

Испит траје 2.5 сата. Није дозвољена употреба литературе.

1. [10] а) Представити шематски изглед *Azure Table storage* у који су смештени подаци о четири студента (име, презиме, бр.индекса, статус финансирања) при чему су два самофинансирајућа, а два буџетска и постоје два складишна чвора један за буџетска, други за самофинансирајуће. Морају се навести конкретне вредности свих релевантних података, који се односе на студенте и које захтева сам *Azure*.
б) Објаснити шта се дешава у случају да *Worker* из *Azure Queue-a* извуче поруку и потом дође до отказа тог *Workera*. Шта се дешава у супротном случају (ако није дошло ни до каквог отказа).
2. [10] а) Специфицирати доменску класу Корисник (име, адреса, активан/неактиван) и одговарајућу класу у *DataMapper* узорку (за класе навести потписе свих јавних метода, имплементација није потребна, само коментар шта метод ради).
б) Приказати секвенцу кода који користи класе дефинисане у тачки а) када се из базе извлаче подаци о свим активним корисницима.
3. [40] Нека је дат веб систем за куповину лоу кост летова. Корисници система су путници, авио компаније, и администратор који одржава систем. На почетној страни сви корисници треба да имају могућност уноса корисничког имена (који је јединствен у систему) и лозинке, и у случају исправно унетих података, треба им омогућити рад са остатком система. Уколико се не унесу неки од података, или уколико се унесу погрешни подаци, треба исписати одговарајуће поруке са грешкама. Приликом логовања, треба проверити да ли је корисник активан (треба да постоји као посебно поље у табели са корисницима).

Путник када се улогује на систем, може видети своје основне податке (име, презиме, адреса е-поште, телефон, датум рођења, тренутно новца на рачуну), листу својих промотивних летова које је већ купио (приказати: назив дестинације, датум/е путовања - један у случају једног смера и два датума у случају повратног лета, број путника и цену по карти), и може претраживати летове преко форме за претрагу. Форма треба да садржи: место од, место до, повратно/једносмерно путовање, датум одласка, датум повратка - ако постоји, тип карте и број путника од 12+ година, испод 12 година и испод 2 године. Тип карте може бити: економска, бизнис или премијум (изабрати из листе која подржава избор више опција). Премијум летови су са 20% попушта и могу их реализовати само путници који су имали најмање 10 летова у протеклих годину у тој компанији која је авиопревозник жељеног лета. Резултат извршавања овог претраживања су сви летови који су у понуди на дан претраге. Путник може резултата да сортира растуће/оппадајуће према цени. За свако путовање, путник може да купи највише 4 карте, уколико има довољно новца на рачуну. За сваку карту, корисник мора да унесе име, презиме, пол и годину

рођења особе која путује. Карта се не може купити мање од 24 сата пре лета, па такве резултате треба избацити из претраге.

Компанија када се улогује на систем, може видети своје основне податке (назив компаније, адреса, телефон, ПИБ, веб сајт, адреса е-поште) и може да унесе нови промотивни лет (дестинација од/до, период летења - односно датум у сезони када постоји први/последњи лет у години, дан у недељи и време када се лет реализује, број карти за сваки лет и цене за сваки тип карте појединачно). Накнадно компанија може да за тачно одређени лет појефтини или поскупи карту. Компанија може да врши и преглед за сваки свој појединачни лет колико има непродатих и продатих карти и колики је приход од њих. Авио компанија може и да откаже лет, када треба свим путницима који су купили карте за тај лет вратити новац.

Администратор је задужен за додавање нових путника и додавање нових компанија у систем. Администратор може да измени и новчани салдо путника, када му из банке јаве да на рачуну има неки нови новац уплаћен.

а) [8] Нацртати *IE* модел релационе базе података, са свим ентитетима и везама, које су потребне за реализацију ове апликације коју пројектујете, према описаној спецификацији.

б) [8] Написати *ССУ* (сценарио случаја употребе) за *Претраживање и куповину авио карте*. Разматрати успешан сценарио и могуће алтернативне токове. Затим скицирати прототип те функционалности које има корисник, методом *storyboard* (илустрација сценарија).

в) [8] Нацртати *УМЛ* дијаграм секвенце за функционалност *Унос новог лета*, од стране авио компаније.

г) [8] Написати *PHP* функцију која се користи у случају отказивања лета од стране авио компаније, при чему свим путницима са купљеним картама треба вратити новац од тих авио карти и обавештава све путнике преко е-поште да је лет отказан. Сматрати да у систему постоји класа која извршава слање поруке електронском поштом са задатом адресом примаоца, насловом (стринг) и телом поруке (стринг): `Mailer::send(recipientEmail, subject, body)`. Дозвољено је користити све табеле које сте дефинисали у моделу базе података, под тачком а). Објаснити сваки податак који се користи као аргумент функције или написати који се подаци прослеђују из форме за отказивање лета и на који начин.

д) [8] Нацртати *WAE* дијаграм који приказује серверске странице, клијентске странице и *HTML* форме дела апликације који остварује путник приликом претраживања летова и куповине авио карте. На првој страници се уносе подаци по којима се претражују летови, на другој се приказује листа свих пронађених летова који задовољавају претрагу, уз линк

која води ка страници за куповину карата, на којој се приказују подаци о изабраном лету, и приказује више линкова: први линк води на страницу за куповину економске авио карте (уколико је путник чекирао у листи), други линк води на страницу за куповину бизнис авио карте (уколико је путник чекирао у листи), а трећи линк води на страницу за куповину премијум авио карте са попустом (уколико је путник чекирао у листи и уколико задовољава услов за куповину премијум карте).

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.