

✨ ВЕБ ПРОГРАМИРАЊЕ ТЕОРИЈА ВТОР КОЛОКВИУМ ✨

good luck good luck

Правила:

1. Секој има право да напише решение/одговор, побарајте **request edit access** доколку сакате да напишете нешто.
2. Доколку некој пишува глупости, става несоодветна содржина, брише и преуредува решенија на други или на било кој начин смета во документов, **ќе биде отстранет** (сечија и секаква активност е зачувана).
3. Решенијата се пишуваат **ПОД** секоја слика каде што е обележано **Решение/Објаснување 1,2,3.**
4. Доколку има некој што има различен одговор или објаснување, пишете во **Решение/Објаснување 2,3,4** итн., ве молам **НЕ пишувайте во решенијата на другите колеги.**
5. Дозволено е да прикачувате и слики од решенијата ако е потребно, само под Решение/Објаснување n, не на рандом место.
6. **Доколу нешто не ви е јасно, прашајте во серверот.** Ве молам не прашувајте прашања и не барајте објаснувања тука.
7. Во документов има сигурно задачи кои се ставени повеќе пати, обележете ги со „**веке решена задача**“, ве молам **не ги бришете.**
8. Ако имате некоја нова задача што ја нема тука, слободно додадете ја најдолу во документов и обележете ја со „**нова задача**“.
9. Доколку сликите не се гледаат јасно, **зумирајте** со опцијата за зумирање под tools.
10. Сликите обележани со „**Земено од првата база:**“ се земени од првиот документ, кои порано биле за прв колоквиум, а сега се за втор. Ако сакате таму да ги видите објаснувањата за овие задачи, може да ги пронајдете на следниов линк: [WP](#)

Помош:

Линк до учебникот од професорите: <http://194.149.135.92/wp-book/full-document.pdf>

Линк до други книги поставени на courses:

O'Reilly - Head First - Servlets and JSP

Craig Walls - Spring in Action, 5th Edition (2018, Manning Publications)

Spring in Action, 3rd Edition

Pro Spring Security

Базата на мацо, објаснети некои задачи од прв и втор колоквиум:

<https://discord.com/channels/810997107376914444/8883993a93986850846/1177687287660806284>

Тип задачи од претходни години:

1. authentication provider i so e tocno za nego, 4 ponudeni
2. za testiranje cekorite da gi podredes, imase 6 ponudeni
3. sto e tocno za spring mvc
4. imase dve klasi definirano student i course i relacija many-to-many, i koja tabela ke se kreira od tia klasi
5. imase koji metod ke go vmetnes u programata za da prebaruvas po opis na kniga i daeno implementacija na nekoja klasa za kniga
6. imase daeno eden metod 10 linii kod nekade i taa funkcija deka treba da se definira u persistence, service, i uste nekoj odgovor
7. entity life cycle da gi napisas metodите kaj sekoga strelka
8. kako ke zemes sesiskio atribut "user" i ima daeno 4 metodi

Земено од првата база:

n 15
ed
out of
question

Нека во веб апликација е поставена следната конфигурација на view-resolver:

```
<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/views/">
    <property name="suffix" value=".html">
</bean>
```

Која ќе биде датотеката која ќе се избере од ViewResolver-от за приказ (view template) при повик на question() методот дефиниран во продолжение?

```
@Controller
@RequestMapping("/exam")
public class ExamController {

    @RequestMapping(value = "/question", method = RequestMethod.POST)
    public String question() {
        return "view";
    }
}
```

Select one:

- a. src/main/resources/views/question.html
- b. Нема да се селектира ниту една датотека, ќе се изврши редирекција
- c. src/main/resources/views/view.html
- d. src/main/resources/WEB-INF/views/view.html
- e. src/main/resources/WEB-INF/views/question.html
- f. Нема да се селектира ниту една датотека, ќе се генерира JSON

Give your reasons

A A B B I I ≡ ≡ % % S S

Решение/Објаснување 1:

ViewResolver мапира името на еден view во соодветената локација до тој view. Во овој случај го прави тоа така што на името на view-то става префикс /WEB-INF/views/ и суфикс .html.

question() методот враќа string (име на view) "view". На тоа се додава префиксот и суфиксот и се добива /WEB-INF/views/view.html.

Решение/Објаснување 2:

Решение/Објаснување 3:

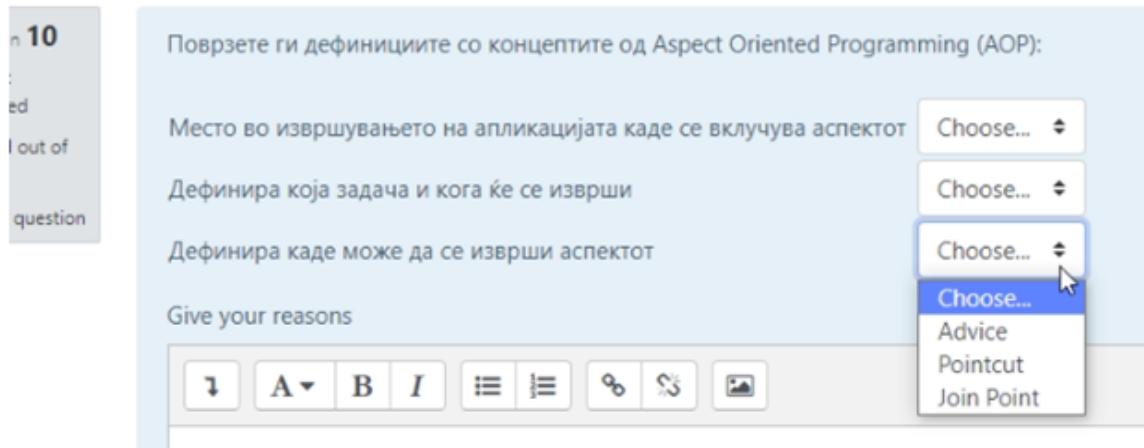
Земено од првата база:

10
ed
I out of
question

Поврзете ги дефинициите со концептите од Aspect Oriented Programming (AOP):

Место во извршувањето на апликацијата каде се вклучува аспектот Choose...
Дефинира која задача и кога ќе се изврши Choose...
Дефинира каде може да се изврши аспектот Choose...
Give your reasons

Choose...
Advice
Pointcut
Join Point



Решение/Објаснување 1: Join Point, Advice, Pointcut

Решение/Објаснување 2:

Решение/Објаснување 3:

Земено од првата база:

n 12
nd
out of
question

REST контролерот ConsultationSlotApi е мапиран на следната локација: /api/consultations. Апликацијата е стартувана на порта 8080. Кои од прикажаните методи успешно ќе го обработат PATCH барањето во следната форма:

```
PATCH http://localhost:8080/api/consultations/{slotId}
professorId: petko.petkov
Content-Type: application/x-www-form-urlencoded

roomName=117&dayOfWeek=2&from=18:00:00&to=20:00:00
```

Како опционален параметар во барањето, може да се прати и date којшто треба да се десеријализира во објект од типот LocalDate.

Select one or more:

a.

```
@RequestMapping(value = "{pathId}", method = RequestMethod.PATCH)
public ConsultationSlot updateSlot(@PathVariable int slotId,
    @RequestHeader String professorId,
    @RequestParam String roomName,
    @RequestParam(value = "dayOfWeek") Integer dayOfWeek,
    @RequestParam(value = "date", required = false) @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
    @RequestParam("from") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
    @RequestParam("to") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
    return consultationSlotService.updateSlot(slotId, professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
}
```

b.

```
@PatchMapping("/{slotId}")
public ConsultationSlot updateSlot(@PathVariable int slotId,
    @RequestHeader String professorId,
    @RequestParam String roomName,
    @RequestParam(value = "dayOfWeek") Integer dayOfWeek,
    @RequestParam(value = "date") @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
    @RequestParam("from") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
    @RequestParam("to") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
    return consultationSlotService.updateSlot(slotId, professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
}
```

c.

```
@PatchMapping("/{slotId}")
public ConsultationSlot updateSlot(@PathVariable int slotId,
    @RequestHeader String professorId,
    @RequestParam String roomName,
    @RequestParam(value = "dayOfWeek", required = false) Integer dayOfWeek,
    @RequestParam(value = "date", required = false) @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
    @RequestParam("from") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
    @RequestParam("to") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
    return consultationSlotService.updateSlot(slotId, professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
}
```

d.

```
@PatchMapping("/{slotId}")
public ConsultationSlot updateSlot(@RequestParam int slotId,
    @RequestHeader String professorId,
    @RequestParam String roomName,
    @RequestParam(value = "dayOfWeek", required = true) Integer dayOfWeek,
    @RequestParam(value = "date", required = false) @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
    @RequestParam("from") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
    @RequestParam("to") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
    return consultationSlotService.updateSlot(slotId, professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
}
```

Give your reasons

Решение/Објаснување 1: Патеката што се бара е /api/consultations/{slotId}.

/api/consultations е мапирањето на контролерот, што значи ни треба метод мапиран на /{slotId} (испаѓа a.)

Соодветно мора да го земеме како параметар slotID кој е дел од патеката (испаѓа d.)

Во задачата пишува дека date е опционален параметар, а и истиот не е во

PATCH (испаѓа b.)

Останува точно да е под **c**.

Решение/Објаснување 2:

Решение/Објаснување 3:

Земено од првата база:

13

d
out of
question

Доколку имаме потреба кон клиентот да вратиме Header-от "user" во рамките на мапиран метод во Spring MVC, во кој од следните методи е можно да го поставиме?

Select one or more:

a.

```
@Controller  
@RequestMapping("/exam")  
public class ExamController {  
  
    @RequestMapping("question")  
    public String question(HttpServletRequest session, ServletContext context) {  
        ...  
    }  
}
```

b.

```
@Controller  
@RequestMapping("/exam")  
public class ExamController {  
  
    @RequestMapping("question")  
    public String question(@Header String session) {  
        ...  
    }  
}
```

c.

```
@Controller  
@RequestMapping("/exam")  
public class ExamController {  
  
    @RequestMapping("question")  
    public String question(HttpServletRequest request) {  
        ...  
    }  
}
```

 d.

```
@Controller  
@RequestMapping("/exam")  
public class ExamController {  
  
    @RequestMapping("question")  
    public String question(@RequestHeader(name="user") String user) {  
        ...  
    }  
}
```

e.

```
@Controller  
@RequestMapping("/exam")  
public class ExamController {  
  
    @RequestMapping("question")  
    public String question(HttpServletRequest context) {  
        ...  
    }  
}
```

Give your reasons



Решение/Објаснување 1:

- a. Не може да се пристапи до header
- b. Не постои @Header анотација
- c. Може да се пристапи со request.getHeader("user")
- d. правилно пристапува до header параметар со @RequestHeader
- e. Response се користи за работа со одговорот, не за добивање вредности од барањето.

Решение/Објаснување 2:

Решение/Објаснување 3:

Земено од првата база:

14
ed
out of
question

Доколку сакаме барањето кон нашиот сервис да биде
GET http://localhost:8080/exam/answer/12?answer=xyz
и истото треба да го имплементираме во контролерот:

```
@Controller("exam")
@RequestMapping("exam")
public class ExamController {
```

}

Кое од следните е валидна дефиниција на метод кој ќе се справи со ова барање?

Select one:

a.

```
@RequestMapping(value = "/exam/answer/{id}?answer={answer}", method = RequestMethod.GET)
public String question(@RequestParam Long id, @RequestParam String answer) {
```

b.

```
@RequestMapping(value = "/answer/{id}?answer={answer}", method = RequestMethod.GET, produces = "application/json")
public String question(@RequestParam Long id, @RequestParam String answer) {
```

c.

```
@RequestMapping(value = "/answer/{id}?{answer}", method = RequestMethod.GET, produces = "application/json")
public String question(@PathVariable Long id, @PathVariable String answer) {
```

d.

```
@RequestMapping(value = "/exam/answer/{id}", method = RequestMethod.GET, produces = "application/json")
public String question(@PathVariable Long id, @RequestParam String answer) {
```

e.

```
@RequestMapping(value = "/answer/{id}", method = RequestMethod.GET, produces = "application/json")
public String question(@PathVariable Long id, @RequestParam String answer) {
```

Give your reasons

Решение/Објаснување 1: Патеката што се бара е /exam/answer/{id} со еден query параметар answer.

/exam е мапирањето на контролерот, што значи ни треба метод мапиран на answer/{id}.

Query параметарот го земаме со @RequestParam String answer.

Точно е под e.

Решение/Објаснување 2:

Решение/Објаснување 3:

Земено од првата база:

Дадена веб апликација е стартувана на `localhost:8080`, контролер класата во која се наоѓа свој метод е анотиран со `@RequestMapping("/api/events")`

```
Time left 0:10:45
@RequestMapping(method = RequestMethod.POST)
public Event createEvent(@RequestHeader String creatorId,
    @RequestParam String location,
    @RequestParam(value="day_of_week", required = false) Integer dayOfWeek,
    @RequestParam(required = false) @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
    @RequestParam("start") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
    @RequestParam @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
    Event result = service.createEvent(creatorId, location, DayOfWeek.of(dayOfWeek), date, from, to);
    return result;
}
```

Пополнете го http барањето со кое ќе се повика методот од погоре:

POST	/api/events	HTTP/1.1					
creatorId	: 345						
ContentType	application/x-www-form-urlencoded						
location	=lab	day_of_week	=2	start	=18:00:00	to	=20:00:00

Како резултат од ова барање ќе се прикаже страницата `createEvent.html` и во моделот ќе се постави својство со име `createEvent` и вредност `result`.

GET		PUT	DELETE	
localhost/api/events	http://localhost:8080	http://localhost:8080/api/events	localhost:8080/api/events	
creator_id		roomName	dayOfWeek	
date	event_date	from		
end		multipart/form-data	text/html	text/css
		content-type	contentType	ContentLength
content-length	contentLength		event	Event

Activate Windows
www.microsoft.com/activate-windows

Решение/Објаснување 1: Се согласувам со решенијата освен со последната реченица. Како прво враќаме објект и не знаеме која страна моментално е прикажана. Потоа, пишува “во моделот ќе се постави својство...” но никаде немаме овде објект “Model”.

Решение/Објаснување 2:

Решение/Објаснување 3:

Земено од првата база:

Question 1
Not yet answered
Marked out of 13.00
Flag question

Клиент пополнува форма за најава и клика на копчето Login. Формата се наоѓа на страницата wp.mk/login-page.html а барањето е сервисот wp.mk/login. По успешната најава, серверот го пренасочува клиентот на страницата wp.mk/index.html.
Пополнете ги полината за барањето кое ги генерира прелистувачот и одговорот кој пристапува од серверот. Клиентот веќе пристапил до сајтот неколку пати.

Request	Response	
	HTTP1.1 HTTP1.1	
Host:		
ContentType:		
ContentLength:		
Referer:		
Cookie:		
Body:		
<input type="button" value="GET"/>	<input type="button" value="POST"/>	<input type="button" value="PUT"/>
<input type="button" value="DELETE"/>	wp.mk/index.html	wp.mk/login
wp.mk	index.html	/login
wp.mk/login-page.html	200 OK	302 Redirect
401 Forbidden	404 Not found	500 Server Error
application/json	application/x-www-form-urlencoded	application/pdf

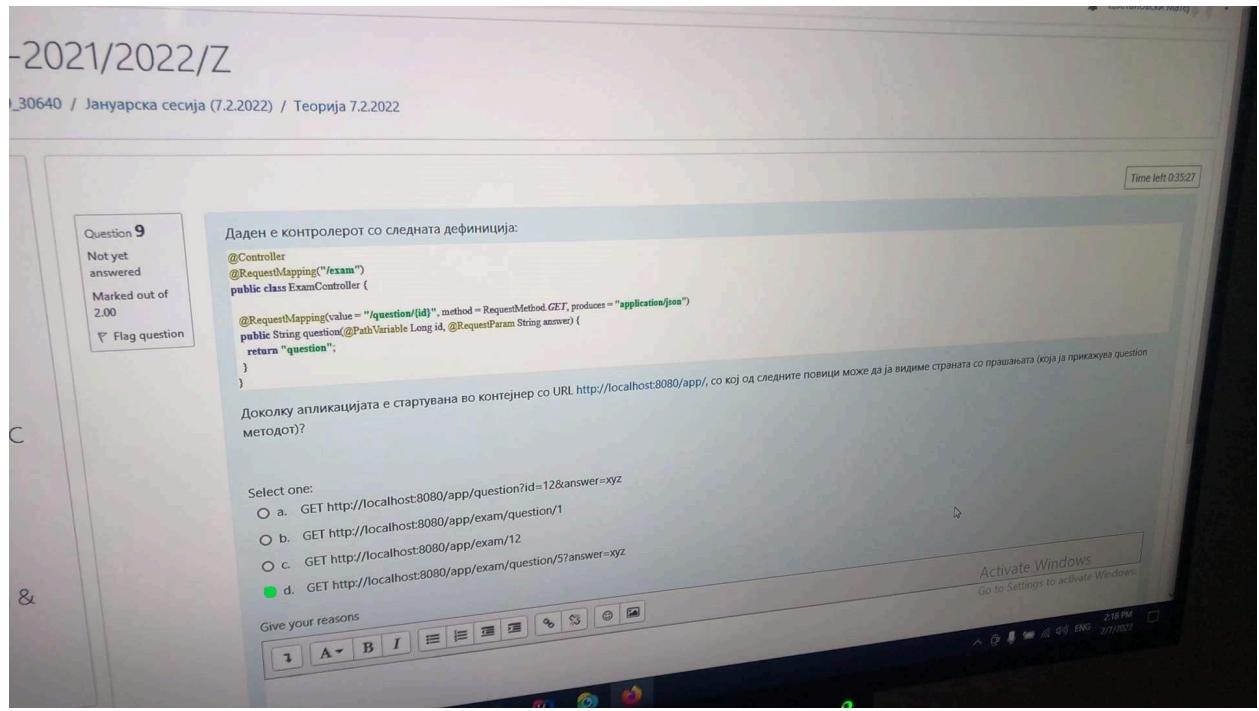
Прв колоквиум??? - Да, од прв е

Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

Земено од првата база:



Решение/Објаснување 1: Апликацијата е стартирана во контејнер со URL

<https://localhost:8080/app/>.

Методот се наоѓа во контролер чије мапирање е /exam.

Методот има мапирање /question/{id}. (/question/12)

Методот бара и query параметар answer. (?answer=xyz)

Се кога ќе се спои се добива

<https://localhost:8080/app/exam/question/12?answer=xyz> (под d.)

Решение/Објаснување 2:

Решение/Објаснување 3:

Земено од првата база:

Question 8
Not yet answered
Marked out of 2.00
Flag question

Доколку имаме потреба да пристапиме до Header-от "user" во рамките на мапиран метод во Spring MVC, во кој од следните методи е можно да го пристапиме?

Select one or more:

a.

```
@Controller  
@RequestMapping("/exam")  
public class ExamController {  
  
    @RequestMapping("question")  
    public String question(HttpServletRequest request) {  
        ...  
    }  
}
```

b.

```
@Controller  
@RequestMapping("/exam")  
public class ExamController {  
  
    @RequestMapping("question")  
    public String question(HttpServletRequest context) {  
        ...  
    }  
}
```

c.

```
@Controller  
@RequestMapping("/exam")  
public class ExamController {  
  
    @RequestMapping("question")  
    public String question(@Header String session) {  
        ...  
    }  
}
```

d.

```
@Controller  
@RequestMapping("/exam")  
public class ExamController {  
  
    @RequestMapping("question")  
    public String question(HttpServletRequest context) {  
        ...  
    }  
}
```

e.

```
@Controller  
@RequestMapping("/exam")  
public class ExamController {  
  
    @RequestMapping("question")  
    public String question(@RequestHeader("user") String user) {  
        ...  
    }  
}
```

Time left 0:33:42

Activate Windows
Go to Settings to activate Windows.

2:20 PM 2/7/2022

https://ispiti.finki.ukim.mk/mod/quiz/attempt.php?attempt=283663&cmid=11075&page=7

80% Цветановски Матеј Time left 0:33:21

Give your reasons

Activate Windows
Go to Settings to activate Windows.

2:21 PM 2/7/2022

Решение/Објаснување 1: Точни се под А и Е

Решение/Објаснување 2:

Решение/Објаснување 3:

Земено од првата база:

2021/2022/Z

640 / Јануарска сесија (7.2.2022) / Теорија 7.2.2022

Time left 0:31:38

Question 7
Not yet answered
Marked out of 2.00
Flag question

REST контролерот ConsultationSlotApi е мапиран на следната локација: /api/consultations. Апликацијата е стартирана на порта 8080. Кои од прикажаните методи успешно ќе го обработат PATCH барањето во следната форма:

```
PATCH http://localhost:8080/api/consultations/{slotId}
professorId: petko.petkov
Content-Type: application/x-www-form-urlencoded
roomName=11&dayOfWeek=2&from=18:00:00&to=20:00:00
```

Како опционален параметар во барањето, може да се прати и date којшто треба да се десеријализира во објект од типот LocalDate.

Select one or more:

a. `@RequestMapping(value = "{pathId}", method = RequestMethod.PATCH)
public ConsultationSlot updateSlot(@PathVariable int slotId,
 @RequestHeader String professorId,
 @RequestParam String roomName,
 @RequestParam(value = "dayOfWeek") Integer dayOfWeek,
 @RequestParam(value = "date", required = false) @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
 @RequestParam("from") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
 @RequestParam("to") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
 @RequestParam("to")
 return consultationSlotService.updateSlot(slotId, professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
}`

b. `@PatchMapping("/{slotId}")
public ConsultationSlot updateSlot(@PathVariable int slotId,
 @RequestHeader String professorId,
 @RequestParam String roomName,
 @RequestParam(value = "dayOfWeek", required = false) Integer dayOfWeek,
 @RequestParam(value = "date", required = false) @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
 @RequestParam("from") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
 @RequestParam("to") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
 @RequestParam("to")
 return consultationSlotService.updateSlot(slotId, professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
}`

c. `@PatchMapping("/{slotId}")
public ConsultationSlot updateSlot(@RequestParam int slotId,
 @RequestHeader String professorId,
 @RequestParam String roomName,
 @RequestParam(value = "dayOfWeek", required = true) Integer dayOfWeek,
 @RequestParam(value = "date", required = false) @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
 @RequestParam("from") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
 @RequestParam("to") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
 @RequestParam("to")
 return consultationSlotService.updateSlot(slotId, professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
}`

d. `@PatchMapping("/{slotId}")
public ConsultationSlot updateSlot(@PathVariable int slotId,
 @RequestHeader String professorId,
 @RequestParam String roomName,
 @RequestParam(value = "dayOfWeek") Integer dayOfWeek,
 @RequestParam(value = "date") @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
 @RequestParam("from") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
 @RequestParam("to") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
 @RequestParam("to")
 return consultationSlotService.updateSlot(slotId, professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
}`

Activate Windows
Go to Settings to activate Windows.

2:22 PM 2/7/2022

Select one or more:

a. `@RequestMapping(value = "{pathId}", method = RequestMethod.PATCH)
public ConsultationSlot updateSlot(@PathVariable int slotId,
 @RequestHeader String professorId,
 @RequestParam String roomName,
 @RequestParam(value = "dayOfWeek") Integer dayOfWeek,
 @RequestParam(value = "date", required = false) @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
 @RequestParam("from") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
 @RequestParam("to") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
 @RequestParam("to")
 return consultationSlotService.updateSlot(slotId, professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
}`

b. `@PatchMapping("/{slotId}")
public ConsultationSlot updateSlot(@PathVariable int slotId,
 @RequestHeader String professorId,
 @RequestParam String roomName,
 @RequestParam(value = "dayOfWeek", required = false) Integer dayOfWeek,
 @RequestParam(value = "date", required = false) @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
 @RequestParam("from") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
 @RequestParam("to") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
 @RequestParam("to")
 return consultationSlotService.updateSlot(slotId, professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
}`

c. `@PatchMapping("/{slotId}")
public ConsultationSlot updateSlot(@RequestParam int slotId,
 @RequestHeader String professorId,
 @RequestParam String roomName,
 @RequestParam(value = "dayOfWeek", required = true) Integer dayOfWeek,
 @RequestParam(value = "date", required = false) @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
 @RequestParam("from") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
 @RequestParam("to") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
 @RequestParam("to")
 return consultationSlotService.updateSlot(slotId, professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
}`

d. `@PatchMapping("/{slotId}")
public ConsultationSlot updateSlot(@PathVariable int slotId,
 @RequestHeader String professorId,
 @RequestParam String roomName,
 @RequestParam(value = "dayOfWeek") Integer dayOfWeek,
 @RequestParam(value = "date") @DateTimeFormat(iso = DateTimeFormat.ISO.DATE) LocalDate date,
 @RequestParam("from") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime from,
 @RequestParam("to") @DateTimeFormat(iso = DateTimeFormat.ISO.TIME) LocalTime to) {
 @RequestParam("to")
 return consultationSlotService.updateSlot(slotId, professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
}`

Activate Windows
Go to Settings to activate Windows.

2:23 PM 2/7/2022

Решение/Објаснување 1: Под Б, објаснување има погре

Решение/Објаснување 2:

Решение/Објаснување 3:

7.

REST контролерот ConsultationSlotApi е мапиран на следната локација: /api/consultations. Апликацијата е стартувана на порта 8080. Кои од прикажаните методи успешно ќе го обработат GET барањето во следната форма:

GET <http://localhost:8080/api/consultations/>
professorId: petko.petkov

Select one or more:

- a. `@GetMapping`
public Page<ConsultationSlot> getAllConsultationSlots(
 @RequestHeader("professorId") String professorId,
 @RequestParam(name = "page") int page,
 @RequestParam(name = "page-size") int size) {
 return consultationSlotService.getAllConsultationSlots(page, size);
 }
- b. `@GetMapping`
public Page<ConsultationSlot> getAllConsultationSlots(
 @RequestHeader("professorId") String professorId,
 @RequestHeader(name = "page", defaultValue = "0", required = false) int page,
 @RequestHeader(name = "page-size", defaultValue = "10", required = false) int size) {
 return consultationSlotService.getAllConsultationSlots(page, size);
 }
- c. `@GetMapping`
public Page<ConsultationSlot> getAllConsultationSlots(
 @RequestHeader("professorId") String professorId,
 @RequestHeader(name = "page") int page,
 @RequestHeader(name = "page-size") int size) {
 return consultationSlotService.getAllConsultationSlots(page, size);
 }
- d. `@RequestMapping(method = RequestMethod.GET)`
public Page<ConsultationSlot> getAllConsultationSlots(
 @RequestHeader(value = "professorId", required = true) String professorId,
 @RequestHeader(name = "page", defaultValue = "0", required = false) int page,
 @RequestHeader(name = "page-size", defaultValue = "10", required = false) int size) {
 return consultationSlotService.getAllConsultationSlots(page, size);
 }
- e. `@GetMapping("/{professorId}/{page}/{page-size}")`
public Page<ConsultationSlot> getAllConsultationSlots(
 @PathVariable("professorId") String professorId,
 @PathVariable(name = "page") int page,
 @PathVariable(name = "page-size") int size) {
 return consultationSlotService.getAllConsultationSlots(page, size);
 }

8.

Доколку имаме потреба да следните методи е можно

Select one or more:

- a. `@Controller`
@RequestMapping
public class Example {
- b. `@ResponseWrapper`
@ResponseWrapper
- c. `@ResponseBody`
@ResponseBody

- d. `@RequestWrapper`
@RequestWrapper
- e. `@ResponseWrapper`
@ResponseWrapper

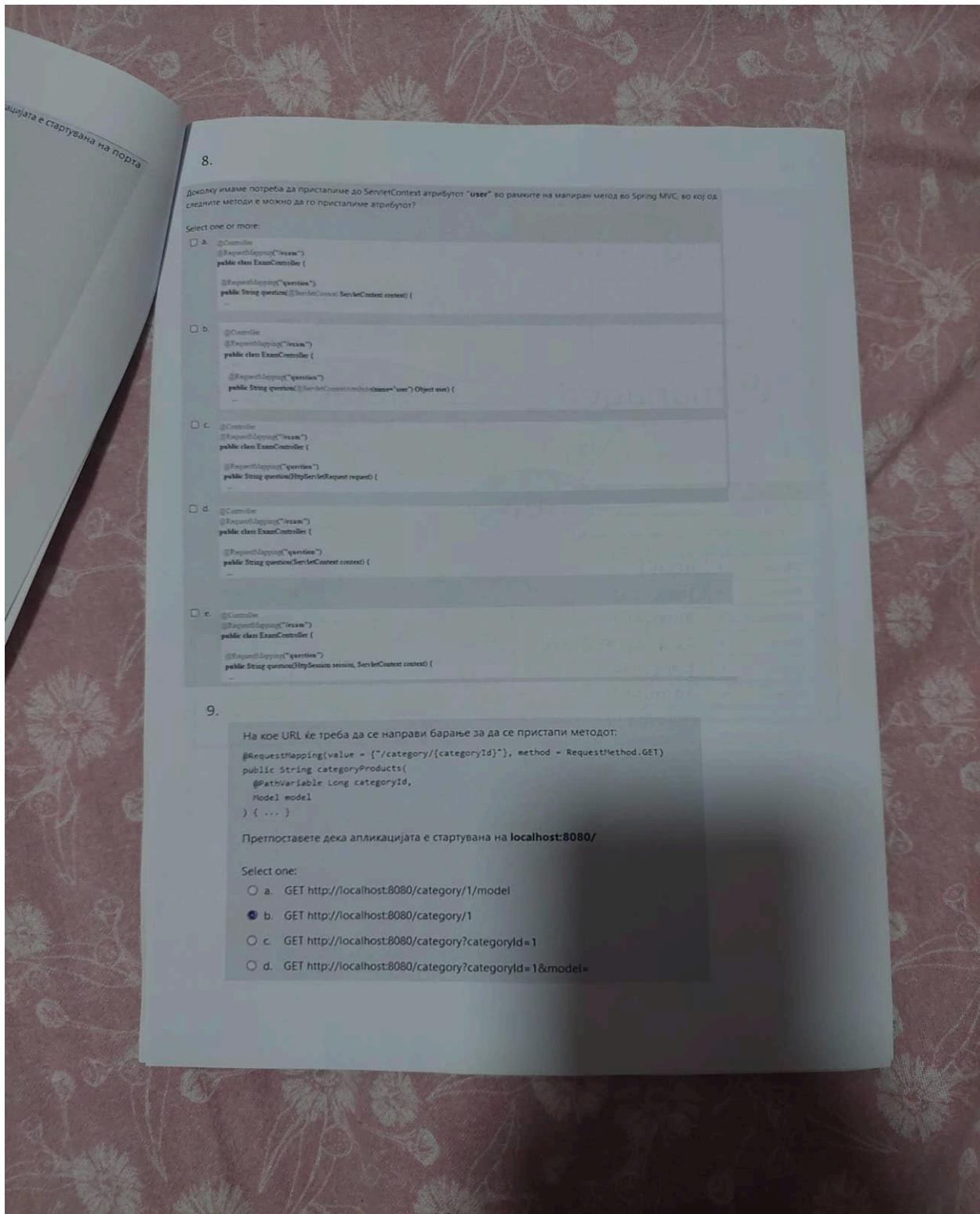
Решение/Објаснување 1: Ако читам добро во барањето, има

/api/consultations/7 , а во решенијата никаде нема path variable

Решение/Објаснување 2:

Решение/Објаснување 3:

Земено од првата база:



Решение/Објаснување 1: Прашање 9 под Б, бидејќи нема @RequestParam

Решение/Објаснување 2: Прашање 8 под с,d,e.

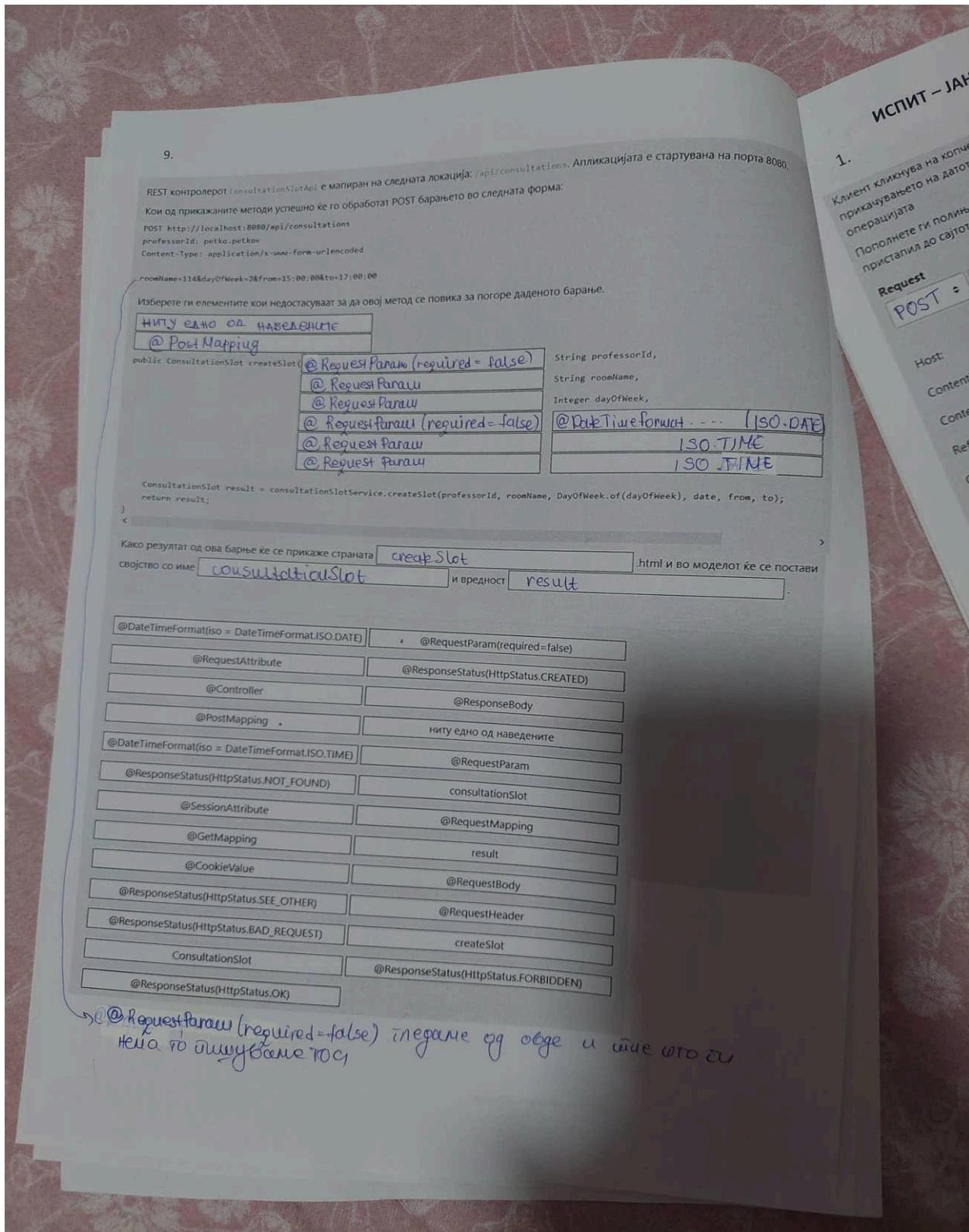
Не постојат анотации за ServletContext како во а и в

Во с може да се стигне до контекстот со request.getServletContext()

Под d и е го имаме контекстот земено

Решение/Објаснување 3:

Земено од првата база:



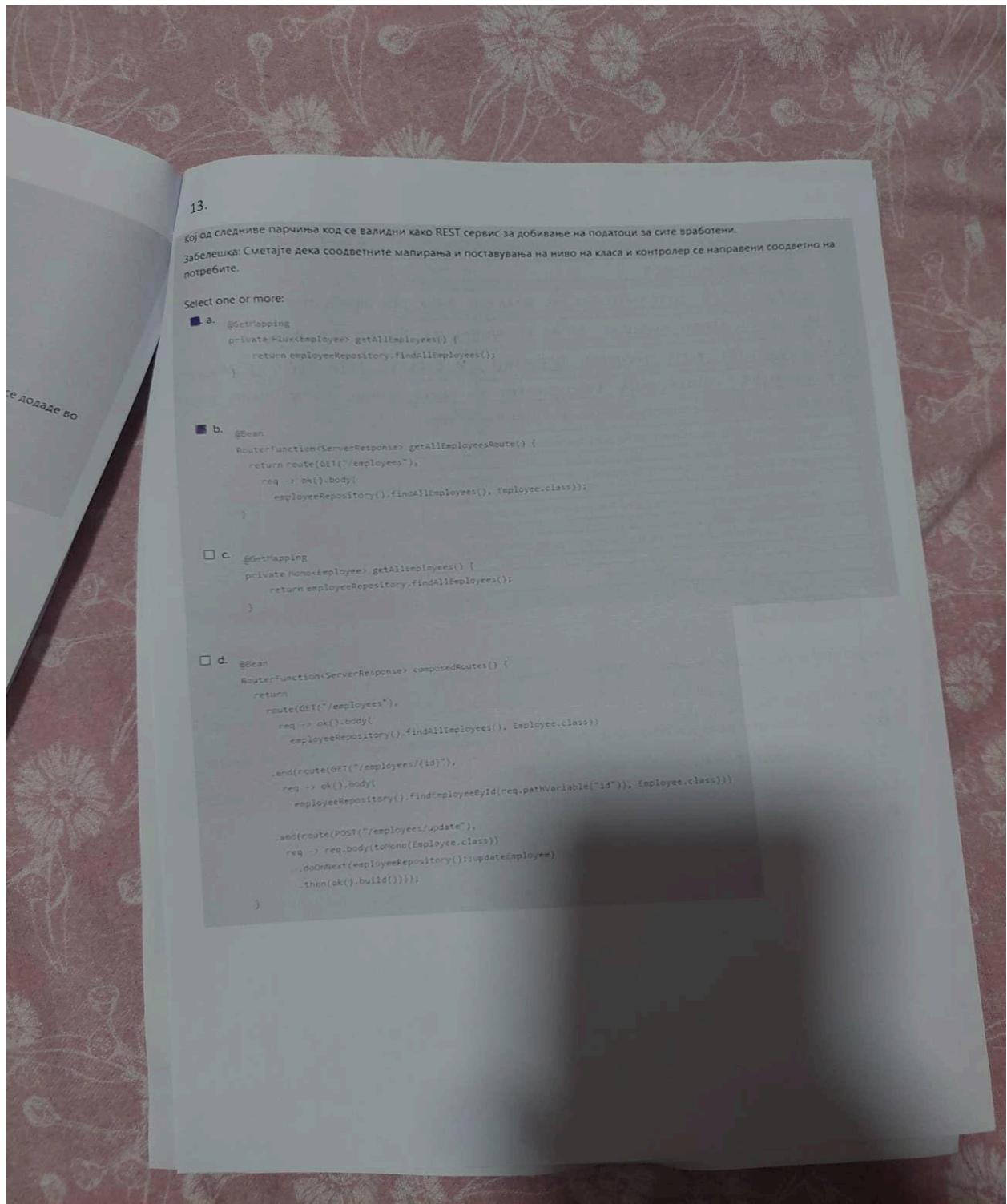
Решение/Објаснување 1: Третото е грешно, треба да е `@RequestHeader`.

Другите мислам се точни. Можда првото треба да е `@ResponseBody` ама не сум сигурен.

Решение/Објаснување 2:

Решение/Објаснување 3:

Земено од првата база:



Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

Земено од првата база:

Question 9
Not yet answered
Marked out of 15.00
Flag question

REST контролерот `ConsultationSlotApi` е мапиран на следната локација: `/api/consultations`. Апликацијата е стартувана на порта 8080.

Кои од прикажаните методи успешно ќе го обработат POST барањето во следната форма:

```
POST http://localhost:8080/api/consultations
professorId: petko.petkov
Content-Type: application/x-www-form-urlencoded

roomName=114&dayOfWeek=2&from=15:00:00&to=17:00:00
```

Изберете ги елементите кои недостасуваат за да овој метод се повика за погоре даденото барање.


```
public ConsultationSlot createSlot(
    String professorId,
    String roomName,
    Integer dayOfWeek,
    Date date,
    Date from,
    Date to
) {
    ConsultationSlot result = consultationSlotService.createSlot(professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
    return result;
}
```

Како резултат од ова барање ќе се прикаже страната .html и во моделот ќе се постави

Time left 0:23:54


```
ConsultationSlot result = consultationSlotService.createSlot(professorId, roomName, DayOfWeek.of(dayOfWeek), date, from, to);
return result;
```

Како резултат од ова барање ќе се прикаже страната .html и во моделот ќе се постави свойство со име и вредност .

@DateTimeFormat(iso = DateTimeFormat.ISO.DATE)	@RequestParam(required=false)
@RequestAttribute	@ResponseStatus(HttpStatus.CREATED)
@Controller	@ResponseBody
@PostMapping	ниту едно од наведените
@DateTimeFormat(iso = DateTimeFormat.ISO.TIME)	@RequestParam
@ResponseStatus(HttpStatus.NOT_FOUND)	consultationSlot
@SessionAttribute	@RequestMapping
@GetMapping	result

Решение/Објаснување 1: Го има погоре решено на лист

Решение/Објаснување 2:

Решение/Објаснување 3:

Земено од првата база:

стартувана на порта 8080.

Time left 0:25:39

Кои од прикажаните методи успешно ќе го обработат POST барањето во следната форма:

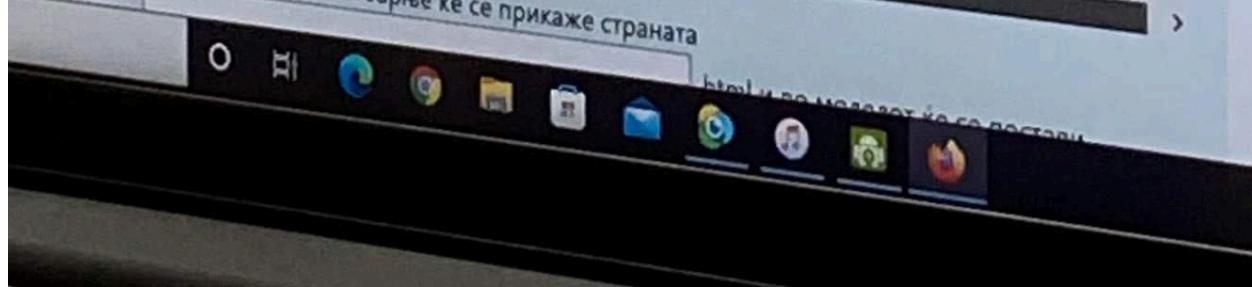
```
POST http://localhost:8080/api/students  
professorId: 345  
Set-Cookie: enrolmentYear: 2019  
Content-Type: application/x-www-form-urlencoded  
  
name=Desetko&studyProgram=PET&enrolmentYear=2019&birthDate=20-11-2001
```

Изберете ги елементите кои недостасуваат за да овој метод се повика за погоре даденото барање.

	String professorId;
	String name;
	String lastName;
	String studyProgram;
	Integer enrolmentYear;
	LocalDate birthDate;

```
, studyProgram, enrolmentYear, birthDate);
```

Како резултат од ова барје ќе се прикаже страната



Решение/Објаснување 1:

@RequestHeader

@RequestParam

```
@RequestParam(required=false)
@RequestParam
@RequestParam/
@RequestParam DateTimeFormat(iso=DateTimeFormat.ISO.DATE)
Решение/Објаснување 2:
Решение/Објаснување 3:
```

Земено од првата база:

2. Доколку имаме потреба да пристапиме до сесиски атрибут "user" во рамките на мапиран метод во Spring MVC, во кој од следните методи е можно да го пристапиме атрибутот?

Select one or more

a.

```
@Controller  
@RequestMapping("/exam")  
public class ExamController {  
    @RequestMapping("question")  
    public String question(HttpServletRequest request) {  
        ...  
    }
```

b.

```
@Controller  
@RequestMapping("/exam")  
public class ExamController {  
    @RequestMapping("question")  
    public String question(ServletContext context) {  
        ...  
    }
```

c.

```
@Controller  
@RequestMapping("/exam")  
public class ExamController {  
    @RequestMapping("question")  
    public String question(HttpSession session, ServletContext context) {  
        ...  
    }
```

d.

```
@Controller  
@RequestMapping("/exam")  
public class ExamController {  
    @RequestMapping("question")  
    public String question(@Session HttpSession session) {  
        ...  
    }
```

e.

```
@Controller  
@RequestMapping("/exam")  
public class ExamController {
```

```
    @RequestMapping("question")  
    public String question(@SessionAttribute(name="user") Object user) {  
        ...  
    }
```

Решение/Објаснување 1: точно се обележани а,с,е.

Под а. може да се земе сесијата со request.getSession(), и преку нејзе да се земе атрибутот со session.getAttribute("user").

Под б. Не може да се земе сесиски атрибут преку ServletContext

Под с. исто со session.getAttribute("user")

Под д. Не е точно дека не постои @Session анотација

Под е. се зема атрибутот директно

Решение/Објаснување 2:

Решение/Објаснување 3:

Земено од првата база:

3.На кое URL ќе треба да се направи барање за да се пристапи методот:

```
@RequestMapping(value = "/search"), method = RequestMethod.GET)
public String search(
    @RequestParam String query,
    Model model
){ ... }
Предпоставете дека апликацијата е стартувана на localhost:8080/
Select one:
a. GET http://localhost:8080/search/model?query=text
b. GET http://localhost:8080/search?query=text&model=
c. GET http://localhost:8080/search?query=text
d. GET http://localhost:8080/search/query
```

Решение/Објаснување 1: Точно е

Решение/Објаснување 2:

Решение/Објаснување 3:

Земено од првата база:

4.Кои искази се точни за Spring MVC.

Select one or more:

- a. Spring MVC ги процесира барањата преку DispatcherServlet-от
- b. Spring MVC ги процесира барањата преку DispatcherFilter-от
- c. Spring MVC ги читува зависностите (bean) преку ContextLoaderListener од локацијата дефинирана во contextConfigLocation параметарот
- d. Spring MVC ги читува зависностите (bean) од локацијата дефинирана во contextConfigLocation серверот иницијализацискиот параметар

Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

Земено од првата база:

5. Нека во веб апликација е поставена следната конфигурација view-resolver:

```
<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/views/" />
    <property name="suffix" value=".jsp" />
</bean>
```

Која ќе биде датотеката која ќе се избере од ViewResolver-от за приказ (view template) при повик на question() методот дефиниран во продолжение?

```
@Controller
@RequestMapping("/exam")
public class ExamController {
    @RequestMapping(value = "/question", method = RequestMethod.POST)
    public String question() {
        return "question";
    }
}
```

Select one:

- a. /views/question.jsp
- b. Нема да се селектира ниту една датотека, ќе се генерира JSON
- c. Нема да се селектира ниту една датотека, ќе се изврши редирекција

-
- d. /WEB-INF/jsp/question.jsp
 - e. /WEB-INF/views/question.jsp

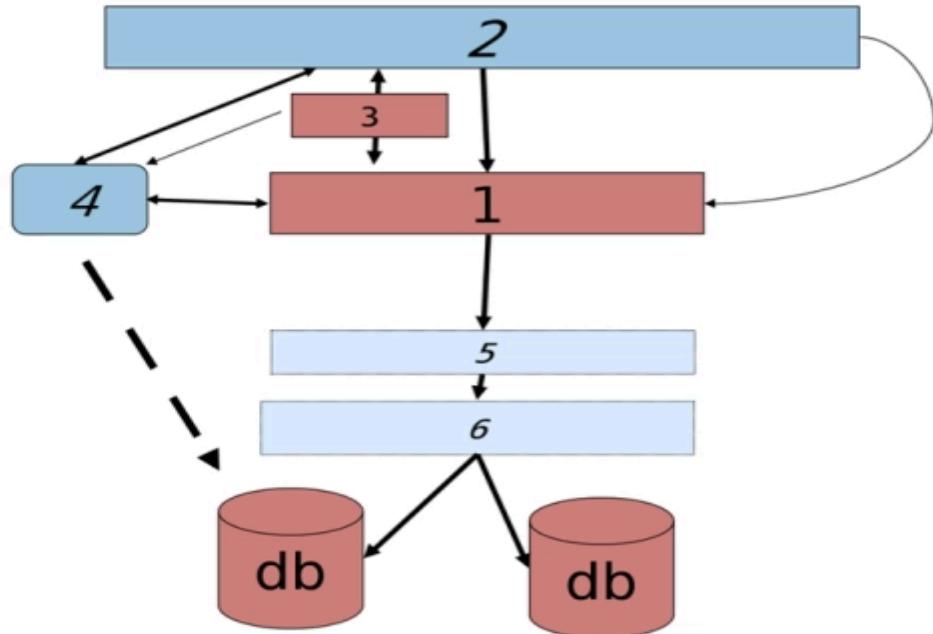
Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

1
d
out of
question

Поврзете ги компонентите кои одговараат на броевите од слика.



- 1 Choose...
- 2 Choose...
- 3 Choose...
- 4 JDBC driver
- 5 Entity Manager
- 6 JPA provider
- 7 EntityManagerFactory
- 8 Entity
- 9 EntityManager
- 10 Query
- 11 your application
- 12 Choose...

Give your reasons



Решение/Објаснување 1: 1.Entity manager, 2.your app, 3.query, 4.entity,

5.JPA provider, 6.JDBC driver, 7.entity manager factory

Решение/Објаснување 2:

Решение/Објаснување 3:

2
d
out of
question

Означете ги JPA анотациите кои треба да се постават за класите да се мапираат во релациона база на податоци.

```
public class Student {
    public String index;
    public String name;
    public String email;
    public StudyProgram program;
}

public class StudyProgram {
    public Long id;
    public String name;
    public List<Student> enrolledStudents;
}
```

Решение/Објаснување 1:

of
ation

Означете ги JPA анотациите кои треба да се постават за класите да се мапираат во релациона база на податоци.

```
@Entity
public class Student {
    @Id
    public String index;
    @Id
    public String name;
    @Id
    public String email;
    @ManyToOne
    public StudyProgram program;
}

@Entity
public class StudyProgram {
    @Id
    public Long id;
    @Id
    public String name;
    @OneToMany(mappedBy="program")
    public List<Student> enrolledStudents;
}
```

Решение/Објаснување 2:

Решение/Објаснување 3:

3
1
out of
question

Во еден проект е конфигурирано да се користи стратегијата за именување на колоните од аудиториските вежби:
`spring.jpa.hibernate.naming.implicit-strategy=org.hibernate.boot.model.naming.ImplicitNamingStrategyJpaCompliantImpl`

Потребно е да ја анотирате класа подолу **со минимален број на JPA анотации** за да се поврзе со табелата **COOPERATE_USER** со следните колони: ("EMBG", "name", "job", "address", "birthDate").

```
public class CooperateUser {  
    public String embg;  
    public String name;  
    public String job;  
    public String address;  
    public LocalDate birthDate;  
}
```

```
@Table(name = "COOPERATE_USER") @Entity  
@Id @Column(name = "EMBG")  
Нема потреба од анотација  
@Entity  
@Id  
@Column(name = "EMBG")  
@Column(name = "name")  
@Column(name = "job")  
@Column(name = "address")  
@Column(name = "birthDate")  
@Column(name = "birth_date")
```

Give your reasons

Решение/Објаснување 1:

1
ed
1 out of
question

Во еден проект е конфигурирано да се користи стратегијата за именување на колоните од аудиториските вежби:
`spring.jpa.hibernate.naming.implicit-strategy=org.hibernate.boot.model.naming.ImplicitNamingStrategyJpaCompliantImpl`

Потребно е да ја анотирате класа подолу **со минимален број на JPA анотации** за да се поврзе со табелата **COOPERATE_USER** со следните колони: ("EMBG", "name", "job", "address", "birthDate").

```
@Entity  
public class CooperateUser {  
    @Id @Column(name="EMBG")  
    public String embg;  
    public String name;  
    public String job;  
    public String address;  
    public LocalDate birthDate;  
}
```

```
@Table(name = "COOPERATE_USER") @Entity  
@Id @Column(name = "EMBG")  
Нема потреба од анотација  
@Entity  
@Id  
@Column(name = "EMBG")  
@Column(name = "name")  
@Column(name = "job")  
@Column(name = "address")  
@Column(name = "birthDate")  
@Column(name = "birth_date")
```

Give your reasons

Решение/Објаснување 2:

Решение/Објаснување 3:

on 4
it
red
id out of
g question

Доколку го имаме следното мапирање на ентитетите, кои табели ќе се изгенерираат од страна на JPA.

```
@Entity  
@Table(name = "STUDENTS")  
public class Student extends BaseEntity {  
  
    @Column(name = "student_index")  
    public String index;  
    public String firstName;  
    public String lastName;  
    @ManyToMany  
    public List<Course> courses;  
}  
  
@Entity  
@Table(name = "COURSES")  
public class Course extends BaseEntity {  
    public String name;  
    @ManyToMany(mappedBy = "courses")  
    public List<Student> students;  
}
```

Притоа ја имаме следната конфигурација:

```
spring.jpa.hibernate.naming.implicit-strategy=org.hibernate.boot.model.naming.ImplicitNamingStrategyJpaCompliantImpl
```

Select one or more:

- a. STUDENT_COURSES
- b. COURSES_STUDENTS
- c. STUDENTS
- d. COURSES

Give your reasons



Решение/Објаснување 1: В, С и D - В поради тоа што имаме mappedBy во Courses, без тоа ќе се креираа сите понудени табели

Решение/Објаснување 2:

Решение/Објаснување 3:

5
ed
1 out of
question

Дадени се ентитетите:

```
@Entity
public class BookDetails {
    @Id @GeneratedValue
    public long id;
    public String description;
    @OneToOne
    public Book book;
}

@Entity
public class Book {
    @Id @GeneratedValue
    public long id;
    public String name;
    public String isbn;
    public Double price;
}
```

Доколку сакаме да ги пронајдеме сите вредности според **isbn** на книгата, кој метод треба да се додаде во следната класа подолу за Spring Data да ни овозможи добивање на валидните резултати:

```
@Repository
public interface BookDetailsRepository extends CrudRepository<BookDetails, Long> {
}
```

Select one:

- a. List<BookDetails> findByisbn(String isbn);
- b. List<BookDetails> searchisbn(String isbn);
- c. List<BookDetails> findBybookDetailsisbnLike(String isbn);
- d. List<BookDetails> findBybookisbn(String isbn);

Give your reasons

Решение/Објаснување 1: Под Д, бидејќи репозиториумот е за BookDetails кој содржи Book, а ISBN е во Book

Решение/Објаснување 2:

Решение/Објаснување 3:

6
d
out of
question

Кој од следниве парчиња код се валидни како REST сервис за добивање на податоци за сите вработени.

Забелешка: Сметајте дека соодветните мапирања и поставувања на ниво на класа и контролер се направени соодветно на потребите.

Select one or more:

a.

```
@GetMapping  
private Mono<Employee> getAllEmployees() {  
    return employeeRepository.findAllEmployees();  
}
```

b.

```
@Bean  
RouterFunction<ServerResponse> getAllEmployeesRoute() {  
    return route(GET("/employees"),  
        req -> ok().body(  
            employeeRepository().findAllEmployees(), Employee.class));  
}
```

c.

```
@GetMapping  
private Flux<Employee> getAllEmployees() {  
    return employeeRepository.findAllEmployees();  
}
```

d.

```
@Bean  
RouterFunction<ServerResponse> composedRoutes() {  
    return  
        route(GET("/employees"),  
            req -> ok().body(  
                employeeRepository().findAllEmployees(), Employee.class))  
  
.and(route(GET("/employees/{id}"),  
        req -> ok().body(  
            employeeRepository().findEmployeeById(req.pathVariable("id")), Employee.class)))  
  
.and(route(POST("/employees/update"),  
        req -> req.body(toMono(Employee.class))  
            .doOnNext(employeeRepository()::updateEmployee)  
            .then(ok().build())));  
}
```

Give your reasons



Решение/Објаснување 1: b,c,d

Решение/Објаснување 2: a- e private znaci ne moze i guess?

Решение/Објаснување 3:

7
ed
out of
question

При веб тестирање, кои се предностите ако се тестира рендерирањето во веб browser-от.

Select one or more:

- a.
Well defined API and data structures – resilient
- b.
Can test for usability, performance, responsive design, etc.
- c.
Unit testing of key modules
- d.
Most realistic testing, validates what the end user experiences



Give your reasons

<input type="button" value="A"/>	<input type="button" value="B"/>	<input type="button" value="I"/>	<input type="button" value="≡"/>	<input type="button" value="≡"/>	<input button"="" type="button" value="S"/>	<input type="button" value=""/>
----------------------------------	----------------------------------	----------------------------------	----------------------------------	----------------------------------	---	---------------------------------

Решение/Објаснување 1: b i d, pisit vo predavaneto za security

Решение/Објаснување 2:

Решение/Објаснување 3:

8
t
d out of
question

Кој од следните тестови се користи за проверка на точноста на одреден метод, изолиран од неговите зависности?

Select one:

- a. Performance Testing
- b. Integration Testing
- c. Unit Testing
- d. Stress Testing
- e. System Testing
- f. User Acceptance Testing

Give your reasons

<input type="button" value="A"/>	<input type="button" value="B"/>	<input type="button" value="I"/>	<input type="button" value="≡"/>	<input type="button" value="≡"/>	<input data-bbox="731 1512 763 1564" type="button" value="%"/>	<input type="button" value="S"/>	<input type="button" value=""/>
----------------------------------	----------------------------------	----------------------------------	----------------------------------	----------------------------------	--	----------------------------------	---------------------------------

Решение/Објаснување 1:Unit testing

Решение/Објаснување 2:

Решение/Објаснување 3:

9

id
out of
question

Селектирајте ги точните тврдења за процесот на автентикација.

Select one or more:

- a. Корисникот е валиден доколку е пронајден во User/Credential Storage
- b. Се идентификува корисникот во рамките на системот
- c. Се дефинира кои ресурси смее одреден корисник да ги пристапи
- d. Корисникот секогаш се идентификува според корисничко име и лозинка

Give your reasons



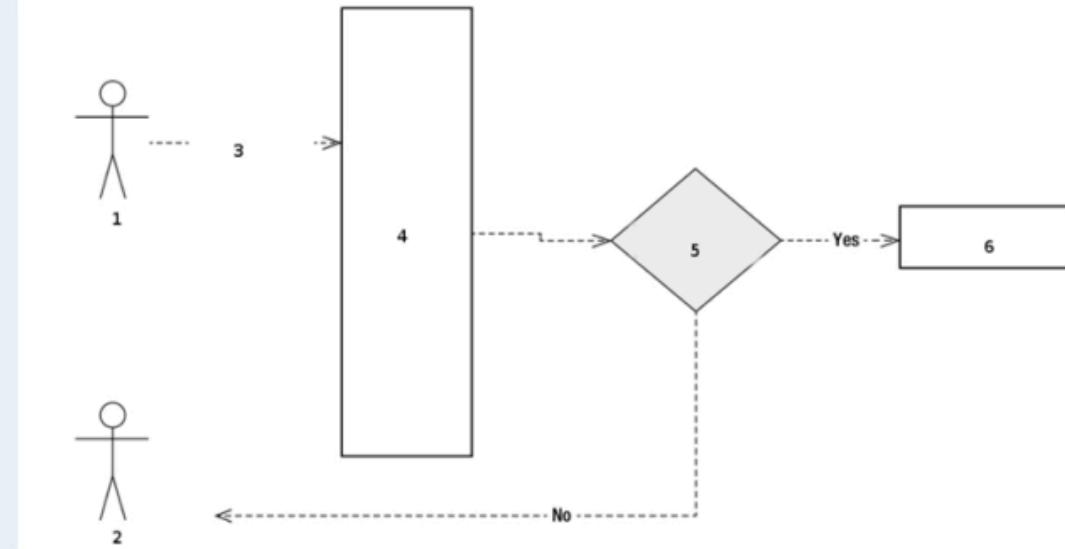
Решение/Објаснување 1: a, b

Решение/Објаснување 2:

Решение/Објаснување 3:

10
d
out of
question

Означете кој термин одговара на бројчето во дијаграмот.



Authorization layer

Choose... ↗

User not allowed access

Choose... ↗

Access Resource

Choose... ↗

Secured Resource

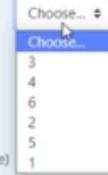
Choose... ↗

Permissions match (user and resource)

Choose... ↗

User

Choose... ↗



Give your reasons:

1 A B I ≡ % S

Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

12.

Даден е ентитетот:

```
@Entity
@Table(name = "book_details")
public class BookDetails extends BaseEntity {

    @Column(length = 5000)
    public String description;

    @OneToOne
    public Book book;
}

Доколку сакаме да ги пронајдеме сите вредности кои содржат одреден опис (description), кој метод треба да се додаде во следната класа подолу за Spring Data да ни овозможи добивање на валидните резултати:
```

@Repository

```
public interface BookDetailsRepository extends CrudRepository<BookDetails, Long> {
}
```

Select one:

- a. List<BookDetails> findByDescriptionLike(String text);
- b. List<BookDetails> findByBookDetailsDescriptionLike(String text);
- c. List<BookDetails> findByDescription(String text);
- d. List<BookDetails> searchDescription(String text);

13.

Кој од следните па
Забелешка: Смет
потребите.

Select one of
a.

Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

11.

Означете ги JPA анотациите кои треба да се постават за класите да се мапираат во релационна база на податоци.

```
@Entity  
public class Vacation {  
    @Id  
    public Long id;  
    // Не мора анотација  
    public LocalDate from;  
    // Не мора анотација  
    public LocalDate to;  
    @ManyToOne  
    public Employee requester;  
}
```

```
@Entity  
public class Employee {  
    @Id  
    public Long id;  
    // Нема анотација  
    public String name;  
    @OneToMany(mappedBy = "requester")  
    public List<Vacation> absences;
```

```
@OneToOne(mappedBy = "requester")  
@ManyToOne(mappedBy = "employee")  
@OneToMany(mappedBy = "absences")  
@Entity  
@Column(name = "x")  
@ManyToOne(mappedBy = "Employee")  
@ManyToOne  
@Id  
@OneToMany(mappedBy = "Vacation")  
Не мора да има анотација
```

Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

се-2021/2022/Z

Z-39_30640 / Јануарска сесија (7.2.2022) / Теорија 7.2.2022

Question 11
Not yet answered
Marked out of 2.00
 Flag question

Time left 0:27:19

Кое од следните мапирања на класите Training и Group е валидно?

Select one or more:

a.

```
@Entity
public class Training {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
    @ManyToOne
    public Group group;

}

@Entity
public class Group {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
    @OneToOne(mappedBy = "TRAINING")
    List<Training> trainings;
}
```

b.

```
@Table(name = "TRAINING")
@Entity
public class Training {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
    @ManyToOne
    public Group group;
}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
    @OneToOne(mappedBy = "trainings")
    List<Training> trainings;
}
```

c.

```
@Table(name = "TRAINING")
@Entity
public class Training {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
    @ManyToOne
    public Group group;
}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
}
```

Activate Windows
Go to Settings to activate Windows.

2:27 PM 2/7/2022

https://ispiti.finki.ukim.mk/mod/quiz/attempt.php?attempt=283663&cmid=11075&page=10

Time left 0:27:14

Цветановски Матеј

b.

```
@Table(name = "TRAINING")
@Entity
public class Training {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
    @ManyToOne
    public Group group;
}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
    @OneToOne(mappedBy = "trainings")
    List<Training> trainings;
}
```

c.

```
@Table(name = "TRAINING")
@Entity
public class Training {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
    @ManyToOne
    public Group group;
}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
}
```

Activate Windows
Go to Settings to activate Windows.

2:27 PM 2/7/2022

The screenshot shows a web browser window with a quiz interface. The URL is <https://ispiti.finki.ukim.mk/mod/quiz/attempt.php?attempt=283663&cmid=11075&page=10>. The page displays Java code for entity classes:

```
@Table(name = "TRAINING")
@Entity
public class Training {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
    @ManyToOne
    public Group group;
}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
    @OneToMany(mappedBy = "group")
    List<Training> trainings;
}

d. @Table(name = "TRAINING")
@Entity
public class Training {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
    @OneToOne
    public Group group;
}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
    @ManyToOne(mappedBy = "group")
    List<Training> trainings;
}
```

The code is part of a multiple-choice question (d). The interface includes a progress bar, a timer (Time left 0:27:08), and a watermark for "Цветановски Матеј".

The screenshot shows a web browser window with a quiz interface. The URL is <https://ispiti.finki.ukim.mk/mod/quiz/attempt.php?attempt=283663&cmid=11075&page=10>. The page displays Java code for entity classes:

```
@Table(name = "TRAINING_GROUP")
@Entity
public class Group {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
    @ManyToOne(mappedBy = "group")
    List<Training> trainings;
}

e. @Table(name = "TRAINING")
@Entity
public class Training {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
    @ManyToOne
    public Group group;
}

@Table(name = "TRAINING_GROUP")
@Entity
public class Group {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Long id;
    public String name;
    @Transient
    List<Training> trainings;
}
```

The code is part of a multiple-choice question (e). The interface includes a progress bar, a timer (Time left 0:26:57), and a watermark for "Цветановски Матеј".

Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

3
1
out of
question

Во еден проект е конфигурирано да се користи стратегијата за именување на колоните од аудиториските вежби:
`spring.jpa.hibernate.naming.implicit-strategy=org.hibernate.boot.model.naming.ImplicitNamingStrategyJpaCompliantImpl`

Потребно е да ја анотирате класа подолу со минимален број на JPA анотации за да се поврзе со табелата **COOPERATE_USER** со следните колони: ("EMBG", "name", "job", "address", "birthDate").

@Table(name = "COOPERATE_USER") @Entity

```
public class CooperateUser {
    @Id @Column(name = "EMBG")
    public String embg;
    Нема потреба од анотација
    public String name;
    Нема потреба од анотација
    public String job;
    Нема потреба од анотација
    public String address;
    Нема потреба од анотација
    public LocalDate birthDate;
}
```

Give your reasons

Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

2
1
out of
question

Означете ги JPA анотациите кои треба да се постават за класите да се мапираат во релациона база на податоци.

@Entity

```
public class Student {
    @Id
    Не мора да има анотација
    public String index;
    Не мора да има анотација
    public String name;
    Не мора да има анотација
    public String email;
    @ManyToOne
    public StudyProgram program;
}
```

@Entity

```
public class StudyProgram {
    @Id
    public Long id;
    Не мора да има анотација
    public String name;
    @OneToMany(mappedBy="program")
    public List<Student> enrolledStudents;
}
```

Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

on 4
it
red
id out of
g question

Доколку го имаме следното мапирање на ентитетите, кои табели ќе се изгенерираат од страна на JPA.

```
@Entity  
@Table(name = "STUDENTS")  
public class Student extends BaseEntity {  
  
    @Column(name = "student_index")  
    public String index;  
    public String firstName;  
    public String lastName;  
    @ManyToMany  
    public List<Course> courses;  
}  
  
@Entity  
@Table(name = "COURSES")  
public class Course extends BaseEntity {  
    public String name;  
    @ManyToMany(mappedBy = "courses")  
    public List<Student> students;  
}
```

Притоа ја имаме следната конфигурација:

```
spring.jpa.hibernate.naming.implicit-strategy=org.hibernate.boot.model.naming.ImplicitNamingStrategyJpaCompliantImpl
```

Select one or more:

- a. STUDENT_COURSES
- b. COURSES_STUDENTS
- c. STUDENTS
- d. COURSES

Give your reasons



Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

on 5
ed
l out of
question

Дадени се ентитетите:

```
@Entity
public class BookDetails {
    @Id @GeneratedValue
    public long id;

    public String description;

    @OneToOne
    public Book book;
}

@Entity
public class Book {
    @Id @GeneratedValue
    public long id;

    public String name;

    public String isbn;

    public Double price;
}

Доколку сакаме да ги пронајдеме сите вредности според isbn на книгата, кој метод треба да се додаде во следната класа подолу за Spring Data да ни овозможи добивање на валидните резултати:
@Repository
public interface BookDetailsRepository extends CrudRepository<BookDetails, Long> {
}

Select one:
 a. List<BookDetails> findByisbn(String isbn);
 b. List<BookDetails> searchisbn(String isbn);
 c. List<BookDetails> findByBookDetailsisbnLike(String isbn);
 d. List<BookDetails> findBybookisbn(String isbn);

Give your reasons
```

Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

on 8
ed
l out of
question

Кој од следните тестови се користи за проверка на точноста на одреден метод, изолиран од неговите зависности?

Select one:

- a. Performance Testing
- b. Integration Testing
- c. Unit Testing
- d. Stress Testing
- e. System Testing
- f. User Acceptance Testing

Give your reasons



Решение/Објаснување 1: c.unit testing

Решение/Објаснување 2:

Решение/Објаснување 3:

9

id
out of
question

Селектирајте ги точните тврдења за процесот на автентикација.

Select one or more:

- a. Корисникот е валиден доколку е пронајден во User/Credential Storage
- b. Се идентификува корисникот во рамките на системот
- c. Се дефинира кои ресурси смее одреден корисник да ги пристапи
- d. Корисникот секогаш се идентификува според корисничко име и лозинка

Give your reasons



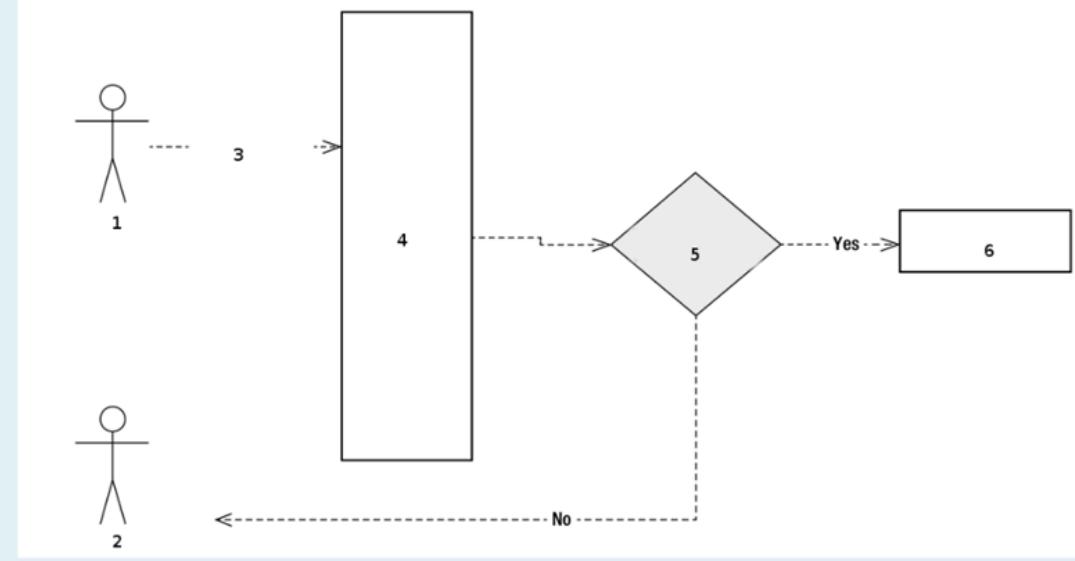
Решение/Објаснување 1: одговорено е погоре

Решение/Објаснување 2:

Решение/Објаснување 3:

10
d
out of
question

Означете кој термин одговара на бројчето во дијаграмот.



Authorization layer

Choose... 4

User not allowed access

Choose... 2

Access Resource

Choose... 3

Secured Resource

Choose... 6

Permissions match (user and resource)

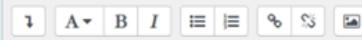
Choose... 5

User

Choose... 1

Choose...
Choose...
3
4
6
2
5
1

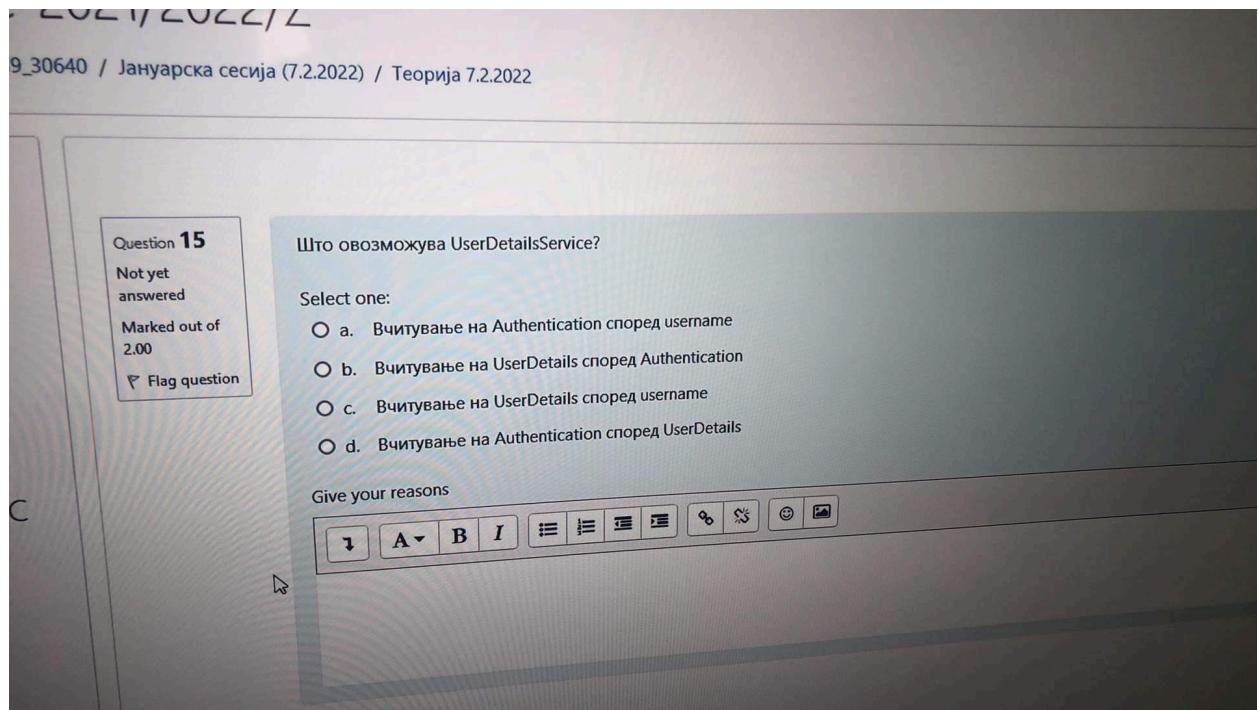
Give your reasons



Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:



Решение/Објаснување 1: под с, ево link

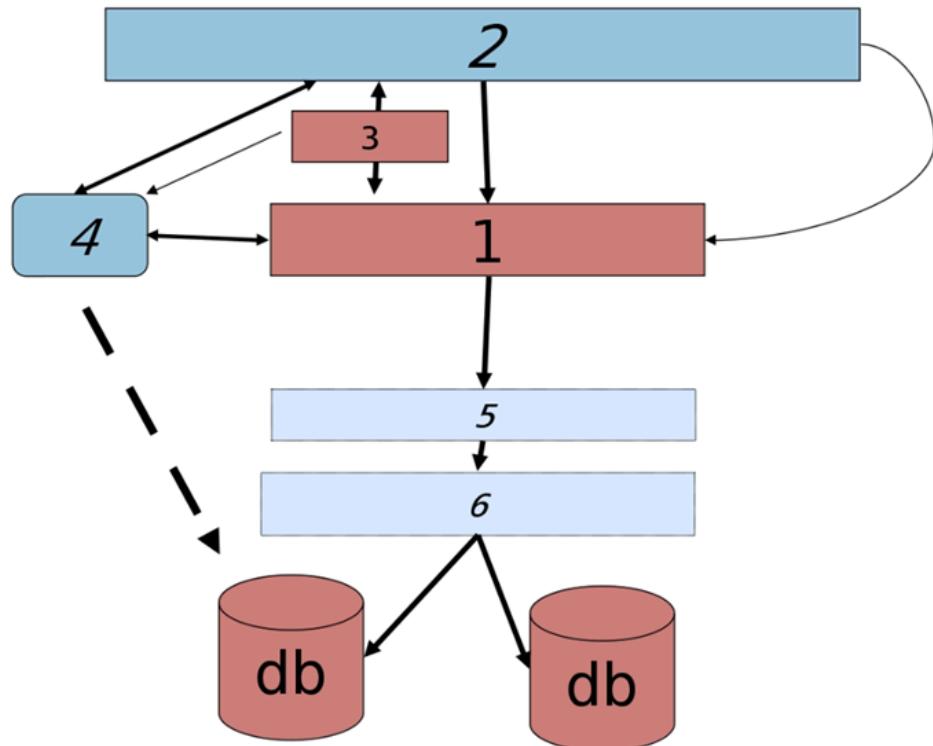
<https://www.baeldung.com/spring-security-authentication-with-a-database#UserDetailsService-1>

Решение/Објаснување 2:

Решение/Објаснување 3:

1
1
out of
question

Поврзете ги компонентите кои одговараат на броевите од сликата.

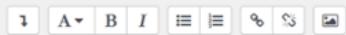


7

- 1 Choose...
- 2 Choose...
- 3 Choose... **JDBC driver**
- 4 Entity Manager
- 5 JPA provider
- 6 EntityManagerFactory
- 7 your applicaton

- 1 - Entity Manager**
2 - your application
3 - Query
4 - Entity
5 - JPA provider
6 - JDBC driver
7 - EntityManager Factory

Give your reasons



Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

14.

Наредете ги чекорите кои се извршуваат при развој на софтвер воден од тестирање (Test driven development)

1. Додадете тест
2. Извршете ги сите тестови и видете дека тестирањето не успева
3. Направете мала промена за да го направите тестот да успее
4. Извршете ги сите тестови повторно и видете дека сите успеваат
5. Направете refactoring на системот за подобрување на неговиот дизајн

*Придружете ги текстовите подоле на чекорите на кои соодветствуваат.

Извршете ги сите тестови и видете дека тестирањето не успева

Направете refactoring на системот за подобрување на неговиот дизајн

Направете мала промена за да го направите тестот да успее

Извршете ги сите тестови повторно и видете дека сите успеваат

Додадете тест

15.

Селектирајте ги точните тврдења за процесот на автентикација.

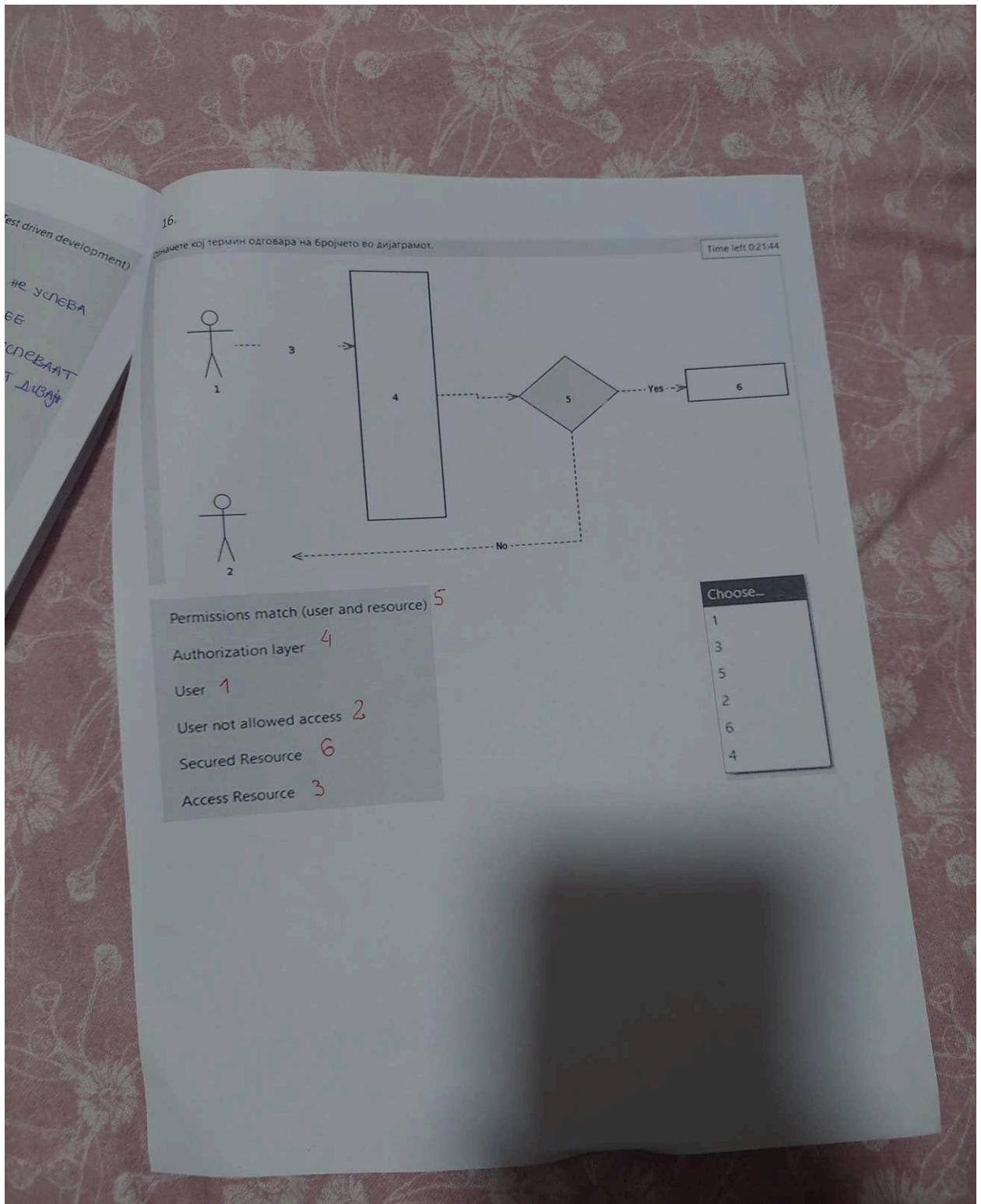
Select one or more:

- a. Корисникот е валиден доколку е пронајден во User/Credential Storage
- b. Корисникот секогаш се идентификува според корисничко име и лозинка
- c. Се дефинира кои ресурси смее одреден корисник да ги пристапи
- d. Се идентификува корисникот во рамките на системот

Решение/Објаснување 1:

Решение/Објаснување 2:

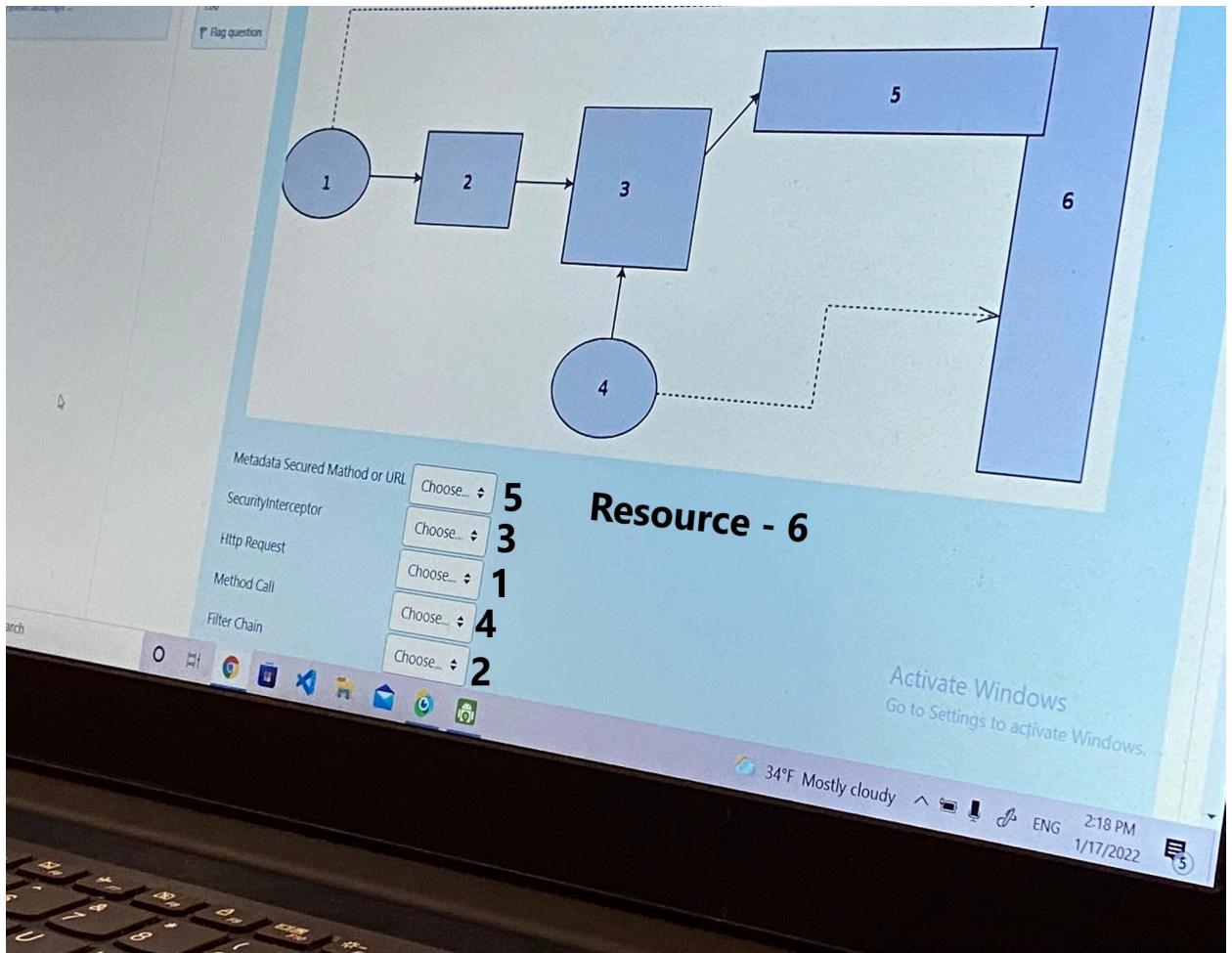
Решение/Објаснување 3:



Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:



Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

Question 14
Not yet answered
Marked out of 1.00
Flag question

Koj од следните тестови се користи за проверка на точноста на апликацијата, со ресурси слични на продукциската околина?

Select one:

a. User Acceptance Testing
 b. Performance Testing
 c. Unit Testing
 d. Integration Testing
 e. System Testing
 f. Stress Testing

Give your reasons

Решение/Објаснување 1: e. System testing

Решение/Објаснување 2:

Решение/Објаснување 3:

Question 6
Not yet answered
Marked out of 1.00
Flag question

Кај Функционалното Реактивно Програмирање кој од следните извори на податоци би се карактеризирале како **Hot Data Stream**?

Select one or more:

a. File data source
 b. Mouse events
 c. Network data stream
 d. Database cursor

Give your reasons

Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

Question 9
Answer saved
Marked out of 1.00
Flag question

Селектирајте ги точните тврдена за процесот на авторизација.

Select one or more:

a. Корисникот е добиен преку процесот на автентикација

b. Се дефинира кои ресурси смеа одреден корисник да ги пристапи

c. Може да се користат листи за контрола на пристап (Access Control List - ACL) во процесот на авторизација

d. Корисникот секогаш се идентификува според корисничко име и лозинка

e. Се идентификува корисникот во рамките на системот

Give your reasons

Previous page

Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

Question 7

Not yet answered

Marked out of 1.00

Flag question

Selenium е алатка за тестирање на:

Select one:

- a. Unit testing
- b. Тестирање на REST сервисите
- c. Front-end во web browser C
- d. Покриеност на кодот со тестови

Give your reasons

=260585&cmid=12381&page=6#

Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

Time left 0:2

6
out of question

Кај Функционалното Реактивно Програмирање кој од следниве извори на податоци би се карактеризирале како **Cold Data Stream**?

Select one or more:

a. Mouse events
 b. Network data stream
 c. File data source
 d. Database cursor

Give your reasons

14:22 17/01/2022

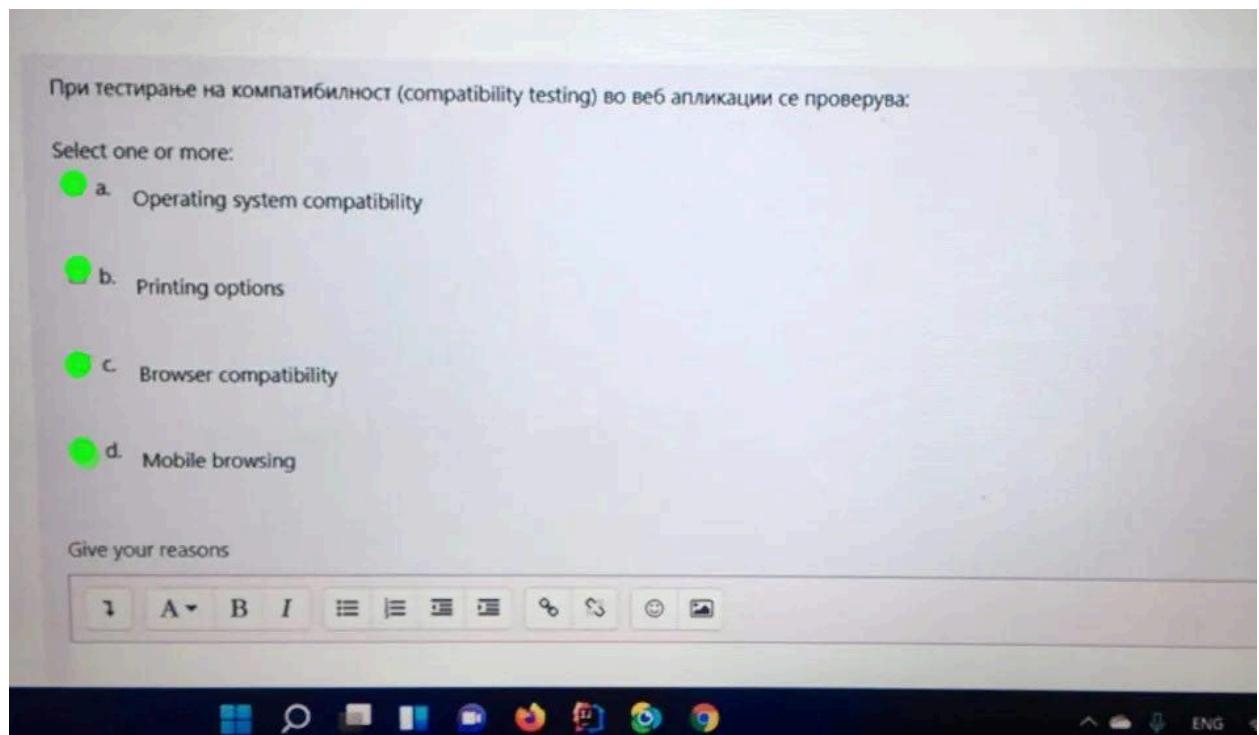
Решение/Објаснување 1: c i d

The correct choices for a Cold Data Stream would likely be:

- c. File data source
- d. Database cursor

Решение/Објаснување 2:

Решение/Објаснување 3:



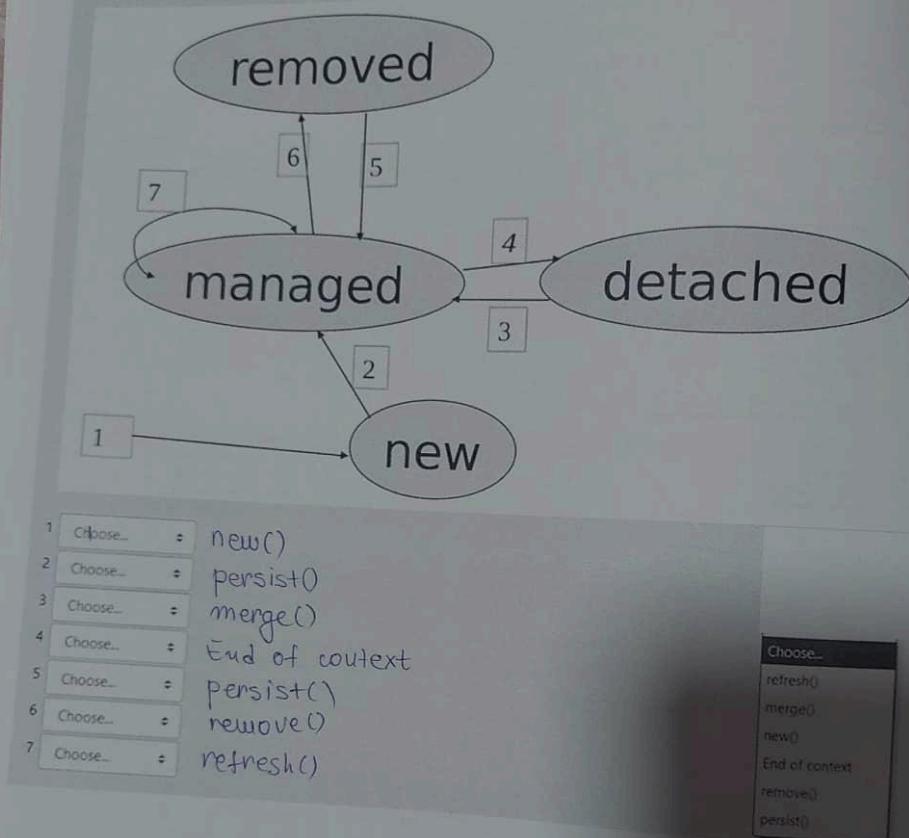
Решение/Објаснување 1: а и с

Решение/Објаснување 2:

Решение/Објаснување 3:

10.

Поврзете ги настапите од животниот циклус на ентитетите со броевите на сликата.



11.

Означете ги JPA анотациите

@Entity
public class Vacat

@Id
public Long

he

publi

h

Решение/Објаснување 2:

Решение/Објаснување 3:

6

out of
question

Кој од следниве парчиња код се валидни како REST сервис за добивање на податоци за сите вработени.

Забелешка: Сметајте дека соодветните мапирања и поставувања на ниво на класа и контролер се направени сојдентно на потребите.

Select one or more:

a.

```
@GetMapping  
private Mono<Employee> getAllEmployees() {  
    return employeeRepository.findAllEmployees();  
}
```

b.

```
@Bean  
RouterFunction<ServerResponse> getAllEmployeesRoute() {  
    return route(GET("/employees"),  
        req -> ok().body(  
            employeeRepository().findAllEmployees(), Employee.class));  
}
```

c.

```
@GetMapping  
private Flux<Employee> getAllEmployees() {  
    return employeeRepository.findAllEmployees();  
}
```

d.

```
@Bean  
RouterFunction<ServerResponse> composedRoutes() {  
    return  
        route(GET("/employees"),  
            req -> ok().body(  
                employeeRepository().findAllEmployees(), Employee.class))  
  
.and(route(GET("/employees/{id}"),  
        req -> ok().body(  
            employeeRepository().findEmployeeById(req.pathVariable("id")), Employee.class)))  
  
.and(route(POST("/employees/update"),  
        req -> req.body(toMono(Employee.class))  
            .doOnNext(employeeRepository()::updateEmployee)  
            .then(ok().build())));  
}
```

Give your reasons



Решение/Објаснување 1:

Решение/Објаснување 2:

Решение/Објаснување 3:

7

ed
out of
question

При веб тестирање, кои се предностите ако се тестира рендерирањето во веб browser-от.

Select one or more:

- a.
Well defined API and data structures – resilient
- b.
Can test for usability, performance, responsive design, etc.
- c.
Unit testing of key modules
- d.
Most realistic testing, validates what the end user experiences



Give your reasons



Решение/Објаснување 1: b, d

Решение/Објаснување 2:

Решение/Објаснување 3: