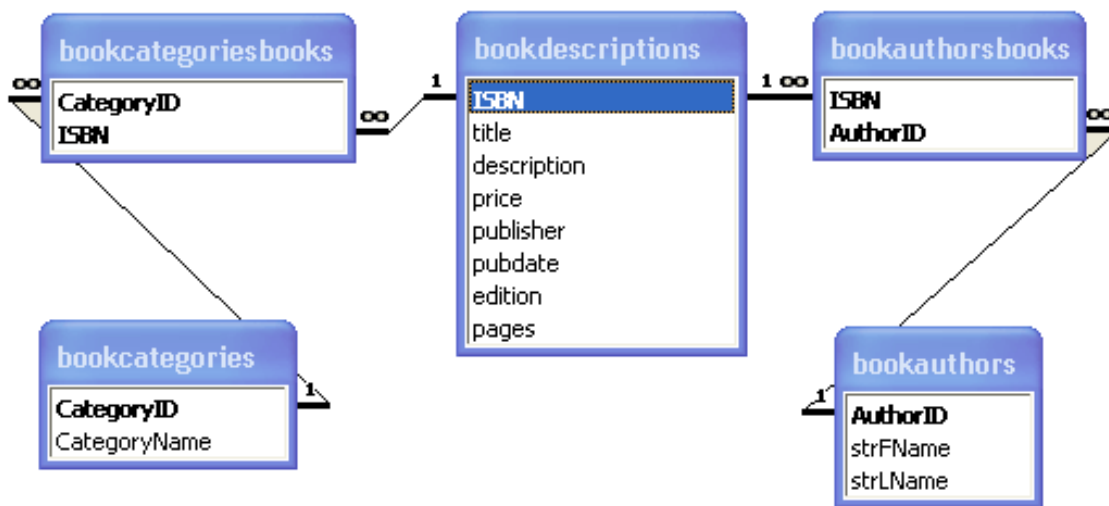


Испит траје 2.5 сата. Није дозвољена употреба литературе.

1. [10] Дата је следећа релациона шема. Потребно је пронаћи описе свих књига које припадају некој од категорија из задатог скупа категорија. Претпоставити да су подаци у бази непроменљиви (не мора се водити рачуна о ефектима промене података од стране другог корисника).



- а) Како би изгледао PHP код, ако је апликација модела 1 и директно се користи PDO за приступ бази.
- б) Како би исто решење изгледало са *Data Mapper* узорком за податке (приказати све релевантне класе, могуће је приказати празна тела метода, али са тачним потписима (параметрима)).

2. [5] Описати дисциплину тестирања у RUP-у (сврха, улоге, производи рада - артефакти).

3. [5] Објаснити чему служи *relay* функција *Service Bus*-а у *Azure* облаку.

4. [40] Нека је дат веб систем за продају, куповину и издавање некретнина. Могуће је претраживати огласе по различитим критеријумима. Резултати претраге садрже све релевантне податке о огласима и оглашивачима. Постоје две врсте корисника: администратор и регистровани корисници. 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).

Након успешног пријављивања корисника на систем, треба омогућити кориснику да дода нови оглас, прегледа огласе, ажурира свој оглас и обрише свој оглас. Код додавања огласа, за сваку некретнину уноси се: тип огласа (продаја, куповина, издавање), локација (град, земља, улица,...), врста некретнине (кућа, стан, пословни простор, викендица,...), површина, спратност објекта, спрат на коме се налази (опционо), број соба, година

изградње, датум постављања огласа, цена, напомена и датум до када је оглас активан. Када ажурира оглас, корисник може да измени само цену (тако што може само да је смањи), напомену и датум до када је оглас активан. Приликом брисања огласа, корисник добија списак својих огласа, а администратор све огласе. За сваки оглас се приказују: врста некретнине, локација, површина, датум постављања огласа и цена. Корисник бира један од огласа који жели да обрише, при чему регистровани корисник може обрисати само своје огласе, а администратор било који. Кориснику треба омогућити и претрагу огласа по типу (понуда, потражња, издавање), претрагу некретнина по локацији (на пример: Београд), претрагу по врсти некретнине (на пример приказати само викендице), претрагу по површини (у границама, на пример од 30м² - до 60м²) и претрагу по години изградње (у границама, на пример од 1989 - до 2004). Претпоставити да се као резултат претраге огласа, на свакој страници излистава по 10 огласа, а уколико има више огласа који задовољавају претрагу, јавља се пагинација по страницама (број страница одговара броју резултујућих огласа подељеним са 10). Када пронађе одговарајући оглас, корисник може да га одабере, а затим прелази на другу веб страницу, на којој може преко форме да упути понуду и додатни коментар продавцу некретнине. Идентификатор огласа прослеђује се овој веб страници као GET параметар.

Администратор је корисник са посебним привилегијама. Поред стандардних опција које има регистровани корисник (преглед огласа, додавање и брисање огласа), администратору треба омогућити унос новог корисника и брисање корисника. Уколико се корисник регистровао, користећи форму на споља видљивој веб страници, администратор треба такве кориснике да прогласи активним, или да одбије њихов захтев за регистрацијом. Такође, администратор треба да има могућност прегледа архиве огласа (огласа чији је датум до када је оглас активан истекао). Администратор има могућност додатних прегледа: 20 најјефтинијих некретнина, 20 најскупљих некретнина и 20 последње објављених огласа.

а) [6] Нацртати *IE* модел релационе базе података, са свим ентитетима и везама, које су потребне за реализацију ове апликације коју пројектујете, према описаној спецификацији.

б) [4] Нацртати дијаграм случајева коришћења (*use-case*) за све типове корисника, према описаној спецификацији.

в) [8] Написати ССУ (сценарио случаја употребе) за функционалност *Регистрација новог корисника*, од стране новог корисника који није био у систему. Неопходно је да нови корисник у форми за регистрацију попуни следећа поља: корисничко име, лозинка, потврда лозинке, име, презиме, адреса е-поште, контакт телефон. Уколико неко од ових поља не попуни. Разматрати успешан сценарио и могуће алтернативне токове. Затим скицирати прототип те функционалности, методом *storyboard* (илустрација сценарија), обухватајући и екране регистрованог корисника и администратора, који треба да потврди захтев за регистрацију и прогласи корисника активним.

г) [8] Написати *PHP* функције које се користи за претраживање огласа са некретнинама и за слање понуде, за случај да регистровани корисник жели да пошаље неку понуду продавцу некретнине. Треба имплементирати и обавештавање е-поштом: продавцу треба путем е-поште да стигне понуда (са именом и презименом потенцијалног купца, ИД огласа за који се интересује, понуђеном ценом и додатним коментаром), а купцу треба исто то послати (да би у мејл архиви знао коју је понуду послао). 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 претраживање некретнина.

д) [8] Нацртати WAE дијаграм који приказује серверске странице, клијентске странице и HTML форме ове веб апликације. Претраживање некретнина треба реализовати преко једне форме. Веб странице које су резултат извршања неке форме могу бити са прослеђеним GET или POST аргументима. 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.

е) [6] Написати јединичне тестове који тестирају описану имплементацију из тачке г) користећи *PHPUnit/CIUnit*.