

*Испит траје 3 сата. Није дозвољена употреба литературе.*

1. (Заједнички текст за питања 1, 2 и 3)  
[9] Разматра се информациони систем за подршку специјалистичке ординације (рад са пацијентима). Пацијенти са упутом долазе на пријемно одељење, где сестра проверава валидност упута (да ли је за тог специјалисту, да ли је датум валидан, важећа здравствена књижица). Ако је упут валидан, пацијент се упућује у чекаоницу. Пацијенте из чекаонице прима специјалиста према следећим приоритетима: хитни случајеви, они који су имали заказано за одређено време, остали пацијенти. После прегледа, специјалиста уноси дијагнозу и опционо пише рецепте. Ови подаци се прослеђују пријемном, где се печатира рецепт и опционо заказује пацијенту нови термин. Лекари и сестре на пријемном (има већи број лекара и већи број сестара) имају по рачунар са веб интерфејсом, а пацијенту се подаци штампају/приказују на инфо екрану/саопштавају.  
Напомена: ако уводите неке сопствене претпоставке ван онога што је експлицитно дефинисано у поставци, јасно напишите шта сте подразумевали.  
а) [5] Нацртати DFD-0 дијаграм описаног система.  
б) [4] Нацртати CRUD матрицу између ентитета система и случајева употребе који улазе у основни сценарио где пацијент долази на преглед, чека, лекар га прима, даје му дијагнозу и заказује нови преглед код сестре.
2. [5] Ако се користи шаблон *Активни запис* приказати имплементацију свега што улази у овај шаблон, а повезано је са функцијом система избора следећег пацијента који треба да буде примљен код лекара.
3. [6] а) Ако систем описан у задатку 1 треба да се имплементира као сервис на *Azure cloud*-у са асинхроним комуникацијом између улога, дефинисати потребне делове сервиса за имплементацију функције система из задатка 2.  
б) Детаљно описати ток догађаја у имплементацији из тачке а), у оквиру сценарија када пацијент долази, чека и на крају буде примљен код лекара.
4. [30] Пројектовати веб апликацију за замену дежурстава асистената на ЕТФ-у. На почетној страни апликације, корисник имају могућност уношења корисничког имена, лозинке и у падајућој листи да одабере тип (асистент или администратор). У случају исправно унетих података, кориснику омогућити рад са остатком система (за сваки тип корисника треба приказати посебну почетну страницу након логовања). Уколико асистент или администратор не унесу неки од података или унесу погрешне податке, потребно је исписати одговарајуће поруке црвеним словима са могућношћу исправљања грешке, уз задржавање коректно унетог корисничког имена.

Асистент, када се успешно улогује на систем, може видети своје личне податке и две листе својих дежурстава - она која су прошла (архива) и она која тек треба да ради (почев од сутрашњег дана). Он може своје дежурство да прогласи да је "за замену". Такође, асистент у посебној форми може да претражи дежурства других асистената, која су они прогласили за замену. Претрага се ради у једној форми и треба да обухвати: сва своја дежурстава за замену, сва дежурства других за замену, дежурства тачно одређеног датума и дежурства одређене особе (из падајуће листе која садржи асистенте, односно њихово име и презиме).

Резултат извршавања ове форме треба да буде листа дежурстава која задовољавају претрагу. Када асистент пронађе туђе дежурство (дежурство које други асистент хоће да мења), ако му одговара он може да га упари са једним својим дежурством, које је одабрао за замену, и тада он шаље понуду другом асистенту о замени. Замена је успешна тек када други асистент потврди ту замену, да и њему одговара. Листа дежурстава која се види када се асистент улогује на систем у сваком тренутку треба да буде ажурирана, односно уколико је неко дежурство замењено и успешно потврђено од стране другог асистента, поред тог дежурства треба да стоји: **ЗАМЕЊЕНО СА** + име и презиме асистента (са којим је извршена замена) + назив испита, датум и време тог другог испита. Та два дежурства онда више нису за замену, јер је извршена замена, и не треба их више приказивати приликом претраживања. Заменае су могуће само 1:1, односно дежурство за дежурство. Асистент може једно своје дежурство да иницијално понуди за замену код више других асистената, али када први прихвати, све остале понуде истог дежурства другим асистентима треба обрисати.

Администратор је особа са посебним привилегијама (нека од особа из Студентског одсека), која може да уноси предмете, термине колоквијума/испита за одређени предмет и да додаје дежурства асистената на тим колоквијумима/испитима. Колоквијум/испит се мора одржати у неком испитном року. Сваки колоквијум/испит може се одржати у више сала, а за сваку салу мора да постоји најмање 1, а највише 4 дежурна. Свака сала има такозвани подразумевани број дежурних (број дежурних оптималан да се дежурство у тој сали одржи). Такође, администратор има могућност уноса нових администратора и асистената (особа које дежурају). Администратора у систему може бити више.

а) [5] Нацртати *IE* модел релационе базе података, са свим ентитетима и везама, које су потребне за реализацију ове апликације коју пројектујете, према описаној спецификацији.

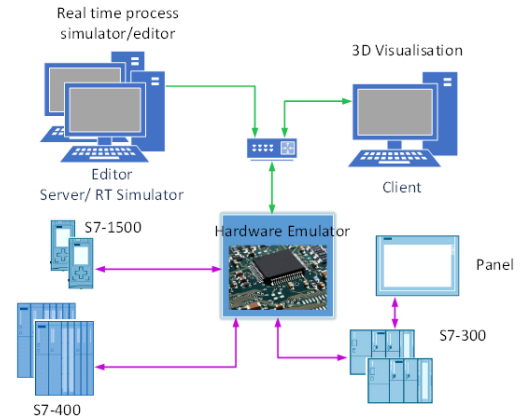
б) [7] Написати дијаграм активности за функционалност *Замена дежурстава*. Разматрати успешан сценарио (од тренутка логовања на систем до тренутка успешне замене дежурства између два асистента) и могуће алтернативне токове. Затим скицирати прототип те функционалности које има корисник, методом *storyboard* (илустрација сценарија).

в) [7] Нацртати *WAE* дијаграм који приказује серверске странице, клијентске странице и *HTML* форме ове апликације. 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.

г) [8] Написати *PHP* програмски код (или у *CodeIgniter*-у), писан у складу са пројектним узорком *MVC*, који се користи приликом поступка замене дежурстава, када се замена успешно изврши и када се у табелама базе података ажурирају све потребне вредности. 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 приказане у тачки б), код илустрације сценарија. Шта би се разликовало у овом коду и коду писаним у складу са узорком *MVVM*? Да ли би тај узорак могао успешно да се примени на овом примеру?

д) [3] Коришћењем документационих коментара према *PHPDoc* синтакси, дефинисати у програмском коду под тачком г) све варијабле, аргументе функција и излазне вредности функција, дефинисаних у оквиру класа за модел и контролер.

5. [10] Развија се софтверски систем за управљање неком фабриком. Прва фаза обухвата израду симулатора/едитора тих процеса у реалном времену, друга фаза израду 3Д виртуелног окружења, трећа фаза моделовање експертског система и израду хардверског емулятора, и четврта фаза тестирање и верификацију целог система. Све информације се чувају на централизованим серверима, а приступ самом систему се реализује преко панела (терминала). У пројекту се користе готове компоненте, означене на слици ознакама S7-X. Сав хардвер је доступан приликом реализације пројекта. Пре почетка развоја система, потребно је извршити анализу и моделовање типичних процеса у фабрици (нулта фаза), коју врше аналитичари.



Ви сте постављени за тим лидера целог пројекта и треба да направите пројектни план. Свака од прве три фазе треба да обухвати анализу корисничких захтева, дизајнирање подсистема, имплементацију и верификацију са тестирањем. По завршетку ових фаза, а пре тестирања целог система, потребно је извршити фазу интеграције свих компоненти у систем. Пре ове три главне фазе, а након нулте фазе, потребно је урадити дизајнирање целог система.

Имате тим од 6 људи, једног аналитичара корисничких захтева (који мора бити и интегратор система), једног дизајнера, 2 програмера (који могу бити и интегратори) и 2 тестера (који могу бити и дизајнери система, односно подсистема).

Анализа процеса у фабрици:	5 човек/дана
Анализа сваког подсистема:	3 човек/дана по подсистему
Дизајнирање система:	6 човек/дана
Дизајнирање подсистема:	3 човек/дана по подсистему
Имплементација подсистема:	5 човек/дана по подсистему
Интеграција система:	2 човек/дана по подсистему (сваки са сваким мора да се интегрише)
Тестирање подсистема:	4 човек/дана по подсистему
Тестирање целог система:	6 човек/дана

У поступку решавања користити:

- модел водопада
- инкрементални модел

Нацртати *GANTT* дијаграме за овај пројекат, за оба модела, и расподелу ресурса у пројектном тиму (на оба дијаграма), у складу са задатом спецификацијом система.