
Krok	Akcia	Očakávaná reakcia	Skutočná reakcia
1	začiatok kódu bude obsahovať skratku DEST	Skratka sa nachádza na začiatku reťazca	
2	pre zvyšok reťazca sa vygeneruje kód	kód je vygenerovaný v správnom formáte	
3	K reťazcu sa musí vygenerovať správny čiarový kód	po naskenovaní kódu je reťazec rovnaký	
4	Reťazce musia byť unikátne a musia sa generovať inkrementálne	Kontrola pomocou vzorca prezeranie celého oboru	

id	2	Názov	Zdieľanie informácií medzi centrami
Pripad použitia	UC2	Úroveň splnenia testu	Musí —mal by— mohol by
rozhranie	Užívateľ, systém		
Účel	Pri zmene údajov v systéme sa musia zmeniť zdieľané dáta		
Vstupné podmienky	Prístup k dátam v systéme		
Výstupné podmienky	Vykonaná zmena v kontrolovaná v inom centre		
Krok	Akcia	Očakávaná reakcia	Skutočná reakcia
1	Vloženie dát	Systém obsahuje dáta	
2	Prístupenie k dátam v inom centre	Dáta sú zdieľané	

id	3	Názov	Výpočet vyváženia lietadla
Pripad použitia	UC3	Úroveň splnenia testu	Musí —mal by— mohol by
rozhranie	Autorizovaný špeditér, systém		
Účel	Pre vstup musí byť vykonaný správny výpočet		
Vstupné podmienky	Testovací vstup váh kontajnerov, pre ktoré je vypočítaný výstup		
Výstupné podmienky	Výstup z testovacích dát vypočítanému výstupu		
Krok	Akcia	Očakávaná reakcia	Skutočná reakcia
1	Vloženie dát	Systém obsahuje dáta	
2	Zadanie výpočtu	Systém reaguje kontrolnými otázkami	
3	Zadanie chybných vstupov	Systém vyvolá chýbné hlásenia	
4	Vykonanie kontrolného výpočtu	Kontrolný výpočet je správny	
5	Kontrola záložného systému	Výpočet je stále vykonávaný	

id	4	Názov	Jazykové menu systému
Pripad použitia	UC4	Úroveň splnenia testu	Musí —mal by— mohol by
rozhranie	Užívateľ, systém		
Účel	Po zmene jazyka sa musí zmeniť obsah gramaticky správne		
Vstupné podmienky	Informačný obsah systému		
Výstupné podmienky	Gramaticky správny preklad		
Krok	Akcia	Očakávaná reakcia	Skutočná reakcia
1	Vloženie dát	Systém obsahuje dáta	
2	Zmena jazyka systému	Obsah systému preloží textový obsah systému do zvoleného jazyka	
3	Preklad obsahu	Expert skontroluje obsah	

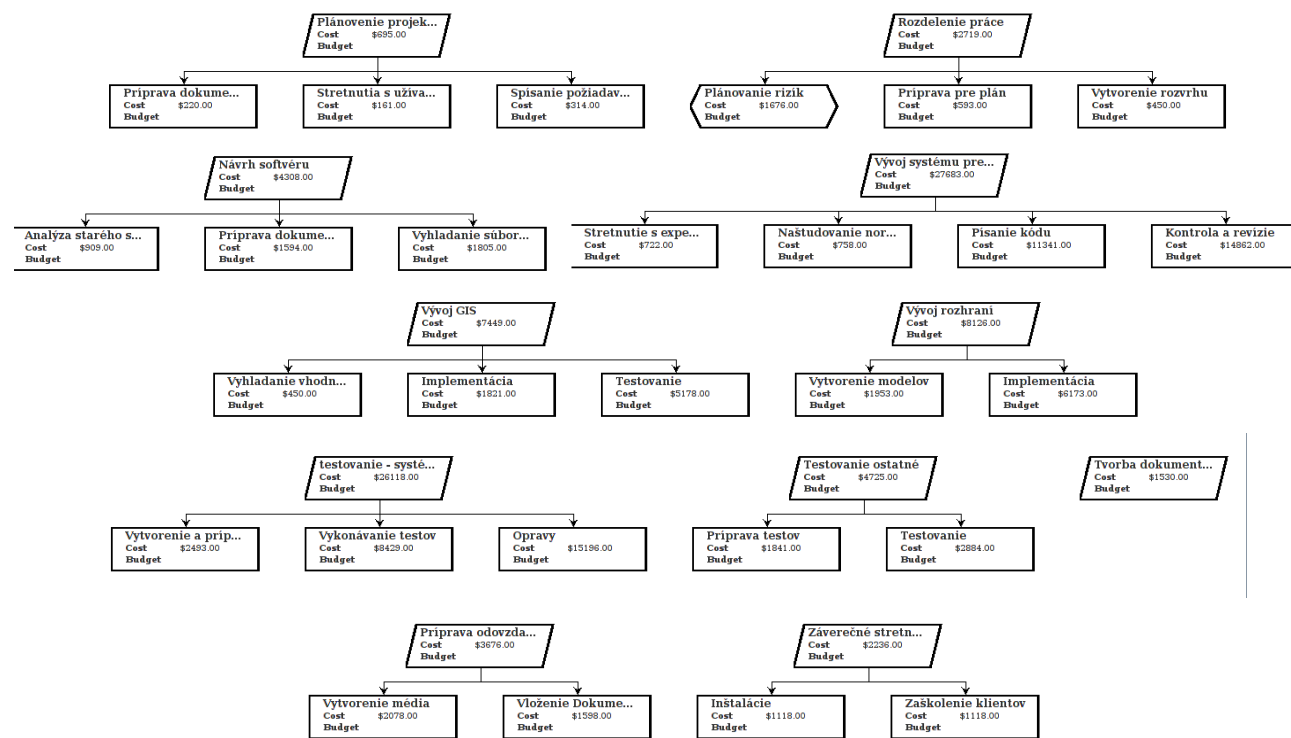
id
Pripad použitia
rozhranie
Účel
Vstupné podmienky
Výstupné podmienky
Krok
1
2

5 Projektové plánovanie

5.0.1 Môj tím

		Name	Type	E-mail Address	Initials	Group	Max. Units	Standard Rate	Overtime Rate	Cost Per Use
1		Dimitrij Vodkov	Work	@ . . .	D	Web DeVelopers	100%	\$12.00/hour	\$20.00/hour	\$100.00
2		Johny Walker	Work	labe@whisky.pub	JW	c.c++ developer	100%	\$18.00/hour	\$30.00/hour	\$17.00
3		Jim Beam	Work	Bourbon@DajSiBoub...	JB	Tester	100%	\$14.00/hour	\$25.00/hour	\$16.00
4		Jack Daniels	Work	D.JJD@jackThe.pub	JD	IT Manager	100%	\$16.00/hour	\$20.00/hour	\$25.00

5.1 WBS-work breakdown schedule



5.2 Ganttov graf a kalendár/plán úloh pre jednotlivých členov tímu

