

Испит траје 2.5 сата. Није дозвољена употреба литературе.

1. [6] Набројати и са по једном реченицом објаснити догађаје и активности Спринта.
2. [6] Нацртати пример дијаграма секвенце који се остварује код узорка *Identity Map* и укратко објаснити.
3. [30] Нека је дат веб систем произвођача аутомобила Фолксваген (*Volkswagen*). Систем обухвата подсистем фабрика, подсистем за мрежу свих овлашћених продаваца-дилера и сервисера и подсистем за клијентски портал.

Сваки аутомобил произведен у фабрици спада у неку од категорија: хечбек, кабрио, купе, караван, цип, комби. У свакој категорији постоје различити модели аутомобила (нпр. у класи хечбек: *Polo, Golf, Jetta,...*). Када се аутомобил произведе у фабрици, он се евидентира у њиховом систему, где се бележе сва произведена возила. Произведени аутомобил има своју боју (из палете боја те фабрике), годину производње, серијски број мотора, број шасије и опционе податке о додатној опреми. Када се произведени аутомобил преда продавцу-дилеру или клијенту (особи која купује ауто), то све треба да буде такође евидентирано у овом систему. Сваки резервни део који се произведе у фабрици, такође се евидентира у њиховом систему, где се бележе подаци о том делу: назив, каталогски број, подаци за који модел/серију аутомобила се тај резервни део везује и количина таквог произведеног резервног дела.

Фабрика увек има увид у све своје продавце и сервисере, али и особе које су купиле аутомобил произвођача Фолксваген (клијенте). Компаније продавци-дилери и сервисери налазе се у целом свету и они могу да претражују стање нових аутомобила (произведених у фабрици) и стање резервних делова. Све компаније које продају и сервисирају аутомобиле морају бити регистроване у систему. Када нешто поруче, дилери/сервисери то раде преко система. Када продају некој особи одређени део то се бележи у систему, као и подаци о тој особи. Аутомобил дакле може бити на стању у фабрици, на стању код неког продавца или продат неком кориснику. Свака фабрика покрива један део мреже продаваца и сервисера (на пример: фабрика у Чешкој за регион југоисточне Европе). Када продавац-дилер наручује неки ауто за одређеног клијента, он отпочиње комуникацију са фабриком (која је надређена том продавцу-дилеру).

Клијенти преко свог портала у сваком тренутку могу видети електронску сервисну књижицу за свој аутомобил, да погледају редовне сервисе које су посетили, када је шта мењано од делова и да виде када треба следећи сервис да ураде. Такође, када се нешто поквари клијенти могу да пријаве преко посебне форме једном од званичних сервиса у свом граду (или неком другом у својој земљи, уколико у граду из кога је клијент нема сервиса). Након пријаве квара, клијент и овлашћени сервисер наизменично размењују поруке у оквиру комуникације, да би утврдили проблем. За сваког клијента се евидентира име, презиме, број телефона и адреса.

а) [8] Нацртати ИЕ модел релационе базе података, са свим ентитетима и везама, које су потребне за реализацију ове апликације коју пројектујете, према описаној спецификацији. Написати како би реализовали у бази податке за додатну опрему сваког аутомобила, ако постоји 30 категорија додатне опреме (размислити и о перформансама претраживања базе). Дати пример како би изгледали записи података у *NoSQL* нерелационој бази података за овај исти систем.

б) [8] Потребно је осмислити архитектуру овог система, реализованог у *cloud* платформи *Azure*, ако у свету постоји 15 фабрика, мрежа од неколико хиљада продаваца и сервисера и мрежа од неколико стотина хиљада клијената. Систем треба да подржи више хиљада конкурентних корисника у једном тренутку. Постоје четири сценарија комуникације: фабрика-дилер, фабрика-сервис, дилер-клијент и сервис-клијент. Они наизменично размењују поруке, док се комуникација не заврши, или успешном реализацијом посла или одустајањем једног члана комуникације. Води се статистика колико је успешно наручено аутомобила/резервних делова из које фабрике и у којој земљи је продато колико

аутомобила, тако да произвођач може да прати колико процентуално има тржиште у свету. Нацртати дијаграм и описати коментарима све *Web/Worker* улоге и све податке које треба разменити/памтити у описаним сценаријима.

в) [8] Написати PHP функције које се користе у процесу куповине аутомобила: претраживање одређеног аутомобила путем форме од стране продавца-дилера на основу неколико основних података (категорија, модел, боја, кубикажа мотора, врста горива), провера да ли постоји такав аутомобил код тог продавца на стању или на стању у његовој надређеној фабрици за тај регион и након одабира аутомобила, попуњавање података о клијенту коме је тај аутомобил продат. За случај постојећег клијента потребно га је само евидентирати, а за случај новог клијента, потребно је извршити његову регистрацију. Дозвољено је користити све табеле које сте дефинисали у моделу базе података, под тачком а).

г) [6] Користећи *PHP Unit*, написати јединични тест (и позитивни и негативни сценарио) који тестира функцију за проверу да ли неког аутомобила са одређеним карактеристикама има на стању.

4. [12] Нацртати **WAE дијаграм** који приказује серверске странице, клијентске странице и HTML форме и **скицу пототипа** за део апликације који остварује претраживање и куповину аутомобила. Сматрати да су претрага аутомобила, провера постојања аутомобила у надрђеној фабрици и попуњавање података о клијенту реализовани кроз засебне странице. На првој страници се уносе подаци по којима се претражују аутомобили, на другој се приказује листа свих пронађених аутомобила, уз линк која води ка страници за куповину, на којој се приказују подаци о изабраном аутомобилу (боја, серијски број мотора и шасије и др.), и приказује два линка: први води на страницу за куповину за постојећег клијента (приказује се листа постојећих, из које се бира један), а други води на страницу на којој се реализује регистрација клијента и одмах затим куповина аутомобила. Обратити пажњу на правилан приказ података који се прослеђују серверским страницама и користећи WAE стереотипе јасно направити разлику између клијентских страница, серверских страница и форми.

5. [6] Тим сачињен од непознатог броја студената Одсека за софтверско инжењерство развија део гореописаног софтвера за компанију Фолксваген. Процењено је да део који се развија има 3500 линија програмског кода. Будући да апликација коју производе ради са базом података и комуницира са другим постојећим подсистемима у оквиру инфраструктуре компаније, за моделовање комплексности система се користи двојни тип. Пошто су програмери неискусни у раду са *Code Igniter*-ом, узето је да је рејтинг (енг. *driver*) за искуство програмера (*PCAP*) 1.17, а, како је систем веома комплексан, рејтинг комплексности (*CPLX*) износи 1.30, док су остали трошкови на номиналној вредности.

Израчунати, користећи методологију COCOMO:

- | | |
|---|--|
| а) Уложени напор (за константе a , b у формули усвојити 3.0 и 1.12) | д) Продуктивност |
| б) Време развоја софтвера ($s = 0.35$) | ђ) Време потребно за реализацију пројекта, уколико на пројекту ради 4 студента |
| в) Процену средњег броја ангажованих студената | |