

Функционални захтеви

1. да би се апликација користила, потребно је имати налог на платформи. Стога је потребно имплементирати систем регистрације корисника.

2. само пријављени корисници могу да креирају форму. Форму могу попуњавати пријављени корисници, док је неке формуларе могуће попунити и без пријаве, зависно од тога да ли је то дозвољено у подешавању самог формулара.

3. пријављени корисници креирају формуларе задајући притом:

(а) име формулара

(б) опис формулара

(в) индикатор да ли форму могу попунити и непријављени корисници или је обавезно да буду пријављени при попуњавању исте

(г) листу питања (где се свако питање дефинише као обавезно или необавезно).

Свако питање има свој тип одговора, и то:

- кратак текстуални одговор - до 512 карактера
- дуг текстуални одговор - до 4096 карактера
- више понуђених одговора, само један се бира
- више понуђених одговора, бира се више одговора (могуће је задати број захтеваних одговора)
- један нумерички одговор. Задаје се листа бројева, или цела листа бројева или скала бројева са кораком. Нпр. захтев -4 до 20 уз корак 3 би генерисао следећу листу бројева: -4, -1, 2, 5, 8, 11, 14, 17, 20
- датум
- време

4. код сваког питања је могуће додати слику у питање или одговор (начин и стил приказа слика у оквиру питања и одговора се оставља вама на избор)

5. свако питање је могуће клонирати

6. свако питање је могуће изменити

7. свако питање је могуће обрисати

8. могуће је променити редослед питања

9. корисник може да подели линк за попуњавање форме

10. корисник може да дода колабораторе на формулар у улози посматрача или уредника, као и да уклони исте. Сви уредници, као и креатор могу да измене све у формулару (измене назив, опис, питања, клонирају и обришу иста), али и да закључају формулар (није могуће попунити такав формулар). Само власник формулара може да управља колабораторима на том формулару.

11. при пријави на сервис, учитавају се сви формулари код којих је корисник креатор или колаборатор

12. сви колаборатори на формулару могу да прегледају резултате и то:

- појединачне одговоре
- групне одговоре (груписане по питању)
- извезу питања и одговоре у `xlsx` фајл

2 Технички и нефункционални захтеви

- Апликација треба да буде имплементирана у форми микросервиса. Сегрегација сервиса и њихове одговорности се остављају студентским тимовима као задатак.

- Сав програмски код треба да буде похрањен у приватном репозиторијуму (једном или више њих) на платформи за верзионисање пројеката GitHub. Апликацију израдити у неком од следећих језика: Python, Java, C#, C++, PHP или JavaScript. За израду графичког интерфејса (сучеља), дозвољено је користити било које оквире и библиотеке.

- подаци о корисницима и формуларима треба да буду трајно сачувани, односно потребно их је складиштити у бази података. Студентима се оставља на избор који систем за управљање базом података ће користити.

- компоненте је потребно контејнеризовати (сервисне компоненте и базе података).

Сервиси треба да буду подигнути користећи Docker compose мултиконејнер алат или Kubernetes оркестрациони алат.

- потребно је аутоматизовати ток извршења континуалне интеграције (CI)
- потребно је покрити код тестовима - unit и интеграционим, и то:
 - unit тестовима (пишу чланови тима који имплементирају backend функционалности) покрити случајеве:
 - * регистрацију и CRUD операције над корисницима
 - * CRUD операције над формуларима
 - * преглед попуњених формулара (појединачни и агрегациони резултати)
 - интеграционим
 - * [UI] регистрација корисника
 - * [UI] пријава корисника
 - * [API] CRUD операције над корисницима
 - * [API] CRUD операције над формуларима
 - * [UI] претрага формулара по називу
 - * [API] попуњавање формулара. Дефинисати формулар који садржи све типове питања.
 - * [API] затварање формулара
 - * [API] додавање колаборатора на формулар
 - * [API] преглед попуњених формулара (појединачни и агрегациони резултати)
 - мануелно тестирати попуњавање формулара на UI и генерисање xlsx извештаја