



Babeş-Bolyai Tudományegyetem, Kolozsvár
Matematika és Informatika Kar
Informatika Szak



Mobil döntéshozatali rendszer biztosítási ügynökök részére

XIII. Erdélyi Tudományos Diákköri Konferencia – Kolozsvár, 2010. május 14–16.

Témavezetők

Dr. Darvay Zsolt, adjunktus
Babeş-Bolyai Tudományegyetem
Matematika és Informatika Kar
Programozási Nyelvek és Módszerek Tanszék

Dipl. Ing. Marcel Mateescu
msg systems Romania
Project Manager

Szerző

Feischmidt László
Babes-Bolyai Tudományegyetem
Matematika és Informatika Kar
Informatika Szak
III. év

2010

Bevezető

Dolgozatomban egy business-alapú feladat megoldására összpontosítok, miközben időbeli- és térbeli korlátok feloldására törekszem. A biztosítások területére térve ki, a legfejlettebb technológiák felhasználásával próbálok olyan alkalmazást létrehozni, melynek segítségével akár külső helyszínről is hozzáférhetők a vállalatnál, adott esetben biztosítótársaságnál futó megoldások. Manapság okos lépésnek bizonyul a mobilitás fejlesztése, hiszen a megcélzott felhasználók, akik leginkább a kereskedelemben dolgozók közül kerülnek ki, egyre több időt töltenek úton, ügyfeleknél, partnereknél, így nekik a lehető legfontosabb, hogy gyorsan, ám biztonságosan használhassák vállalati erőforrásaikat a vállalattól távol is.

Napról napra újabb technológiák jelennek meg, mindez arra ösztönzi a gazdasági szereplőket, hogy lépést tartsanak az előrehaladással a versenyképesség fenntartása érdekében. Éppen ezért a leginkább mobilitással jellemezhető webszolgáltatások (Web Service-ek) használata kerül előtérbe az általam kifejlesztett alkalmazás esetében. Választásom azért esett kimondottan a webszolgáltatásokra, mert ezek esetén egyes alkalmazások térben egymástól el lehetnek választva, függetlenül futhatnak egymástól, több más rendszerből származó kérést is ki tudnak elégíteni, ugyanakkor könnyen újrafelhasználhatók. Továbbá kiemelendő az SAP rendszer és Android operációs rendszerrel ellátott mobiltelefon közötti kommunikáció létrehozásának jelentősége a Web Service-ek segítségével.

Dolgozatom célja a biztosítási kárfelmérő ellenőrök munkájának megkönnyítése, a biztosító cég költségének jelentős csökkentése mellett. Az ügynökök bonyolult munkájának egyik része a baleset esetén a kárt szenvedettek felmérése és annak eldöntése, hogy mennyire jogos a követelt kártérítési összeg. Ehhez az ellenőrnek napi rendszerességű irodai munkát kell elvégeznie (SAP rendszer, ahonnan az ellenőrizendő helyszínrre vonatkozó információkhoz juthat), továbbá a különböző helyszínekre való ellátogatás, valamint a kár felmérése is az ő feladata. Ilyenformán az általam bemutatott alkalmazás mobilitásfejlesztésre irányul, felhasználva a legmodernebb technológiákat: SAP, Android, SOA (Web Services), Java, ABAP, XML, Google Maps API, HTML, JavaScript, kiküszöbölve a számos esetben időigényesnek bizonyuló irodai munkát.

Tartalomjegyzék

Bevezető.....	2
Tartalomjegyzék.....	3
1. Az <i>msg systems ag</i> bemutatása	4
2. Az SAP biztosítási termék	4
2. Az SAP biztosítási termék	5
3. Gépjármű-biztosításról röviden.....	7
4. Az üzleti folyamat.....	8
5. Létező és tervezett funkcionális leírások	10
5.1 Létező funkcionális leírás	11
5.2 Tervezett funkcionális leírás	12
6. A rendszer áttekintése és leírása	13
6.1. Mobil rendszer	13
6.2. JAVA Web Server	14
6.3. SAP WebAppServer / SAP FS-CM.....	14
6.4. Adatbázisrendszer	14
7. Használt technológiák és azok jellemzői	15
7.1. ABAP programozási nyelv	15
7.2. Java és Android operációs rendszer	16
7.3. Web Service-ek (SOA)	16
7.4. Google Maps API, HTML, JavaScript	17
8. Az Android mobil alkalmazás rövid leírása.....	18
8.1. Bejelentkezés	18
8.2. Bejelentkezett.....	18
8.3. Fontossági sorrend beállítása	19
8.4. Ellenőrizendő helyszín kiválasztása.....	19
8.5. Fontossági sorrend kiszámítása.....	19
8.6. Kiválasztott helyszín	20
8.7. Két helyszín közötti útvonal	20
8.8. Részletes információ az útvonalról	21
8.9. Ellenőrzés.....	21
8.10. Utolsó lépés.....	22
Következtetés	23
Bibliográfia	24

1. Az *msg systems ag* bemutatása

Az 1980-ban alakult *msg systems ag* Németország informatikai piacán vezető pozíciót foglal el.

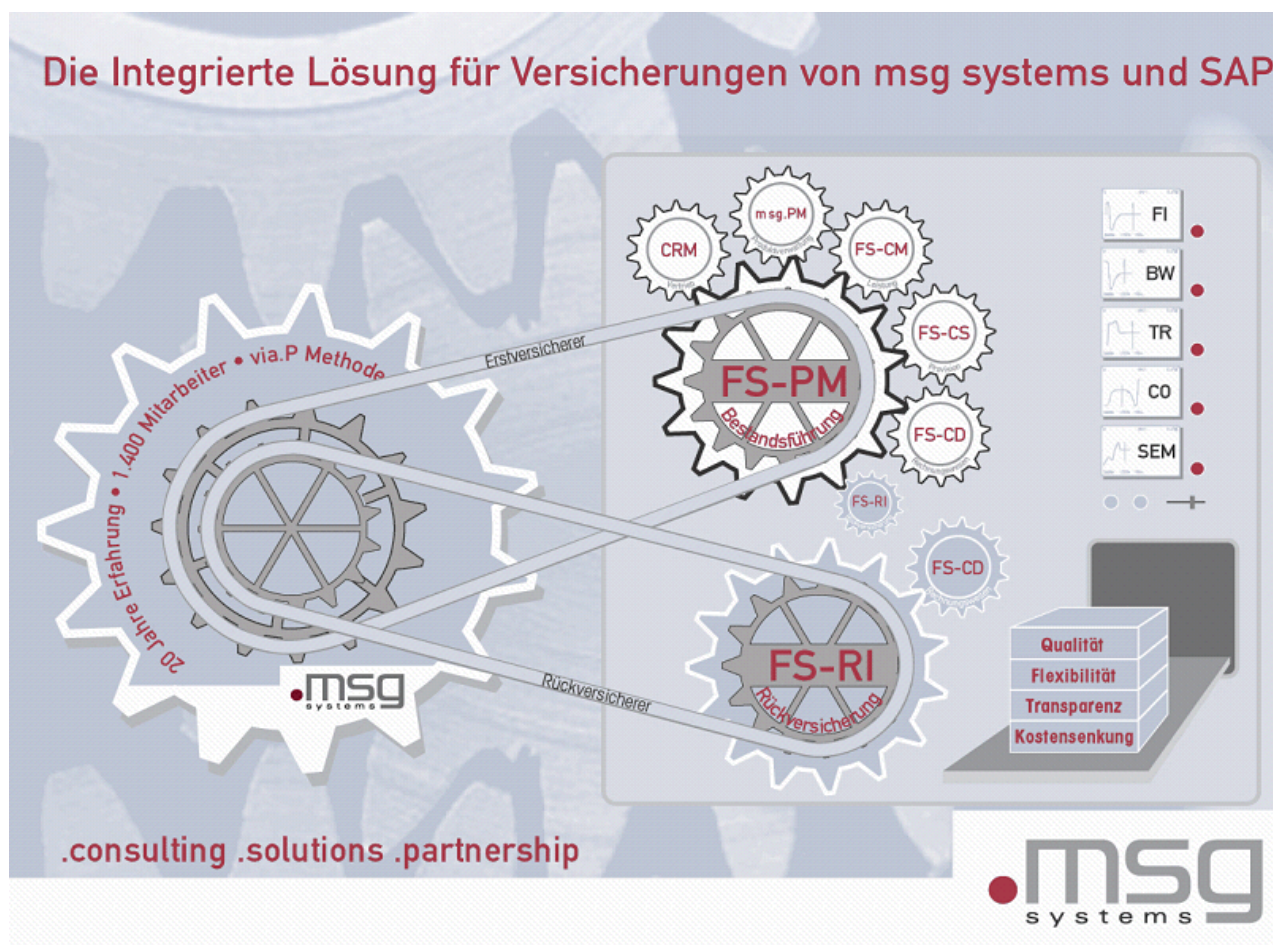
A cég alaptevékenységei közé sorolhatók a következő területek: biztosítás, viszontbiztosítás, autóiipari-, illetve egyéb ipárgspecifikus megoldások, ideértve IT-tanácsadást, rendszerintegrációt, valamint pénzügyi szoftverek készítését.

A németországi gyökerű cég központja Münchenben található, és olyan vezető egyéniségeket tarthat számon, mint Hans Zehetmaier, Karl-Martin Klein, Volker Reichenbach, dr. Dirk Taubner, ugyanakkor Frank Plechinger.

Németország határait túllépve, a világ több országában is sikerült terjeszkedni, így Ausztriában, Svájcban, Hollandiában, Szingapúrban, az Amerikai Egyesült Államokban, Indiában, Spanyolországban, és nem utolsó sorban Romániában. A jelenleg világszerte több, mint 3000 alkalmazottal rendelkező cég nyereségessége növekvő tendenciát mutat évről évre. Mindehhez a Romániában működő leányvállalat kolozsvári székhelye 50 alkalmazott foglalkoztatásával járul hozzá.

Az *msg systems ag* tanúsított vállalat, azaz az ISO 9001-es tanúsított minőségirányítási rendszerrel rendelkezik 1996 óta.

2. Az SAP biztosítási termék



2. Ábra: A biztosítások integrált megoldása az msg systems és az SAP közreműködése által
[Forrás: msg systems]

Az FS-PM egy átfogó, termék-vezérelt kötvény menedzselő¹ rendszer, amely biztosítási kötvény megalkotására és adminisztrálására fekteti a hangsúlyt. Egyik fő eleme az SAP rendszernek az FS-PM biztosítási megoldás, amely különböző modulokat integrál, így például az FS-CD inkasszáló/kifizetési rendszert; az FS-CM kárrendezési szoftvert; FS-CS tartalékképző megoldást, és az FS-RI viszontbiztosítási megoldást vagy éppenséggel a Business Partner és a Business Warehouse modulokat. Mindez lehetővé teszi, hogy a biztosító társaság minden egyes üzleti folyamata optimálisan lefedhető legyen egyetlen integrált megoldásban.

Az FS-PM rendszer lényege az objektumorientált megoldás, amelyet az msg.ProductManager (msg.PM) biztosít. Ez lehetővé teszi a biztosítási termékek egyszerű modellezését és leírását, valamint ezek integrálását a meglévő termékekbe. Ezáltal a létrehozott modulok bármikor felhasználhatók a folyamatosan megjelenő új termékeknel. Az msg.PM biztosítja a háttérben lévő,

¹ Policy Management.

de kihagyhatatlan, biztosításra vonatkozó matematikai képleteket, és azok végrehajtását a megfelelő pillanatban. Ezenkívül a mintacsomag tartalmaz számos előre meghatározott biztosítási terméket.

Továbbá az FS-PM rendkívül rugalmas eszközöket biztosít az üzleti folyamatok és tranzakciók kialakításában. A standard termék számos olyan tranzakciót tartalmaz, amely a kliens egyedi igényeinek felel meg, vagy, hogyha nem, akkor ki lehet egészíteni az új ügyletekkel. Az FS-PM az összes standard irányelv menedzselési funkciót magában foglalja, ugyanakkor széleskörű kiterjesztett funkciókat, beleértve a kockázatértékelést, valamint az alapkezelést biztosítási termékek esetén. Ez a megoldás egyfajta napló funkciót is kínál, az események időbeli irányítását, amelyek zökkenőmentessé teszik az adatok ellenőrzését és elfogadhatóságát.

Az FS-CM modul egy hatékony kárrendezési rendszert kínál az élet-, felelősség-, baleset- és gépjárműbiztosításoktól különböző biztosítások esetében. A követelés feldolgozásának egyszerűsítésével és gyorsításával, az FS-CM nemcsak jelentős megtakarítást képez, hanem fokozza az ügyfelek elégedettségét és hűségét is az adott termékhez.

Az FS-RI rendszer egy olyan integrált viszontbiztosítási megoldást nyújt, amely hamar vezető szoftverré alakult a viszontbiztosítások területén, az elsődleges biztosítóknak és viszontbiztosítóknak egyaránt, az msg rendszerek és az SAP közti szakmai együttműködésnek köszönhetően. Napjainkban számos nagy viszontbiztosító társaság, úgy Európában, mint az Amerikai Egyesült Államokban is sikeresen használja az FS-RI-t, beleértve a következő biztosítási cégeket: Zurich Financial Services, Allstate Insurance, AXA, Gerling és Victoria.[4]

3. Gépjármű-biztosításról röviden

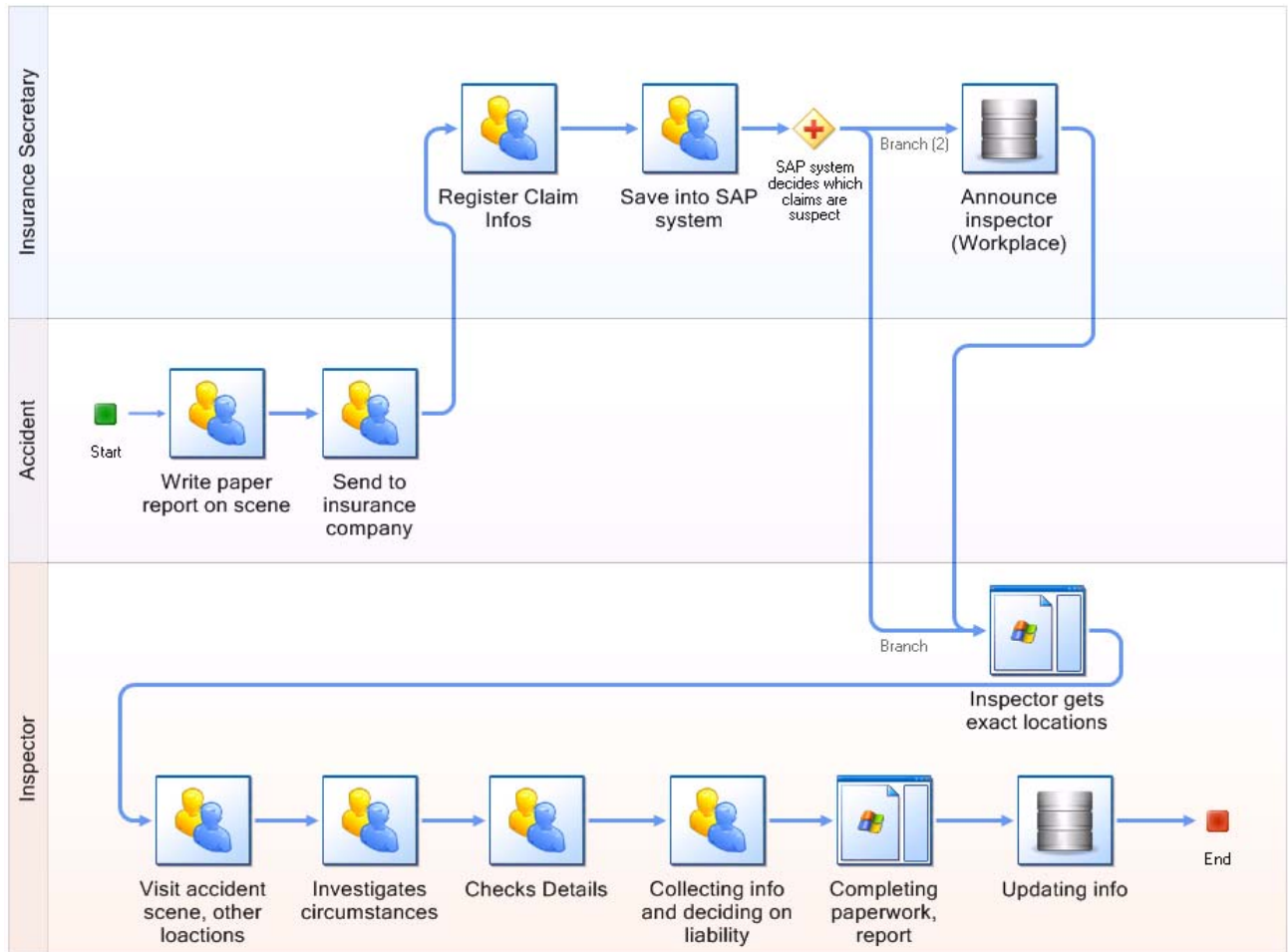
Egy gépjármű veszély forrása is lehet, a balesetekben úgy az utasok, mint a gépjárművek komoly sérüléseket szenvedhetnek el. Hogyan is működnek a gépjármű-biztosítások? Köztudott, hogy a biztosítási esemény avagy baleset bekövetkezése esetén a biztosítótársaság az, aki jótáll az elszenvedett károkért, melyek az előzőleg kockázati fedezet alá vett biztosítás tárgyát érintik. Ilyenformán a biztosítás megkötésétől, a biztosítótársaság átvállalja a kockázatot a biztosítottól, egy biztosítási díj fejében. A gépjármű-biztosítás a felelősségbiztosítások kategóriába tartozik, biztosításkötéskor a kárösszeg mértéke nem határozható meg egyértelműen, ezért egy úgynevezett maximálisan kifizethető biztosításösszegig térít a biztosító.

A biztosítható kockázati fedezetek különböző módon kategorizálhatók. Az alábbiakban bemutatunk ezek közül néhányat.

- **Felelősségbiztosítás:** egy gépjármű üzemeltetője által mások vagyonában okozott károkra és személyi sérülések esetére nyújt fedezetet a kötelező gépjármű-biztosítás, melynek célja az okozott károk megtérítése attól függetlenül, hogy azt a biztosított képes kifizetni vagy sem;
- **Casco biztosítás,** amely a saját tulajdonban levő gépjárműben keletkezett károk megtérítését vállalja. Ilyenformán kötelező biztosítás esetén akkor fizet a biztosító, ha a balesetért egyértelműen a biztosított okolható, míg a casco esetében mindegy kinek a hibájából következett be az esemény;
- **Tűz- és elemi kár biztosítása** (tűz, szél, jégeső, árvíz, vandalizmus, lopás);
- **Személyi sérülésre vonatkozó biztosítás** (orvosi ellátás költségeit téríti meg).[1]

4. Az üzleti folyamat

Az alábbi ábrán a biztosításra vonatkozó üzleti folyamat vagy más néven a Business Process jelenítődik meg.



3.Ábra: Üzleti folyamat

A biztosítási esemény, azaz a baleset, bekövetkezése hosszadalmas folyamatot von maga után, amely végső soron a károsult anyagi kielégítésével fejeződik be, vagy sem.

Adott baleset helyszínén hivatalos papírok kitöltésére kerül sor, amelyek a biztosító cég számára szükségesek a kártérítési folyamat elindítása érdekében. A biztosítónál ezen információk adatok formájában egy adatbázisba kerülnek, jelen esetben a SAP rendszerbe. Az adatbázis segítségével az ügynevezett biztosítási ellenőr a számára szükséges és megfelelő információkhoz jut. Az ellenőrzésre kinevezett ügynök EMAIL kliens-hez hasonló felhasználói felülettel rendelkezik, ahol az INBOX-ban az újonnan bekövetkezett balesetekről szóló értesítők jelennek meg. Az ellenőr feladata ezen történések ellenőrzése, miáltal meggyőződik arról, hogy cége csalás áldozata adott esetben vagy sem.

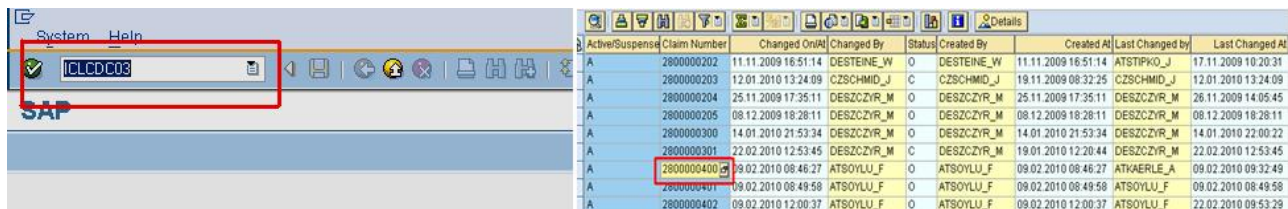
Az ellenőr feljegyzi az ellenőrzés tárgyát képező biztosítási követelés helyszínét. A helyszínre való látogatás során dönti el, valóban jogos-e a biztosított által követelt pénzösszeg nagysága.

A helyszínen felmért és megállapított kár függvényében elfogadhatja vagy visszautasíthatja a biztosított követelését.

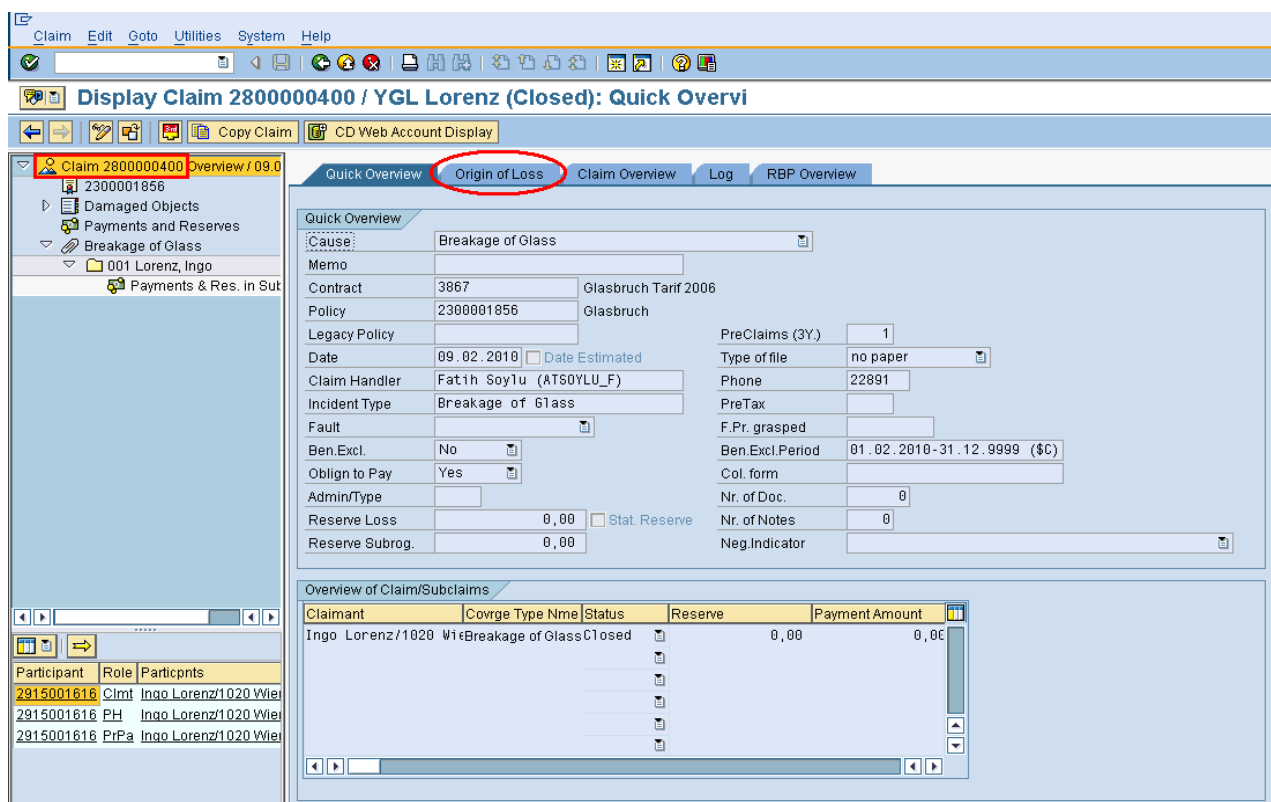
A hivatalos dokumentumok kitöltése után módosításra kerülnek az illető követelés adott jellemzői a rendszerben, ilyenformán új állapot jön létre.

5. Létező és tervezett funkcionális leírások

A követelések kezelését az SAP rendszerben végzi a biztosítási ellenőr. A rendszerbe való belépésekor az ICLCDC03 tranzakcióval megtekintheti a követelések adatait.



4. Ábra: A követelések listája



5.Ábra: Követelésre vonatkozó információk

A származási hely megtekintéséhez kiválasztja az “Origin Of Loss” fület.

Itt a helyszínről részletes információt talál(ország, helység, utca, házszám, postakód).

The screenshot displays a web application for managing claims. The title bar reads "Change Claim 2800000400 / YGL Lorenz (Closed): Origin of Los". The left sidebar shows a tree view with "Claim 2800000400 Overview / 09.0" selected. A red arrow points from the text "Claim NR." to this selection. Another red arrow points from "Loss Location Info" to the "Loss Location" section. The "Loss Location" section contains the following data:

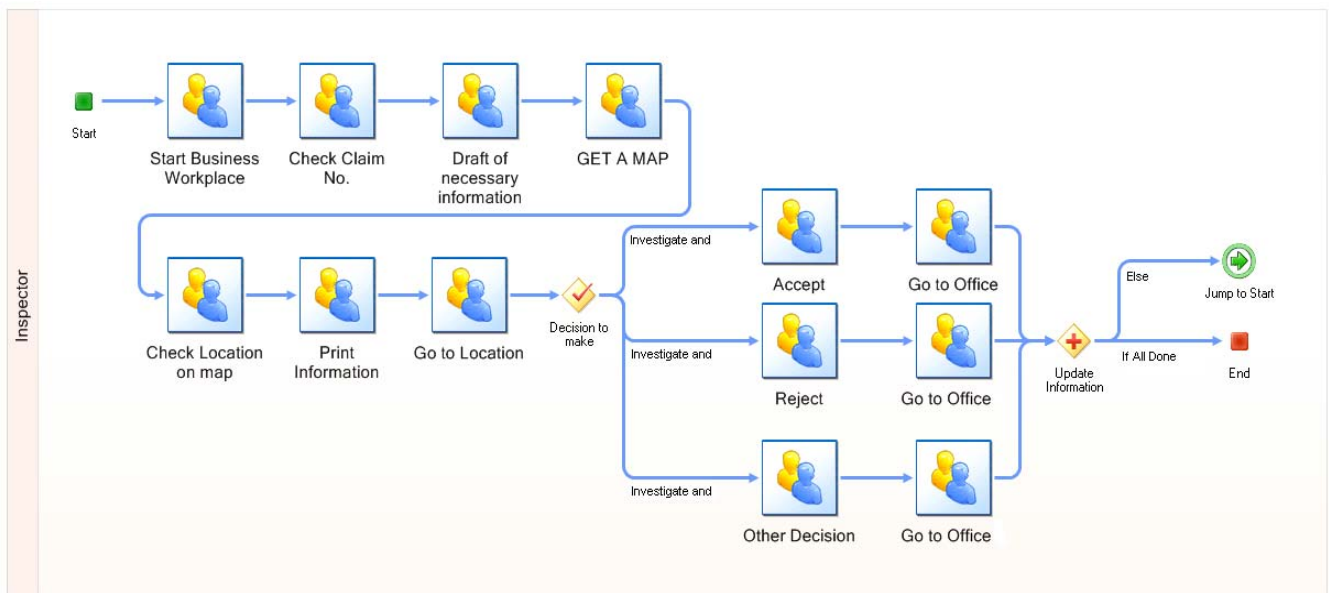
Street/House No	Hamburg Strasse	12
PostalCode/City	4503	HRTEST1
Country/Region	HR	Austria 01

Below this is a "Comments on Loss Location" text area. At the bottom, a table lists participants:

Participant	Role	Participants
2915001616	Clmt	Ingo Lorenz/1020 Wier
2915001616	PH	Ingo Lorenz/1020 Wier
2915001616	PrPa	Ingo Lorenz/1020 Wier

6. Ábra: Helyszínre vonatkozó adatok

5.1 Létező funkcionális leírás

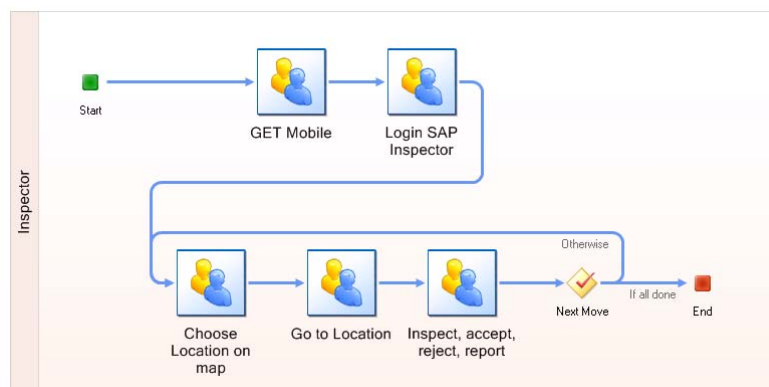


7.Ábra: A biztosítási ellenőr feladatai

Az alábbiakban ismertetem a biztosítási ellenőr feladatainak jellemvonásait, amelyek általában tetszőleges biztosítási követelésre vonatkozhatnak:

- a rendszerben létező és nyilvántartott követeléseket tudomásul veszi; az újonnan bejegyzett követeléseket a rendszer osztja szét az ellenőrök között;
- minden ellenőr saját programmal rendelkezik, ahol az ellenőrzésre váró követelések számai jelennek meg;
- a fentiekben bemutatott módszerrel tudomására jut a helyszín, ahol a felmérésre sor fog kerülni, a követelés számának megfelelően;
- ismeretlen cím esetén egy térkép áll rendelkezésre;
- szükséges a megfelelő dokumentumok megléte a helyszínen való adatfeljegyzés érdekében;
- a kiszállás célja a helyszínfelmérést követő döntéshozatal;
- a megfigyelt eseményre vonatkozó információk rendszerbe való bejegyzése, amely a központhoz való visszatérést igényli.

5.2 Tervezett funkcionális leírás

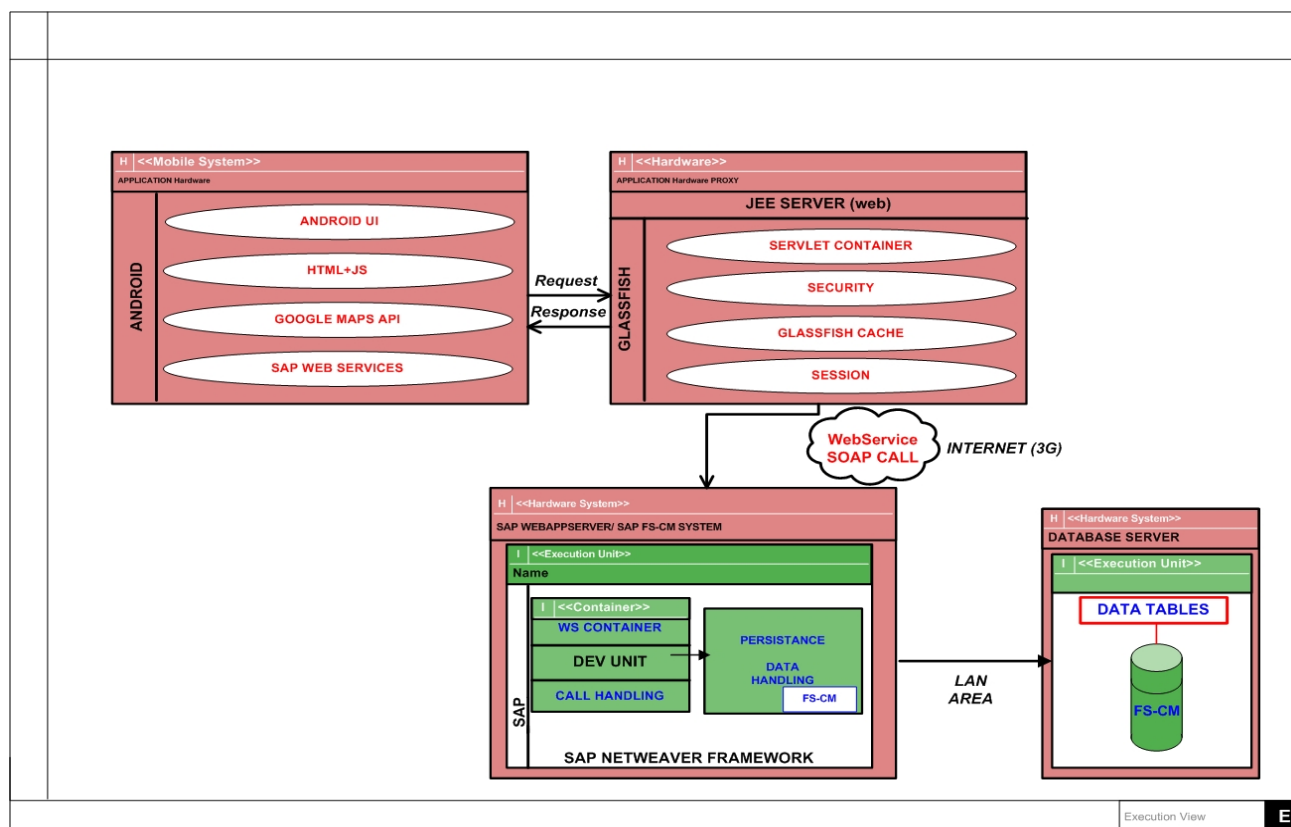


8. Ábra: A Mobile Inspector által ajánlott munkamenet

A biztosítási ellenőr új lépései a Mobile Inspector használatával az alábbiak.

- Fő munkaeszköz a mobiltelefon.
- A mobiltelefonról könnyen be lehet lépni a SAP rendszerbe, az adott ellenőrnek kiosztott biztosítási követelés megtekintése érdekében. A biztosítási követeléshez automatikusan hozzárendelődik a meglátogatandó helyszín koordinátája.
- A mobiltelefonba beépített térkép segítségével könnyen azonosítható a biztosítási esemény helyszíne.
- Az úgynevezett terepszemlélt követően meghozott döntés közvetlenül bejegyezhető a rendszerbe, a központba való visszatérés nélkül.

6. A rendszer áttekintése és leírása



9. Ábra: Rendszerdiagramm

A teljes rendszer 3 fő részből áll:

1. Mobil rendszer,
2. JAVA Web Server,
3. SAP WebAppServer, illetve SAP FS-CM (Claim Management) rendszer,
4. Adatbázisrendszer.

6.1. Mobil rendszer

A mobil rendszerrel kapcsolatban megjegyzendő, hogy 2.1-es verziójú Android operációs rendszerrel rendelkező mobiltelefon szükséges a program installálásához és a felhasználói felület helyes működéséhez. Továbbá a térkép használatához a Google Maps alkalmazásprogramozási felületre (API) is szükség van. A felhasználói felület nagy része a standard Android felhasználói felület elemeit tartalmazza, emellett HTML és JavaScript elemeket is.[3]

A mobiltelefonról egyszerű kérések fognak távozni a Java Web Server irányába, ott ezek a kérések értelmeződnek, és Web Service formájában továbbítódnak az SAP Web Serverhez. Innen válasz érkezik a Java Serverhez, majd ez azt visszaküldi a mobiltelefonnak.

6.2. JAVA Web Server

A Java Web Server egy gateway szerepét tölti be az alkalmazáson belül. Ez a Web Service hívások központja.

A Web Service²-ek töltik be a legfontosabb szerepet az alkalmazásban, hiszen rajtuk keresztül teremődik meg a kapcsolat a SAP rendszerrel. Lényegében különböző programozási nyelven írott programok hívhatják meg egymást általuk. Ilyenformán függetlenné válik a két végpont, gyorsabbá téve a programozó munkáját azáltal, hogy pontosan megadott feladatok párhuzamosan végezhetők.

A kérések boríték (Envelope) formájában továbbítódnak a szerverhez, ahol értelmezve lesznek.

A mobiltelefon és a szerver közötti kommunikáció 3G Internet csatlakozáson keresztül valósul meg.

6.3. SAP WebAppServer / SAP FS-CM

Az SAP szerver a következő négy fontos szerepet tölti be.

- Konténerként szolgál a Web Service-eknek.
- Itt vannak a BO (Business Object)-k, az üzleti logika és a Web Service-ek implementálva .
- A kívülről érkező kéréseket a megfelelő célhelyre irányítja, a kérésre választ küld vissza Envelope formájában.
- Kommunikációt hoz létre az adatbázis szerverrel.

6.4. Adatbázisrendszer

Az adatbázisrendszer az adatbázistáblák tárhelyéül szolgál. Az SAP rendszer bármilyen adatbázisrendszerhez képes kapcsolódni. Újabban a saját fejlesztésű SAP MaxDB adatbázis-kezelő rendszert támogatja leginkább, viszont ez is változhat kienstől függően.

² Egyedülálló, önálló alkalmazások, melyek egy külső webes interfészen keresztül egy kérést szolgáltatnak ki. A kérés hatására az átdott paraméterekkel a Web-service elvégez egy adott műveletet és valamilyen adattal tér vissza. Jelentősége abban áll, hogy az egyes alkalmazások térben egymástól el lehetnek választva, függetlenül futhatnak egymástól, több más rendszerből származó kérést is ki tudnak elégíteni, újrafelhasználhatók.

7. Használt technológiák és azok jellemzői

- ABAP,
- JAVA,
- Android,
- SOA (Web Service-ek),
- Google Maps API,
- HTML, JavaScript.

7.1. ABAP programozási nyelv

Az SAP AG 1972-ben alakult Németországban, rendszerelemzés és programfejlesztés céljából, később viszont átértelmezték feladatukat: rendszerek, alkalmazások és termékek az adatfeldolgozásban³ címen. Jelenleg a legnagyobb szoftvercégek közé sorolható a világon, több vetélytársra is szert tévén, így az IBM, Microsoft, Google, Oracle Corporation vagy az Apple. Körülbelül 20 millió felhasználó élvezi az SAP rendszer előnyeit, mindez az üzletviteli megoldások terén kimagasló eredmény.[6]

Az ABAP az SAP szoftvercég által kifejlesztett programozási nyelv⁴. Az évek során többször is átdolgozásra került, a mai napig tartó fejlesztések folynak a programozási nyelvet illetően. Kifejezetten negyedik generációs⁵ programnyelv, amelyet üzleti alkalmazások fejlesztésére készítettek. Mindezek lehetővé teszik, hogy az ABAP nyelven írt programok a rendszerkörnyezettől nagymértékben függetlenül működjenek. Továbbá megjegyzendő, az ABAP szerves részét képezi az ABAP Workbench, amely egy saját integrált fejlesztőeszközt jelöl. Az úgynevezett Object Navigator-ból⁶ érhető el, és nagy előnye integráltságában rejlik, hisz az ABAP kódban duplán kattintva az objektumok nevére, rögtön megkapható a nekik megfelelő bejegyzés ABAP

³ Német nyelvről fordítva: Systeme, „Anwendungen und Produkte in der Datenverarbeitung”.

⁴ Eredeti neve a 70-es években: „Allgemeiner Berichts -Aufbereitungs - Prozessor“ (általános jelentéskészítő processzor), később pedig az: „Advanced Business Application Programming“ lett.

⁵ Grafikus felületű, objektumorientált programfejlesztő eszközök közé sorolható, amelyek komponens alapú fejlesztési módszert alkalmaznak; megjegyzendő az első generációs nyelvek még gépi bináris kódok, a második alacsonyszintű, gépközel-, míg a harmadik már magasszintű feladatorientált, strukturált programozási nyelvek.

⁶ SE80 tranzakcióból.

Dictionary⁷-ben, és fordítva. Az ABAP programozási nyelv független az adatbázis-kezelőtől és operációs rendszertől, a nyelv része az Open SQL.

7.2. Java és Android operációs rendszer

“Az Android a Google Linux alapokra helyezett, de Java nyelven programozható mobil platformja. Az Android érdekességét éppen ez a két jellemző adja: az alapszintű működését és az eszközök kezelését egy módosított Linux kernel biztosítja, amelyen egy Java virtuális gép fut, és az operációs rendszer szolgáltatásait már Java nyelven írt programok használják.” [7]

7.3. Web Service-ek (SOA)

Az SAP világában az elmúlt évek egyik fő témája a webszolgáltatás volt. Ennek a területnek jellemző kifejezései a SOA⁸, UDDI⁹, WSDL¹⁰, SOAP¹¹. A webszolgáltatások különböző platformon futó alkalmazások között teremtik meg az adatcserét.

Legjellemzőbb vonása a különböző programozási nyelvben megírt szoftverek közötti kapcsolatteremtés. A webszolgáltatások szabványosításáért a W3C és az OASIS felelős. Számos előnye közül megemlítendő az alacsony fejlesztési és karbantartási költség, könnyű és gyors újrafelhasználhatóság.

Az SAP webszolgáltatások által használt szabványok:

- XML
- WSDL
- WSDL
- UDDI [5]

Az ABAP programozási nyelvben különböző elemekből lehet Web Service-t készíteni. A két legelterjedtebb módja, az RFC¹²-re alkalmas funkciós modulból (távoli függvény hívására alkalmas modul) és RFC-re alkalmas funkciós csoportból, (ami tartalmaz RFC-re alkalmas funkciós

⁷ Struktúrák, adatbázistáblák, adatobjektumok, belső táblák, adattípusok, függvénykönyvtárak tárhelye.

⁸ Service Oriented Architecture.

⁹ Universal Description Discovery and Integration.

¹⁰ Web Service Definition Language.

¹¹ Simple Object Access Protocol.

¹² Remote Function Call.

modulokat), való Web Service meghatározása. A funkciós modul kimeneti és bemeneti paraméterei alkotják a kérésnek és a válasznak megfelelő paramétereket. A modulon belül hívódik meg az üzleti logika.[6]

7.4. Google Maps API, HTML, JavaScript

Mivel az Android operációs rendszer fejlesztését a Google tervezte és valósította meg, a Google Maps beépítése a telefonokba egyértelmű lépésnek számított (az Apple Iphone-on már működött). A legújabb operációs rendszer kiadásakor már a Google Earth is elérhetővé vált, a desktop alkalmazásnak szinte minden opcióját felhasználva.

Ennek az előnyét a programozók nagy mértékben kihasználhatják, mivel bármilyen Android-os alkalmazásba beépíthető, JAVA nyelven programozható. Az egyik legérdekesebb és legfontosabb opciót viszont nem tették publikussá: két vagy több cím közötti legrövidebb útvonal meghatározását. Ez komoly fejfájást okoz azoknak a programozóknak, akik ilyen jellegű alkalmazást szeretnének írni. Egyik általam is használt megoldás erre, a standard Google Maps paramétereivel kitöltött kérésnek (request) megfelelő, KML (XML) formátumban visszatérített válasz értelmezése (koordináták listája, utcanevek, távolságok, időtartamok).

A **HTML** és a **JavaScript** nyelveket a táblázatok könnyű kezelhetősége érdekében használtam. Ezeket az úgynevezett WebView segítségével jelenítem meg. A WebView-k beépíthetők az alkalmazásokba, így Android „desktop” alkalmazás keretében webes alkalmazást futtathatunk.

8. Az Android mobil alkalmazás rövid leírása

8.1. Bejelentkezés

Az alkalmazás indításakor a biztosítási ellenőr a felhasználói felülettel szembesül. A felhasználói felület három fő elemet tartalmaz: felhasználónév, jelszó, valamint a belépés jóváhagyása. A szerveren a szerződött ellenőrök tárolva vannak, a megfelelő adatbázistáblában.

A megfelelő felhasználói név és jelszó megadásával létrejön az első kapcsolat az SAP rendszerrel. Ekkor lépnek működésbe a Web Service-ek. Ha a megadott jelszó és a felhasználói név helyes, akkor a következő oldalra ugrik az alkalmazás a LOG gomb lenyomása által. Ha a megadott adatok helytelenek, azaz nem léteznek, akkor egy megfelelő hibaüzenet jelenik meg a képernyőn.

A mobiltelefon és a szerver közötti kapcsolat 3G Internet csatlakozás által valósul meg.

8.2. Bejelentkezett

Sikeres bejelentkezés esetén számos menüből választhat a felhasználó. Az általa elfoglalt aktuális pozíció meg fog jelenni a mobiltelefonba beépített térképen (beépített GPS, Google Maps API).

Az egyik legfontosabb kiválasztható menüpont a **PRIORITIES**. Itt állítandók be a megfelelő vizsgálati kritériumok. Ezen opciók mentés révén a továbbiakban is érvényesek maradnak.

A **POSITION** menüpont további információt nyújt a felhasználó aktuális pozíciójáról és a térségről, ahol található.



10. Ábra: Bejelentkezés



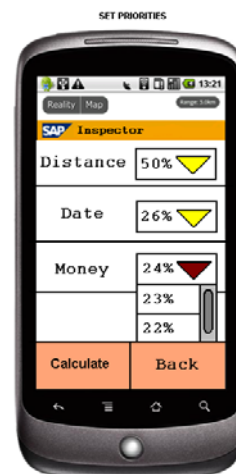
11. Ábra: Bejelentkezett oldal

A felhasználó kiválaszthatja a térkép megtekintési módját (műhold, utak, nagyítás, illetve távolítás).

8.3. Fontossági sorrend beállítása

Kötelezővé válik három kritérium beállítása:

- Távolság,
- Kártérítési összeg,
- A kártérítésre vonatkozó követelés bejelentésének időpontja.



Mindezen értékek százalékos nagyságok, a fontossági sorrendet a magasabb százalékarány adja. Ez utóbbi képezi majd a helyszínek megvizsgálásának sorrendjét.

12. Ábra: Fontossági sorrend

8.4. Ellenőrizendő helyszín kiválasztása

Különböző keresési feltételek megadásával a felhasználó lekérheti a rendszerből a helyszínek listáját. Dátum szerinti keresés esetén, a megadott időpont után bejegyzett helyszínek listája jelenik meg. Két különböző dátum megadásával az ezek között bejegyzett helyszínek listája térítődik vissza. Továbbá ország szerinti keresés is lehetséges.

Az eredményül kapott listából kiválasztható az illető ellenőr számára



elsőbbséget élvező helyszín. Ebben az esetben két opció állhat rendelkezésre:

- Kiválasztható az elsőszámú helyszín, amely vizsgálatra kerül;
- A fontossági kritériumok alapján, a rendszer által kiválasztott helyszínek vizsgálatára kerülhet sor.

13. Ábra: Helyszín lista

8.5. Fontossági sorrend kiszámítása

A helyszínek látogatásának fontossági sorrendje változhat a megadott kritériumok módosításával: távolság, kártérítési összeg, a kártérítésre vonatkozó követelés bejelentésének időpontja szerint.

Egyszerű algoritmus:

A meglévő helyszínek és az azoknak megfelelő adatok normalizálásával, egy nagyon egyszerű algoritmus határozható meg a helyszínek látogatási sorrendjére. Százalékban kifejezhetők a meglévő kifizetendő összeg, távolság, idő értékei. Az összes helyszínek megfelelő előbb felsorolt adatok külön-külön összeadódnak, majd a helyszínenként változó értékkel ez az összeg elosztódik. Ekkor százalékban kifejezett értékek kaphatók. Megadott prioritásokkal ezen értékek beszorzódnak és összeadódnak. Így kapható meg a végső fontossági sorrendet.

Distance	Money	Date	Location Name	Distance Prio %	Money Prio %	Date Prio %	VISITING PRIOS
100	100	1	Passau	14,29%	1,45%	25,00%	15,14%
50	4000	2	Frankfurt	7,14%	57,97%	50,00%	20,80%
250	2500	1	Munchen	35,71%	36,23%	25,00%	33,62%
300	300	0	Berlin	42,86%	4,35%	0,00%	30,43%
700	6900	4		100,00%	100,00%	100,00%	100,00%
				Distance Prio	Money Prio	Date Prio	Highest Prio
				70,00%	10,00%	20,00%	

14. Ábra: Prioritás meghatározása

8.6. Kiválasztott helyszín

A fentiekben bemutatott módszerrel kiszámított fontossági sorrend a következő képernyőn levő listában jelenítődik meg. Ekkor a megfelelő menügomb kiválasztásával az ellenőr útnak indulhat. Az első vizsgálandó helyszín az lesz, amely a listában a legnagyobb prioritással rendelkezik. A rákövetkezők prioritás szerint csökkenő sorrendben

követik egymást.



15. Ábra: Prioritási sorrend

16. Ábra: Útvonal kirajzolása

8.7. Két helyszín közötti útvonal

A Google Maps segítségével könnyedén meghatározható két pont közti útvonal. A Google Maps standard paramétereinek helyes megadásával bármilyen információ leszűrhető az aktuális útvonalról. Kérésre a válasz



XML formátumban jut el hozzánk. Ennek az XML-nek a pontos neve: KML¹³. Ezzel a formátummal a Google Earth program esetében is találkozhatunk. A KML pontos vizsgálatával az útvonal összes koordinátája kiolvasható. Ezekre főként akkor van szükség, amikor magát az útvonalat rajzoljuk, jelenítjük meg (mivel a mobiltelefon Google API-ja nem tartalmazza az optimális út kirajzolását, meghatározását).

Kirajzoláskor a telefonvászon és az arra kirajzolandó vonalhalmaz-opció használatos. Vászonnként a térkép szolgál, a koordináták pedig a szakaszok végpontjai. Mivel a vászonra rajzoláskor egy pontos A(x1, y1) és B(x2, y2) koordinátájú végpontokkal rendelkező szakaszt rajzolunk, ezért a mi geokoordinátáinkat a vászonnak megfelelő(x,y) koordinátává kell alakítanunk.

A teljes útvonal hossza (km, min) szintén a KML válaszból kapható meg.

A menüből kiválasztható, hogy erre az információra van-e szükségünk vagy részletes leírást szeretnénk az útvonallról.

8.8. Részletes információ az útvonallról

A részletes információ mint kifejezés a következő elemeket takarja:

- Utcanév,
- Irány,
- Úthossz,
- Időtartam.

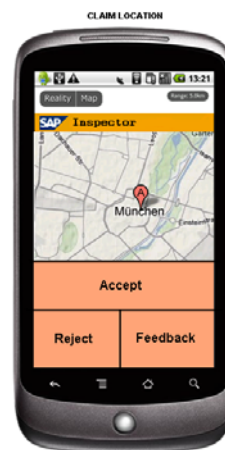
Ezen információk minden irányváltottnál megkaphatók, szintén a KML-válaszból olvashatók ki.

8.9. Ellenőrzés

A helyszínen a biztosítási ellenőr felméri a kár nagyságát és eldönti, hogy a követelt kártérítési összeg jogos-e vagy sem. Ennek függvényében elfogadhatja, visszautasíthatja a kártérítési kérést. Ekkor a döntését egyenesen a SAP-szerverhez küldi, ahol felülíródik a kártérítési kérés korábbi helyzete. Ez egy nagyon fontos lépés, amit anélkül tesz meg, hogy



17. Ábra: Útvonal információk



18. Ábra: Döntéshozatal

¹³ Keyhole Markup Language

visszatérne a központhoz, ezzel értékes időt és pénzt takarítva meg.

8.10. Utolsó lépés

Abban az esetben, ha az ellenőr az összes kitűzött helyszínt meglátogatta, a napi munkájának vége; különben kiválasztja a következő helyszínt és folytatja útját.

Következtetés

A dolgozatomban bemutatott alkalmazás célja a biztosítási kárfelmérő ellenőrök munkájának megkönnyítése volt, a biztosító cég költségének jelentős csökkentése mellett.

Az ügynökök bonyolult munkájának része baleset esetén a kárt szenvedettek felmérése és annak eldöntése, hogy mennyire jogos a követelt kártérítési összeg. Mindezt az SAP Mobile Inspector nevű alkalmazás prototípussal sikerült megvalósítani, kiszűrve az időigényes irodai munkát, és lehetővé téve a biztosítási ügynökök hatékonyabb munkavégzését.

Ezen prototípus más alkalmazások alapját is képezheti a Web Service-eknek köszönhetően. Az ehhez megírt üzleti logika bármely felhasználói felülethez csatlakoztatható (webes-, desktop- vagy mobil alkalmazások) a legújabb technológiákhoz alkalmazkodva.

A vállalati alkalmazások mobiltelefonon való használata nem annyira elterjedt, mint a mobiltelefonos játékoké. Az ilyen jellegű alkalmazásokon kívül az EMAIL kliensek, a különböző böngészők (mobiltelefonra készített weboldalak) azok, amelyek a leginkább használtak. Vállalati alkalmazásokat főleg PDA-kra fejlesztenek, ezek elérhetősége viszont nem annyira egyszerű és kézenfekvő, mint a mobiltelefonoké.

A vállalati alkalmazások mobiltelefonra való fejlesztési folyamatának a felgyorsításához járul hozzá a Mobile Inspector nevű prototípus is, amely nemcsak anyagi megtakarítási lehetőséget nyújt a biztosító cégeknek, de a biztosítási kárfelmérő ellenőrök munkájának hatékonyságát is növeli úgy időben, mint térben, előnyös funkcióinak köszönhetően.

Bibliográfia

- [1] Cristina Ciumas : “*Asigurari generale*”, Casa Cartii de Stiinta, Cluj-Napoca, Romania, 2009
- [2] Eliotte Rusty Harold: “*Java Network Programming, 3rd Edition*”, O'Reilly, USA, 2004
- [3] Jerome (J. F.) DiMarzio: “*Android, A Programmers Guide*”, Mc Graw Hill, USA, 2008
- [4] msg systems ag, FS-PM: <http://msg.de/insurance.0.html>
- [5] SOA: <http://www.oracle.com/global/hu/middleware/soa.html>
- [6] Thorsten Franz, Tobias Trapp: “*Anwendungsentwicklung mit ABAP Objects*”, Galileo Press, Bonn, Deutschland, 2006
- [7] <http://www.javaforum.hu/javaforum/8/android>