Semestrálne zadanie SWI [32b]

Cieľom zadania je vypracovať dokument, ktorý bude obsahovať špecifikáciu softvérového systému, ktorý vám bol pridelený. Samotný dokument má byť prehľadný, vhodne štruktúrovaný a formátovaný, s obsahom a prednou stranou. Dokumentu musí byť písaný odborným štýlom. Použité skratky vysvetlite, prípadne môžete zaviesť aj malý slovník použitých pojmov. Dokument sa odovzdáva do AIS do 12.12.2023 23:59 (vrátane) vo formáte PDF a tiež aj so zdrojom (napr. .doc, .docx, .odt .tex, …) z ktorého to PDF vzniklo. V dokumente použite normálne riadkovanie (1.0), Times font veľkosti 12pt a okraj 2,5cm (1″). Vymyslite aj svoj vlastný, marketingovo pôsobivý, komerčne úspešný a jedinečný názov pre vami vyvíjaný softvérový systém. Odovzdaný dokument obsahuje nasledovné časti:

1. Používateľská špecifikácia (cca. 2-3 strany textu) [body: 1b/0,5b]

Vžite sa do role majiteľa softvérovej firmy, ktorého zákazník požiadal o vytvorenie systému a vytvorte používateľskú špecifikáciu, ktorá bude slúžiť ako súčasť zmluvy.

1.1. *Stručný úvod do problematiky.* Tu treba popísať, čo sa v danej oblasti robí, aké sú tam pravidlá, ciele, postupy, aká je business logika (doménová logika) atď. Heslovite je táto informácia zhrnutá v zadaní, ktoré ste dostali, avšak treba ju rozvinúť a doplniť nespomenuté časti a súvislosti. Použite vlastnú inteligenciu, tvorivosť, externé zdroje a diskusiu s inými ľuďmi, aby ste zistili, ako funguje daná doménová oblasť.
1.2. *Používateľské požiadavky.* Definujte zákazníkove ciele a prepíšte ich na merateľné požiadavky. Identifikujte a popíšte funkcionálne, nefunkcionálne a doménové požiadavky (ak existujú).

2. Systémová špecifikácia

V diagramoch použite notáciu UML verzie 2.x

2.1. *Diagramy prípadov použitia.* Nakreslite diagram(y) prípadov použitia pre daný softvérový systém. Diagram (minimálne jeden, prípadne viacej ak sa to hodí), bude pomocou prípadov použitia obsahovať *celú hlavnú* funkcionalitu systému. Každý prípad použitia by mal, v rámci svojej realizácie, poskytovať svojmu hráčovi (alebo hráčom) niečo hodnotné, nejakú užitočnú funkcionalitu, nejaký pozorovateľný výsledok alebo zmenu. [body: 3b/1,5b]
2.2. *Use-case tabuľky*. K **trom** najzložitejším prípadom použitia vytvorte use-case tabuľku, ktorá bude obsahovať [2b]:
   - názov prípadu použitia
   - identifikátor - ako identifikátor môžete použiť svoje vlastné číslovanie, ktoré bude spájať jednotlivé prípady použitia z diagramu prípadov použitia.
   - opis prípadu použitia (stručný)
   - hráčov
   - vstupné podmienky
   - inicializácia
   - hlavnú postupnosť udalostí
   - alternatívnu postupnosť udalostí
   - výstupné podmienky

   VZOR: tutoriál č.2 – use-case tabuľka

2.3. *Diagram tried.* Vytvorte jeden detailný dátový model pre celý váš systém, ktorý bude zahŕňať *všetky* atribúty, vzťahy, násobnosti a *aspoň niektoré* metódy. Zobrazte ho ako jeden UML 2.x diagram tried vo vašej výslednej dokumentácii. Ak je systém príliš komplexný, môžete rozčleniť diagram na viacero menších diagramov, ktoré budú reprezentovať len príslušný podsystém. Vynechajte triedy obslužného jadra systému (tzv. „Controller“), ktoré nie sú relevantné pri modelovaní dátových entít. [body: 2b/5b]

2.4. *Diagramy aktivít a sekvenčné diagramy.* K vybraným **netriviálnym** prípadom použitia nakreslite diagramy graficky popisujúce tieto prípady použitia. Nakreslite 2 sekvenčné diagramy a 2 diagramy aktivít. Diagram aktivít môže zodpovedať aj viacerým prípadom použitia. Sekvenčné diagramy a diagramy aktivít nemusia zapovedať tomu istému scenáru. [4b+4b]

2.5. *Stavový diagram.* Nakreslite stavový diagram pre vami vyvíjaný systém. Môžete vytvoriť aj viacero menších stavových diagramov namiesto jedného veľkého. [3b]

3. Akceptačné testy [2b]

Vytvorte testy, na základe ktorých sa rozhodne o tom, či vytvorený systém spĺňa alebo nespĺňa požiadavky – teda či ho zákazník akceptuje alebo nie. Každý test by mal v tabuľke obsahovať minimálne tieto časti:

- identifikátor
- prípad použitia, ku ktorému test prislúcha
- cieľ testu (čo overujeme – nanajvýš stručne)
- vstupné podmienky
- výstupné podmienky
- jednotlivé kroky testu

Kroky testu reprezentujú sekvenciu testovania a ku každému kroku prislúcha a je v teste popísaná určitá akcia (podnet od aktéra) a určitá reakcia systému na tento podnet. Aby bol výsledný systém zákazníkom akceptovaný, musí splniť všetky testy. Keďže v tomto zadaní systém neprogramujeme, ale len navrhujeme, jednotlivé očakávané reakcie je potrebné si vymyslieť.

Do dokumentácie doplňte aspoň 4 akceptačné testy:

- **tri**, ktoré súvisia s funkcionálnymi požiadavkami a
- **jeden**, ktorý overuje nefunkcionálne požiadavky.

PRÍKLAD: AkceptacneTestyPriklad.pdf

4. Projektové plánovanie [3b]

Vytvorte plán tvorby (realizácie) vášho systému.

1. Rozdeľte prácu na aspoň 10 úloh a rozdeľte úlohy pre aspoň 4 ľudí tvoriacich váš tím. Počet si zvoľte podľa náročnosti témy, ale minimálne musí mať váš tím aspoň 4 členov.
2. Odhadnite časovú náročnosť úloh, naplánujte postupnosť úloh do kalendára.
3. V dokumente v kapitole 4.1 zobrazte Ganttov graf aj s tabuľkou závislostí a postupnosti vykonávania úloh, s míľnikmi a s WBS (work breakdown schedule).
4. V dokumente v kapitole 4.2. zobrazte sieťový graf pre postupnosti vykonávania úloh.

Na túto úlohu použite vami zvolený systém na projektový manažment (či už offline, lokálny program alebo ľubovoľný/dostupný online produkt). Zoznam je napr. na:

http://en.wikipedia.org/wiki/Comparison_of_project-management_software). Úlohou je oboznámiť sa so systémom na projektový manažment.

Odporúčame: Microsoft Project, Project Libre alebo google: alternatives to ms project

Zaznamenávanie zmien [1b]

Na začiatku dokumentu, ešte pred obsahom, v tabuľke zaznamenajte dátumy vykonania zmien a stručne popíšte, aké zmeny boli vykonané, tak **aby bolo jasné, kedy nastala zmena v dokumente a aká zmena to bola**.

PRÍKLAD: document_change_control_example

(napr. tabuľka na strane 4) alebo google: document change control