

Испит траје 3 сата. Није дозвољена употреба литературе.

1. [6] а) Конструисати табелу одлучивања за следећу спецификацију:
Муштеријама се шаљу обавештења о кредитном дисконту. Садржај обавештења је везан за тренутни ниво дисконта и потенцијалне нивое дисконта. Садржај је различит за различите категорије муштерија. Категорије А, В и С добијају стандардно писмо осим муштерија категорије С, који добијају посебно писмо. Било која муштерија са 2 или више кредитних линија или са кредитним рејтингом Х добија посебан додатни параграф са понудом да се пријави на други ниво дисконта.
б) Конвертовати добијену табелу одлучивања у стабло одлучивања.
2. [4] Која од следећих тврђења су тачна за MVC узорак?
 - а) Објекат погледа директно комуницира (на пример, прави инстанцу или позива методе) са објектом модела.
 - б) Објекат модела директно комуницира са објектом погледа.
 - в) Објекат контролера директно комуницира и са објектом погледа и са објектом модела.
 - г) Ако објекту погледа требају подаци он прослеђује захтев објекту модела преко објекта контролера.
3. [4] Објаснити појмове отказних домена и домена надоградње код *Azure* платформе.
4. Дата је спецификација веб апликације за информације о пројектима и задацима у некој српској ИТ компанији. Апликација је урађена према MVC архитектури.

Корисници апликације су администратори, шефови и инжењери. За сваког корисника у систему се памте основни подаци: име, презиме, ЈМБГ, адреса, контакт телефон, адреса е-поште, број радне књижице. На почетној страни апликације, корисник треба да има могућност уношења корисничког имена и лозинке, и у случају исправно унетих података, омогућава рад са остатком система. Уколико корисник не унесе неки од података или унесе погрешне податке, потребно је исписати одговарајуће поруке: нисте унели корисничко име/лозинку, непостојеће корисничко име у систему, лозинка није у реду, корисник није активан у систему.

Компанија има већи број пројеката. Пројекат има свој идентификатор, назив, тип (предефинисани у посебној табели базе *TipProjekta*), датум почетка, очекивани датум краја и тренутни број ангажованих особа на пројекту. За сваки пројекат дефинише се већи број инжењерских тимова. На једном пројекту, инжењери припадају тачно једном инжењерском тиму, али могу имати више пројеката на којима учествују (МАКС=3). На врху хијерархије пројекта налази се главни шеф (координатор пројекта, надређени за све шефове тимова). Састави тимова, као и шефови који их предводе, могу се разликовати од пројекта до пројекта. Сваки пројекат може имати задатке, који морају да се реализују. Задаци се додељују на нивоу тима или појединца. Ако се доделе на нивоу тима од стране главног шефа, онда ти задаци морају да се поделе на подзадатке на нивоу појединаца, односно инжењера, чланова тог тима. Подзадатак је тимски задатак који се састоји од мањих задатака.

Сваки шеф када се улогује, може извршити: унос пројекта (којим руководи), унос тимова на пројекту, додељивање инжењера (из те компаније) одређеном тиму и постављање шефова у тим тимовима. Након тога он додељује задатке тимовима, а те задатке шефови тимова деле на подзадатке члановима својих тимова. Један подзадатак може бити додељен само једном, али и додељен више чланова тима. Сваки шеф може и да врши претраживање пројеката

(према називу/делу назива и временском интервалу: датум почетка - датум краја), којима руководи или на којима учествује, претражује тимове на таквим пројектима (према називу/делу назива) и чланове тима (према имену и презимену и позицији), да прегледа за сваког инжењера на том пројекту задатке који су му додељени, прегледа статус тих задатака и види статистички колико је процентуално задатака урађено на нивоу целог пројекта. Позиција у тиму може бити различита: архитекта система, програмер, тестер, дизајнер, итд. Шеф приликом додељивања инжењера тиму, додељује и позицију (која је предефинисана у посебној табели базе *PozicijaUTimu*).

Сваки задатак када се додели појединцу има статус: 1 - „додељен“ и додељује се тренутни датум и време када је задатак додељен. Преостали статуси су: 2 - „у току“ и 3 - „завршен“. Шеф тима може задатак доделити само особама које су већ распоређене у тимовима на том пројекту, односно којима је он надређен.

Инжењер (особа која није шеф), када се улогује у систем, види све своје пројекте, тимове којима припада на тим пројектима, као и задатке. За сваки свој задатак који је додељен, инжењер може да промени статус, и то са 1 на 2, и са 2 на 3, уз постављање опционог коментара о том задатку. Када се промени статус у 3, памти се и датум завршетка задатка.

Администратор система може да додаје нове кориснике у систем (нови инжењер у компанији) и да их деактивира у систему (ако напусте компанију). Приликом логовања обавезно проверавати да ли је корисник у систему активан или није. Администратор може да отвара и нове пројекте и да на тим пројектима додели главног шефа (координатора).

а) [10] Нацртати UML дијаграме случајева коришћења (*use-case*) за све типове корисника ове апликације. Нацртати IE модел базе података, са свим ентитетима и везама, које су потребне за реализацију ове апликације коју пројектујете, према описаној спецификацији.

б) [12] Нацртати WAE дијаграм који приказује серверске странице, клијентске странице и HTML форме, за описану веб апликацију.

в) [12] Написати сценарио случаја употребе формирање тима на пројекту и додавање инжењера у тим. Разматрати успешан сценарио и могуће алтернативне токове.

Нацртати дијаграм секвенце који одговара случају коришћења „Додавање задатка инжењеру“ од стране шефа, описаном у тачки а), ако систем аутоматски након тога шаље инжењеру потврду додељеног задатка, као и датум и време доделе задатка. 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)**. Узети у обзир ограничење да у једном тренутку инжењер не сме да има више од 5 додељених задатака на једном пројекту.

г) [8] Написати HTML *front-end* форму за логовање, а затим коришћењем *PHP* (или *CodeIgniter* фрејмворка) и MVC пројектног узорка имплементирати компоненте потребне за логовање корисника на систем. Потребно је обезбедити и успешно и неуспешно логовање, уз исписивање свих потребних порука. За случај када је лозинка погрешна, потребно је обезбедити чување већ унетог корисничког имена у форми за унос. Дозвољено је користити све табеле које сте дефинисали у моделу базе података, под тачком а).

д) [4] Написати јединичне тестове који тестирају описану имплементацију из тачке г) користећи *PHPUnit/CIUnit*.