Универзитет у Београду

Електротехнички факултет

Одсек СИ



Принципи софтверског инжењерства

Тестирање пројекта *hoceneko.com* – извештај

Студент:

Никола Миличевић 0387/17

Тим:

*Nebula*

Београд, школска година 2019/2020

**Тестирање веб апликације техникама црне кутије**

Веб апликација *hoceneko.com* тестирана је применом техника класе еквиваленције, комбиновано са граничним случајевима.

У табели 1 може се видети функционалности које ће бити истестиране.

Табела 1. Функционалности које се тестирају

|  |  |
| --- | --- |
| Редни број | Назив |
|  | Преглед свих објава |
|  | Филтрирање |
|  | Постављање објава |
|  | Уклањање објава |
|  | Моје објаве приказ |
|  | Прихватање и одбацивање пријаве на догађај |
|  | Одјава са догађаја |

У наставку ћу детаљно описати начин тестирања сваке функционалности. То ћу урадити тако што ћу за сваку функционалност приказати две табеле:

1. Табела - детаљан план тестирања са списком свих класа еквиваленције, при чему ће бити наведен редни број класе, назив и тип (да ли је у питању легална класа (лк) или нелегална класа (нк))
2. Табела - детаљан план тестирања са свим осмишљеним тест примерима, при чему ће бити наведен редни број тест примера, назив, кратак опис тест примера, поступак извршавања теста, предуслови и постуслови за извршење теста, као и начин имплементације.

**Пар напомена**

* У табели тест плана, верификационе тачке су обележене зеленом бојом.
* Сви тестови у Селениуму су названи по редном броју функционалности и редном броју у табели, тако да је могуће упоредо пратити извештај и тестове у Селениуму.
* Ако неки тест падне закључујемо да та функционалност има ману.

1. **Прва функционалност – Преглед свих објава**

Табела 2. Класе еквиваленције

|  |  |  |
| --- | --- | --- |
| Редни број | Назив | Тип |
|  | Страница са објавама се приказује | Лк |
|  | Страница са објавама се не приказује | Нк |
|  | Објаве на страници су из будућности тј. релевантне | Лк |

Табела 3. Тест план

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Редни бр. | Назив | Опис (покривене класе) | Поступак извршавања | Предуслови/ постуслови | Начин имплементације |
|  | Приказивање странице са свим објавама | 1,2,3 | 1. Налазимо се на сајту у роли госта 2. Кликнемо на „Све објаве“ 3. Проверимо у складу са базом да ли су све објаве које су актуелне излистане | Пре#1:  У улози госта смо  Пост#1: | Аутоматски + ручно |

1. **Друга функционалност – Филтрирање**

Табела 4. Класе еквиваленције

|  |  |  |
| --- | --- | --- |
| Редни број | Назив | Тип |
|  | Филтер исправно филтрира објаве по датуму | Лк |
|  | Филтер неисправно филтрира објаве по датуму | Нк |
|  | Филтер исправно филтрира објаве по месту | Лк |
|  | Филтер неисправно филтрира објаве по месту | Нк |
|  | Филтер исправно филтрира објаве по времену | Лк |
|  | Филтер неисправно филтрира објаве по времену | Нк |

Табела 5. Тест план

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Редни бр. | Назив | Опис (покривене класе) | Поступак извршавања | Предуслови/ постуслови | Начин имплементације |
| 1 | Филтер по датуму | 1,2 | 1. Идемо на страницу „Филтери“ 2. Бирамо датум између датума најраније и најкасније објаве у бази 3. Проверавамо да ли су излистане само објаве из будућности | Пре#1:  У бази постоје објаве са датумом у прошлости и у будућности | Аутоматски + ручно |
| 2 | Филтер по месту | 3,4 | 1. Идемо на страницу „Филтери“ 2. Бирамо место неке објаве која је у бази 3. Проверавамо да ли је исписана објава која садржи унет назив у свом месту | Пре#1:  У бази постоје објаве са датумом у прошлости и у будућности | Аутоматски + ручно |
| 3 | Филтер по времену коректно време | 5 | 1. Идемо на страницу „Филтери“ 2. Бирамо један датум за који постоји објава у бази 3. Бирамо време које је пре времена те објаве 4. Проверавамо да ли је исписана одговарајућа објава | Пре#1:  У бази постоје објаве са датумом у прошлости и у будућности | Аутоматски + ручно |
| 4 | Филтер по времену некоректно време | 6 | 1. Идемо на страницу „Филтери“ 2. Бирамо један датум за који постоји објава у бази 3. Бирамо време које је иза времена те објаве 4. Проверавамо да ли није исписана одговарајућа објава | Пре#1:  У бази постоје објаве са датумом у прошлости и у будућности | Аутоматски + ручно |
|  |  |  |  |  |  |

1. **Трећа функционалност – Постављање објава**

Табела 6. Класе еквиваленције

|  |  |  |
| --- | --- | --- |
| Редни број | Назив | Тип |
|  | Назив објаве унет | Лк |
|  | Назив објаве није унет | Нк |
|  | Категорија одабрана | Лк |
|  | Категорија није одабрана | Нк |
|  | Број потребних чланова није унет | Нк |
|  | Број потребних чланова није природан број | Нк |
|  | Број потребних чланова унет и природан број | Лк |
|  | Датум у прошлости | Нк |
|  | Датум у будућности | Лк |
|  | Време у прошлости | Нк |
|  | Време у будућности | Лк |
|  | Место унето | Лк |
|  | Место није унето | Нк |
|  | Опис унет | Лк |
|  | Опис није унет | Нк |

Табела 7. Тест план

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Редни бр. | Назив | Опис (покривене класе) | Поступак извшавања | Предуслови/ постуслови | Начин имплементације |
|  | Назив објаве | 1,2 | 1. Улогујемо се 2. Идемо на страницу „Додај објаву“ 3. Не попунимо поље „Назив“ 4. Попунимо сва остала поља исправно | Пре#1:  Улоговани смо | Аутоматски |
|  | Категорија објаве | 3,4 | 1. Улогујемо се 2. Идемо на страницу „Додај објаву“ 3. Не одаберемо ништа из листе „Категорија“ 4. Попунимо сва остала поља исправно | Пре#1:  Улоговани смо | Аутоматски |
|  | Број потребних чланова није унет | 5 | 1. Улогујемо се 2. Идемо на страницу „Додај објаву“ 3. Не попунимо поље „Број пот. чл.“ 4. Попунимо сва остала поља исправно | Пре#1:  Улоговани смо | Аутоматски |
|  | Број потребних чланова унет добро | 6,7 | 1. Улогујемо се 2. Идемо на страницу „Додај објаву“ 3. Унесемо природан број у поље „Број пот. чл.“ 4. Попунимо сва остала поља исправно | Пре#1:  Улоговани смо | Аутоматски |
|  | Датум објаве | 8,9 | 1. Улогујемо се 2. Идемо на страницу „Додај објаву“ 3. Одаберемо датум из прошлости 4. Попунимо сва остала поља исправно | Пре#1:  Улоговани смо | Аутоматски |
|  | Време објаве | 10,11 | 1. Улогујемо се 2. Идемо на страницу „Додај објаву“ 3. Одаберемо за данашни датум време из прошлости 4. Попунимо сва остала поља исправно | Пре#1:  Улоговани смо | Аутоматски |
|  | Место објаве | 12,13 | 1. Улогујемо се 2. Идемо на страницу „Додај објаву“ 3. Не попунимо поље „Место“ 4. Попунимо сва остала поља исправно | Пре#1:  Улоговани смо | Аутоматски |
|  | Опис објаве | 14,15 | 1. Улогујемо се 2. Идемо на страницу „Додај објаву“ 3. Не попунимо поље „Опис“ 4. Попунимо сва остала поља исправно | Пре#1:  Улоговани смо | Аутоматски |

1. **Четврта функционалност – Уклањање објава**

Табела 8. Класе еквиваленције

|  |  |  |
| --- | --- | --- |
| Редни број | Назив | Тип |
|  | Дугме за уклањање објаве се приказује Госту | Нк |
|  | Дугме за уклањање објаве се приказује Кориснику | Нк |
|  | Дугме за уклањање објаве се приказује Админу | Лк |
|  | Админ кликом на дугме уклони објаву брише објаву из базе | Лк |
|  | Админ кликом на дугме уклони објаву НЕ брише објаву из базе | Нк |

Табела 9. Тест план

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Редни бр. | Назив | Опис (покривене класе) | Поступак извшавања | Предуслови/ постуслови | Начин имплементације |
|  | Дугме се приказује Госту | 1 | 1. У улози госта смо на сајту 2. Идемо на неку објаву из листе свих објава 3. Проверавамо да ли је приказано дугме „Обриши објаву“ | Пре#1:  Излоговани смо | Ручно |
| 1. Дуг | Дугме се приказује Кориснику | 2 | 1. У улози корисника смо на сајту 2. Идемо на неку објаву из листе свих објава 3. Проверавамо да ли је приказано дугме „Обриши објаву“ | Пре#1:  Улоговани смо као корисник | Ручно |
| 1. Дуг | Дугме се приказује Админу | 3 | 1. У улози админа смо на сајту 2. Идемо на неку објаву из листе свих објава 3. Проверавамо да ли је приказано дугме „Обриши објаву“ | Пре#1:  Улоговани смо као админ | Ручно |

1. **Пета функционалност – Моје објаве приказ**

Табела 10. Класе еквиваленције

|  |  |  |
| --- | --- | --- |
| Редни број | Назив | Тип |
|  | Приказане све објаве за корисника из базе | Лк |
|  | Не приказују се објаве за корисника које су у бази | Нк |

Табела 11. Тест план

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Редни бр. | Назив | Опис (покривене класе) | Поступак извшавања | Предуслови/ постуслови | Начин имплементације |
|  | Приказивање објава | 1,2 | 1. Улогујемо се 2. Уверимо се да постоје објаве у бази за нас тј. улогованог корисника 3. Кликнемо на „Моје објаве“ 4. Проверимо конзистентност података | Пре#1:  Улоговани смо | Ручно |

1. **Шеста функционалност – Прихватање и одбацивање пријаве на догађај**

Табела 12. Класе еквиваленције

|  |  |  |
| --- | --- | --- |
| Редни број | Назив | Тип |
| 1. | Корисник се пријављује на догађај и чека одобрење | Лк |
| 2. | Корисник се пријављује на догађај и одмах се додаје на догађај | Нк |
| 3. | Корисник се појављује у пријавама на догађај код корисника чија је објава | Лк |
| 4. | Корисник се не појављује у пријавама на догађај код корисника чија је објава | Нк |

Табела 13. Тест план

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Редни бр. | Назив | Опис (покривене класе) | Поступак извшавања | Предуслови/ постуслови | Начин имплементације |
|  | Додавање на догађај | 1,2 | 1. Улогујемо се 2. Уђемо на догађај на који хоћемо да се пријавимо 3. Додамо се на догађај 4. Проверимо да се није увећао број пријављених у том тренутку | Пре#1:  Улоговани смо | Ручно |
|  | Прихватање корисника | 3,4 | 1. Улогујемо се као корисник који има пријаве за одобрење 2. Идемо на „Моје објаве“ 3. Идемо на „Пријажи пријаве“ од објаве на коју се очекују пријаве 4. Потврдимо или одбијемо корисника 5. Проверимо број пријављених у складу са претходном одлуком | Пре#1:  Улоговани смо | Ручно |
|  |  |  |  |  |  |

1. **Седма функционалност** – **Одјава са догађаја**

Табела 14. Класе еквиваленције

|  |  |  |
| --- | --- | --- |
| Редни број | Назив | Тип |
|  | Дугме за одјаву се појављује улогованом кориснику који је пријављен на догађај | Лк |
|  | Дугме за одјаву се појављује госту | Нк |
|  | Дугме за одјаву се појављује улогованом кориснику који није пријављен на догађај | Нк |

Табела 15. Тест план

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Редни бр. | Назив | Опис (покривене класе) | Поступак извшавања | Предуслови/ постуслови | Начин имплементације |
|  | Појава дугмета госту | 2 | 1. Уђемо на сајт у улози госта 2. Уђемо на неку објаву 3. Проверимо да ли постоји дугме за одјаву | Пре#1:  Нисмо улоговани | Ручно |
|  | Појава дугмета улогованом |  | 1. Уђемо на сајт у улози корисника или админа 2. Уђемо на неку објаву 3. Пријавимо се на ту објаву 4. Проверимо да ли постоји дугме за одјаву | Пре#1:  Улоговани смо | Ручно |
|  | Појава дугмета непријављеном кориснику | 3 | 1. Уђемо на сајт у улози корисника или админа 2. Уђемо на неку објаву 3. Проверимо да ли постоји дугме за одјаву | Пре#1:  Улоговани смо | Ручно |