

УНИВЕРЗИТЕТ У БЕОГРАДУ
ФАКУЛТЕТ ОРГАНИЗАЦИОНИХ НАУКА

ЗАВРШНИ РАД

**Развој софтверског система за
евиденцију филмских рецензија
употребом ASP.NET Core оквира**

Ментор

Студент

Проф. Др Саша Лазаревић, Милош Вујић, 0081/2017

Редовни професор

Београд, 2022. године

Садржај

1. УВОД	11
2. ПРЕГЛЕД КОРИШЋЕНИХ ТЕХНОЛОГИЈА	12
2.1. ОБЈЕКТНО ОРИЈЕНТИСАНО ПРОГРАМИРАЊЕ (ООП)	12
2.2. C#	13
2.2.1. Еволуција програмског језика C#	13
C# 1.0 – Управљани код	14
C# 2.0 – Генерички типови	14
C# 3.0 – Упит интегрисан у језик	14
C# 4.0 – Динамички типови	15
C# 5.0 – Асинхроно програмирање	15
C# 6.0 – Компајлер као сервис	15
C# 7.0	15
2.3. .NET	16
2.3.1. .NET архитектура и компоненте	18
2.4. ASP.NET Core оквир	19
2.5. MVC ДИЗАЈН ПАТЕРН	21
2.5.1. ASP.NET Core MVC архитектура	21
Одговорности <i>Model</i> -а	22
Одговорности <i>View</i> -а	22
Одговорности контролера	23
2.6. RAZOR	24
2.7. REPOSITORY ОБРАЗАЦ	25
2.8. JAVASCRIPT	26
2.8.1. Типови променљивих	28
2.9. AJAX	30
Традиционална метода	31
AJAX метода	31
2.10. BOOTSTRAP	32
2.11. ENTITY FRAMEWORK CORE	33
2.11.1. Модел	34
2.12. ПОСТАВЉАЊЕ УПИТА	34
2.12.1. LINQ – УПИТИ	36
3. СТУДИЈСКИ ПРИМЕР	37

4.	ПРИКУПЉАЊЕ КОРИСНИЧКИХ ЗАХТЕВА	38
4.1.	ВЕРБАЛНИ ОПИС МОДЕЛА	38
4.2.	СПЕЦИФИКАЦИЈА ЗАХТЕВА ПОМОЋУ СЛУЧАЈЕВА КОРИШЋЕЊА	39
	СК1: Случај коришћења – Пријава на систем	41
	СК2: Случај коришћења – Креирање филма	42
	СК3: Случај коришћења – Креирање стриминг сервиса	43
	СК4: Случај коришћења – Креирање особља	44
	СК5: Случај коришћења – Претраживање филмова	45
	СК6: Случај коришћења – Промена података стриминг сервиса	47
	СК7: Случај коришћења – Промена података особља	49
	СК8: Случај коришћења – Брисање филма	51
	СК9: Случај коришћења – Оцењивање филма	53
	СК10: Случај коришћења – Претраживање особља	55
5.	АНАЛИЗА	57
5.1.	ПОНАШАЊЕ СОФТВЕРСКОГ СИСТЕМА - СИСТЕМСКИ ДИЈАГРАМИ СЕКВЕНЦИ	57
	ДС1: Дијаграм секвенци случаја коришћења – Пријава на систем	57
	ДС2: Дијаграм секвенци случаја коришћења – Креирање филма	59
	ДС3: Дијаграм секвенци случаја коришћења – Креирање стриминг сервиса	62
	ДС4: Дијаграм секвенци случаја коришћења – Креирање особља	64
	ДС5: Дијаграм секвенци случаја коришћења – Претраживање филмова	66
	ДС6: Дијаграм секвенци случаја коришћења – Промена података стриминг сервиса	68
	ДС7: Дијаграм секвенци случаја коришћења – Промена података особља	72
	ДС8: Дијаграм секвенци случаја коришћења – Брисање филма	76
	ДС9: Дијаграм секвенци случаја коришћења – Оцењивање филма	80
	ДС10: Дијаграм секвенци случаја коришћења – Претраживање особља	84
	Списак системских операција	86
5.2.	ПОНАШАЊЕ СОФТВЕРСКОГ СИСТЕМА – ДЕФИНИСАЊЕ УГОВОРА О СИСТЕМСКИМ ОПЕРАЦИЈАМА	88
5.3.	СТРУКТУРА СОФТВЕРСКОГ СИСТЕМА-КОНЦЕПТУАЛНИ МОДЕЛ	96
6.	ФАЗА ПРОЈЕКТОВАЊА	98
6.1.	ПРОЈЕКТОВАЊЕ КОРИСНИЧКОГ ИНТЕРФЕЈСА	99
	СК1: Случај коришћења – Пријава на систем	100
	СК2: Случај коришћења – Креирање филма	103
	СК3: Случај коришћења – Креирање стриминг сервиса	105

СК4: Случај коришћења – Креирање особља	108
СК5: Случај коришћења – Претраживање филмова	112
СК6: Случај коришћења – Промена података стриминг сервиса	114
СК7: Случај коришћења – Промена података особља	122
СК8: Случај коришћења – Брисање филма	129
СК9: Случај коришћења – Оцењивање филма	132
СК10: Случај коришћења – Претраживање особља	137
6.2. ПРОЈЕКТОВАЊЕ АПЛИКАЦИОНЕ ЛОГИКЕ	142
6.2.1. Пројектовање контролера апликационе логике	142
6.2.2. Пројектовање пословне логике	142
6.2.3. Пројектовање слоја приступа подацима	153
6.3. ПРОЈЕКТОВАЊЕ СКЛАДИШТА ПОДАТАКА	155
6.4. КОНАЧАН ИЗГЛЕД АРХИТЕКТУРЕ СОФТВЕРСКОГ СИСТЕМА	164
7. ИМПЛЕМЕНТАЦИЈА	166
7.1. СТРУКТУРА СОФТВЕРСКОГ РЕШЕЊА	166
7.2. ИМПЛЕМЕНТАЦИЈА ПОСЛОВНЕ ЛОГИКЕ	169
7.3. ИМПЛЕМЕНТАЦИЈА СЛОЈА ПРИСТУПА ПОДАЦИМА	179
7.4. ИМПЛЕМЕНТАЦИЈА ПРЕЗЕНТАЦИОНОГ СЛОЈА	182
СК1: Случај коришћења – Пријава на систем	182
СК2: Случај коришћења – Креирање филма	185
СК3: Случај коришћења – Креирање стриминг сервиса	187
СК4: Случај коришћења – Креирање особља	190
СК5: Случај коришћења – Претраживање филмова	194
СК6: Случај коришћења – Промена података стриминг сервиса	196
СК7: Случај коришћења – Промена података особља	204
СК8: Случај коришћења – Брисање филма	211
СК9: Случај коришћења – Оцењивање филма	214
СК10: Случај коришћења – Претраживање особља	219
8. ТЕСТИРАЊЕ	224
9. ЗАКЉУЧАК	225
10. ЛИТЕРАТУРА	226

Списак слика

Слика 1. .NET 5	16
Слика 2. Визуелни преглед Опште језичке инфраструктуре.....	17
Слика 3. .NET Core	20
Слика 4. Интеракције у MVC апликацији	22
Слика 5. Поређење између традиционалне методе (лево) и коришћењем AJAX методе (десно).....	31
Слика 6. AJAX метода.....	32
Слика 7. Пример поделе странице у <i>Bootstrap</i> мрежу.....	33
Слика 8. Дијаграм случаја коришћења за корисника	40
Слика 9. ДС Пријава на систем	58
Слика 10. ДС неуспешна пријава	58
Слика 11. ДС креирање филма	60
Слика 12. ДС неуспешно креирање филма	61
Слика 13. ДС креирање стриминг сервиса	62
Слика 14. ДС неуспешно креирање стриминг сервиса	63
Слика 15. ДС креирање особља.....	64
Слика 16. ДС неуспешно креирање особља	65
Слика 17. ДС претрага филма.....	66
Слика 18. ДС неуспешна претрага филма	67
Слика 19. ДС неуспешно учитавање филма.....	67
Слика 20. ДС Измена стриминг сервиса.....	69
Слика 21. ДС неуспешна претрага стриминг сервиса	70
Слика 22. ДС неуспешно учитавање стриминг сервиса.....	70
Слика 23. ДС неуспешна измена стриминг сервиса.....	71
Слика 24. ДС измена особља	73
Слика 25. ДС неуспешна претрага особља.....	74
Слика 26. ДС неуспешно учитавање особља	74
Слика 27. ДС неуспешна измена особља.....	75
Слика 28. ДС брисање филма	76
Слика 29. ДС неуспешна претрага филмова	77
Слика 30. ДС неуспешно учитавање филма.....	78
Слика 31. ДС неуспешно брисање филма	78
Слика 32. ДС оцењивање филма	81

Слика 33. ДС неуспешна претрага филмова	82
Слика 34. ДС неуспешно читавање филма.....	82
Слика 35. ДС неуспешно додавање оцене.....	83
Слика 36. ДС претрага особља	84
Слика 37. ДС неуспешна претрага особља.....	85
Слика 38. ДС неуспешно читавање особља	85
Слика 39. Концептуални модел структуре софтверског система	96
Слика 40. Структура и понашање софтверског система.....	97
Слика 41. Архитектура софтверског система	98
Слика 42. Повезаност контролера и корисничког интерфејса у архитектури софтверског система.....	99
Слика 43. Форма за пријаву на систем	100
Слика 44. Успешна пријава на систем (преусмеравање на почетну страницу).....	101
Слика 45. Неуспешна пријава на систем	102
Слика 46. Форма за креирање новог филма	103
Слика 47. Страница успешно креираног филма	104
Слика 48. Неуспешно креирање филма	104
Слика 49. Форма за креирање стриминг сервиса.....	105
Слика 50. Успешно креирање стриминг сервис (преусмеравање на страницу новокреираног стриминг сервиса).....	106
Слика 51. Неуспешно креирање стриминг сервиса.....	107
Слика 52. Форма за креирање особља	109
Слика 53. Успешно креирано особље (преусмеравање на страницу особља)	110
Слика 54. Неуспешно креирање особља.....	111
Слика 55. Страница са филмовима	112
Слика 56. Систем је успешно прочитао филм.....	113
Слика 57. Систем није пронашао филмове по задатој вредности.....	113
Слика 58. Систем не може да прочита филм.....	114
Слика 59. Страница са стриминг сервисима	115
Слика 60. Приказ претражених стриминг сервиса	116
Слика 61. Успешно прочитан стриминг сервис	117
Слика 62. Успешно измењен стриминг сервис	118
Слика 63. Систем није пронашао стриминг сервис по задатој вредности	119
Слика 64. Неуспешно читавање стриминг сервиса	120
Слика 65. Неуспешна измена стриминг сервиса	121
Слика 66. Страница са особљем	122

Слика 67. Систем је нашао особље по задатој вредности.....	123
Слика 68. Успешно учитано особље	124
Слика 69. Успешно измењено особље	125
Слика 70. Систем није пронашао особље по задатој вредности	126
Слика 71. Неуспешно учитавање особља.....	127
Слика 72. Систем не може да запамти особље	128
Слика 73. Страница са филмовима	129
Слика 74. Позив система да обрише филм.....	130
Слика 75. Неуспешно брисање филма	131
Слика 76. Страница са филмовима	132
Слика 77. Филм	133
Слика 78. Форма за оцењивање филма	134
Слика 79. Неуспешно учитавање филма	135
Слика 80. Неисправно попуњена форма.....	136
Слика 81. Страница са особљем	137
Слика 82. Систем је нашао особље по задатој вредности.....	138
Слика 83. Страница нађеног особља.....	139
Слика 84. Не постоји особа са задатим именом.....	140
Слика 85. Неуспешно учитавање особља.....	141
Слика 86. Приказ животног циклуса MVC захтева.....	142
Слика 87. ДС – Уговор: GetAll(List<StreamingService>)	143
Слика 88. ДС – Уговор: GetAll(List<Studio>)	143
Слика 89. ДС – Уговор: GetAll(List<Actor>).....	144
Слика 90. ДС – Уговор: GetAll(List<Personnel>)	144
Слика 91. ДС – Уговор: GetAll(List<Genre>)	145
Слика 92. ДС – Уговор: Add(Movie)	145
Слика 93. ДС – Уговор: Add(StreamingService)	146
Слика 94. ДС – Уговор: GetAll(Country)	146
Слика 95. ДС – Уговор: Add(Personnel).....	147
Слика 96. ДС – Уговор: GetAll(List<Movie>)	147
Слика 97. ДС – Уговор: Find(Criteria, List<Movie>).....	148
Слика 98. ДС – Уговор: GetSingle(Movie)	148
Слика 99. ДС – Уговор: Find(Criteria, List<StreamingService>).....	149
Слика 100. ДС – Уговор: GetSingle(StreamingService)	149
Слика 101. ДС – Уговор: Update(StreamingService).....	150

Слика 102. ДС – Уговор: Find(Criteria, List<Personnel>)	150
Слика 103. ДС – Уговор: GetSingle(Personnel).....	151
Слика 104. ДС – Уговор: Update(Personnel)	151
Слика 105. Пројектовање слоја приступа подацима	154
Слика 106. Коначна архитектура софтверског система – први део.....	164
Слика 107. Коначна архитектура софтверског система – други део	165
Слика 108. Библиотека доменских класа.....	166
Слика 109. Библиотека класа DataAccesLayer	167
Слика 110. Пројекат Filmofil.....	167
Слика 111. Фолдер wwwroot.....	167
Слика 112. Фолдер Controllers	168
Слика 113. Фолдер Models	168
Слика 114. Фолдер Views.....	168
Слика 115. <i>Acting</i>	179
Слика 116. <i>Actor</i>	179
Слика 117. <i>Country</i>	179
Слика 118. <i>Genre</i>	179
Слика 119. <i>MovieGenre</i>	179
Слика 120. <i>Movie</i>	180
Слика 121. <i>Person</i>	180
Слика 122. <i>Personnel</i>	180
Слика 123. <i>Position</i>	180
Слика 124. <i>Review</i>	180
Слика 125. <i>StreamingService</i>	181
Слика 126. <i>Studio</i>	181
Слика 127. Форма за пријаву на систем	182
Слика 128. Успешна пријава на систем (преусмеравање на почетну страницу).....	183
Слика 129. Неуспешна пријава на систем	184
Слика 130. Форма за креирање новог филма	185
Слика 131. Страница успешно креираног филма	186
Слика 132. Неуспешно креирање филма	186
Слика 133. Форма за креирање стриминг сервиса.....	187
Слика 134. Успешно креирање стриминг сервис (преусмеравање на страницу новокреираног стриминг сервиса).....	188
Слика 135. Неуспешно креирање стриминг сервиса.....	189
Слика 136. Форма за креирање особља	191

Слика 137. Успешно креирано особље (преусмеравање на страницу особља)	192
Слика 138. Неуспешно креирање особља.....	193
Слика 139. Страница са филмовима	194
Слика 140. Систем је успешно прочитао филм.....	195
Слика 141. Систем није пронашао филмове по задатој вредности.....	195
Слика 142. Систем не може да прочита филм.....	196
Слика 143. Страница са стриминг сервисима	197
Слика 144. Приказ претражених стриминг сервиса	198
Слика 145. Успешно учитан стриминг сервис	199
Слика 146. Успешно измењен стриминг сервис	200
Слика 147. Систем није пронашао стриминг сервис по задатој вредности	201
Слика 148. Неуспешно читавање стриминг сервиса	202
Слика 149. Неуспешна измена стриминг сервиса	203
Слика 150. Страница са особљем	204
Слика 151. Систем је нашао особље по задатој вредности.....	205
Слика 152. Успешно читано особље	206
Слика 153. Успешно измењено особље.....	207
Слика 154. Систем није пронашао особље по задатој вредности	208
Слика 155. Неуспешно читавање особља.....	209
Слика 156. Систем не може да запамти особље	210
Слика 157. Страница са филмовима	211
Слика 158. Позив система да обрише филм.....	212
Слика 159. Неуспешно брисање филма	213
Слика 160. Страница са филмовима	214
Слика 161. Филм	215
Слика 162. Форма за оцењивање филма.....	216
Слика 163. Неуспешно читавање филма	217
Слика 164. Неисправно попуњена форма.....	218
Слика 165. Страница са особљем	219
Слика 166. Систем је нашао особље по задатој вредности.....	220
Слика 167. Страница нађеног особља.....	221
Слика 168. Не постоји особа са задатим именом.....	222
Слика 169. Неуспешно читавање особља.....	223

Списак табела

Табела 1. Persons	156
Табела 2. Country.....	156
Табела 3. Personnel.....	157
Табела 4. Actor.....	157
Табела 5. Acting.....	158
Табела 6. Positions	158
Табела 7. Movies.....	159
Табела 8. Genres	160
Табела 9. MovieGenre.....	160
Табела 10. StreamingService	161
Табела 11. Studio	162
Табела 12. Reviews.....	163
Табела 13. AspNetUsers	163

1. УВОД

Савремени развој софтвера укључује у великој мери развој веб апликација различитих намена. Један од мотива за избор теме овог дипломског рада је управо теоријско и практично изучавање проблематике модуларног развоја веб апликација кроз примену сервисно-орјентисане архитектуре.

Приликом израде дипломског рада направљена је апликација намењена за евиденцију података о филмовима коришћењем упрошћене Ларманове методе. Описана је идентификација корисничких захтева, приказана је фаза анализе, пројектовања и тестирања апликације.

Пре свега, на почетку рада биће описане технологије који се користе за развој веб апликација, а то су: HTML, C# и JavaScript. Велики удео у теоријском делу овог рада има опис ASP.NET оквира са фокусом на MVC образац. MVC (енг. *eng. model-view-controller*) је образац који се данас сматра за један од најкоришћенијих узора за израду апликација.

У оквиру студијског примера идентификовано је десет случајева коришћења, за које су у фази анализе дефинисана одговарајућа структура и понашање софтверског система – пословна логика. У фази пројектовања, пројектују се апликациона логика, складиште података и изглед корисничког интерфејса. Фаза имплементације резервисана је за приказ софтверског система имплементираних помоћу програмског језика C# и *Microsoft SQL Server* система за управљање базом података. У слоју приступа подацима имплементирани су *Unit of Work* и *Repository* патерни.

2. ПРЕГЛЕД КОРИШЋЕНИХ ТЕХНОЛОГИЈА

Као оквир за имплементацију софтверског система коришћен је ASP.NET Core којег извршава .NET Core оквир који је део .NET екосистема. Као програмски језик у оквиру .NET екосистема коришћен је C#. Креирана веб апликација прати MVC архитектуру. За израду корисничког интерфејса употребљене су технологије и у оквиру Razor страница – Bootstrap, JavaScript, JQuery и Ajax.

2.1. ОБЈЕКТНО ОРИЈЕНТИСАНО ПРОГРАМИРАЊЕ (ООП)

Програмски језик се може назвати објектно оријентисан ако пружа четири основне могућности (Taher, 2019):

- Наслеђивање: концепт у Објектно оријентисаном програмирању (ООП), који означава да класа може имати своје изведене класе – подкласе
- Енкапсулација: Енкапсулација представља нивое заштите атрибута или функција у оквиру једне класе. Постоји више нивоа заштите:

1. *Public* - одређена функција/атрибут је доступна било где у оквиру програма, односно и изван саме класе у којој су дефинисани

2. *Protected* - функција/атрибут је доступна у оквиру класе и њених изведених класа

3. *Private* - функција/атрибут је доступна само у оквиру класе;

Ова три нивоа заштите су најчећи за ООП језике, али су у коришћеном C#-у присутни и:

4. *Internal* - функција/атрибут је доступна у оквиру асемблија

5. *Protected internal* - функција/атрибут је доступна у оквиру асемблија или у оквиру класе и њених изведених класа.

6. *Private protected* - функција/атрибут је доступна у оквиру класе и њених изведених класа које су у оквиру истог асемблија

- Апстракција: означава својство које служи за поједностављивање карактеристика стварног објекта, занемарују се детаљи и узимају се само заједничке карактеристике класе у зависности од потреба програма.
- Полиморфизам: представља концепт према коме се метода понаша различито у различитим објектима.

2.2. C#

C# је објектно оријентисан (ООП - *Object-oriented programming*), строго-типизирани, језик опште намене који је изграђен на врху *.NET* оквира који је развио *Microsoft* и који су одобриле Европско удружење произвођача рачунара (*ECMA - European Computer Manufacturers Association*) и Међународна организација за стандарде (*ISO - International Standards Organization*). Направљен је да ради на заједничкој језичкој инфраструктури и може да комуницира са било којим другим језицима који су изграђени на истој архитектури. Инспирисан C++, језик је богат у испоруци најбољих апликација без руковања превише сложености кода. (Тагер, 2019)

2.2.1. Еволуција програмског језика C#

C# је био један језика који се највише развио у последње време. Овај језик отвореног кода, је у последње време претрпео низ измена које су биле усмерене ка повећавању једноставности и олакшавању рада програмера. Нека од главних побољшања која су предложена за језик укључује *Generics*, *LINQ*, *Dynamics* и *async/await* патерн. (Тагер, 2019). У поглављу 2.11. ће бити више речи о *LINQ* упитима.

Фазе развоја:

- C# 1.0 – Управљани код
- C# 2.0 – Генерички типови
- C# 3.0 – Упит интегрисан у језик

- C# 4.0 – Динамички типови
- C# 5.0 – Асинхроно програмирање
- C# 6.0 – Компајлер као сервис
- C# 7.0 и више

C# 1.0 – Управљани код

C# 1.0 објављен је 2002. године и укључивао је све важне карактеристике савременог објектно оријентисаног програмског језика. У моменту објављивања био је јако налик на програмски језик *Java*.

C# 2.0 – Генерички типови

Генерички параметар представља тип вредности који ће се ускладиштити у колекцији. Сви типови који имају могућност задавања генеричких параметара се зову генерички типови (енгл. *Generics*).

Употреба генеричких типова доноси мноштво бенефита. Неки од њих су:

- Повећан интегритет типова – не постоји могућност уноса погрешног типа вредности. Генеричка колекција има генерички параметар који гарантује да ће колекција садржати само тип који смо задали
- Боља искоришћеност кода – могућност прављења сопствених генеричке типов омогућава знатно повећање фикасности написаног кода.
- Постављање ограничења генеричких типова – могућност декларисања ограничења на врсте типова који се могу задати

C# 3.0 – Упит интегрисан у језик

Упит интегрисан у језик (енгл. *Language Intergrated Query – LINQ*). јесте језик структурираних упита за претраживање и локалних колекција објекта, али и удаљених извора података на такав начин да не нарушава безбедност података. Поред *LINQ* упита C# 3.0 је са собом донео и измене попут ламбда израза и анонимних типова.

C# 4.0 – Динамички типови

C# 4.0 је објављен 2010. године и фокусиран је на побољшање интероперабилности са динамичким језицима као што су F# и Python. Главни новитет представљали су динамички типови.

Динамички типови користе се за избегавање провере типа у време компајлирања. Компајлер не проверава тип променљиве динамичког типа у време компајлирања, уместо тога, преводилац добија тип током извршавања. Променљива динамичког типа се креира помоћу кључне речи *dynamic*.

C# 5.0 – Асинхроно програмирање

Асинхроно програмирање представља посебно компликован задатак у сваком програмском језику. Када се баве овим проблемом, програмери морају да наиђу на многе сложености које смањују читљивост и могућност одржавања кода. Са функцијом *async/await* у језику C#, програмирање на асинхрони начин је једноставно као и синхроно програмирање. Програмирање је поједностављено, са свим сложеностима које интерно рукују компајлер и оквир.

C# 6.0 – Компајлер као сервис

Microsoft је радио на томе да омогући да се неки делови изворног кода компајлера буду део заједнице отвореног кода. Сходно томе, као програмер, могу се постављати упити компајлеру о неким од његових интерних принципа рада. C# 6.0 је увео велики број библиотека које омогућавају програмеру да стекне увид у компајлер, повезивач, синтаксно стабло програма, и тако даље. Иако су ове карактеристике дуго развијане као *Roslyn* пројекат, *Microsoft* их је коначно јавно објавио.

C# 7.0

C# 7.0 је објављен у марту 2017. и фокусиран је на додавање функционалних карактеристика језику као што су *tuples*, *out* варијабле, шаблони подударача (eng. *pattern matching*), као и мања побољшања језик. (Price, 2019)

2.3. .NET

.NET је платформа отвореног кода и платформа за развој многих врста апликација. Дизајнирана од стране *Microsoft*-а, платформа подржава више програмских језика и библиотека за прављење веб, мобилних, десктоп, IoT (*Internet of Things*) апликација и још много тога.

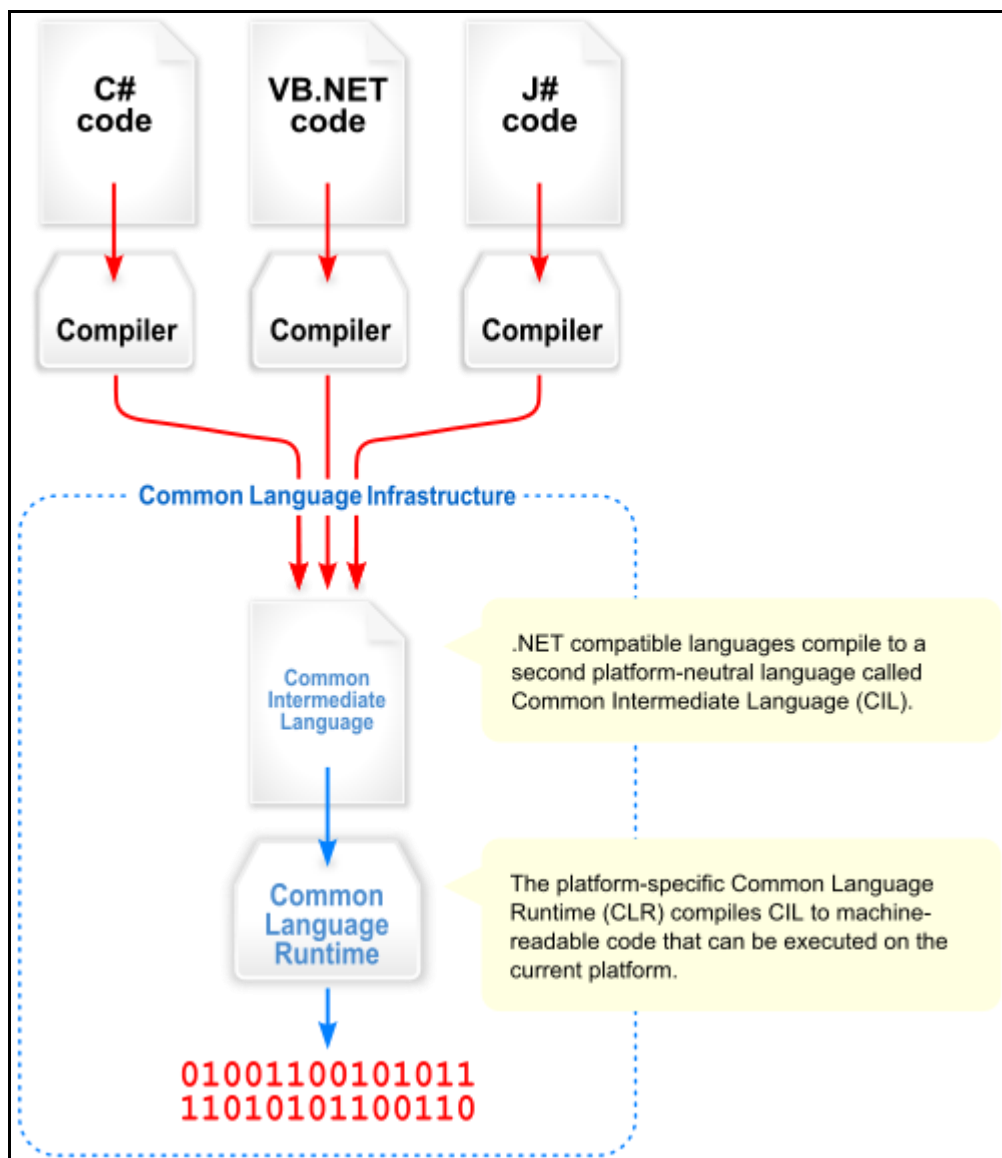


Слика 1. .NET 5

Чак и ако се C# сматра главним програмским језиком за .NET, може се користити много других језика по сопственом избору. Језици које *Microsoft* директно подржава су:

- C#: Савремени објектно оријентисани програмски језик који припада породици језика C.
- F#: функционално оријентисан програмски језик, члан породице ML (енгл. *Meta Language*) језика. Такође подржава објектно оријентисану парадигму програмирања.
- *Visual Basic*: историјски *Microsoft*-ов програмски језик. Постао је потпуни објектно оријентисани програмски језик у контексту .NET-а.

.NET подржава Општа језичка инфраструктура (*Common Language Infrastructure (CLI)*), тако да се изворни код компајлира у *Common Intermediate Language (CIL)*, независно од програмског језика који је коришћен. Ово гарантује велику интероперабилност између језика на платформи. (Chiarelli, 2021). Општа језичка инфраструктура је отворена спецификација развијена од стране компаније *Microsoft* који описује извршни код и окружење за његово извршење које чини језгро *Microsoft .NET Framework*-а и имплементације бесплатног софтвера отвореног кода *Mono* и *Portable .NET*. Спецификација дефинише окружење које омогућава да више језика на високом нивоу могу да се користе на другим рачунарским платформама без писања програма за одређену софтверску архитектуру.



Слика 2. Визуелни преглед Опште језичке инфраструктуре

2.3.1. .NET архитектура и компоненте

.NET архитектура је заснована на две главне компоненте:

1. *CoreCLR*: Ово је .NET *runtime*. Одговоран је за извршавање CLI програма и укључује just-in-time компајлер.
2. *CoreFX*: *API (Application Programming Interface)* платформе који имплементира CLI стандардне библиотеке, односно скуп библиотека које пружају најчешће функционалности, као што су управљање системом датотека, руковање изузетима, мрежна комуникација, нити, рефлексција итд.

Поред основних компоненти, постоје и различити оквири модела апликације, односно библиотеке које нуде подршку за развој различитих типова апликација. Па тако постоје:

- *ASP.NET*: Оквир који вам омогућава да правите веб апликације и веб API-је.
- *Windows Presentation Foundation (WPF)*: графички кориснички интерфејс за *Windows* десктоп апликације.
- *Xamarin*: Оквир за изградњу мобилних, ТВ и десктоп апликација.
- *Blazor*: Оквир за прављење клијентских веб апликација коришћењем C#. Такође омогућава прављење клијентских веб апликација у WebAssembly коду.
- *ML.NET*: Оквир за машинско учење који поједностављује интеграцију модела машинског учења у .NET апликације.

.NET такође нуди и подршку за већину уобичајених програмских задатака: од управљања датотекама до мрежне комуникације, од безбедности до приступа бази података. На пример, на страни умрежавања, подржава програмирање сокета и HTTP комуникацију. (Chiarelli, 2021)

2.4. ASP.NET Core оквир

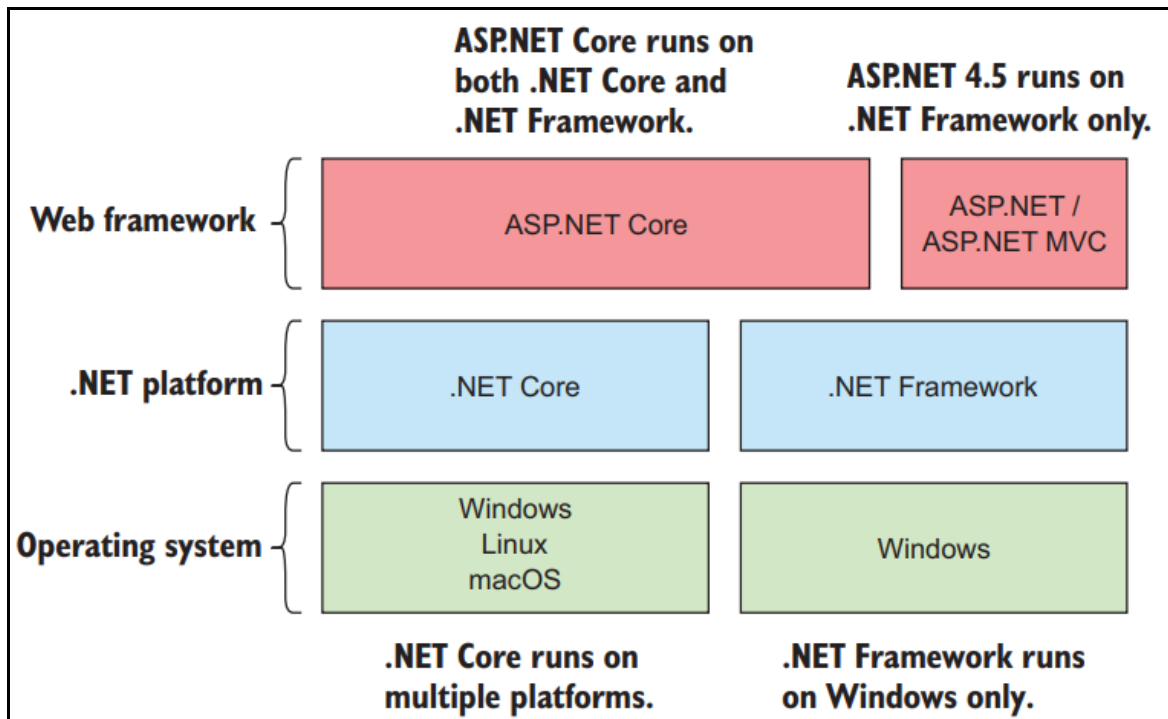
ASP.NET Core је најновија еволуција *Microsoft*-овог популарног ASP.NET веб оквира. Недавне верзије ASP.NET-а су имале многа инкрементална побољшања, фокусирајући се на високу продуктивност програмера и давање приоритета компатибилности уназад. ASP.NET Core подржава тај тренд тако што прави значајне архитектонске промене које преиспитују начин на који је веб оквир дизајниран и изграђен. ASP.NET Core много дугује свом ASP.NET наслеђу и многе функције су пренете из претходних верзија, али ASP.NET Core је нови оквир. Читав технолошки стек је поново написан, укључујући и веб оквир и основу платформе. У сржи промена је филозофија по којој ASP.NET треба да буде у стању да држи корак и предњачи у неким аспектима у односу на друге модерне оквире. (Lock, 2021)

Развој ASP.NET Core-а је мотивисан жељом да се креира веб оквир са четири главна циља:

- Да се покреће и развија на више платформи
- Да има модуларну архитектуру ради лакшег одржавања
- Да се развија у потпуности као софтвер отвореног кода
- Да буде применљив на тренутне трендове у развоју веба

Да би постигао све ове циљеве, *Microsoft*-у је била потребна платформа која би могла да обезбеди основне библиотеке за креирање основних објеката као што су листе и речници, као и извођење, на пример, једноставних операција са датотекама.

До ове тачке развоја, ASP.NET је био фокусиран и зависан од .NET оквира само за *Windows*. За ASP.NET Core, *Microsoft* је креирао платформу која ради и на *Windows*-у, *Linux*-у и *macOS*-у који се називају .NET Core, као што је приказано на слици.



Слика 3. .NET Core

.NET Core дели многе исте API-је као .NET оквир, али је мањи и тренутно имплементира само подскуп функција које .NET оквир пружа, са циљем обезбеђивања једноставнијег модела имплементације и програмирања. То је потпуно нова платформа, а не форк .NET оквира, иако користи слични код за многе његове API-је. Само са .NET Core, могуће је направити апликације за конзоле које покрећу више платформи. Једина платформа која се активно развија и данас је управо .NET Core.

ASP.NET Core је дизајниран од нуле да подржава уметање зависности између класа и њихових зависности. ASP.NET Core умеће објекте класа зависности кроз конструктор или метод користећи уграђени IoC (*Inversion of control*) контејнер.

2.5. MVC ДИЗАЈН ПАТЕРН

Model — View — Controller (MVC) је архитектурални патерн који се користи у развоју софтвера. У модерним сложеним апликацијама које приказују кориснику огромне количине података добра пракса је раздвајање кода који се бави подацима од оног који се бави интерфејсом, а све са циљем да развој оба постане лакши и једноставнији. MVC дели све што једна апликација треба да ради на три дела, са циљем јасно разграничења задужења.

MVC подразумева постојање три врсте компоненти:

- *Model* - садржи податке у облику погодном за конкретну примену притом дефинишући шта све можемо да урадимо са датим подацима.
- *View* – приказује кориснику податке из модела у формату погодном за интеракцију, најчешће као компоненту корисничког интерфејса. У оквиру једне апликације може постојати више *View*-ова прилагођених различитим ситуацијама који приказују податке из истог модела.
- *Controller* - координише моделе и *view*-ове, углавном на основу корисничког уноса.

Поред MVC патерна, у данашњој пракси често се користе и архитектуре настале по узору на овај - MVP (*Model — View — Presenter*) и MVVM (*Model — View — View — Model*). Кључна разлика између поменутих патерна је зависност сваког слоја од других слојева, као и колико су чврсто повезани једни са другима. (Vidović, 2017)

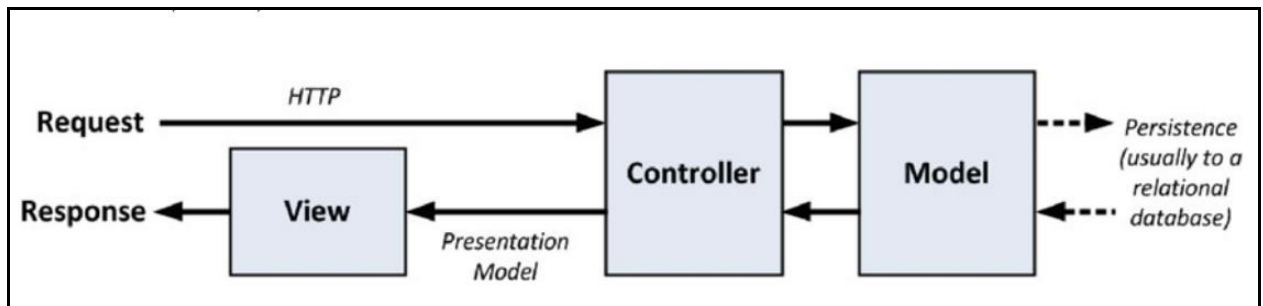
2.5.1. ASP.NET Core MVC архитектура

ASP.NET Core MVC прати је архитектурални патерн MVC, који одређује облик ASP.NET веб апликација и интеракције између компоненти које садржи.

Битно је истаћи разлику између MVC архитектуралног образаца и ASP.NET Core MVC имплементације.

У ASP.NET Core MVC-у контролери су C# класе, обично изведене из *Microsoft*-ове *AspNetCore.Mvc.Controller* класе. Свака јавна метода у контролеру је повезана са одговарајућим URL-ом (*Uniform Resource Locator*). Када се захтев пошаље на URL

повезан са методом, наредбе у тој методи се извршавају са циљем да се изврше операције над модел, а затим да се изаберете приказ клијента. Следећа слика илуструје интеракције између контролер, модел и приказ.



Слика 4. Интеракције у MVC апликацији

ASP.NET Core MVC користи механизам за приказ, познат као *Razor*. *Razor* компоненте су одговорне за процесуирање приказа са циљем генерисања одговора за претраживач. *Razor* прикази су HTML шаблони који садрже C# логику која се користи за обраду података модела и генерисање динамичкој садржаја страница које реагују на промене у моделу. ASP.NET Core MVC не примењује никаква ограничења на имплементацију модела домена. Могуће је креирати модел користећи обичне C# објекте и имплементирати перзистентност користећи било коју од база података, оквира за релационо мапирање објеката или других алата за управљање подацима које подржава .NET. (*Freeman, Pro ASP.NET Core MVC, 2022*)

Одговорности *Model*-а

Модел представља стање апликације и било које пословне логике или операција које он треба да изврши. Пословна логика треба да буде инкапсулирана у моделу, заједно са било којом имплементацијом било које логике која је задужена за одржавање перзистентности апликације.

Одговорности *View*-а

View-и су одговорни за представљање садржаја кроз кориснички интерфејс. Они користе машину за приказ *Razor*-а да би уградили .NET код у HTML ознаке. У њима треба да постоји минимална пословна логика, док сва логика треба да се односи на представљање садржаја.

Одговорности контролера

Контролери су компоненте које управљају интеракцијом корисника, раде са моделом и на крају бирају приказ за рендеровање. У MVC апликацији, *View* приказује само информације; контролер управља и реагује на кориснички унос и интеракцију. Контролер је почетна улазна тачка и одговоран је за избор типова модела са којима ће се радити и који приказ ће се приказати. (Smith, 2022)

2.6. RAZOR

Razor је синтакса за означавање која се користи за уграђивање у веб странице. *Razor* синтакса се састоји од *Razor* ознаке, C# и HTML-а. Датотеке које садрже *Razor* обично имају екстензију датотеке *.cshtml*. Синтакса *Razor*-а је слична машинама за шаблонирање различитих JavaScript оквира апликација за једну страницу (СПА), као што су *Angular*, *React*, *VueJs*, и *Svelte*.

Подразумевани језик *Razor*-а је HTML. Рендеровање HTML-а из *Razor*-а се не разликује од приказивања HTML -а из HTML датотеке. (Anderson, Mullen, & Vicarel, 2022)

Примарна корист од *Razor*-а јесте додавање C# (серверског кода) на поглед, односно писање C# кода у комбинацији са самим HTML кодом. Циљ је овога да се може писати чистији HTML и CSS код, као и да се могу правити петље, услови и сличне ствари у HTML документу, што може бити врло корисно. Овакав начин рада поприлично је сличан начину рада у PHP језику који се исто тако може укључити у сам HTML документ. Уз то што се сам код може писати у HTML документу, могуће је и креирати методе које се затим могу користити на више места (постоје три врсте метода које могу бити креиране: HTML помагачи, *Razor* функције и *Razor* делегати). Наравно, све методе се могу применити локално у погледу, али и глобално.

2.7. REPOSITORY ОБРАЗАЦ

Repository образац је образац дизајна који посредује податке од и до доменског слоја и слоја приступа подацима (као што је *Entity Framework Core*). Репозиторијуми су класе које скривају логику потребну за складиштење или преузимање података. Дакле, апликацији није важно која конкретно врста Објектно-орјентисаних мапа (ОПМ) се користи, пошто се све што је у вези са ОПМ-ом обрађује унутар слоја репозиторијума.

Repository образац помаже код смањења броја упита и дуплирања кода. У ситуацији да су потребни подаци из складишта података, када би се исти код писао сваког пута када су потребни одређени подаци из базе, завршило би се са великом количином дуплог кода који се у малој мери разликује међусобно. Зато је добра пракса овај код сместити у репозиторијуме и позивати их колико год је пута поребно. (Dijkstra, 2022)

Постоји доста ОПМ алата доступних за ASP.NET Core. Тренутно најпопуларнији је *Entity Framework Core*. Да би се одржао корак са технологијама које се развијају и да би решења била ажурна, кључно је изградити апликације које могу да пређу на нову технологију приступа подацима са минималним утицајем на базу кода апликације. Такође могу постојати случајеви када може постојати потреба за више ОПМ-ова у једном решењу.

Repository патерн помаже да се ово постигне, креирањем апстрације преко слоја приступа подацима. Сада више не постоји зависност од ОПМ алата за креирање апликације. Сваки ОПМ алат постаје једна од опција, а никако једина опција. (Murugan, 2021)

2.8. JAVASCRIPT

JavaScript је скриптни језик на страни клијента који се у потпуности покреће унутар веб претраживача. За позивање унутар HTML-а потребно додати отварајуће `<script>` и затварајуће `</script>` HTML тагове. У комбинацији са HTML језиком и CSS-ом, *JavaScript* чини динамички HTML - DHTML (енгл. Dynamic HTML). *JavaScript* се обично користи у веб прегледачу као интерактивни или аутоматизовани садржај, у распону од искачећих порука до сложених веб апликација. Овај програмски језик садржи и могућност рада са текстом, датумима и регуларним изразима за шта има одговарајући API.

JavaScript се такође може користити у програмирању на страни сервера. То је омогућено уз помоћ окружења попут *Node.js*. После популаризације и све већег броја сервиса написаних у *Node.js* –у 2010. године, појавио се менаџер пакета под називом *npm* који је програмерима олакшао објављивање и преузимање делова отвореног кода библиотеке *Node.js*. Данас, *npm* има преко 350.000 пакета отвореног кода и олакшава развој клијентских и серверских апликације.

Када говоримо о *JavaScript*-у контексту клијентских апликација, наилазимо на изазов мулти-платформских решења – то су оне апликације које се могу покретати и користити на више различитих платформи и које се углавном се односе на различите оперативне системе. Овај програмски језик успешно покрива све најпопуларније оперативне системе – покреће се на: *Windows* -у, *Linux* -у, *macOS*-у, *Android*-у, *iOS*-у... Поред различитих оперативних система, *JavaScript* је такође компатибилан са свим веб претраживачима, захваљујући стандардизацији *ECMAScript*. Након сваке нове функционалности коју је објавила *ECMA* организација, сви веб прегледачи морају да имплементирају ове исте могућности.

Иако на први поглед делује, *JavaScript* није повезан са програмским језиком *Java*. Имена ова два језика су заштитни знаци *Oracle* компаније, али их прате различити принципи дизајна. (Nixon, 2014)

У наставку текста биће наведене неке од најважнијих карактеристика овог програмског језика:

1. Универзална подршка: Помоћу одговарајућих интерпретера, омогућена је подршка у свим коришћеним вебпрегледачима
2. Динамички типови: JavaScript подржава динамичке типове што значи да се типови променљиве дефинишу на основу сачуване вредности. На пример, ако се декларише променљива `k`, она може бити ускладиштена као стринг, или као вредност типа броја, или као низ, или као објекат. Ово је познато као динамички типови. У *JavaScript*-у, је неопходно користити кључну реч `var` или `let` испред имена променљиве да би била проглашена променљива без бриге о њеном типу.
3. Подршка за ООП: У *JavaScript*-у, објекат представља асоцијативни низ, наглашен помоћу прототипа; сваки кључ у облику низа карактера представља назив својства објекта за који постоји два синтаксна начина за дефинисање: нотација са тачком, `обј.х = 10` и нотација са заградама, `обј['х'] = 10`. Било које поље се може додати или обрисати за време извршавања кода.
4. Функционални приступ: Ово имплицира да *JavaScript* користи функционални приступ, чак се и објекти креирају из конструкторских функција и свака функција конструктора представља јединствени тип објекта. Такође, функције у *JavaScript*-у се могу користити као објекти и могу се прослеђивати и другим функцијама. (Zakas, 2014)

Наведене карактеристике су заједничке за све ECMAScript имплементације, осим ако није експлицитно другачије наведено.

У овом језику је на GitHub-у је написана јако велика количина кода у свим репозиторијумима корисника. У ствари, према Stack Overflow-у, скоро 70% професионалних програмера који су одговорили на анкету 2020. кодира је у JavaScript-у.

Неки од могућих разлога за овако велику популарност су:

1. Способност да се користи у свим аспектима развоја веб апликације, како у *front-end* делу, тако и у *back-end*-у . Ова свестраност омогућава креирање веб апликације користећи само JavaScript.
2. Његова природа скриптног језика чини његову синтаксу лако разумљивом; такође, није потребно посебно подешавање да би се покренуо, јер се лако покреће у претраживачу.
3. Због прве две тачке има снажну подршку заједнице. Огромна заједница чини да постоји много библиотека отвореног кода доступних за бесплатно коришћење. Како се повећава број јавних библиотека отвореног кода, тако се повећава и број корисника, повећавајући подршку заједнице. (Nixon, 2014)

2.8.1. Типови променљивих

JavaScript не захтева декларисање променљивих унапред, као други (строга типизирани) језици. Променљива се уводи на оном месту где је потребна.

Ипак, декларација променљиве постоји у *JavaScript*-у и врши се навођењем исказа *var*, *let* или *const* и листе променљивих. Истовремено се може декларисати (и дефинисати) више променљивих – довољно је навести листу променљивих, одвојених зарезом. Унутар декларације се може извршити и дефиниција променљиве (додела вредности).

JavaScript је јако специфичан што се тиче променљивих и поља доступности (*scope*) истих у односу на остале језике. Како је већ наведено, постоји више типова променљивих: *var* који дефинише променљиву на нивоу функције, а не на нивоу блока кода као код већине језика, *let* је доступна само унутар блока кода где је дефинисана, нпр. унутар *if* наредбе. Чим се напусти тренутни блок, неће бити могућности да се тој променљивој приступи. Тип *const* има исти ниво доступности као и *let* али за разлику од њега не може се променити или предефинисати накнадно.

JavaScript најпре прође кроз контекст (функцију) и "евидентира" све променљиве које су декларисане у њему, а тек онда извршава наредбе (у које спада и оператор доделе, односно дефиниција променљиве).

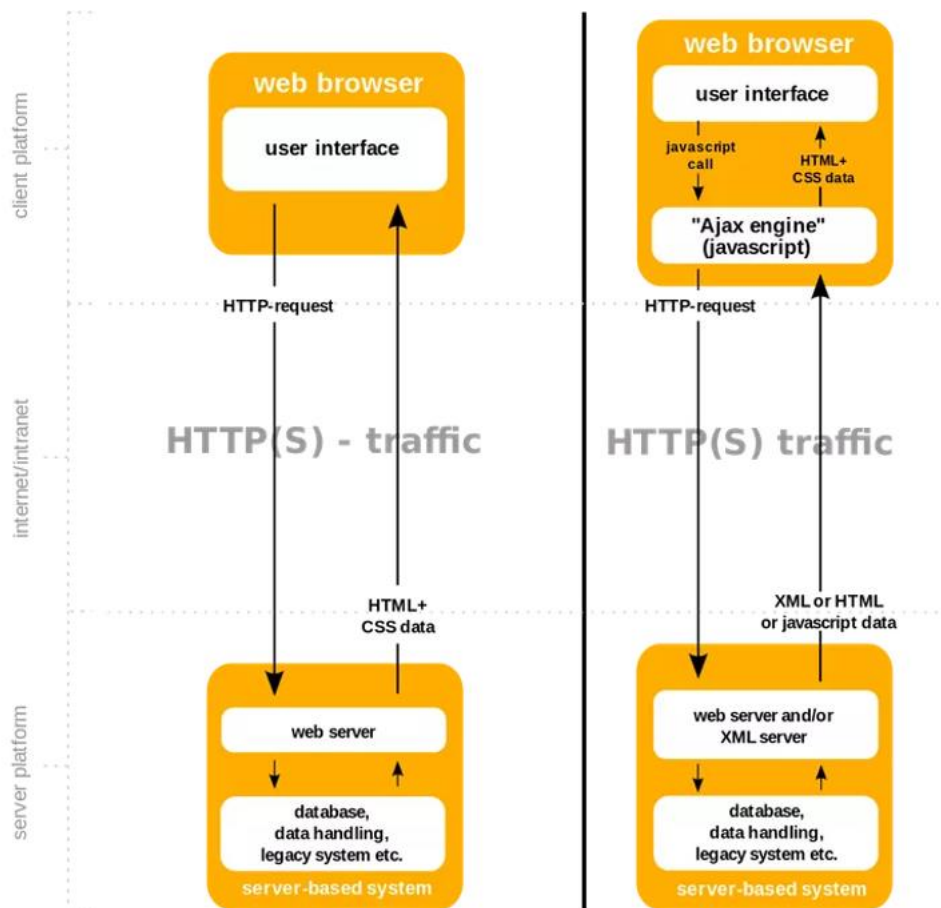
Променљиву можемо увести и без *var* исказа. У том случају, где год да је променљива дефинисана, биће везана за глобални објекат. У веб апликацијма, све глобалне променљиве су у ствари својства објекта *window*. Непостојећа или декларисана али недефинисана променљива ће имати вредност *undefined*. (Nixon, 2014)

2.9. AJAX

AJAX је скраћеница за Асинхрони *JavaScript* и *XML* која се односи на скуп техника веб развоја, а не на стварни програмски језик. *AJAX* се, међутим, широко користи у програмирању на страни клијента (нпр. *JavaScript*) како би се омогућило слање и примање података у и са базе података/сервера. Оно што је јединствено код *AJAX* програмирања је то што можете да размењујете податке у позадини без ометања корисничког искуства.

Одвајањем слоја за размену података од слоја приказивања, *AJAX* омогућава веб страницама, а тиме и веб апликацијама, да динамички мењају садржај без потребе за поновним учитавањем целе странице. Овај метод је изузетно користан за перформансе веб странице. Са становишта корисничког искуства, посетиоци могу имати користи од тога да виде одређене информације генерисане без потребе да поново учитавају страницу. Ово је огроман корак напред ка побољшању перципираних перформанси.

Како би било лакше описивање, на шеми на слици под редним бројем 5 приказано је поређење између традиционалне методе слања захтева за подацима са веб сервера и коришћењем *AJAX* методе. У наставку текста ће бити објашњене сличности и разлике.



Слика 5. Поређење између традиционалне методе (лево) и коришћењем AJAX методе (десно)

Традиционална метода

Почевши од врха традиционалне методе, може се видети да претраживач покушава да упути захтев веб серверу ка подацима.

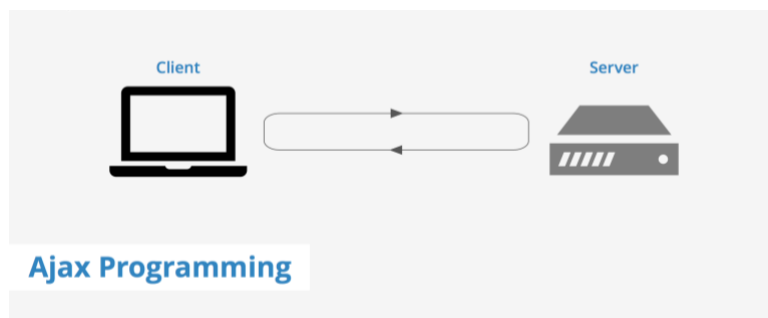
1. *Hypertext Transfer Protocol* (HTTP) захтев се шаље од веб претраживача ка веб серверу. Корисник мора да сачека да се овај захтев обради и врати одговор пре него што може да види тражене податке.
2. Захтев стиже до веб сервера и преузима одговарајуће податке.
3. Тражени подаци се затим шаљу назад у претраживач и корисник може да види те податке.

AJAX метода

Следеће се дешава када се захтевају исти подаци, међутим, овај пут користећи AJAX метод.

1. Претраживач шаље *JavaScript* позив *AJAX* методи. Другим речима, креира се *XMLHttpRequest* објекат.
2. У позадини се шаље *HTTP* захтев серверу и преузимају се одговарајући подаци.
3. *HTML*, *XML* или *JavaScript* подаци се враћају *AJAX* методи која затим испоручује тражене податке претраживачу.

Користећи ову методу, корисник не доживљава застоје од тренутка када је поднео захтев до тренутка када је добио стварне информације.



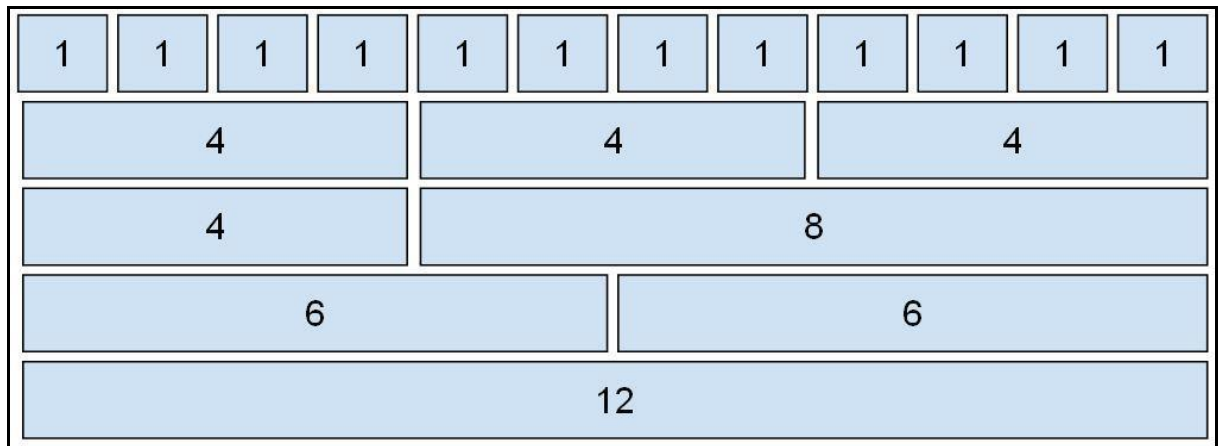
Слика 6. *AJAX* метода

2.10. *BOOTSTRAP*

Bootstrap 4 представља бесплатни веб оквир отвореног кода, за креирање веб сајтова и веб апликација. Као оквир, *Bootstrap* укључује основе за респонзивни веб развој, тако да програмери само треба да убаце код у унапред дефинисани систем мреже. *Bootstrap* оквир је израђену *HTML* -у, *CSS*-у и *JavaScript* -у.

Мрежни систем у *Bootstrap* -у дели страницу на редове и колоне, при чему сваки ред има 12 колона. Може се искористити свих 12 колона, међутим, није неопходно. Могу се колоне се могу и комбиновати како би се направило мање ширих колона. Оно што мора бити испоштовано је да је збир 12 или мањи од 12.

На слици 7 је приказан један пример поделе странице у *Bootstrap* мрежу.



Слика 7. Пример поделе странице у *Bootstrap* мрежу

Овај систем мреже је састављен од *flexbox*-а, што значи да елементи присутни у мрежи мењају своју величину у зависности од резолуције екрана на ком се приказују. Омогућава веб страници или апликацији да открије величину и оријентацију екрана посетиоца и аутоматски прилагоди приказ у складу са тим. Приступ заснован на мобилним уређајима претпоставља да су паметни телефони, таблети и мобилне апликације за специфичне задатке примарни алати запослених за обављање посла. *Bootstrap* се бави захтевима тих технологија у дизајну и укључује компоненте корисничког интерфејса, распореде, *JavaScript* алате за имплементацију оквира. Софтвер је доступан унапред компајлиран или као изворни код.

2.11. ENTITY FRAMEWORK CORE

Entity Framework (EF) Core је верзија отвореног кода популарне *Entity Framework* технологије приступа подацима. Између осталог, *EF Core* се може користити и као објектно-релациони мапер (ОРМ), који:

- Омогућава .NET програмерима да раде са базом података користећи .NET објекте.
- Елиминише потребу за већином кода за приступ подацима који обично треба да се напише. (Vickers, Rojansky, & Anderson, *Entity Framework Core*, 2021)

EF Core подржава многе механизме базе података. Неки NuGet пакети за различите механизме базе података су:

- Npgsql.EntityFrameworkCore.PostgreSQL – PostgreSQL
- Pomelo.EntityFrameworkCore.MySql – MariaDB
- MySql.EntityFrameworkCore – SQL Server 2012 (и све новије верзије)
- Microsoft.EntityFrameworkCore.SqlServer – PostgreSQL (Vickers, Cincura, & Rojansky, Database Providers, 2022)

2.11.1. Модел

Са EF Core, приступ подацима се врши помоћу модела. Модел се састоји од класа ентитета и контекстног објекта који представља сесију са базом података. Контекстни објекта омогућава постављање упита и чување података.

EF подржава следеће приступе развоју модела:

- Генерисање модела из постојеће базе података.
- Ручно кодирање модел који одговара бази података.
- Када је модел креиран, користи се ЕФ миграција да би се креирала база података из модела. Миграције омогућавају развој базе података како се модел мења. (Vickers, Rojansky, & Anderson, Entity Framework Core, 2021)

2.12. ПОСТАВЉАЊЕ УПИТА

Инстанце класа ентитета се преузимају из базе података помоћу упита интегрисан у језик (енгл. *Language Integrated Query* - LINQ).

Миграције су групе команди које припремају базе података за употребу *Entity Framework Core* апликације. Користе се за креирање базе података, а затим је чување синхронизованости са променама у моделу података. Миграције аутоматизују процес креирања и одржавања база података. Без миграција, морала би се креирати база података користећи SQL команде. Такође би се морао ручно конфигурирати *Entity Framework Core* да би база могла да га користи.

Посматрајући миграције, две најзначајније методе за посматрање су:

1. *Up* метода је одговорна за надоградњу базе података. У ситуацији новокреиране базе података, примена миграције покреће *EF Core* да креира табелу за свако својство *DbSet<T>* у класи контекста.
2. *Down* метода се користи за враћање базе података у њено претходно стање, поништавајући ефекат *Up* методе. На пример, када се ради о миграцији која ће бити примењена на празну базу података, прелазак на старију верзију само мора да уклони табелу која је креирана да би складиштили податке ентитета у бази података. (Freeman, Pro Entity Framework Core 2 for ASP.NET Core MVC, 2018)

Seeder-и су посебне класе која се користи за генерисање и уметање узорака података у базу података. Ово је важна карактеристика у развојним окружењима, јер омогућава да се поново креира апликацију са новом базом података, користећи вредности узорака које би се иначе морале ручно убацити сваки пут када се база података поново креира. (Vickers, Cincura, & Rojansky, Database Providers, 2022)

2.12.1. LINQ – УПИТИ

LINQ (*Language Integrated Query* – упит интегрисан у језик) јесте скуп могућности које језик *C#* и *.NET* пружају за писање структурираних упита за претраживање локалних колекција објеката и удаљених извора података, на начин који не нарушава безбедност типова. *LINQ* се састоји од скупа оператора који су на много начина слични *SQL* командама. Уз помоћ *LINQ* оператора, време сортирања, груписања, филтрирања података, које су до сада морале да имплементирају или користе библиотеке трећих страна, знатно се смањује. *LINQ* је уведен у верзији *C#* 3.0 и *.NET* 3.5. *LINQ* омогућава претраживање сваке колекције која имплементира интерфејс *IEnumerable*, без обзира на то да ли је у питању низ вредности, листа или *XML DOM* документ, као и удаљене изворе података, као што су табеле на *SQL* Серверу. *LINQ* пружа предности и проверавања исправне употребе типова у време превођења и динамичког састављања упита. Ово поглавље описује архитектуру *LINQ*-а и основе писања упита. Сви основни типови су дефинисани у именским просторима *System.Linq* и *System.Linq.Expressions*. Акроним *LINQ* би се најбоље могао превести као *SQL* карактеристике интегрисане у програмски језик.

3. СТУДИЈСКИ ПРИМЕР

На студијском примеру је приказан развој софтверског система за евиденцију података о филмовима употребом ASP .NET Core оквира.

Приликом реализације пројекта коришћена је упрошћена Ларманова метода развоја софтвера. Ларманова метода за развој софтвера се заснива на итеративно-инкременталном моделу животног циклуса софтвера и састоји се од следећих фаза (Vlajić, 2020):

1. Прикупљање корисничких захтева;
2. Анализа;
3. Пројектовање;
4. Имплементација;
5. Тестирање.

4. ПРИКУПЉАЊЕ КОРИСНИЧКИХ ЗАХТЕВА

Прикупљање корисничких захтева је прва фаза развоја софтверског система Лармановом методом. Ово јако важна фаза јер се у њој дефинишу својства и услови које софтверски систем или шире гледајући пројекат треба да задовољи. Јако је битно идентификовати корисничке захтеве на прави начин, јер у супротном може доћи до проблема у наредним фазама развоја. Корисничке захтеве делимо на функционалне и нефункционалне захтеве. Функционални захтеви дефинишу захтеване функције система, док нефункционални захтеви дефинишу све остале захтеве (Vlajić, 2020).

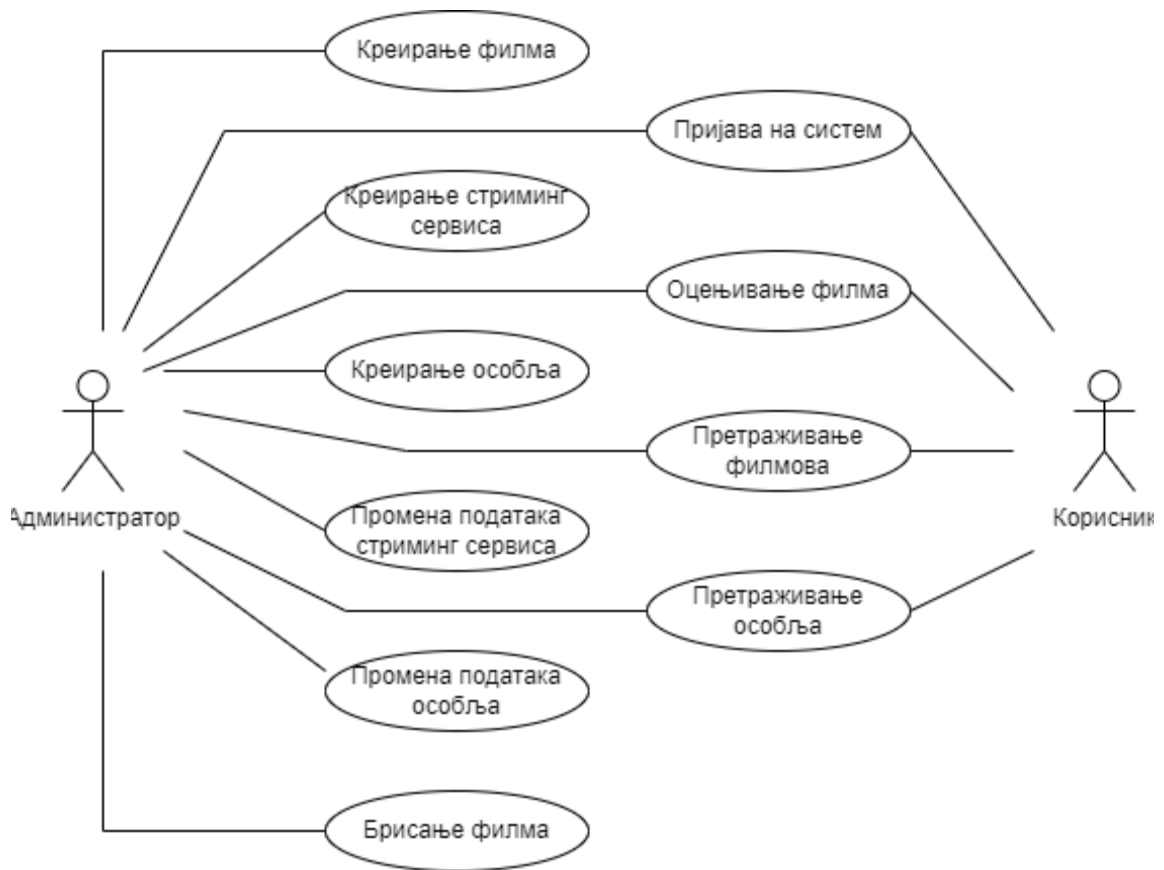
4.1. ВЕРБАЛНИ ОПИС МОДЕЛА

Основна идеја сајта је да олакша корисницима одабир филмова које ће погледати у блиској будућности, тако што ће им пружити информације о самим филмовима и њихове рецензије од стране корисника. Поред информација о филмовима, корисницима ће бити доступне информације о глумцима, издавачким кућама, платформама за приказивање филмова и о сарадницима на филму (режисери, сценаристи, костимографи, итд.). Ради лакшег сналажења корисник ће имати могућност да претражује ставке (филмове, глумце, сараднике на филму, итд.) по називу, као и да сортира филмове по жанровима. Такође, након одгледаног филма корисник ће моћи да га оцени и да остави свој коментар о самом филму како би приближио свој доживљај неким другим корисницима.

Поред обичног корисника постоји и администратор. Администратор поседује следеће привилегије које обичан корисник нема, а то су: креирање нових ставки, брисање ставки, промена података и промовисање обичног корисника у админа.

4.2. СПЕЦИФИКАЦИЈА ЗАХТЕВА ПОМОЋУ СЛУЧАЈЕВА КОРИШЋЕЊА

1. Пријава на систем
2. Креирање филма
3. Креирање стриминг сервиса
4. Креирање особља
5. Претраживање филмова
6. Промена података стриминг сервиса
7. Промена података особља
8. Брисање филма
9. Оцењивање филма
10. Претраживање особља



Слика 8. Дијаграм случаја коришћења за корисника

СК1: Случај коришћења – Пријава на систем

Назив СК: Пријава на систем

Актери СК: Корисник/Администратор

Учесници СК: Корисник/Администратор и систем (програм)

Предуслов: Систем је укључен у приказана је форма за пријаву на систем.

Основни сценарио СК

1. Корисник/Администратор **уноси** податке за пријаву на систем. (АПУСО)
2. Корисник/Администратор **контролише** да ли је коректно унео податке.(АНСО)
3. Корисник/Администратор **позива** систем да нађе пријављеног корисника/администратора. (АПСО)
4. Систем **тражи** пријављеног корисника/администратора. (СО)
5. Систем **приказује** кориснику/администратору поруку: “Успешна пријава”.(ИА)

Алтернативна сценарија

- 5.1. Уколико систем није успео да нађе пријављеног корисника/администратора, он **приказује** поруку: „Нисте унели одговарајуће податке“. (ИА)

СК2: Случај коришћења – Креирање филма

Назив СК: Креирање филма

Актери СК: Администратор

Учесници СК: Администратор и систем (програм)

Предуслов: Систем је укључен и администратор је улогован под својом шифром. Систем приказује форму за рад са филмом. Учитана је листа студија, стриминг сервиса, глумаца, жанрова и особља.

Основни сценарио СК

1. Администратор **уноси** податке о филму. (АПУСО)
2. Администратор **контролише** да ли је коректно унео податке о филму. (АНСО)
3. Администратор **позива** систем да запамти филм. (АПСО)
4. Систем **памти** филм. (СО)
5. Систем **приказује** администратору запамћени филм и поруку: “Систем је запамтио филм”.(ИА)

Алтернативна сценарија

- 5.1. Уколико систем не може да креира филм он приказује администратору поруку “Систем не може да запамти филм”. (ИА)

СК3: Случај коришћења – Креирање стриминг сервиса

Назив СК: Креирање стриминг сервиса

Актери СК: Администратор

Учесници СК: Администратор и систем (програм)

Предуслов: Систем је укључен и администратор је улогован под својом шифром. Систем приказује форму за рад са стриминг сервисом.

Основни сценарио СК

1. Администратор **уноси** податке о стриминг сервису. (АПУСО)
2. Администратор **контролише** да ли је коректно унео податке о стриминг сервису. (АНСО)
3. Администратор **позива** систем да запамти стриминг сервис. (АПСО)
4. Систем **памти** стриминг сервис. (СО)
5. Систем **приказује** администратору запамћени студио и поруку: “Систем је запамтио стриминг сервис”.(ИА)

Алтернативна сценарија

- 5.1. Уколико систем не може да креира стриминг сервис он приказује администратору поруку “Систем не може да запамти стриминг сервис”.
(ИА)

СК4: Случај коришћења – Креирање особља

Назив СК: Креирање особља

Актери СК: Администратор

Учесници СК: Администратор и систем (програм)

Предуслов: Систем је укључен и администратор је улогован под својом шифром. Систем приказује форму за рад са особљем.

Основни сценарио СК

1. Администратор **уноси** податке о особљу. (АПУСО)
2. Администратор **контролише** да ли је коректно унео податке о особљу. (АНСО)
3. Администратор **позива** систем да запамти особље. (АПСО)
4. Систем **памти** особље. (СО)
5. Систем **приказује** администратору запамћено особље и поруку: “Систем је запамтио особље”. (ИА)

Алтернативна сценарија

- 5.1. Уколико систем не може да креира особље он приказује администратору поруку “Систем не може да запамти особље”. (ИА)

СК5: Случај коришћења – Претраживање филмова

Назив СК: Претраживање филмова

Актори СК: Корисник/Администратор

Учесници СК: Корисник/Администратор и систем (програм)

Предуслов: Систем је укључен и корисник/администратор је улогован под својом шифром. Систем приказује форму за рад са филмовима. Учитана је листа филмова.

Основни сценарио СК

1. Корисник/Администратор **уноси** вредност по којој претражује филмове. (АПУСО)
2. Корисник/Администратор **позива** систем да нађе филмове по задатој вредности. (АПСО)
3. Систем **тражи** филмове по задатој вредности. (СО)
4. Систем **приказује** кориснику/администратору податке о филмовима и поруку: “Систем је нашао филмове по задатој вредности”. (ИА)
5. Корисник/Администратор **бира** филм. (АПУСО)
6. Корисник/Администратор **позива** систем да прочита филм. (АПСО)
7. Систем **учитава** филм. (СО)
8. Систем **приказује** кориснику/администратору податке о филму и поруку: “Систем је прочитао филм”. (ИА)

Алтернативна сценарија

4.1 Уколико систем не може да нађе филмове он приказује кориснику/администратору поруку: “Систем не може да нађе филмове по задатој вредности”. Прекида се извршење сценарија. (ИА)

8.1 Уколико систем не може да прочита филм он приказује кориснику/администратору поруку “Систем не може да прочита филм”. (ИА)

СК6: Случај коришћења – Промена података стриминг сервиса

Назив СК: Промена података стриминг сервиса

Актери СК: Администратор

Учесници СК: Администратор и систем (програм)

Предуслов: Систем је укључен и администратор је улогован под својом шифром. Систем приказује форму за рад са стриминг сервисом. Учитана је листа стриминг сервиса.

Основни сценарио СК

1. Администратор уноси вредност по којој претражује стриминг сервисе. (АПУСО)
2. Администратор позива систем да нађе стриминг сервис по задатој вредности. (АПСО)
3. Систем тражи стриминг сервисе по задатој вредности. (СО)
4. Систем приказује студије и поруку: “Систем је нашао стриминг сервисе по задатој вредности”. (ИА)
5. Администратор бира стриминг сервис. (АПУСО)
6. Администратор позива систем да прочита стриминг сервис.(АПСО)
7. Систем учитава стриминг сервис. (СО)
8. Систем показује администратору податке о стриминг сервису и поруку “Систем је прочитао стриминг сервис“. (ИА)
9. Администратор **уноси(мења)** податке о стриминг сервису. (АПУСО)

10. Администратор **контролише** да ли је коректно унео податке о стриминг сервису. (АНСО)
11. Администратор **позива** систем да запамти стриминг сервис. (АПСО)
12. Систем **памти** стриминг сервис. (СО)
13. Систем **приказује** администратору запамћени стриминг сервис и поруку: “Систем је запамтио стриминг сервис”.(ИА)

Алтернативна сценарија

- 4.1. Уколико систем не може да нађе стриминг сервис он приказује администратору поруку: “Систем не може да нађе стриминг сервис по задатој вредности”. Прекида се извршење сценарија. (ИА)
- 8.1. Уколико систем не може да учита стриминг сервис он приказује администратору поруку “Систем не може да учита стриминг сервис”. Прекида се извршење сценарија. (ИА)
- 13.1. Уколико систем не може да запамти податке о стриминг сервису он приказује администратору поруку “Систем не може да запамти стриминг сервис”. (ИА)

СК7: Случај коришћења – Промена података особља

Назив СК: Промена података особља

Актери СК: Администратор

Учесници СК: Администратор и систем (програм)

Предуслов: Систем је укључен и администратор је улогован под својом шифром. Систем приказује форму за рад са особљем. Учитана је листа особља.

Основни сценарио СК

1. Администратор **уноси** вредност по којој претражује особље. (АПУСО)
2. Администратор **позива** систем да нађе особље по задатој вредности. (АПСО)
3. Систем **тражи** особље по задатој вредности. (СО)
4. Систем **приказује** глумце и поруку: “Систем је нашао особље по задатој вредности”. (ИА)
5. Администратор **бира** особље. (АПУСО)
6. Администратор **позива** систем да учита особље.(АПСО)
7. Систем **учитава** особље. (СО)
8. Систем **показује** администратору податке о особљу и поруку “Систем је учитао особље“. (ИА)
9. Администратор **уноси(мења)** податке о особљу. (АПУСО)
10. Администратор **контролише** да ли је коректно унео податке о особљу. (АНСО)
11. Администратор **позива** систем да запамти особље. (АПСО)

12. Систем **памти** особље. (СО)

13. Систем **приказује** администратору запамћеног припадника особља и поруку:
“Систем је запамтио особље”. (ИА)

Алтернативна сценарија

4.1. Уколико систем не може да нађе особље он приказује администратору поруку:
“Систем не може да нађе особље по задатој вредности”. Прекида се извршење сценарија. (ИА)

8.1. Уколико систем не може да учита припадника особља он приказује администратору поруку “Систем не може да учита припадника особља”. Прекида се извршење сценарија. (ИА)

13.1. Уколико систем не може да запамти податке о припаднику особља он приказује администратору поруку “Систем не може да запамти припадника особља”. (ИА)

СК8: Случај коришћења – Брисање филма

Назив СК: Брисање филма

Актери СК: Администратор

Учесници СК: Администратор и систем (програм)

Предуслов: Систем је укључен и администратор је улогован под својом шифром. Систем приказује форму за рад са филмовима. Учитана је листа филмова.

Основни сценарио СК

1. Администратор **уноси** вредност по којој претражује филм. (АПУСО)
2. Администратор **позива** систем да нађе филм по задатој вредности. (АПСО)
3. Систем **тражи** филм по задатој вредности. (СО)
4. Систем **приказује** кориснике и поруку: “Систем је нашао филм по задатој вредности”. (ИА)
5. Администратор **бира** филм. (АПУСО)
6. Администратор **позива** систем да обрише филм. (АПСО)
7. Систем **брише** филм. (СО)
8. Систем **приказује** администратору поруку: “Систем је обрисао филм”. (ИА)

Алтернативна сценарија

4.1. Уколико систем не може да нађе филм он приказује администратору поруку: “Систем не може да нађе филм по задатој вредности”. Прекида се извршење сценарија. (ИА)

8.1. Уколико систем не може да обрише филм он приказује администратору поруку “Систем не може да обрише филм”. (ИА)

СК9: Случај коришћења – Оцењивање филма

Назив СК: Оцењивање филма

Актори СК: Корисник / Администратор

Учесници СК: Корисник / Администратор и систем (програм)

Предуслов: Систем је укључен и корисник / администратор је улогован под својом шифром. Систем приказује форму за рад са филмовима. Учитана је листа филмова.

Основни сценарио СК

1. Корисник/Администратор **уноси** вредност по којој претражује филмове. (АПУСО)
2. Корисник/Администратор **позива** систем да нађе филмове по задатој вредности. (АПСО)
3. Систем **тражи** филмове по задатој вредности. (СО)
4. Систем **приказује** филмове и поруку: “Систем је нашао филмове по задатој вредности”. (ИА)
5. Корисник/Администратор **бира** филм. (АПУСО)
6. Корисник/Администратор **позива** систем да прочита филм. (АПСО)
7. Систем **учитава** филм. (СО)
8. Систем **показује** кориснику/администратору податке о филму и поруку “Систем је прочитао филм“. (ИА)
9. Корисник/Администратор **уноси(мења)** оцену и коментар о филму. (АПУСО)

10. Корисник/Администратор **контролише** да ли је коректно унео оцену и коментар о филму. (АНСО)
11. Корисник/Администратор **позива** систем да запамти оцену и коментар о филму. (АПСО)
12. Систем **памти** оцену и коментар о филму. (СО)
13. Систем **приказује** кориснику/администратору запамћену оцену и коментар о филму и поруку: “Систем је запамтио оцену и коментар о филму”. (ИА)

Алтернативна сценарија

4.1. Уколико систем не може да нађе филмове он приказује кориснику/администратору поруку: “Систем не може да нађе филмове по задатој вредности”. Прекида се извршење сценарија. (ИА)

8.1. Уколико систем не може да учита филм он приказује кориснику/администратору поруку “Систем не може да учита филм”. Прекида се извршење сценарија. (ИА)

13.1. Уколико систем не може да запамти оцену и коментар о филму он приказује кориснику/администратору поруку “Систем не може да запамти оцену и коментар о филму”. (ИА)

СК10: Случај коришћења – Претраживање особља

Назив СК: Претраживање особља

Актори СК: Корисник/Администратор

Учесници СК: Корисник/Администратор и систем (програм)

Предуслов: Систем је укључен и корисник/администратор је улогован под својом шифром. Систем приказује форму за рад са особљем. Учитана је листа особља.

Основни сценарио СК

1. Корисник/Администратор **уноси** вредност по којој претражује особље. (АПУСО)
2. Корисник/Администратор **позива** систем да нађе припаднике особља по задатој вредности. (АПСО)
3. Систем **тражи** припаднике особља по задатој вредности. (СО)
4. Систем **приказује** кориснику/администратору податке о особљу и поруку: “Систем је нашао припаднике особља по задатој вредности”. (ИА)
5. Корисник/Администратор **бира** припадника особља. (АПУСО)
6. Корисник/Администратор **позива** систем да прочита припадника особља. (АПСО)
7. Систем **учитава** припадника особља. (СО)
8. Систем **приказује** кориснику/администратору податке о припаднику особља и поруку: “Систем је прочитао припадника особља”. (ИА)

Алтернативна сценарија

4.1 Уколико систем не може да нађе припаднике особље он приказује кориснику/администратору поруку: “Систем не може да нађе припаднике особље по задатој вредности”. Прекида се извршење сценарија. (ИА)

8.1 Уколико систем не може да учита припадника особља он приказује кориснику/администратору поруку “Систем не може да учита припадника особља”. (ИА)

5. АНАЛИЗА

У другој фази развоја софтверског система описује се његова пословна логика, односно структура и понашање тог система. Понашање је у даљем раду приказано кроз системске дијаграме секвенци и системске операције, а структура помоћу концептуалног и релационог модела.

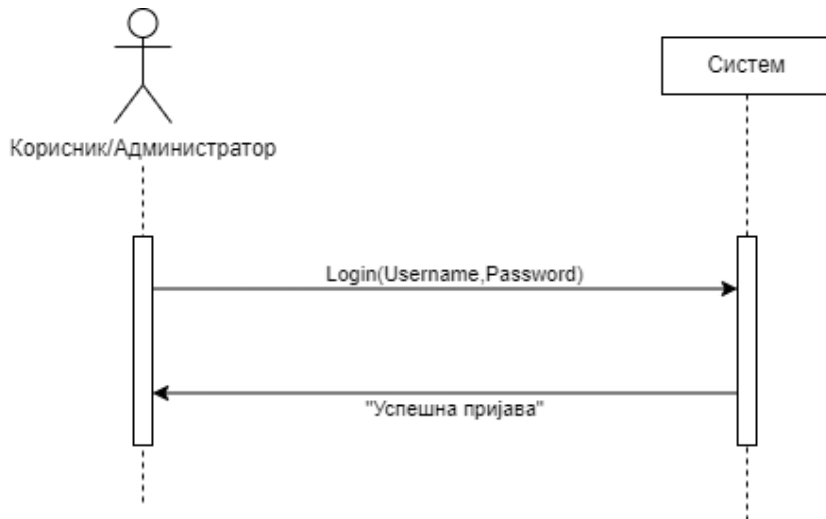
5.1. ПОНАШАЊЕ СОФТВЕРСКОГ СИСТЕМА - СИСТЕМСКИ ДИЈАГРАМИ СЕКВЕНЦИ

Системски дијаграми секвенци приказују двосмерну комуникацију између актора и система. Актори праве догађаје, а након извршења одговарајуће системске операције на страни система, њима се враћа одговор на тај догађај. Догађаји се одвијају у одређеном редоследу и тај редослед је приказан на дијаграмима секвенци.

ДС1: Дијаграм секвенци случаја коришћења – Пријава на систем

Основни сценарио СК

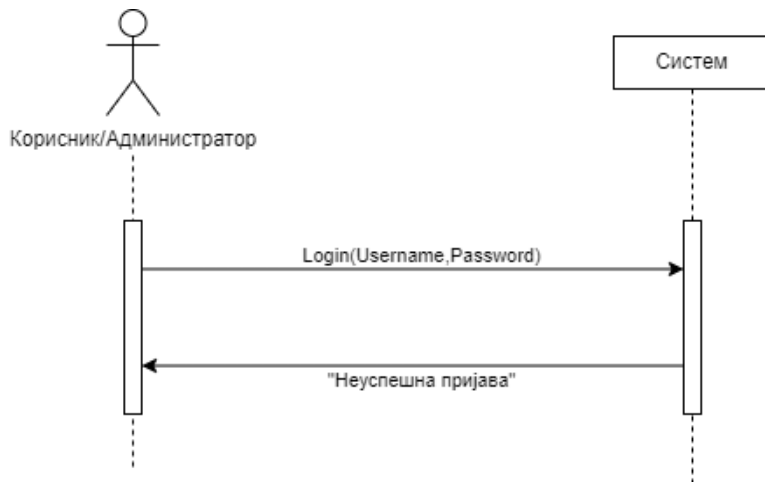
1. Корисник/Администратор **позива** систем да **нађе** пријављеног корисника/администратора. (АПСО)
2. Систем **приказује** кориснику/администратору поруку: “Успешна пријава”.(ИА)



Слика 9. ДС Пријава на систем

Алтернативна сценарија

2.1. Уколико систем није успео да нађе пријављеног корисника/администратора, он **приказује** поруку: „Нисте унели одговарајуће податке“. (ИА)



Слика 10. ДС неуспешна пријава

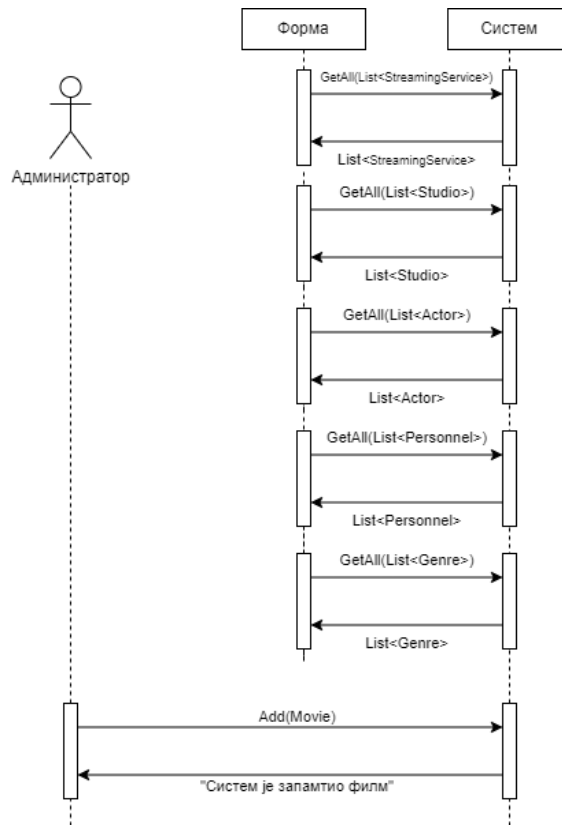
Са наведених секвенцијалних дијаграма уочена је једна системска операција:

1. Signal **Login(Username, Password)**

ДС2: Дијаграм секвенци случаја коришћења – Креирање филма

Основни сценарио СК

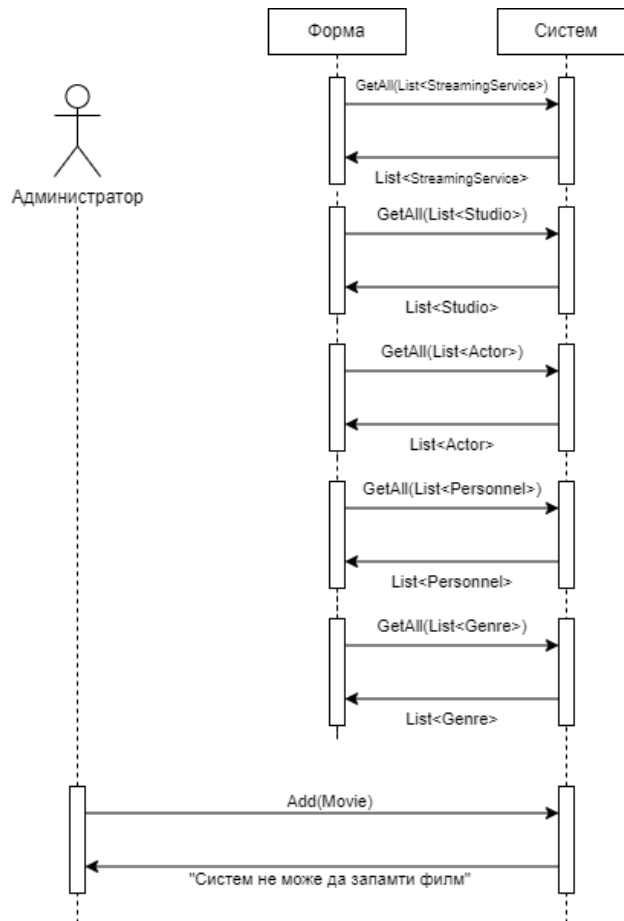
1. Форма **позива** систем да учита листу стриминг сервиса. (АПСО)
2. Систем **враћа** форми листу стриминг сервиса. (ИА)
3. Форма **позива** систем да учита листу студија. (АПСО)
4. Систем **враћа** форми листу студија. (ИА)
5. Форма **позива** систем да учита листу глумаца. (АПСО)
6. Систем **враћа** форми листу глумаца. (ИА)
7. Форма **позива** систем да учита листу филмских сарадника. (АПСО)
8. Систем **враћа** форми листу филмских сарадника. (ИА)
9. Форма **позива** систем да учита листу жанрова. (АПСО)
10. Систем **враћа** форми листу жанрова. (ИА)
11. Администратор **позива** систем да запамти филм. (АПСО)



Слика 11. ДС креирање филма

Алтернативна сценарија

- 1.1. Уколико систем не може да креира филм он приказује администратору поруку “Систем не може да запамти филм”. (ИА)



Слика 12. ДС неуспешно креирање филма

Са наведених секвенцијалних дијаграма уочено је шест системских операција:

1. Signal **GetAll(List<StreaminService>)**
2. Signal **GetAll(List<Studio>)**
3. Signal **GetAll(List<Actor>)**
4. Signal **GetAll(List<Personnel>)**
5. Signal **GetAll(List<Genre>)**
6. Signal **Add(Movie)**

ДС3: Дијаграм секвенци случаја коришћења – Креирање стриминг сервиса

Основни сценарио СК

1. Администратор **позива** систем да запамти стриминг сервис. (АПСО)
2. Систем **приказује** администратору запамћени студио и поруку: “Систем је запамтио стриминг сервис”.(ИА)



Слика 13. ДС креирање стриминг сервиса

Алтернативна сценарија

- 2.1. Уколико систем не може да креира стриминг сервис он приказује администратору поруку “Систем не може да запамти стриминг сервис”. (ИА)



Слика 14. ДС неуспешно креирање стриминг сервиса

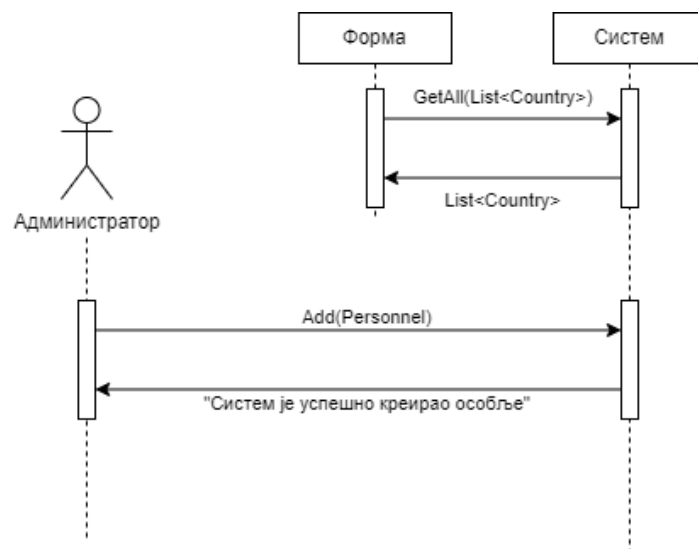
Са наведених секвенцијалних дијаграма уочена је једна системска операција:

1. Signal **Add(StreamingService)**

ДС4: Дијаграм секвенци случаја коришћења – Креирање особља

Основни сценарио СК

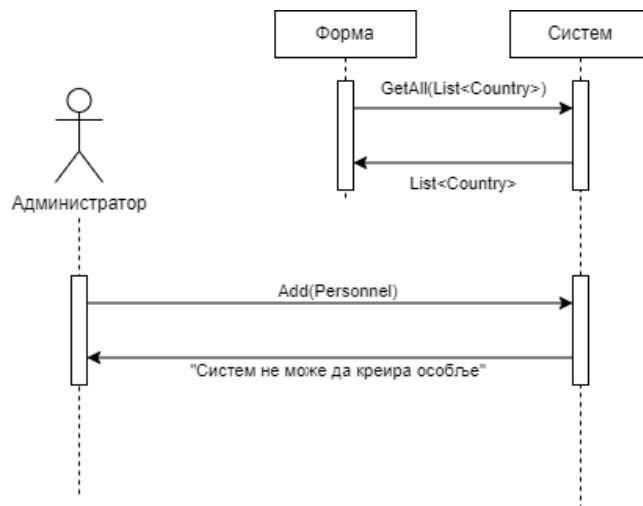
1. Форма **позива** систем да учита листу држава. (АПСО)
2. Систем **враћа** форми листу држава. (ИА)
3. Администратор **позива** систем да запамти особље. (АПСО)
4. Систем **приказује** администратору запамћено особље и поруку: “Систем је запамтио особље”. (ИА)



Слика 15. ДС креирање особља

Алтернативна сценарија

- а. Уколико систем не може да креира особље он приказује администратору поруку “Систем не може да запамти особље”. (ИА)



Слика 16. ДС неуспешно креирање особља

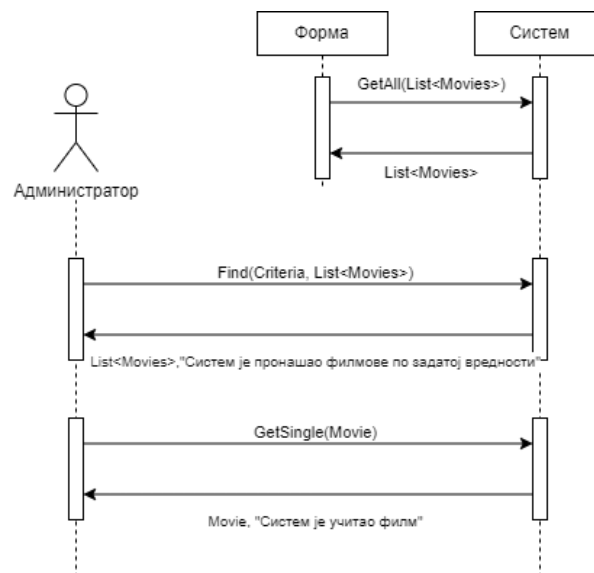
Са наведених секвенцијалних дијаграма уочено је две системске операција:

1. Signal **GetAll(List<Country>)**
2. Signal **Add(Personnel)**

ДС5: Дијаграм секвенци случаја коришћења – Претраживање филмова

Основни сценарио СК

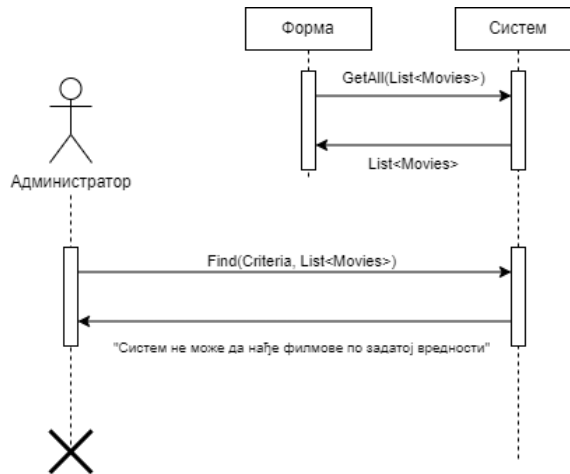
1. Форма **позива** систем да учита листу филмова. (АПСО)
2. Систем **враћа** форми листу филмова. (ИА)
3. Корисник/Администратор **позива** систем да нађе филмове по задатој вредности. (АПСО)
4. Систем **приказује** кориснику/администратору податке о филмовима и поруку: “Систем је нашао филмове по задатој вредности”. (ИА)
5. Корисник/Администратор **позива** систем да учита филм. (АПСО)
6. Систем **приказује** кориснику/администратору податке о филму и поруку: “Систем је прочитао филм”. (ИА)



Слика 17. ДС претрага филма

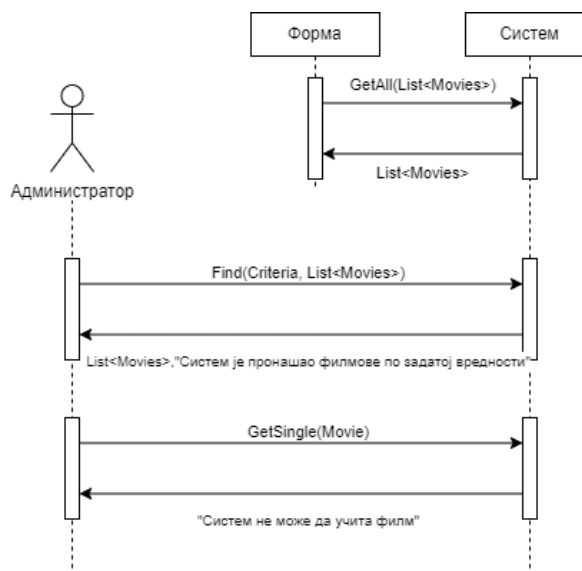
Алтернативна сценарија

4.1 Уколико систем не може да нађе филмове он приказује кориснику/администратору поруку: “Систем не може да нађе филмове по задатој вредности”. Прекида се извршење сценарија. (ИА)



Слика 18. ДС неуспешна претрага филма

6.1 Уколико систем не може да учита филм он приказује кориснику/администратору поруку “Систем не може да учита филм”. (ИА)



Слика 19. ДС неуспешно учитавање филма

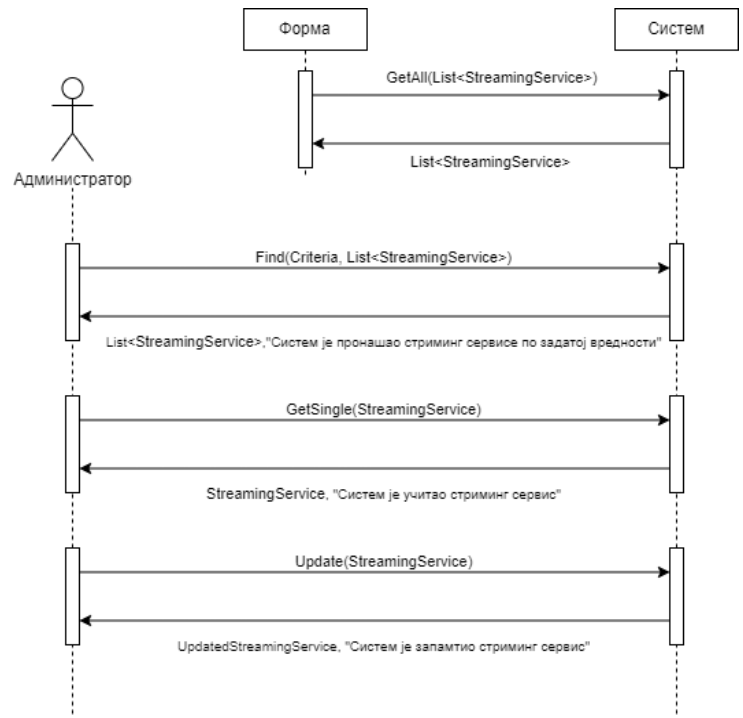
Са наведених секвенцијалних дијаграма уочено је три системске операције:

1. Signal **GetAll(List<Movie>)**
2. Signal **Find(Criteria, List< Movie>)**
3. Signal **GetSingle(Movie)**

ДС6: Дијаграм секвенци случаја коришћења – Промена података стриминг сервиса

Основни сценарио СК

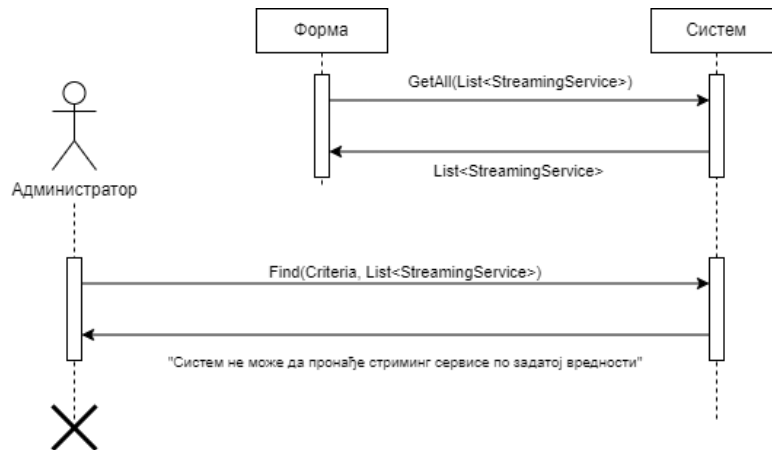
1. Администратор позива систем да нађе стриминг сервис по задатој вредности. (АПСО)
2. Систем приказује студије и поруку: “Систем је нашао стриминг сервисе по задатој вредности”. (ИА)
3. Администратор позива систем да прочита стриминг сервис.(АПСО)
4. Систем показује администратору податке о стриминг сервису и поруку “Систем је прочитао стриминг сервис“. (ИА)
5. Администратор **позива** систем да запамти стриминг сервис. (АПСО)
6. Систем **приказује** администратору запамћени стриминг сервис и поруку: “Систем је запамтио стриминг сервис”.(ИА)



Слика 20. ДС Измена стриминг сервиса

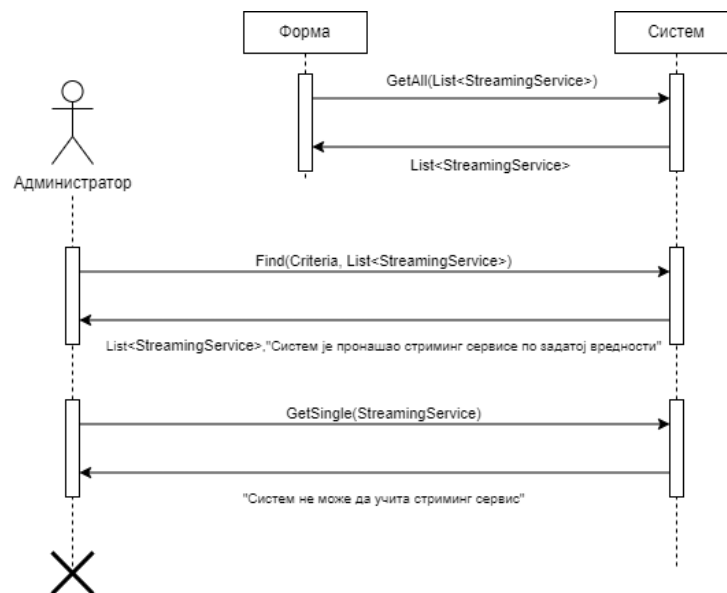
Алтернативна сценарија

4.1. Уколико систем не може да нађе стриминг сервис он приказује администратору поруку: “Систем не може да нађе стриминг сервис по задатој вредности”. Прекида се извршење сценарија. (ИА)



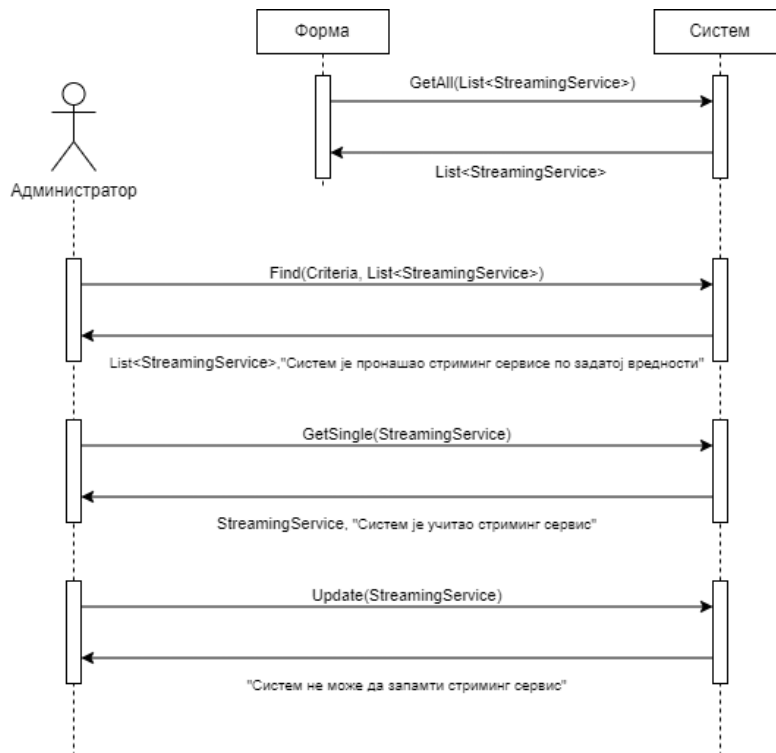
Слика 21. ДС неуспешна претрага стриминг сервиса

8.1. Уколико систем не може да учита стриминг сервис он приказује администратору поруку “Систем не може да учита стриминг сервис”. Прекида се извршење сценарија. (ИА)



Слика 22. ДС неуспешно учитавање стриминг сервиса

13.1. Уколико систем не може да запамти податке о стриминг сервису он приказује администратору поруку “Систем не може да запамти стриминг сервис”. (ИА)



Слика 23. ДС неуспешна измена стриминг сервиса

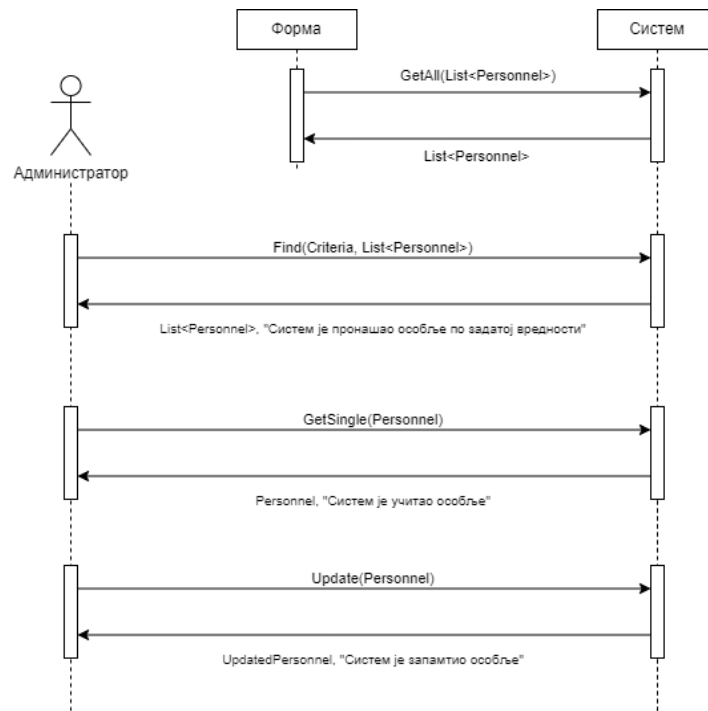
Са наведених секвенцијалних дијаграма уочено је четири системске операције:

1. Signal **GetAll(List<StreamingService>)**
2. Signal **Find(Criteria, List<StreamingService>)**
3. Signal **GetSingle(StreamingService)**
4. Signal **Update(StreamingService)**

ДС7: Дијаграм секвенци случаја коришћења – Промена података особља

Основни сценарио СК

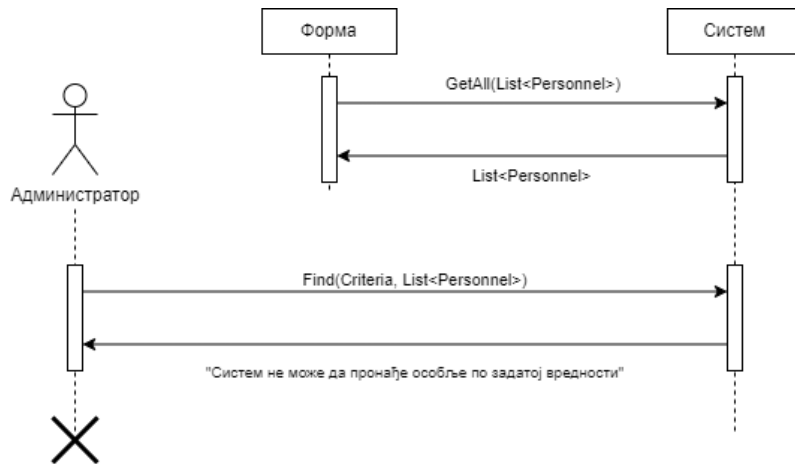
1. Форма позива систем да учита листу студија. (АПСО)
2. Систем враћа форми листу студија. (ИА)
3. Администратор **позива** систем да нађе особље по задатој вредности. (АПСО)
4. Систем **приказује** глумце и поруку: “Систем је нашао особље по задатој вредности”. (ИА)
5. Администратор **позива** систем да учита особље.(АПСО)
6. Систем **показује** администратору податке о особљу и поруку “Систем је прочитао особље“. (ИА)
7. Администратор **позива** систем да запамти особље. (АПСО)
8. Систем **приказује** администратору запамћеног припадника особља и поруку: “Систем је запамтио особље”. (ИА)



Слика 24. ДС измена особља

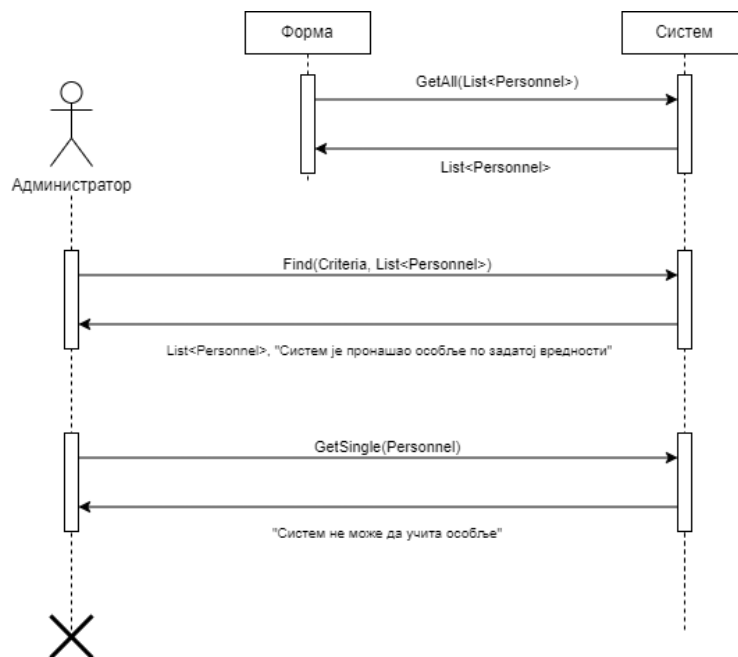
Алтернативна сценарија

4.1. Уколико систем не може да нађе особље он приказује администратору поруку: “Систем не може да нађе особље по задатој вредности”. Прекида се извршење сценарија. (ИА)



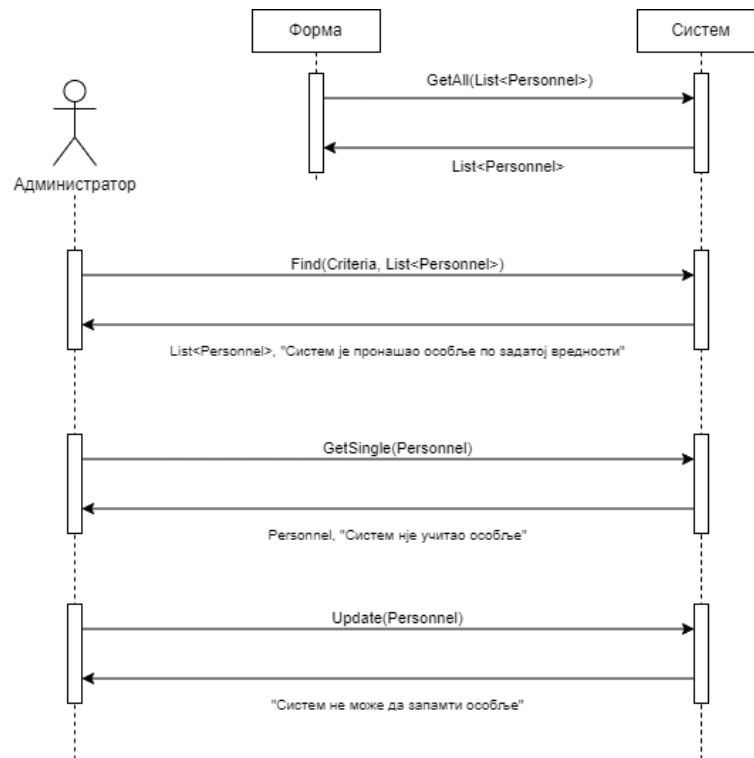
Слика 25. ДС неуспешна претрага особља

6.1. Уколико систем не може да учита припадникаособља он приказује администратору поруку “Систем не може да учита припадникаособља”. Прекида се извршење сценарија. (ИА)



Слика 26. ДС неуспешно учитавање особља

8.1. Уколико систем не може да запамти податке о припаднику особља он приказује администратору поруку “Систем не може да запамти припадника особља”. (ИА)



Слика 27. ДС неуспешна измена особља

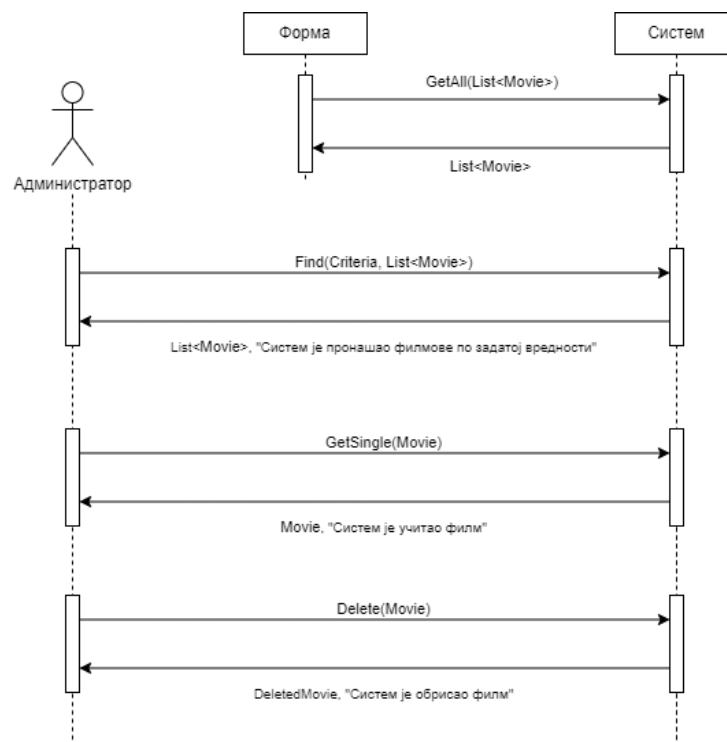
Са наведених секвенцијалних дијаграма уочено је четири системске операције:

1. Signal **GetAll(List<Personnel>)**
2. Signal **Find(Criteria, List<Personnel>)**
3. Signal **GetSingle(Personnel)**
4. Signal **Update(Personnel)**

ДС8: Дијаграм секвенци случаја коришћења – Брисање филма

Основни сценарио СК

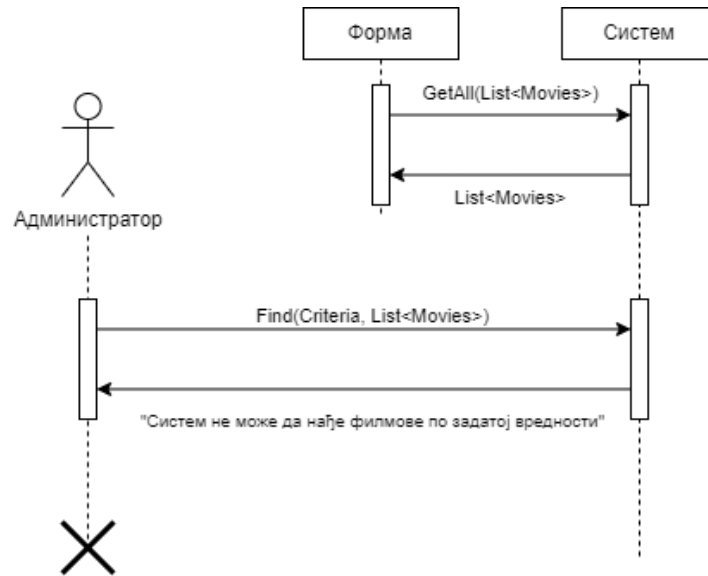
1. Форма **позива** систем да учита листу филмова. (АПСО)
2. Систем **враћа** форми листу филмова. (ИА)
3. Администратор **позива** систем да нађе филм по задатој вредности. (АПСО)
4. Систем **приказује** кориснике и поруку: “Систем је нашао филм по задатој вредности”. (ИА)
5. Администратор **позива** систем да обрише филм. (АПСО)
6. Систем **приказује** администратору поруку: “Систем је обрисао филм”. (ИА)



Слика 28. ДС брисање филма

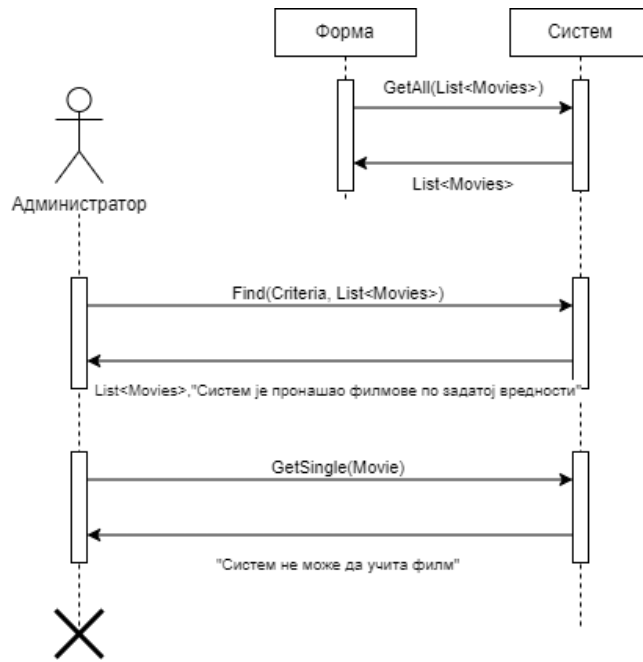
Алтернативна сценарија

4.1. Уколико систем не може да нађе филм он приказује администратору поруку: “Систем не може да нађе филм по задатој вредности”. Прекида се извршење сценариа. (ИА)



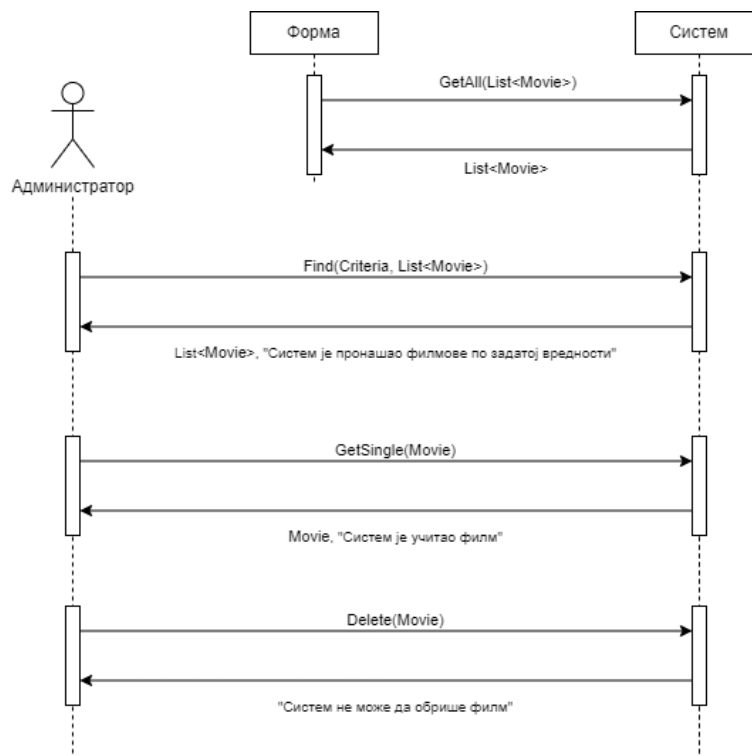
Слика 29. ДС неуспешна претрага филмова

6.1. Уколико систем не може да учита филм он приказује администратору поруку “Систем не може да учита филм”. Прекида се извршење сценариа. (ИА)



Слика 30. ДС неуспешно учитавање филма

8.1. Уколико систем не може да обрише филм он приказује администратору поруку “СИСТЕМ не може да обрише филм”. (ИА)



Слика 31. ДС неуспешно брисање филма

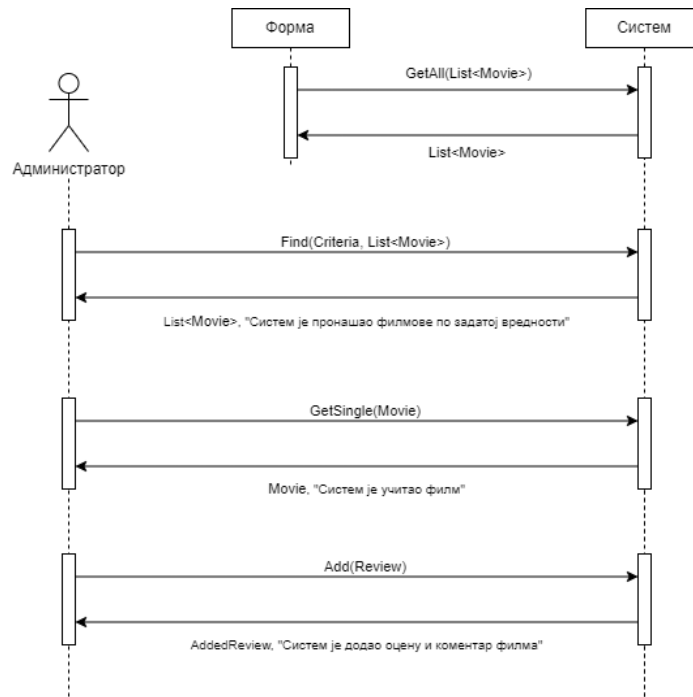
Са наведених секвенцијалних дијаграма уочено је четири системске операције:

1. Signal **GetAll(List<Movie>)**
2. Signal **Find(Criteria, List<Movie>)**
3. Signal **GetSingle(Movie)**
4. Signal **Delete(Movie)**

ДС9: Дијаграм секвенци случаја коришћења – Оцењивање филма

Основни сценарио СК

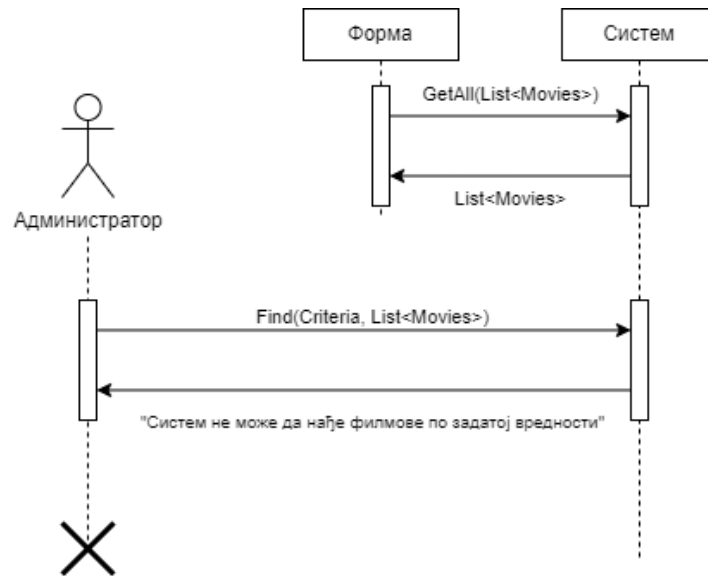
1. Форма позива систем да учита листу особља. (АПСО)
2. Систем враћа форми листу особља. (ИА)
3. Корисник/Администратор **позива** систем да нађе филмове по задатој вредности. (АПСО)
4. Систем **приказује** филмове и поруку: “Систем је нашао филмове по задатој вредности”. (ИА)
5. Корисник/Администратор **позива** систем да учита филм.(АПСО)
6. Систем **показује**кориснику/администратору податке о филму и поруку “Систем је учитао филм“. (ИА)
7. Корисник/Администратор **позива** систем да запамти оцену и коментар о филму. (АПСО)
8. Систем **приказује**кориснику/администраторузапамћену оцену и коментар о филму и поруку: “Систем је запамтио оцену и коментар о филму”. (ИА)



Слика 32. ДС оцењивање филма

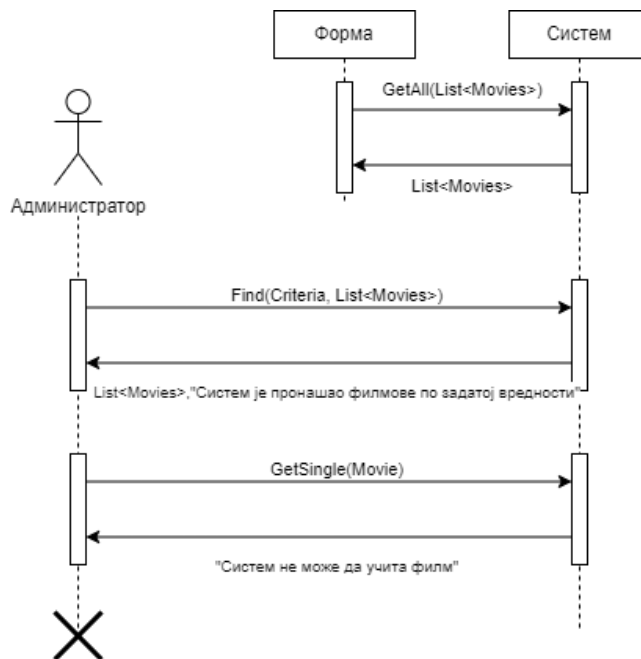
Алтернативна сценарија

4.1. Уколико систем не може да нађе филмове он приказује кориснику/администратору поруку: “Систем не може да нађе филмове по задатој вредности”. Прекида се извршење сценарија. (ИА)



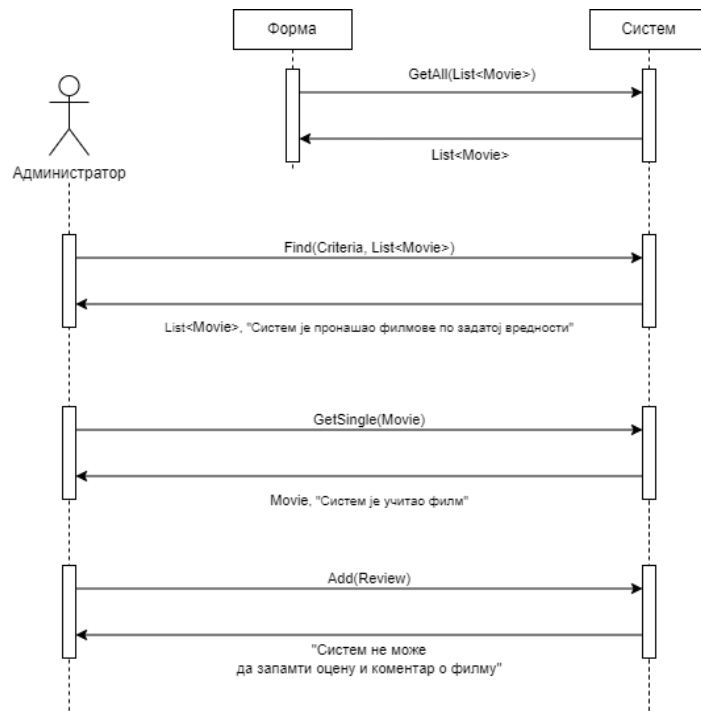
Слика 33. ДС неуспешна претрага филмова

6.1. Уколико систем не може да учита филм он приказује кориснику/администратору поруку “Систем не може да учита филм”. Прекида се извршење сценарија. (ИА)



Слика 34. ДС неуспешно учитавање филма

8.1. Уколико систем не може да запамти оцену и коментар о филму он приказује кориснику/администратору поруку “Систем не може да запамти оцену и коментар о филму”. (ИА)



Слика 35. ДС неуспешно додавање оцене

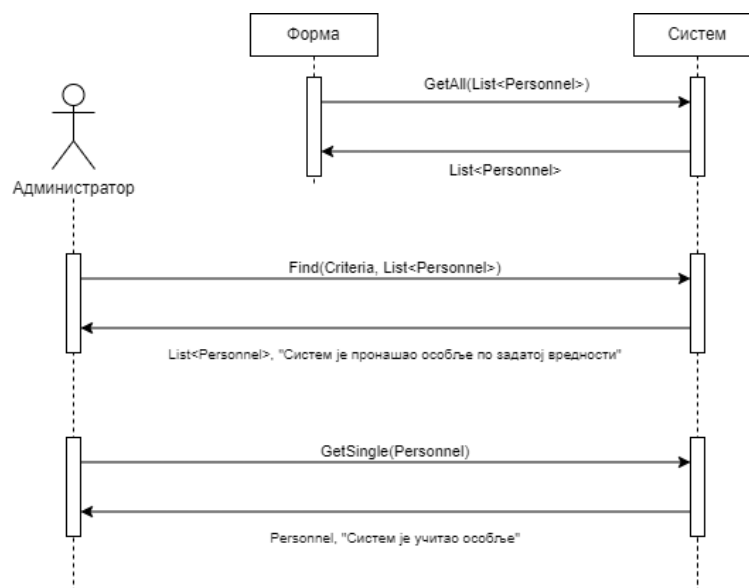
Са наведених секвенцијалних дијаграма уочено је четири системске операције:

1. Signal **GetAll(List<Movie>)**
2. Signal **Find(Criteria, List<Movie>)**
3. Signal **GetSingle(Movie)**
4. Signal **Add(Review)**

ДС10: Дијаграм секвенци случаја коришћења – Претраживање особља

Основни сценарио СК

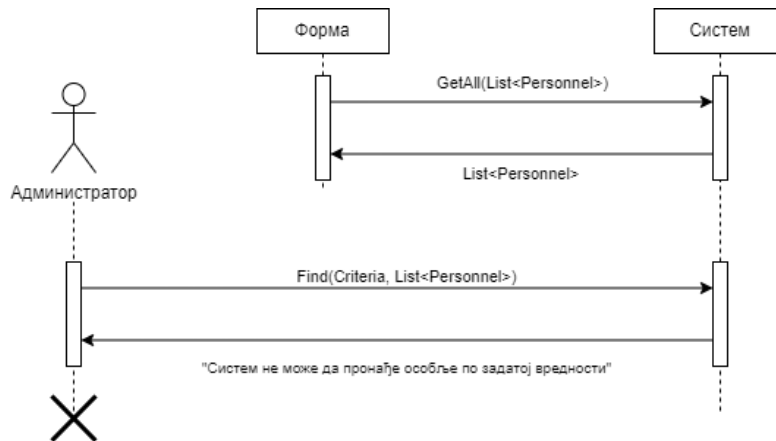
1. Форма позива систем да учита листу особља. (АПСО)
2. Систем враћа форми листу особља. (ИА)
3. Корисник/Администратор **позива** систем да нађе припаднике особља по задатој вредности. (АПСО)
4. Систем **приказује** кориснику/администратору податке о особљу и поруку: “Систем је нашао припаднике особља по задатој вредности”. (ИА)
5. Корисник/Администратор **позива** систем да учита припадника особља. (АПСО)
6. Систем **приказује** кориснику/администратору податке о припаднику особља и поруку: “Систем је учитао припадника особља”. (ИА)



Слика 36. ДС претрага особља

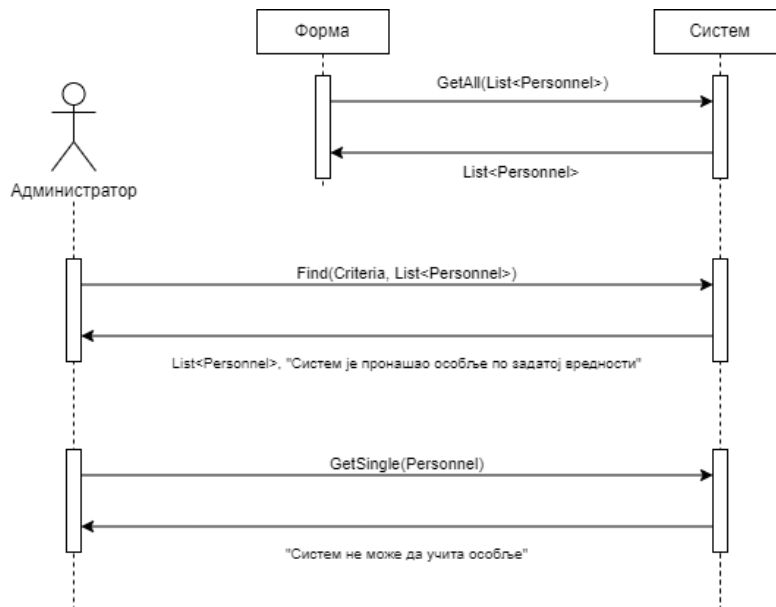
Алтернативна сценарија

4.1 Уколико систем не може да нађе припаднике особље он приказује кориснику/администратору поруку: “Систем не може да нађе припаднике особље по задатој вредности”. Прекида се извршење сценарија. (ИА)



Слика 37. ДС неуспешна претрага особља

6.1 Уколико систем не може да учита филм он приказује кориснику/администратору поруку “Систем не може да учита припадника особља”. (ИА)



Слика 38. ДС неуспешно учитавање особља

Са наведених секвенцијалних дијаграма уочено је четири системске операције:

1. Signal **GetAll(List<Personnel>)**
2. Signal **Find(Criteria, List<Personnel>)**
3. Signal **GetSingle(Personnel)**

Списак системских операција

Као резултат анализе сценарија добијено је укупно 25 системских операција које треба пројектовати:

1. Signal **Login(Username, Password)**
2. Signal **GetAll(List<StreamingService>)**
3. Signal **GetAll(List<Studio>)**
4. Signal **GetAll(List<Actor>)**
5. Signal **GetAll(List<Personnel>)**
6. Signal **GetAll(List<Genre>)**
7. Signal **Add(Movie)**
8. Signal **Add(StreamingService)**
9. Signal **GetAll(List<Country>)**
10. Signal **Add(Personnel)**
11. Signal **GetAll(Movie)**
12. Signal **Find(Criteria, List<Movie>)**
13. Signal **GetSingle(Movie)**
14. Signal **Find(Criteria, List<StreamingService>)**
15. Signal **GetSingle(StreamingService)**

16. Signal **Update(StreamingService)**
17. Signal **Find(Criteria, List<Personnel>)**
18. Signal **GetSingle(Personnel)**
19. Signal **Update(Personnel)**
20. Signal **Delete(Movie)**
21. Signal **Add(Review)**

5.2. ПОНАШАЊЕ СОФТВЕРСКОГ СИСТЕМА – ДЕФИНИСАЊЕ УГОВОРА О СИСТЕМСКИМ ОПЕРАЦИЈАМА

За сваку системску операцију прави се уговор који описује њено понашање. Они описују шта операција треба да ради, а не начин на који ће нешто да изврши. Један уговор је везан искључиво за једну системску операцију (Vlajić, 2020).

Уговор УГ1: Login

Операција: Login(Username, Password): сигнал

Веза са СК: СК1

Предуслови: Вредносна и структурна ограничења над објектом SiteUser морају бити задовољена.

Постуслови: Корисник/Администратор је пријављен

Уговор УГ2: GetAll

Операција: GetAll(List<StreamingService>): сигнал

Веза са СК: СК2, СК6

Предуслови:

Постуслови:

Уговор УГ3: GetAll

Операција: GetAll(List<Studio>): сигнал

Веза са СК: СК2

Предуслови:

Постуслови:

Уговор УГ4: GetAll

Операција: GetAll(List<Actor>): сигнал

Веза са СК: СК2

Предуслови:

Постуслови:

Уговор УГ5: GetAll

Операција: GetAll(List<Personnel>): сигнал

Веза са СК: СК2, СК7, СК10

Предуслови:

Постуслови:

Уговор УГ6: GetAll

Операција: GetAll(List<Genre>): сигнал

Веза са СК: СК2

Предуслови:

Постуслови:

Уговор УГ7: Add

Операција: Add(List<Movie>): сигнал

Веза са СК: СК2

Предуслови: Вредносна и структурна ограничења над објектом Movie морају бити задовољена.

Постуслови: Филм је сачуван.

Уговор УГ8: Add(SS)

Операција: Add(List<StreamingService>): сигнал

Веза са СК: СК3

Предуслови: Вредносна и структурна ограничења над објектом StreamingService морају бити задовољена.

Постуслови: Стриминг сервис је сачуван.

Уговор УГ9: GetAll

Операција: GetAll(List<Country>): сигнал

Веза са СК: СК4

Предуслови:

Постуслови:

Уговор УГ10: Add

Операција: Add(Personnel): сигнал

Веза са СК: СК4

Предуслови: Вредносна и структурна ограничења над објектом Personnel морају бити задовољена.

Постуслови: Особље је сачувано.

Уговор УГ11: GetAll

Операција: GetAll(List<Movie>): сигнал

Веза са СК: СК5, СК8, СК9

Предуслови:

Постуслови:

Уговор УГ12: Find

Операција: Find(Criteria, List<Movie>): сигнал

Веза са СК: СК5, СК8, СК9

Предуслови:

Постуслови:

Уговор УГ13: GetSingle

Операција: GetSingle(Movie): сигнал

Веза са СК: СК5, СК8, СК9

Предуслови:

Постуслови:

Уговор УГ14: Find

Операција: Find(Criteria, List<StreamingService>): сигнал

Веза са СК: СК6

Предуслови:

Постуслови:

Уговор УГ15: GetSingle

Операција: GetSingle(StreamingService): сигнал

Веза са СК: СК6

Предуслови:

Постуслови:

Уговор УГ16: Update

Операција: Update(StreamingService): сигнал

Веза са СК: СК6

Предуслови: Вредносна и структурна ограничења над објектом StreamingService морају бити задовољена.

Постуслови: Стриминг сервис је сачуван

Уговор УГ17: Find

Операција: Find(Criteria, List<Personnel>): сигнал

Веза са СК: СК7, СК10

Предуслови:

Постуслови:

Уговор УГ18: GetSingle

Операција: GetSingle(Personnel): сигнал

Веза са СК: СК7, СК10

Предуслови:

Постуслови:

Уговор УГ19: Update

Операција: Update(Personnel) : сигнал

Веза са СК: СК7

Предуслови: Вредносна и структурна ограничења над објектом Actor морају бити задовољена.

Постуслови: Особље је сачувано.

Уговор УГ20: Delete

Операција: Delete(Movie) : сигнал

Веза са СК: СК8

Предуслови: Вредносна и структурна ограничења над објектом SiteUser морају бити задовољена.

Постуслови: Филм је обрисан.

Уговор УГ21: Add

Операција: Add(Review) : сигнал

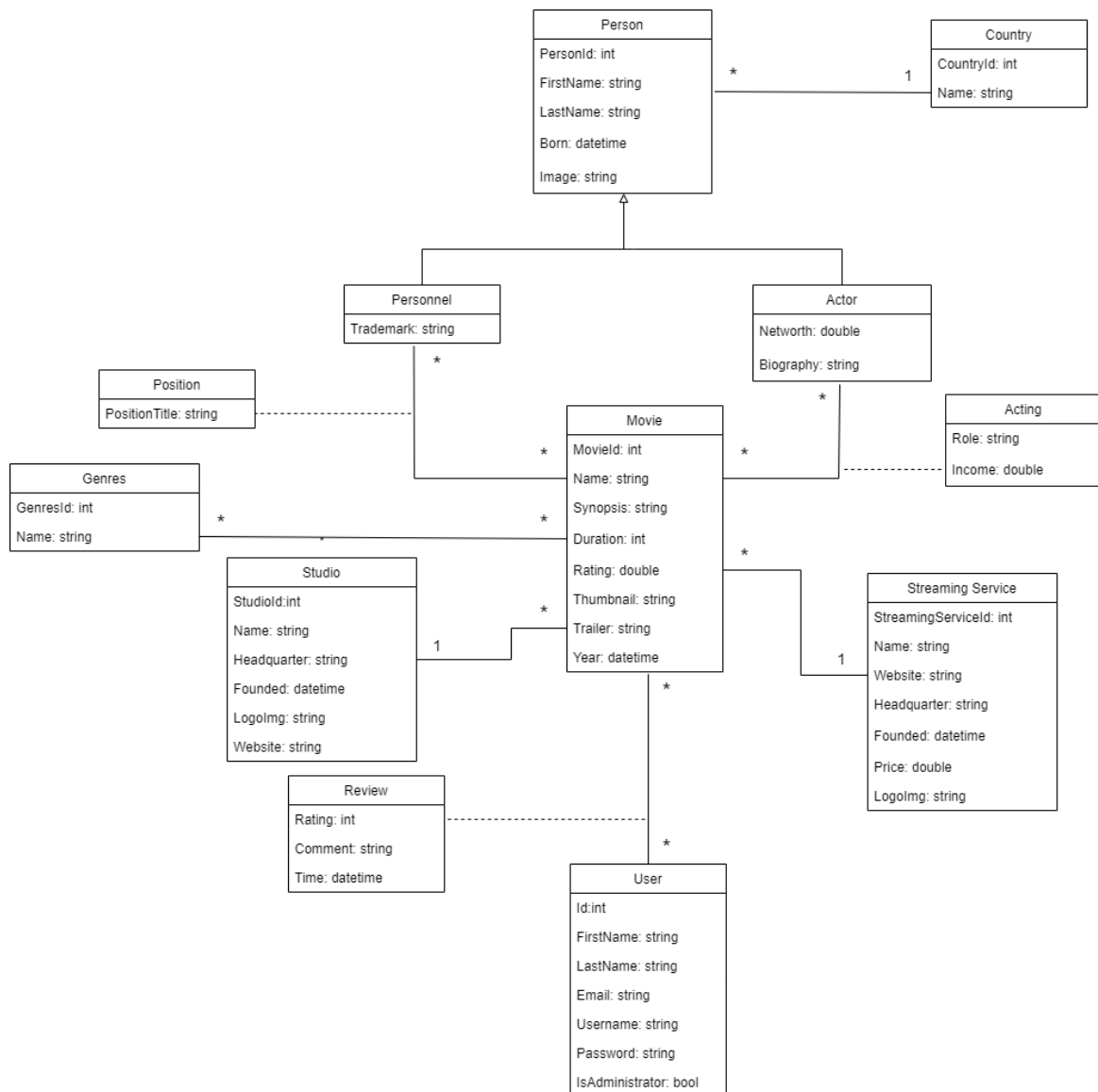
Веза са СК: СК9

Предуслови: Вредносна и структурна ограничења над објектом Review морају бити задовољена.

Постуслови: Рецензија је сачувана.

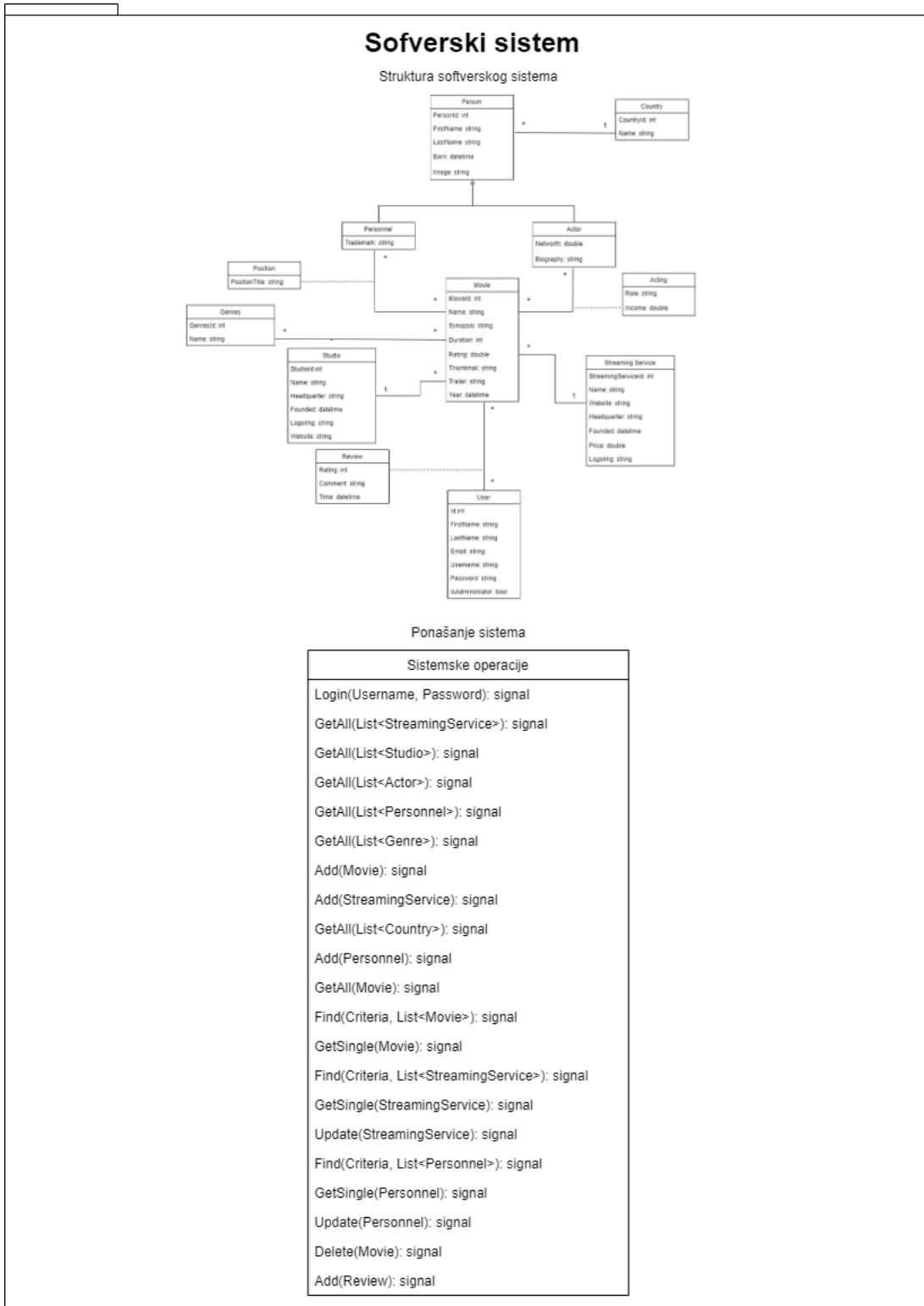
5.3. СТРУКТУРА СОФТВЕРСКОГ СИСТЕМА-КОНЦЕПТУАЛНИ МОДЕЛ

Структура софтверског система се описује помоћу концептуалног модела. Концептуални модел садржи концептуалне класе (доменске објекте) и асоцијације између концептуалних класа. Често се за концептуалне моделе каже да су то доменски модели или модели објектне анализе (Vlajić, 2020).



Слика 39. Концептуални модел структуре софтверског система

Добијена логичка структура и понашање софтверског система (који чине пословну логику резултат су фазе анализе:

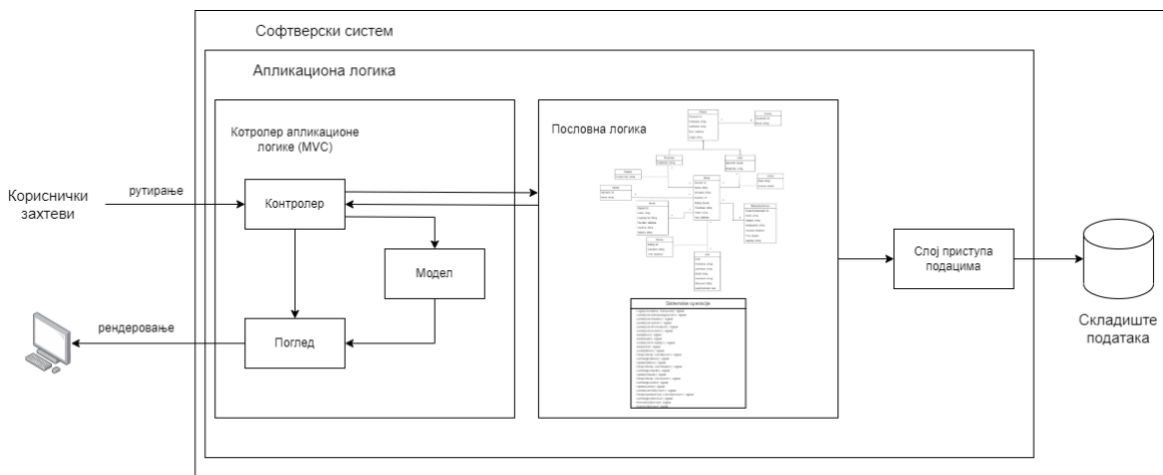


Слика 40. Структура и понашање софтверског система

6. ФАЗА ПРОЈЕКТОВАЊА

Фаза пројектовања описује физичку структуру и понашање софтверског система (архитектуру софтверског система). Архитектура софтверског система је у већини случајева тронивојска и састоји се из следећих слојева (Vlajić, 2020):

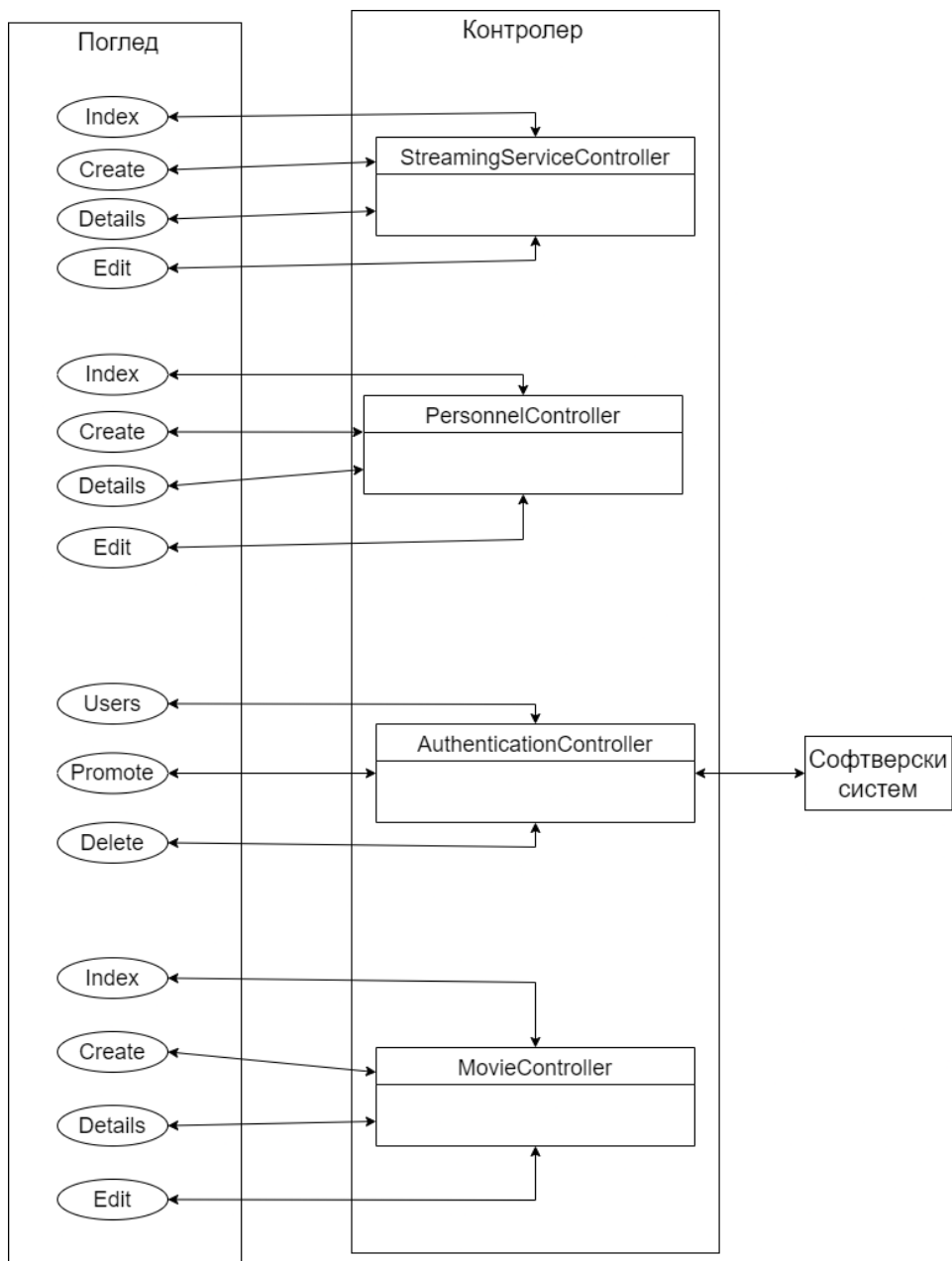
- **Кориснички интерфејс** - који представља улазно–излазну репрезентацију софтверског система.
- **Апликациона логика** - описује структуру и понашање софтверског система. Састоји се из контролера апликационе логике, пословне логике и слоја приступа подацима
- **Складиште података** - који чува стање атрибута софтверског система



Слика 41. Архитектура софтверског система

6.1. ПРОЈЕКТОВАЊЕ КОРИСНИЧКОГ ИНТЕРФЕЈСА

Кориснички интерфејс се састоји од скупа веб страница које прихватају податаке које корисник уноси, прослеђују их до одговарајућег контролера који захтев прослеђује слоју приступа подацима. Након што контролер добије одговор, прослеђује захтеване податке (модел) до одговарајућег погледа.



Слика 42. Повезаност контролера и корисничког интерфејса у архитектури софтверског система

СК1: Случај коришћења – Пријава на систем

Назив СК: Пријава на систем

Актори СК: Корисник/Администратор

Учесници СК: Корисник/Администратор и систем (програм)

Предуслов: Систем је укључен у приказана је форма за пријаву на систем.

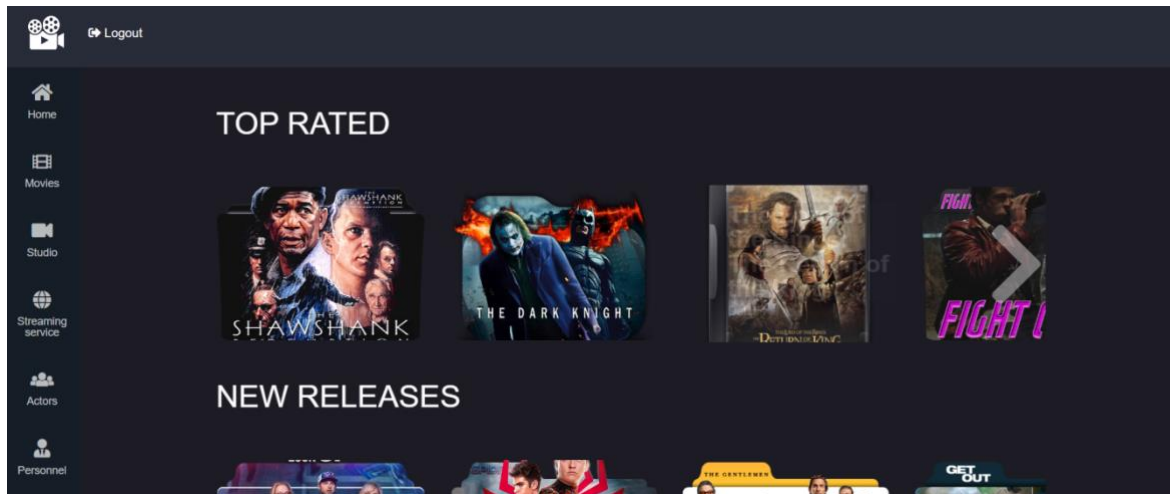
The image shows a user interface for a system login and registration. At the top, there are icons for a film reel and a play button, followed by 'Login' and 'Sign up' links. A sidebar on the left lists navigation options: Home, Movies, Studio, Streaming service, Actors, and Personnel. The main content area contains a 'Username' field, a 'Password' field, and a blue 'Login' button. Below the login fields is a 'Registration' section. The footer includes the text '© 2022 - Filmofil - Privacy'.

Слика 43. Форма за пријаву на систем

Основни сценарио СК

1. Корисник/Администратор **уноси** податке за пријаву на систем. (АПУСО)

2. Корисник/Администратор **контролише** да ли је коректно унео податке.(АНСО)
3. Корисник/Администратор **позива** систем да нађе пријављеног корисника/администратора. (АПСО)
Опис акције: Кликком на дугме „Login” корисник/администратор позива системску операцију *Login(Username, Password)*
4. Систем **тражи** пријављеног корисника/администратора. (СО)
5. Систем **приказује** кориснику/администратору поруку: “Успешна пријава”.(ИА)



Слика 44. Успешна пријава на систем (преусмеравање на почетну страницу)

Алтернативна сценарија

- 5.1. Уколико систем није успео да нађе пријављеног корисника/администратора, он **приказује** поруку: „Нисте унели одговарајуће податке“. (ИА)

A screenshot of a login interface on a dark background. It features two white input fields: the first is labeled 'Username' and contains the text 'PetarPe'; the second is labeled 'Password' and is empty. Below the fields is a blue 'Login' button. A red error message, 'Wrong username or password!', is displayed below the button. At the bottom of the form area, the word 'Registration' is written in white.

Слика 45, Неуспешна пријава на систем

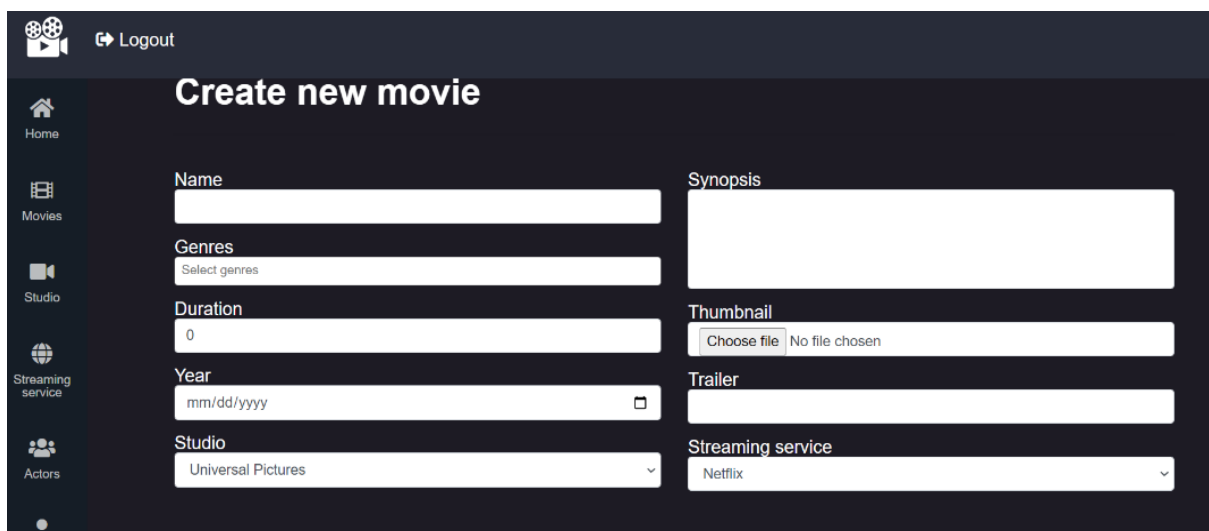
СК2: Случај коришћења – Креирање филма

Назив СК: Креирање филма

Актори СК: Администратор

Учесници СК: Администратор и систем (програм)

Предуслов: Систем је укључен и администратор је улогован под својом шифром. Систем приказује форму за рад са филмом. Учитана је листа студија, стриминг сервиса, глумаца, жанрова и особља.



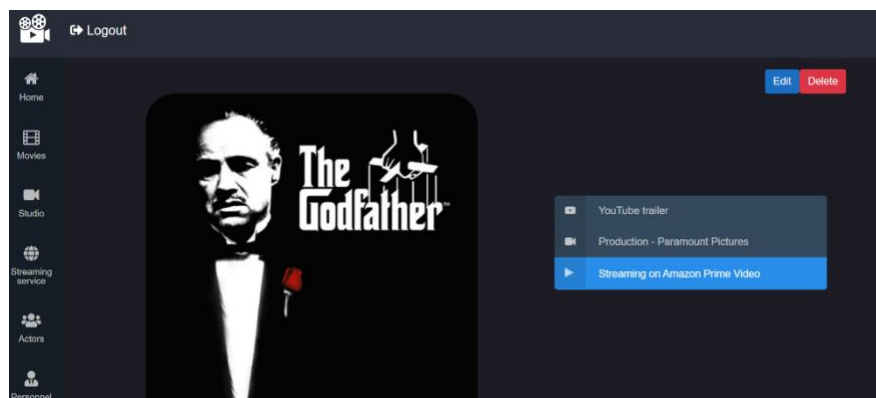
Слика 46. Форма за креирање новог филма

Основни сценарио СК

1. Администратор **уноси** податке о филму. (АПУСО)
2. Администратор **контролише** да ли је коректно унео податке о филму. (АНСО)
3. Администратор **позива** систем да запамти филм. (АПСО)

Опис акције: Администратор притиском на дугме “*Create*” позива системску операцију *Add(Movie)*

4. Систем **памти** филм. (СО)
5. Систем **приказује** администратору запамћени филм и поруку: “Систем је запамтио филм”.(ИА)



Слика 47. Страница успешно креираног филма

Алтернативна сценарија

- 5.1. Уколико систем не може да креира филм он приказује администратору поруку “Систем не може да запамти филм”. (ИА)

A screenshot of a web application interface for creating a new movie. The form is titled 'Create new movie' and contains several input fields with validation errors. The 'Name' field has the error 'The Name field is required.' The 'Genres' field has the error 'The Movie must have at least one genre!'. The 'Year' field has the error 'The Year field is required.' The 'Studio' field is set to 'Universal Pictures'. Other fields include 'Synopsis', 'Thumbnail' (with a 'Choose file' button and 'No file chosen' text), 'Trailer', and 'Streaming service' (set to 'Netflix'). The left sidebar and top navigation are identical to the previous screenshot.

Слика 48. Неуспешно креирање филма

СКЗ: Случај коришћења – Креирање стриминг сервиса

Назив СК: Креирање стриминг сервиса

Актери СК: Администратор

Учесници СК: Администратор и систем (програм)

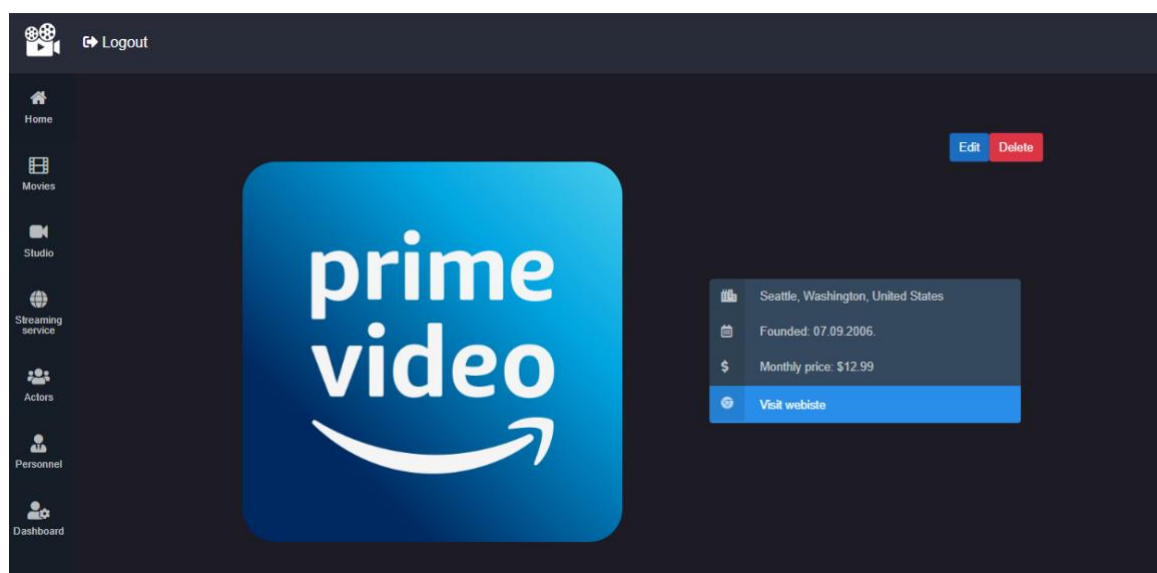
Предуслов: Систем је укључен и администратор је улогован под својом шифром.
Систем приказује форму за рад са стриминг сервисом.

The screenshot shows a web application interface for creating a streaming service. The interface is dark-themed with a sidebar on the left containing navigation icons for Home, Movies, Studio, Streaming service, Actors, Personnel, and Dashboard. The main content area is titled "Create StreamingServiceCreateModel" and contains several input fields: Name, Website, Headquarter, Founded (with a date picker), Price, and Img (with a file upload button). A "Create" button is at the bottom, and a "Back to List" link is below it.

Слика 49. Форма за креирање стриминг сервиса

Основни сценарио СК

1. Администратор **уноси** податке о стриминг сервису. (АПУСО)
2. Администратор **контролише** да ли је коректно унео податке о стриминг сервису. (АНСО)
3. Администратор **позива** систем да запамти стриминг сервис. (АПСО)
Опис акције: Администратор притиском на дугме “*Create*” позива системску операцију *Add(StreamingService)*
4. Систем **памти** стриминг сервис. (СО)
5. Систем **приказује** администратору запамћени студио и поруку: “Систем је запамтио стриминг сервис”.(ИА)



Слика 50. Успешно креирање стриминг сервис (преусмеравање на страницу новокреираног стриминг сервиса)

Алтернативна сценарија

- 5.1. Уколико систем не може да креира стриминг сервис он приказује администратору поруку “Систем не може да запамти стриминг сервис”. (ИА)


Create

StreamingServiceCreateModel

Name

Website

Headquarter

Founded
 

The value " " is invalid.

Price

Img

[Back to List](#)

Слика 51. Неуспешно креирање стриминг сервиса

СК4: Случај коришћења – Креирање особља

Назив СК: Креирање особља

Актери СК: Администратор

Учесници СК: Администратор и систем (програм)

Предуслов: Систем је укључен и администратор је улогован под својом шифром.
Систем приказује форму за рад са особљем.

Logout

Home

Movies

Studio

Streaming service

Actors

Personnel

Dashboard

Create new Personnel

Personnel

First name

Last name

Born

mm/dd/yyyy

Country

United States

Trademark

Image

Choose File No file chosen

Create

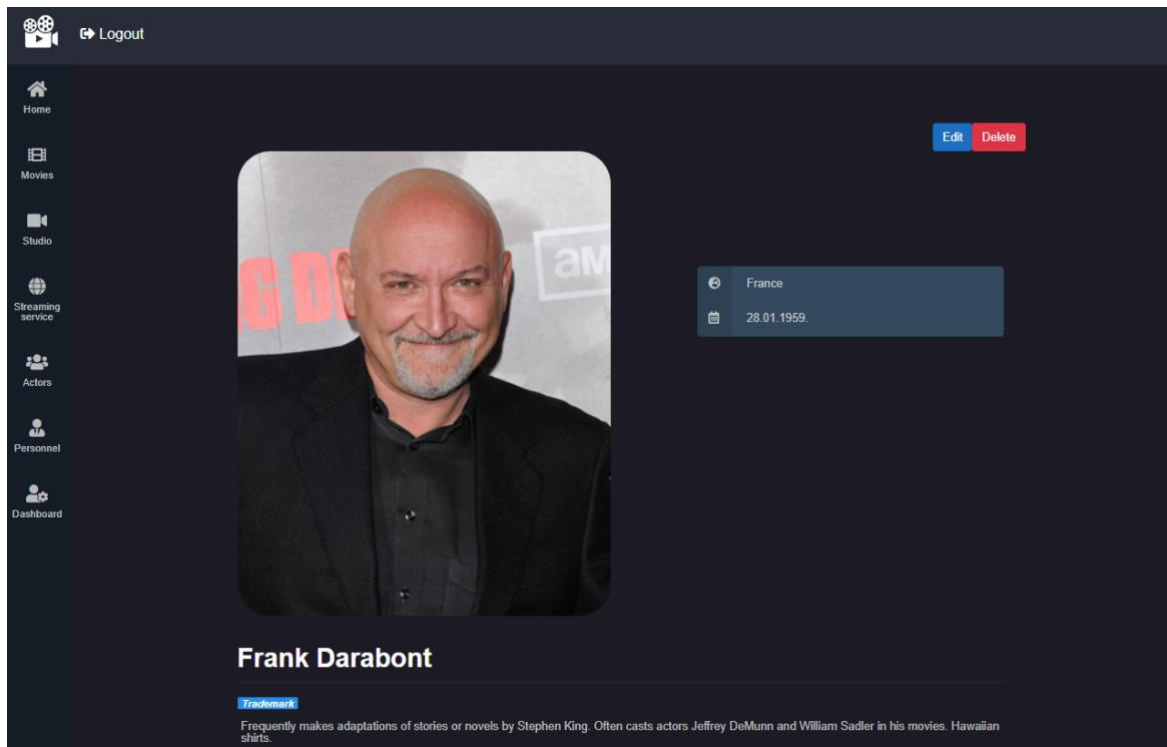
Back to List

Слика 52. Форма за креирање особља

Основни сценарио СК

1. Администратор **уноси** податке о особљу. (АПУСО)
2. Администратор **контролише** да ли је коректно унео податке о особљу. (АНСО)
3. Администратор **позива** систем да запамти особље. (АПСО)
Опис акције: Администратор притиском на дугме “*Create*” позива системску операцију *Add(Personnel)*
4. Систем **памти** особље. (СО)

5. Систем **приказује** администратору запамћено особље и поруку: “Систем је запамтио особље”. (ИА)



Слика 53. Успешно креирано особље (преусмеравање на страницу особља)

Алтернативна сценарија

- 5.1. Уколико систем не може да креира особље он приказује администратору поруку “Систем не може да запамти особље”. (ИА)

Create new Personnel

Personnel

First name

The FirstName field is required.

Last name

The LastName field is required.

Born

The value " is invalid.

Country

Trademark

Image

Create

Слика 54. Неуспешно креирање особља

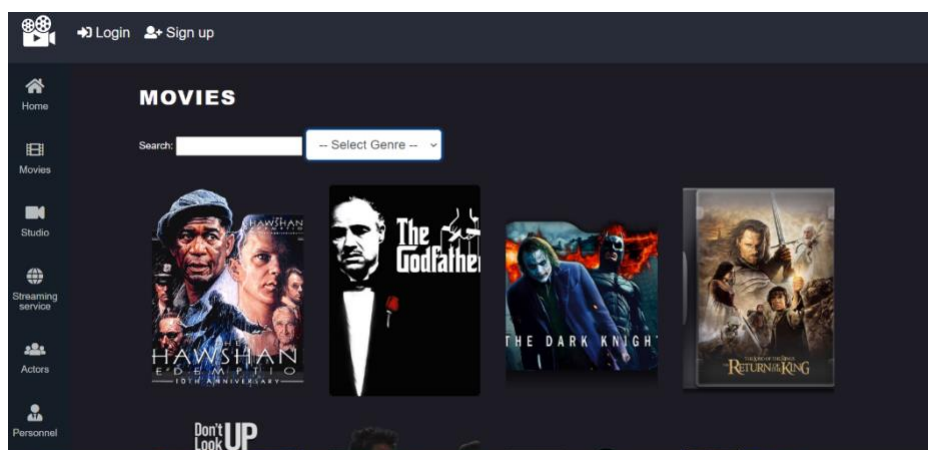
СК5: Случај коришћења – Претраживање филмова

Назив СК: Претраживање филмова

Актори СК: Корисник/Администратор

Учесници СК: Корисник/Администратор и систем (програм)

Предуслов: Систем је укључен и корисник/администратор је улогован под својом шифром. Систем приказује форму за рад са филмовима. Учитана је листа филмова.



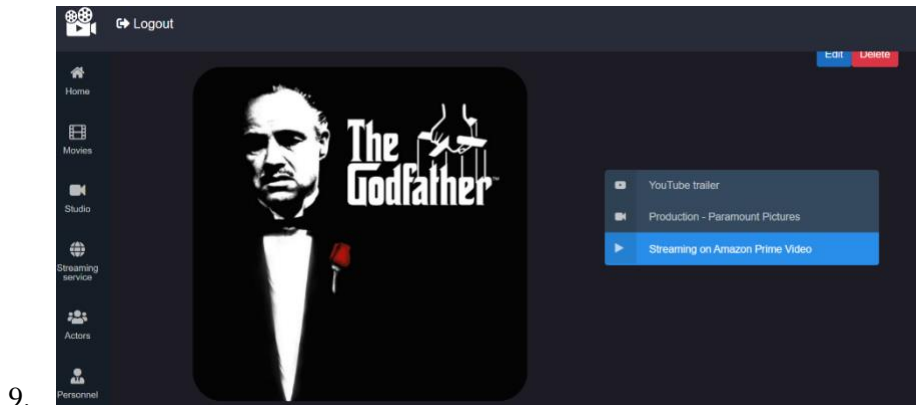
Слика 55. Страница са филмовима

Основни сценарио СК

1. Корисник/Администратор **уноси** вредност по којој претражује филмове. (АПУСО)
2. Корисник/Администратор **позива** систем да нађе филмове по задатој вредности. (АПСО)
Опис акције: Корисник/Администратор уносом текста у “Textbox” позива системску операцију *Find(Movie)*
3. Систем **тражи** филмове по задатој вредности. (СО)
4. Систем **приказује** кориснику/администратору податке о филмовима и поруку: “Систем је нашао филмове по задатој вредности”. (ИА)
5. Корисник/Администратор **бира** филм. (АПУСО)
6. Корисник/Администратор **позива** систем да прочита филм. (АПСО)

Опис акције: Корисник/Администратор притиском на филм позива системску операцију *GetSingle(Movie)*

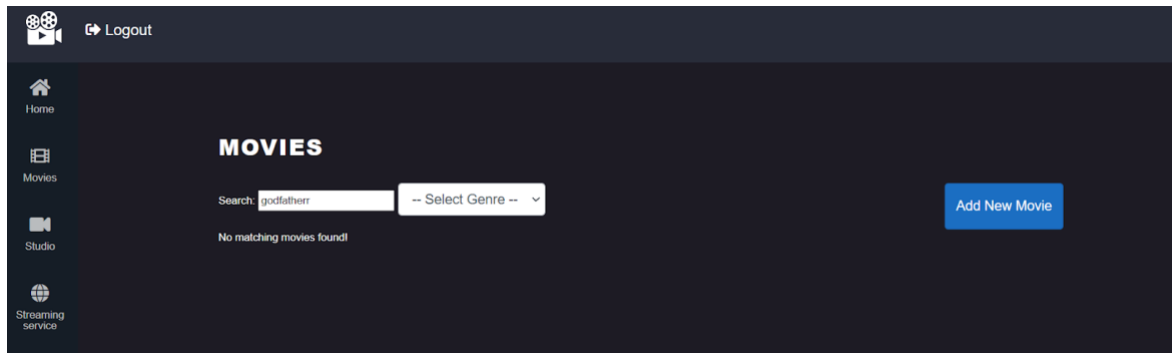
7. Систем **учитава** филм. (СО)
8. Систем **приказује** кориснику/администратору податке о филму и поруку: “Систем је учитао филм”. (ИА)



Слика 56. Систем је успешно учитао филм

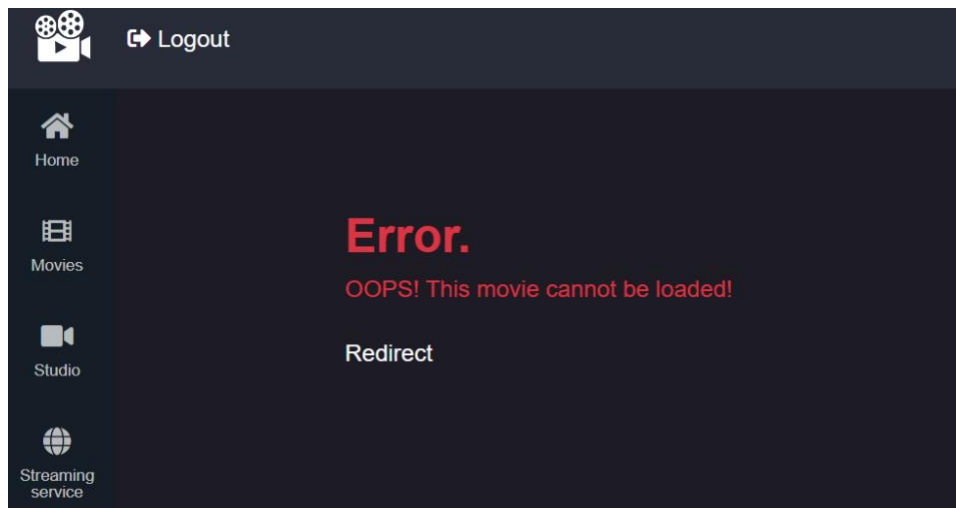
10. Алтернативна сценарија

4.1 Уколико систем не може да нађе филмове он приказује кориснику/администратору поруку: “Систем не може да нађе филмове по задатој вредности”. Прекида се извршење сценарија. (ИА)



Слика 57. Систем није пронашао филмове по задатој вредности

8.1 Уколико систем не може да учита филм он приказује кориснику/администратору поруку “Систем не може да учита филм”. (ИА)



Слика 58. Систем не може да учита филм

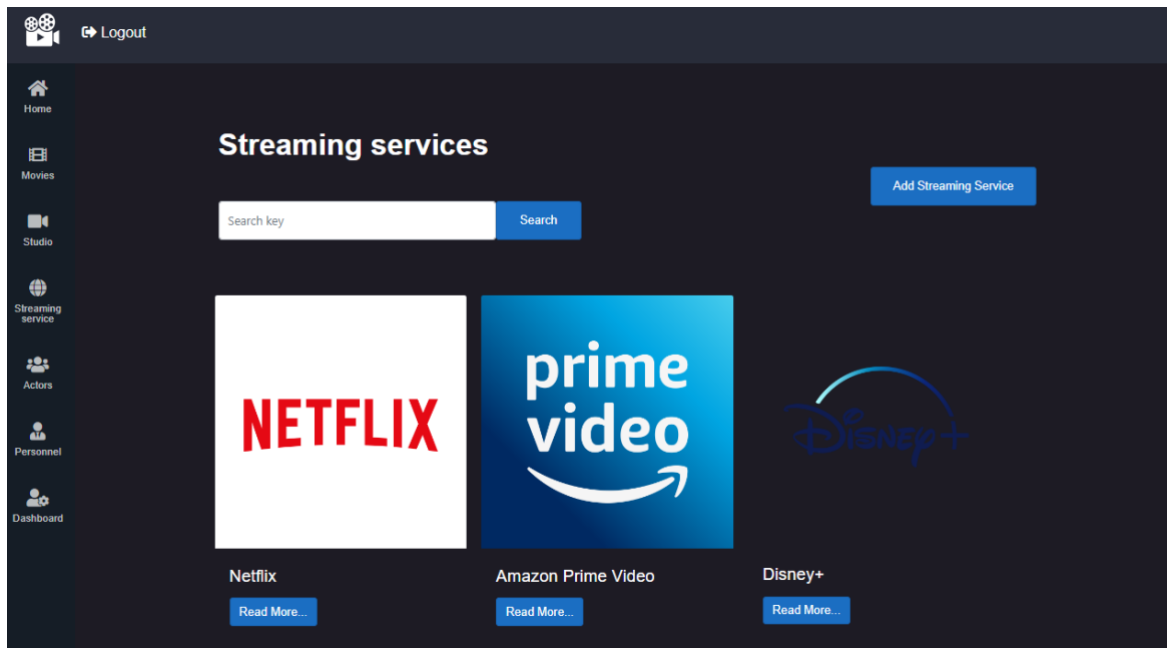
СК6: Случај коришћења – Промена података стриминг сервиса

Назив СК: Промена података стриминг сервиса

Актери СК: Администратор

Учесници СК: Администратор и систем (програм)

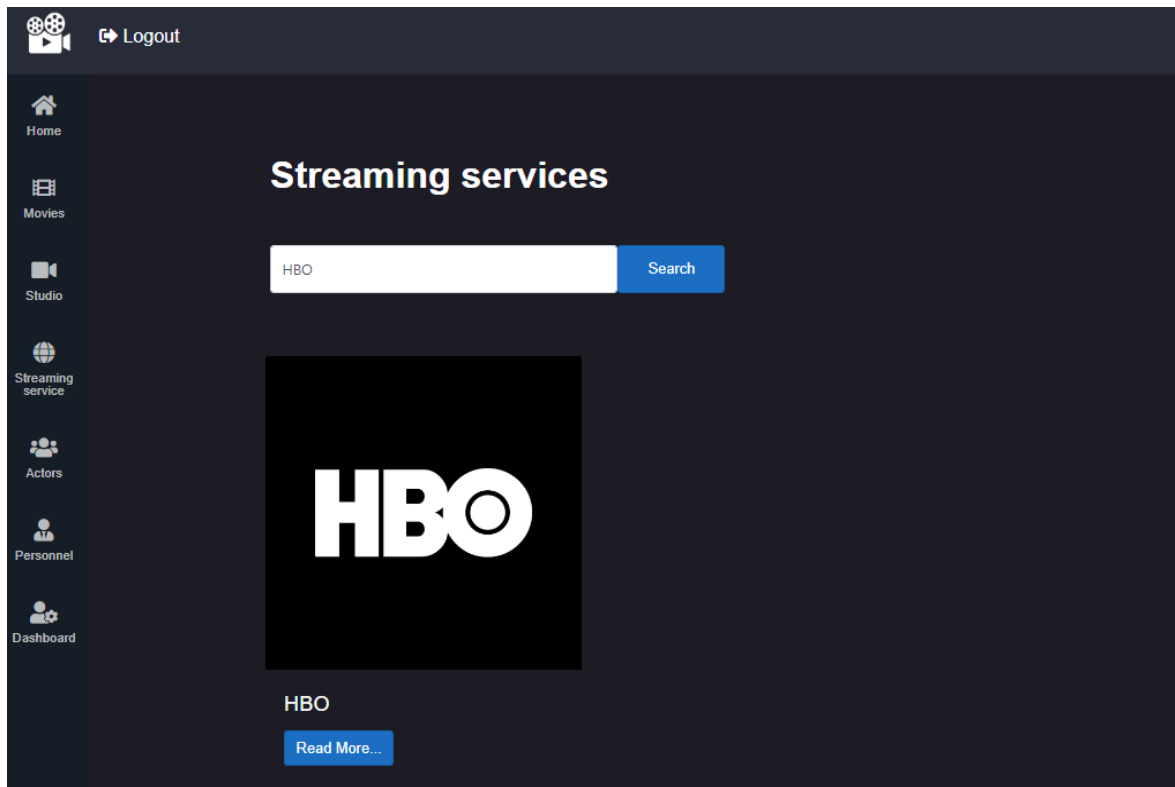
Предуслов: Систем је укључен и администратор је улогован под својом шифром. Систем приказује форму за рад са стриминг сервисом. Учитана је листа стриминг сервиса.



Слика 59. Страница са стриминг сервисима

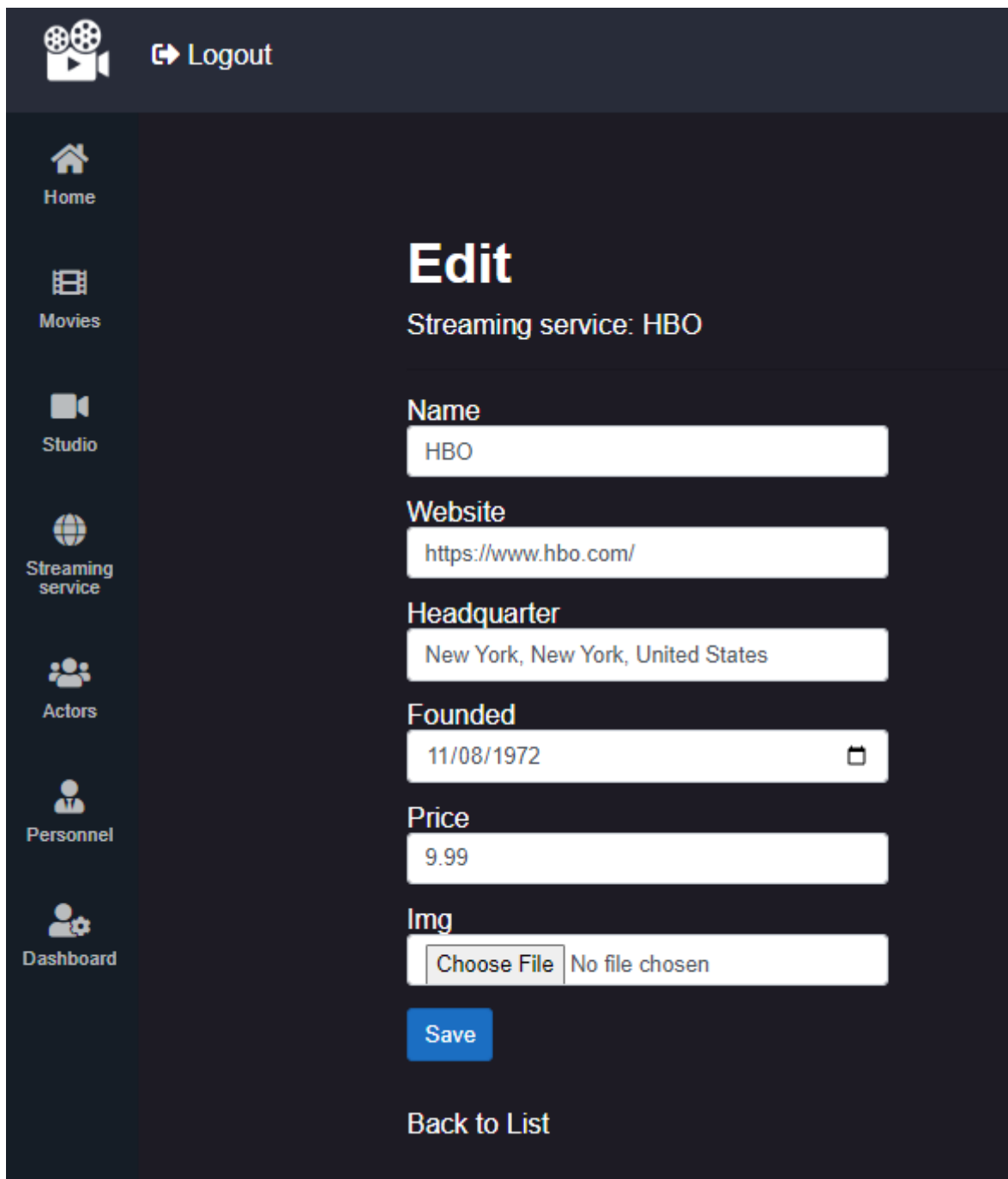
Основни сценарио СК

1. Администратор уноси вредност по којој претражује стриминг сервисе. (АПУСО)
2. Администратор позива систем да нађе стриминг сервис по задатој вредности. (АПСО)
Опис акције: Администратор притиском на дугме “*Search*” позива системску операцију *Find(StreamingService)*
3. Систем тражи стриминг сервисе по задатој вредности. (СО)
4. Систем приказује студије и поруку: “Систем је нашао стриминг сервисе по задатој вредности”. (ИА)



Слика 60. Приказ претражених стриминг сервиса

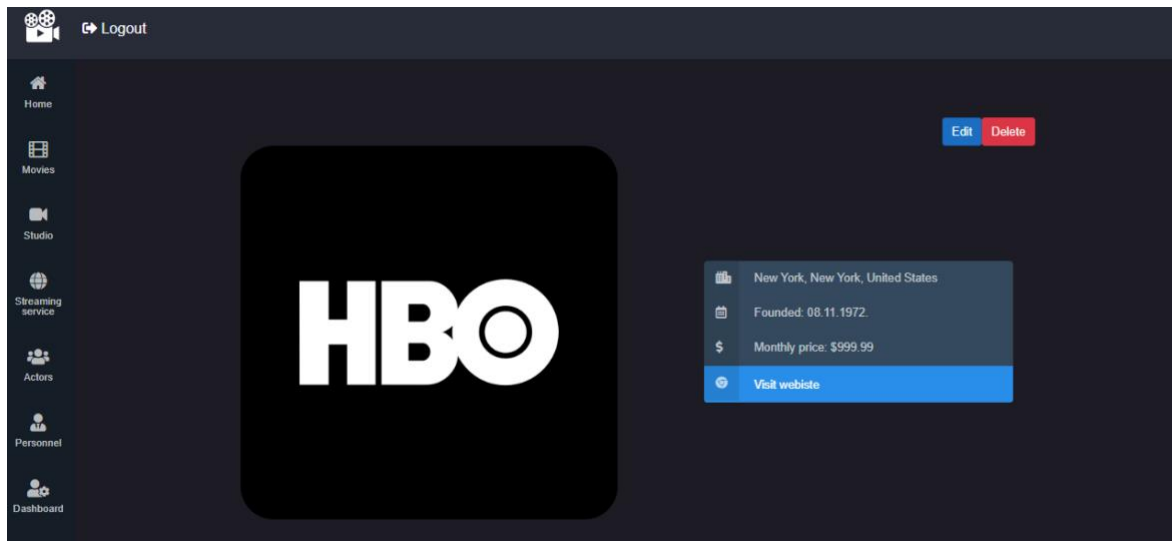
5. Администратор бира стриминг сервис. (АПУСО)
6. Администратор позива систем да учита стриминг сервис.(АПСО)
Опис акције: Администратор притиском на дугме “*Edit*” позива системску операцију *GetSingle(StreamingService)*
7. Систем учитава стриминг сервис. (СО)
8. Систем показује администратору податке о стриминг сервису и поруку “Систем је учитао стриминг сервис“. (ИА)



Слика 61. Успешно учитан стриминг сервис

9. Администратор **уноси(мења)** податке о стриминг сервису. (АПУСО)
10. Администратор **контролише** да ли је коректно унео податке о стриминг сервису. (АНСО)
11. Администратор **позива** систем да запамти стриминг сервис. (АПСО)
Опис акције: Администратор притиском на дугме “Save” позива системску операцију *Update(StreamingService)*
12. Систем **памти** стриминг сервис. (СО)

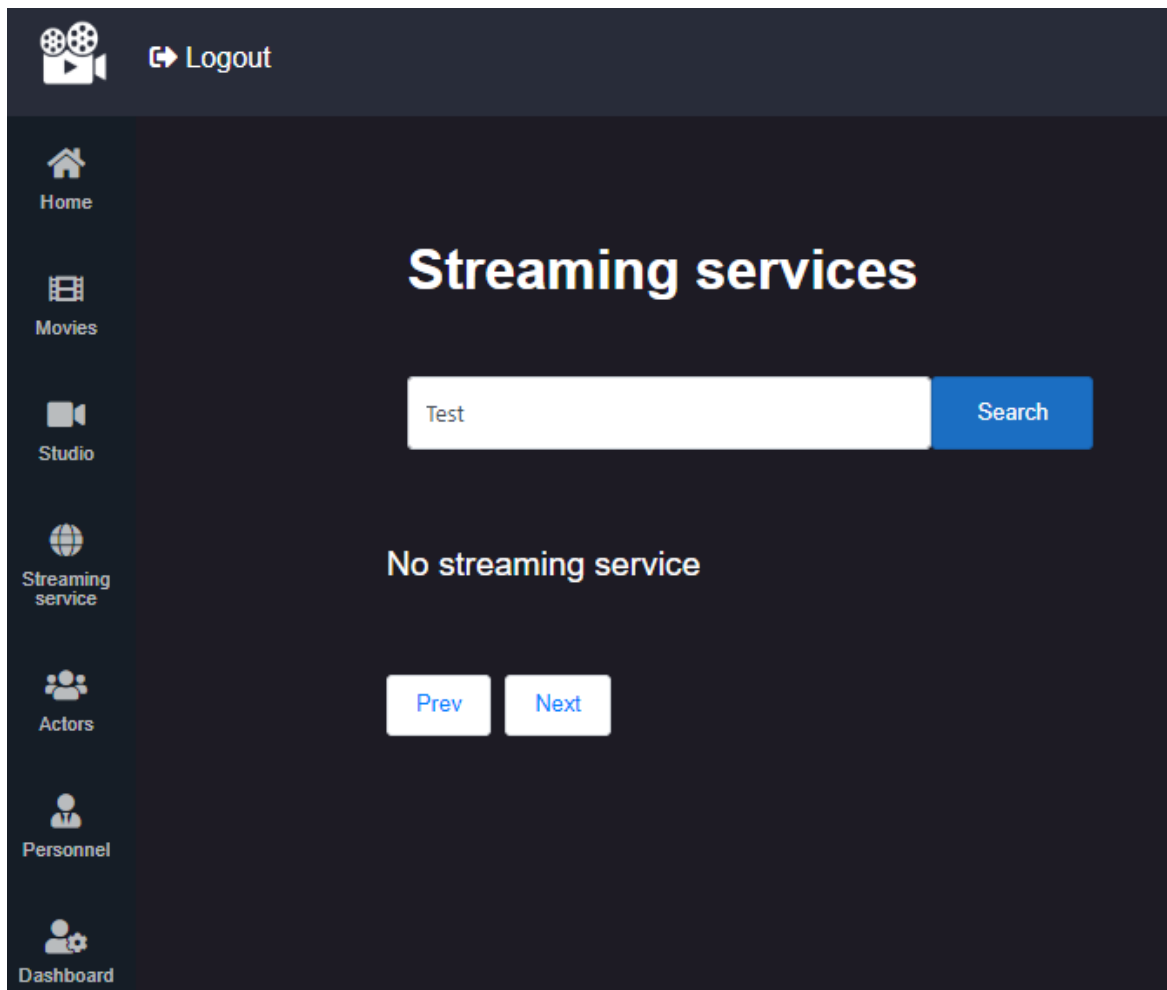
13. Систем **приказује** администратору запамћени стриминг сервис и поруку:
“Систем је запамтио стриминг сервис”.(ИА)



Слика 62. Успешно измењен стриминг сервис

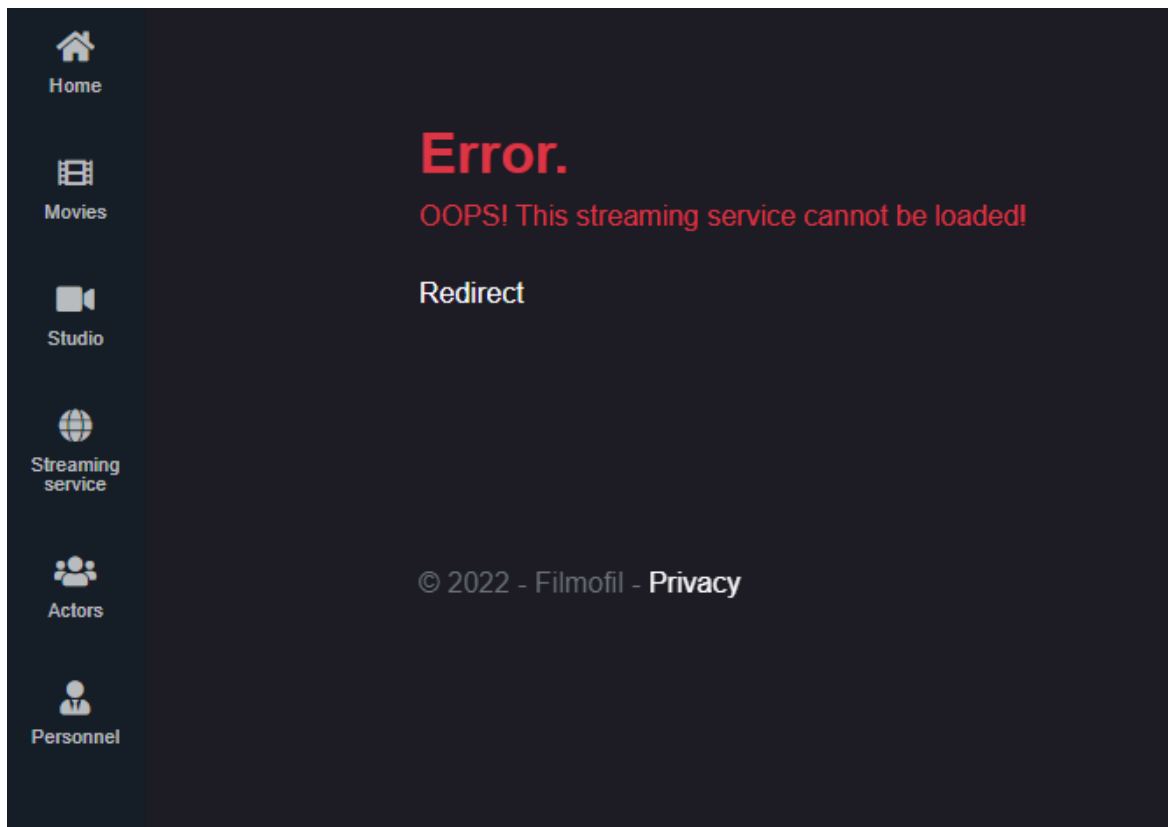
Алтернативна сценарија

4.1. Уколико систем не може да нађе стриминг сервис он приказује администратору поруку: “Систем не може да нађе стриминг сервис по задатој вредности”. Прекида се извршење сценарија. (ИА)



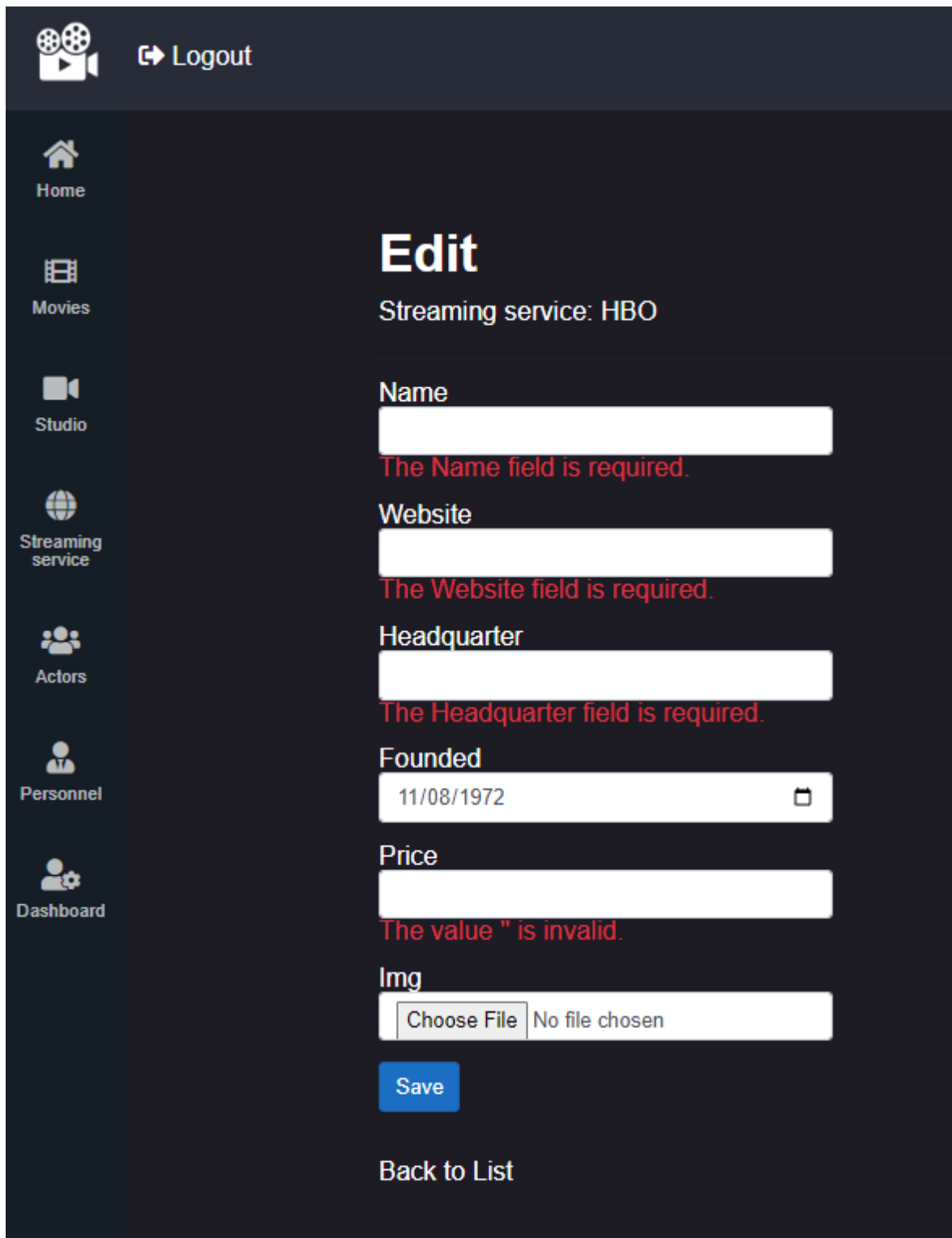
Слика 63. Систем није пронашао стриминг сервис по задатој вредности

8.1. Уколико систем не може да учита стриминг сервис он приказује администратору поруку “Систем не може да учита стриминг сервис”. Прекида се извршење сценариа. (ИА)



Слика 64. Неуспешно учитавање стриминг сервиса

13.1. Уколико систем не може да запамти податке о стриминг сервису он приказује администратору поруку “Систем не може да запамти стриминг сервис”. (ИА)



Слика 65. Неуспешна измена стриминг сервиса

СК7: Случај коришћења – Промена података особља

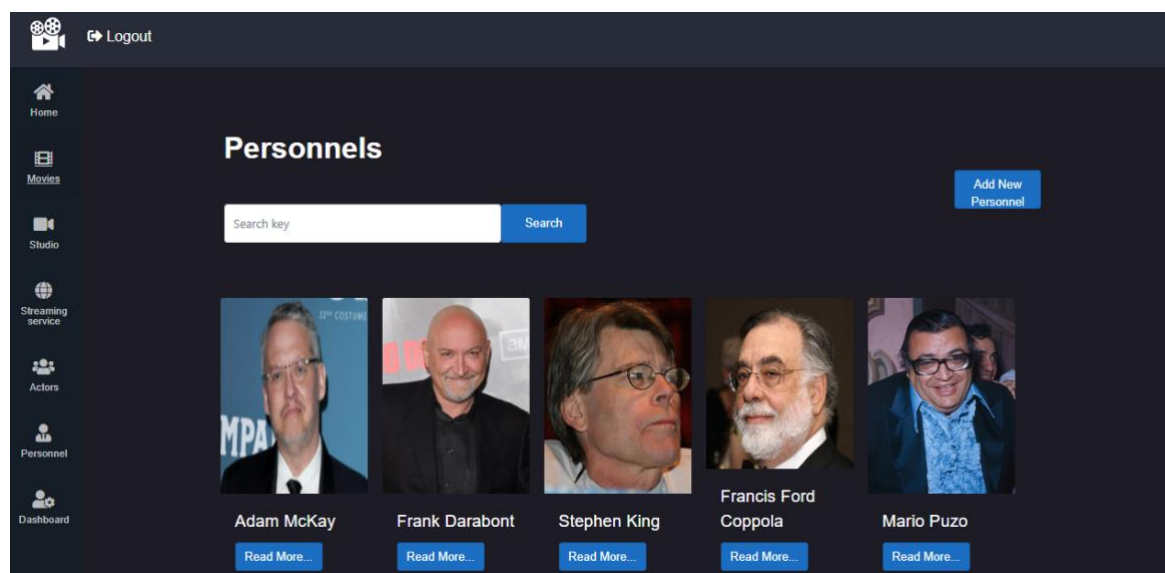
Назив СК: Промена података особља

Актери СК: Администратор

Учесници СК: Администратор и систем (програм)

Предуслов: Систем је укључен и администратор је улогован под својом шифром.

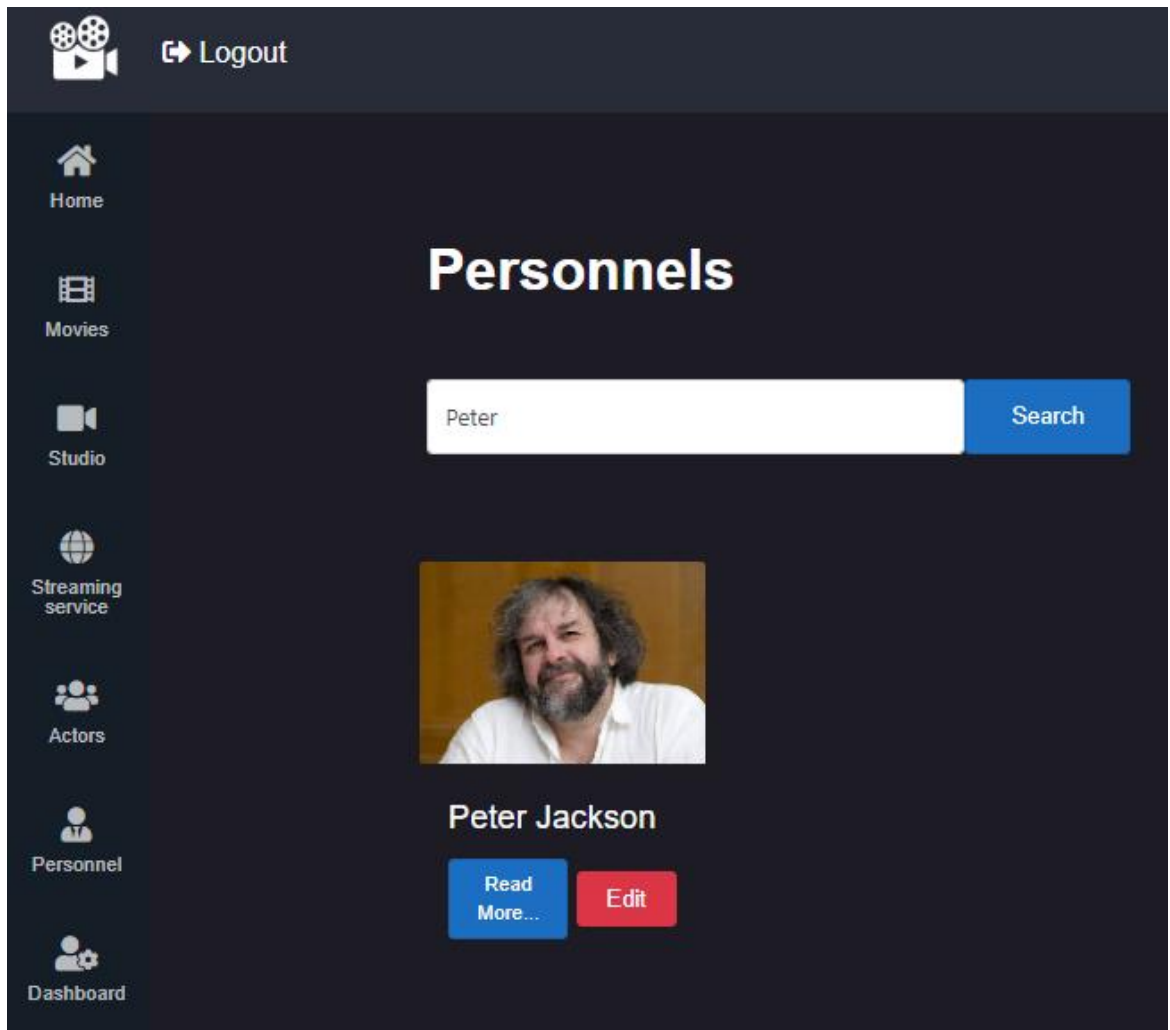
Систем приказује форму за рад са особљем. Учитана је листа особља.



Слика 66. Страница са особљем

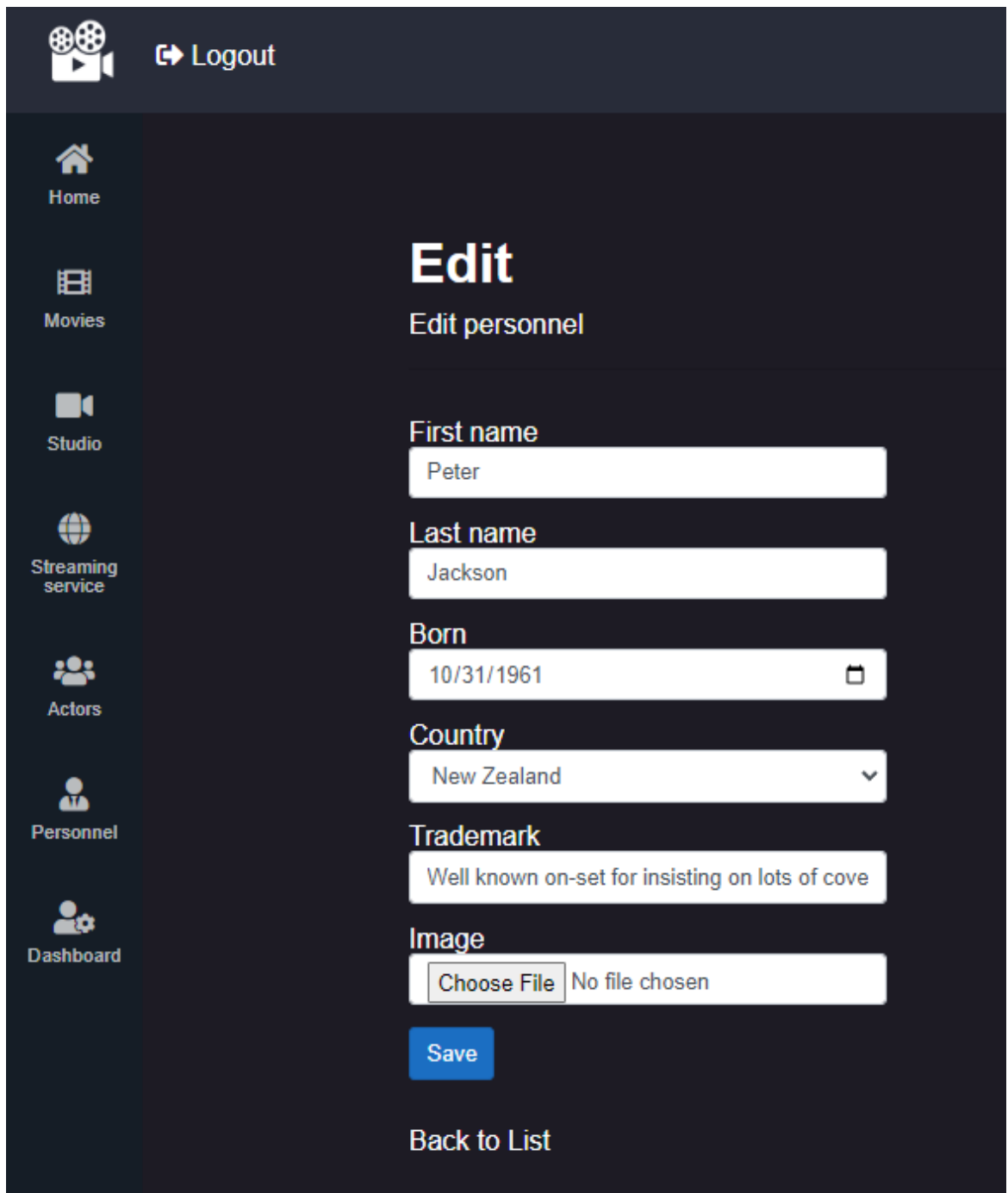
Основни сценарио СК

1. Администратор **уноси** вредност по којој претражује особље. (АПУСО)
2. Администратор **позива** систем да нађе особље по задатој вредности. (АПСО)
Опис акције: Администратор притиском на дугме “*Search*” позива системску операцију *Find(Personnel)*
3. Систем **тражи** особље по задатој вредности. (СО)
4. Систем **приказује** глумце и поруку: “Систем је нашао особље по задатој вредности”. (ИА)



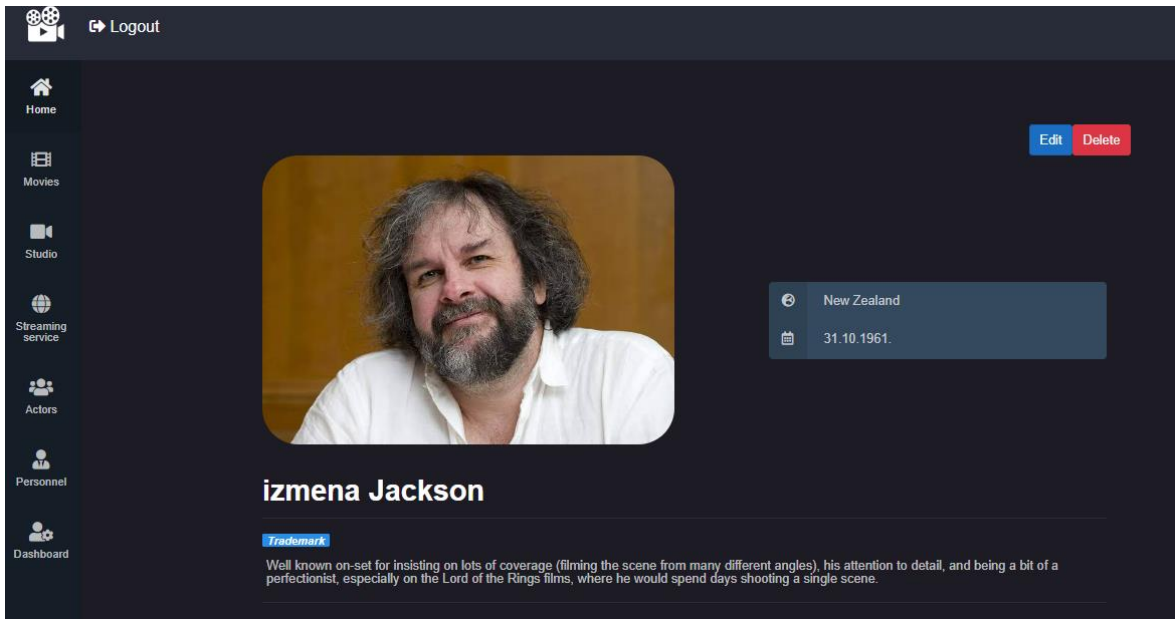
Слика 67. Систем је нашао особље по задатој вредности

5. Администратор **бира** особље. (АПУСО)
6. Администратор **позива** систем да учита особље.(АПСО)
Опис акције: Администратор притиском на дугме “*Edit*” позива системску операцију *GetSingle(Personnel)*
7. Систем **учитава** особље. (СО)
8. Систем **показује** администратору податке о особљу и поруку “Систем је учитао особље“. (ИА)



Слика 68. Успешно учитано особље

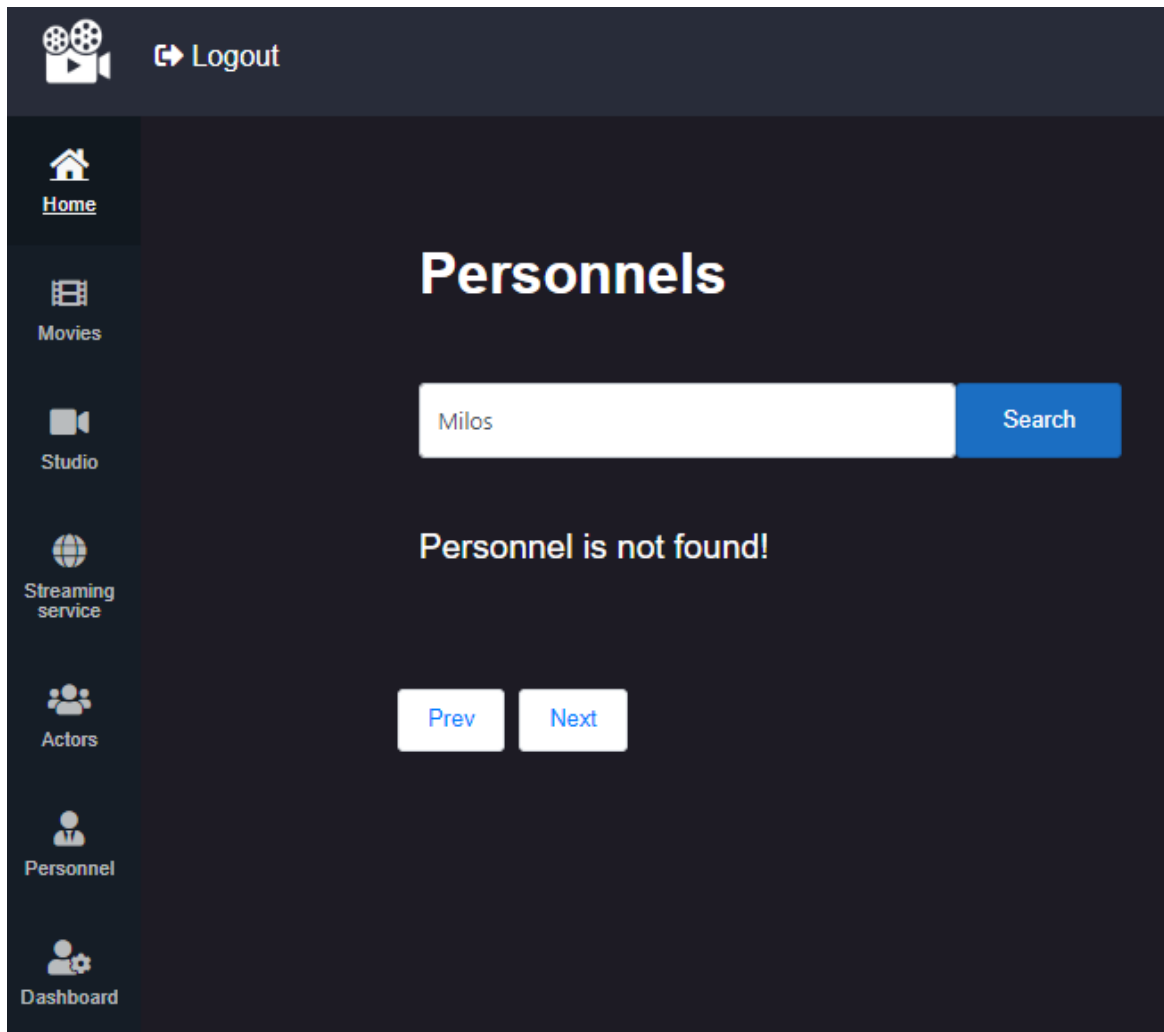
9. Администратор **уноси(мења)** податке о особљу. (АПУСО)
10. Администратор **контролише** да ли је коректно унео податке о особљу. (АНСО)
11. Администратор **позива** систем да запамти особље. (АПСО)
12. Систем **памти** особље. (СО)
13. Систем **приказује** администратору запамћеног припадника особља и поруку:
“Систем је запамтио особље”. (ИА)



Слика 69. Успешно измењено особље

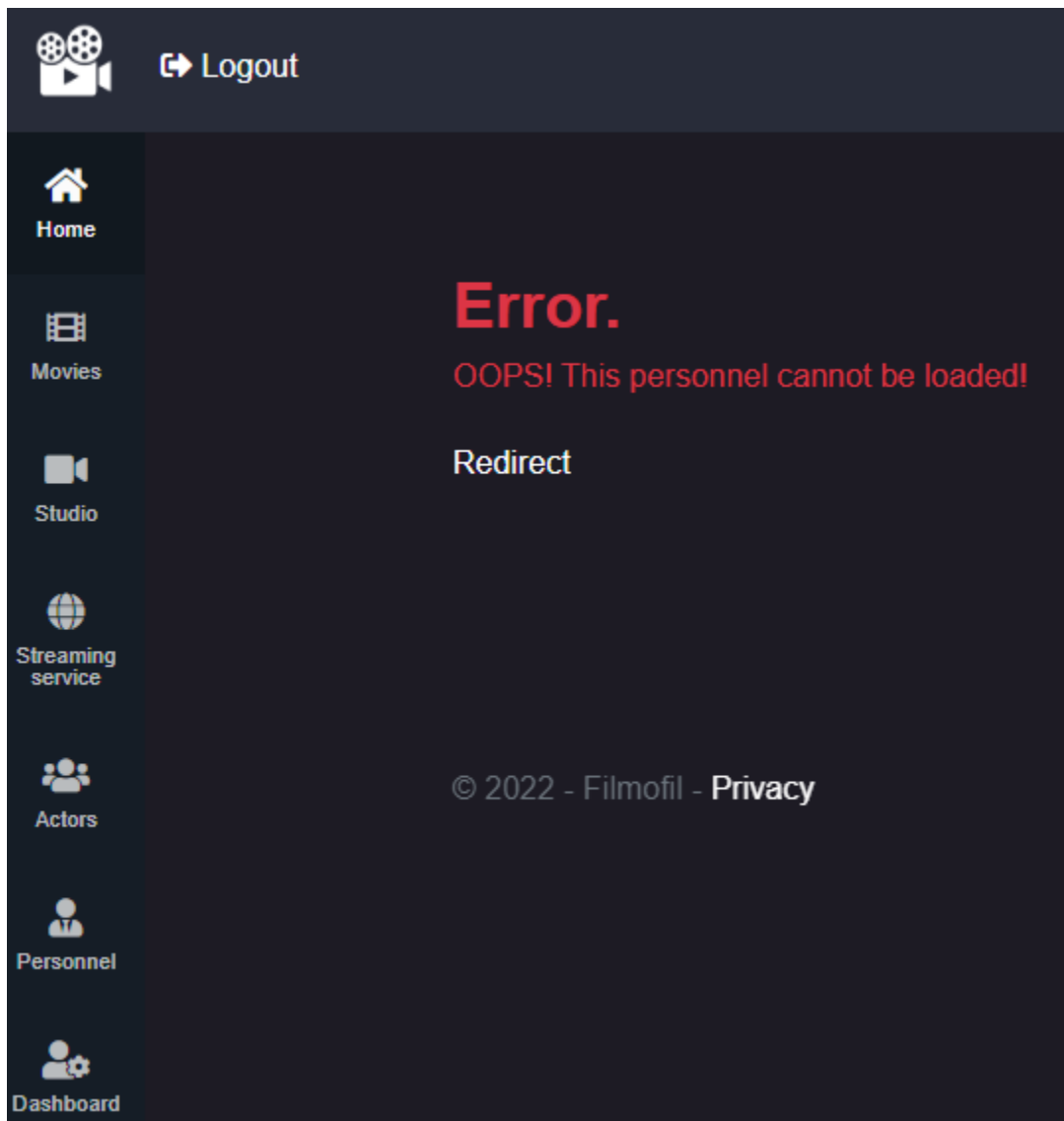
Алтернативна сценарија

4.1. Уколико систем не може да нађе особље он приказује администратору поруку: “Систем не може да нађе особље по задатој вредности”. Прекида се извршење сценарија. (ИА)



Слика 70. Систем није пронашао особље по задатој вредности

8.1. Уколико систем не може да учита припадника особља он приказује администратору поруку “Систем не може да учита припадника особља”. Прекида се извршење сценарија. (ИА)



Слика 71. Неуспешно учитавање особља

13.1. Уколико систем не може да запамти податке о припаднику особља он приказује администратору поруку “Систем не може да запамти припадника

особља”. (ИА)

Logout

Home

Movies

Studio

Streaming service

Actors

Personnel

Dashboard

Edit

Edit personnel

First name

The FirstName field is required.

Last name

The LastName field is required.

Born

10/31/1961

Country

New Zealand

Trademark

Well known on-set for insisting on lots of cove

Image

Choose File No file chosen

Save

Back to List

Слика 72. Систем не може да запамти особље

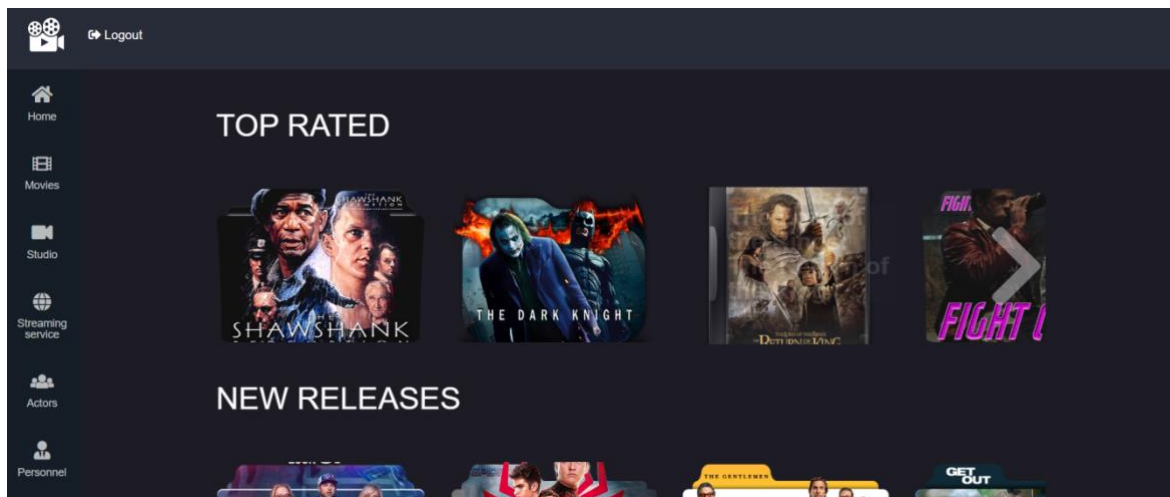
СК8: Случај коришћења – Брисање филма

Назив СК: Брисање филма

Актери СК: Администратор

Учесници СК: Администратор и систем (програм)

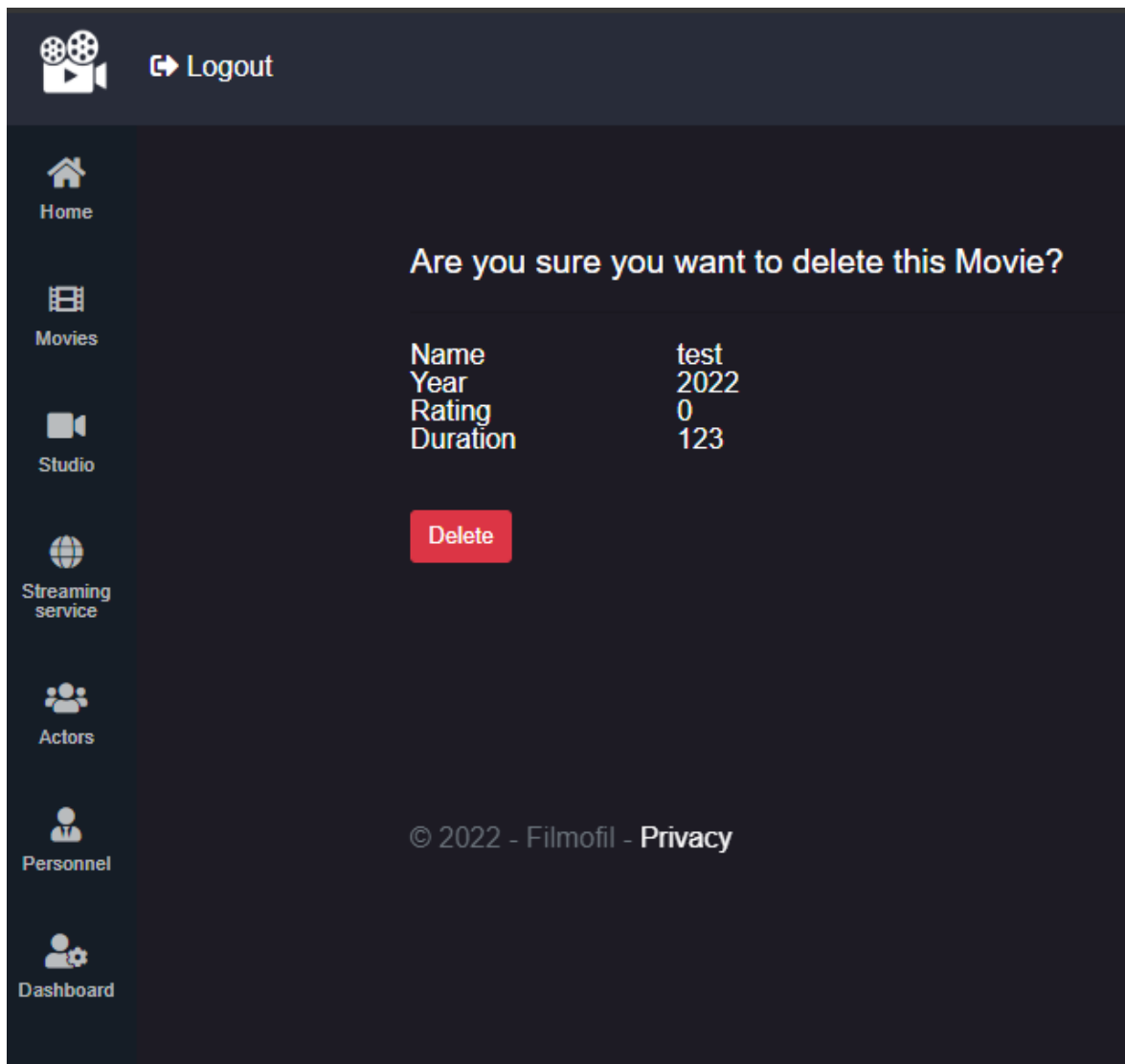
Предуслов: Систем је укључен и администратор је улогован под својом шифром. Систем приказује форму за рад са филмовима. Учитана је листа филмова.



Слика 73. Страница са филмовима

Основни сценарио СК

1. Администратор **уноси** вредност по којој претражује филм. (АПУСО)
2. Администратор **позива** систем да нађе филм по задатој вредности. (АПСО)
3. Систем **тражи** филм по задатој вредности. (СО)
4. Систем **приказује** кориснике и поруку: “Систем је нашао филм по задатој вредности”. (ИА)
5. Администратор **бира** филм. (АПУСО)
6. Администратор **позива** систем да обрише филм. (АПСО)



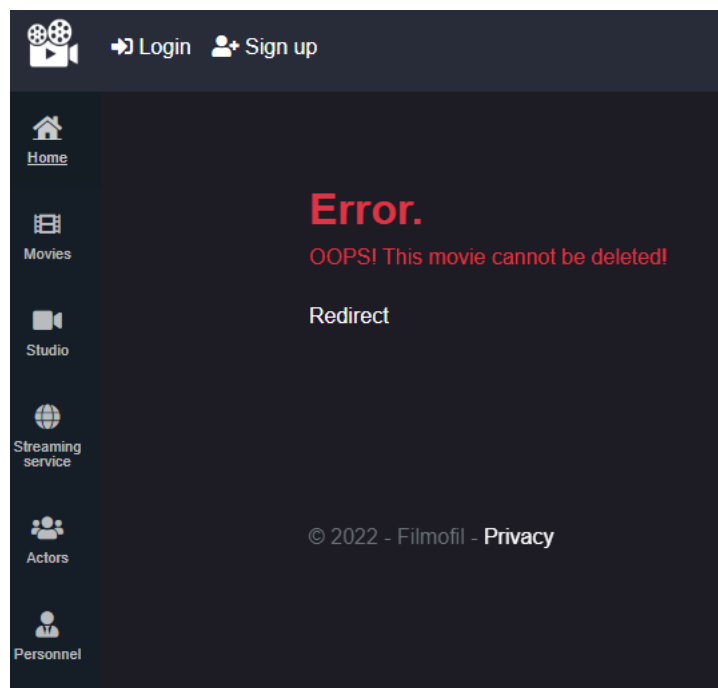
Слика 74. Позив система да обрише филм

7. Систем **брише** филм. (СО)
8. Систем **приказује** администратору поруку: “Систем је обрисао филм”. (ИА)

Алтернативна сценарија

4.1. Уколико систем не може да нађе филм он приказује администратору поруку: “Систем не може да нађе филм по задатој вредности”. Прекида се извршење сценарија. (ИА)

8.1. Уколико систем не може да обрише филм он приказује администратору поруку “Систем не може да обрише филм”. (ИА)



Слика 75. Неуспешно брисање филма

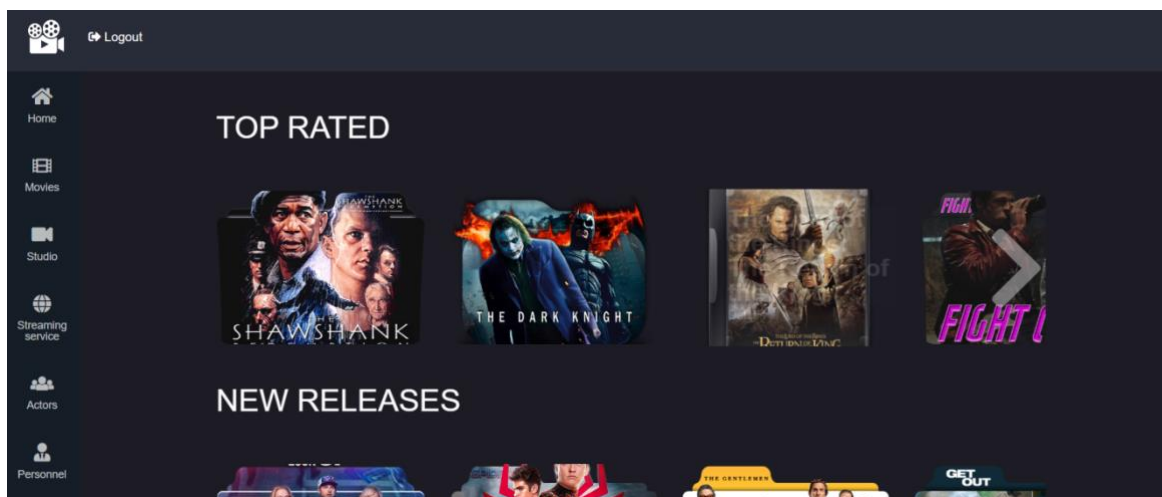
СК9: Случај коришћења – Оцењивање филма

Назив СК: Оцењивање филма

Актери СК: Корисник / Администратор

Учесници СК: Корисник / Администратор и систем (програм)

Предуслов: Систем је укључен и корисник / администратор је улогован под својом шифром. Систем приказује форму за рад са филмовима. Учитана је листа филмова.

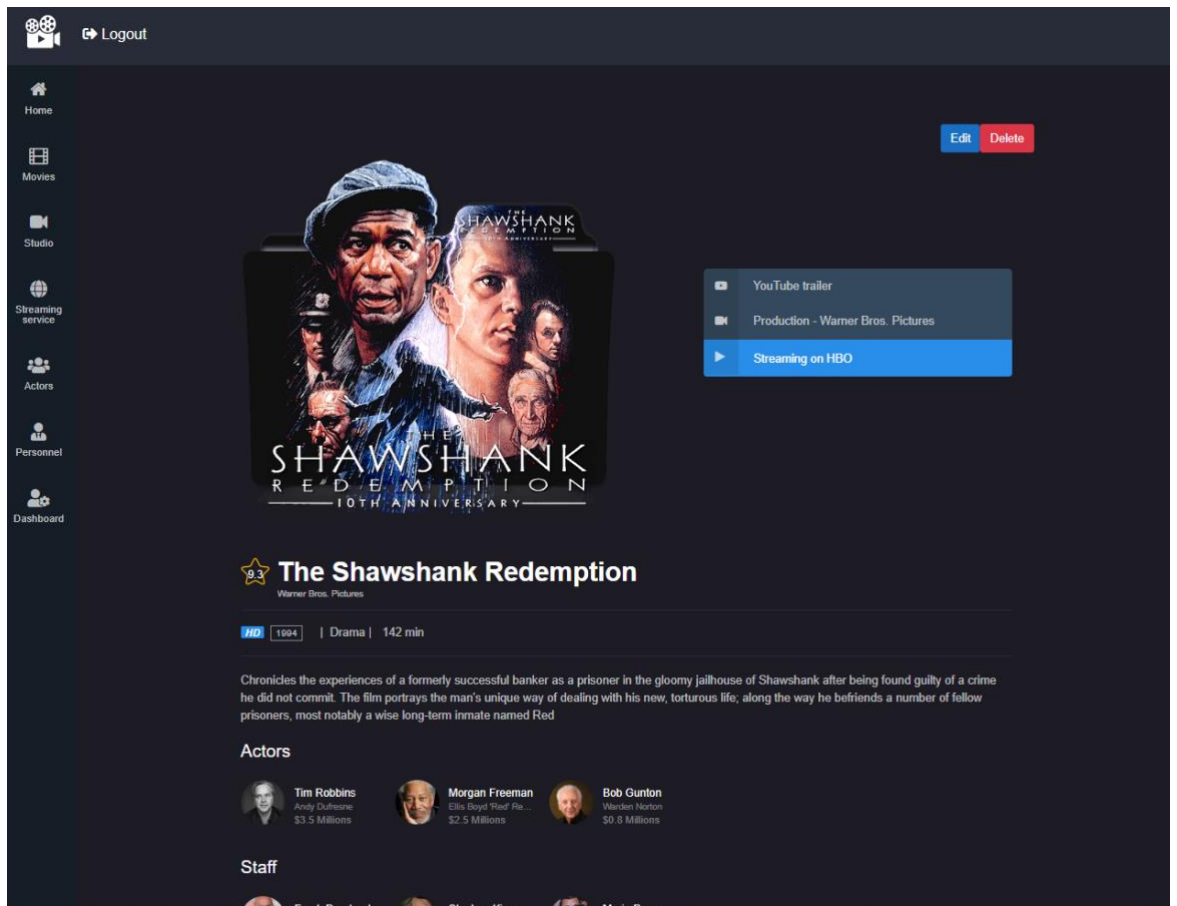


Слика 76. Страница са филмовима

Основни сценарио СК

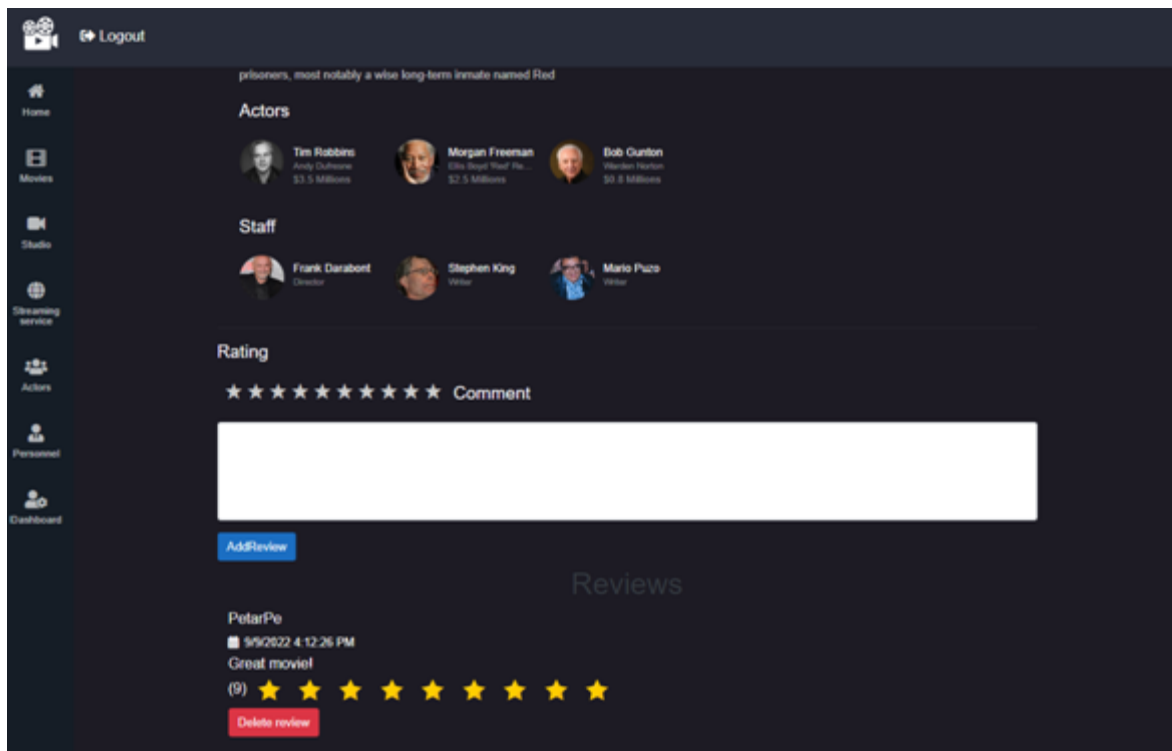
1. Корисник/Администратор **уноси** вредност по којој претражује филмове. (АПУСО)
2. Корисник/Администратор **позива** систем да нађе филмове по задатој вредности. (АПСО)
Опис акције: Администратор уносом текста у Textbox позива системску операцију Find(Movie)
3. Систем **тражи** филмове по задатој вредности. (СО)
4. Систем **приказује** филмове и поруку: “Систем је нашао филмове по задатој вредности”. (ИА)
5. Корисник/Администратор **бира** филм. (АПУСО)
6. Корисник/Администратор **позива** систем да учита филм.(АПСО)

7. Систем **учитава** филм. (СО)
8. Систем **показује** кориснику/администратору податке о филму и поруку “Систем је прочитао филм“. (ИА)



Слика 77. Филм

9. Корисник/Администратор **уноси(мења)**оцену и коментар о филму. (АПУСО)
10. Корисник/Администратор **контролише** да ли је коректно унео оцену и коментар о филму. (АНСО)
11. Корисник/Администратор **позива** систем да запамти оцену и коментар о филму. (АПСО)
12. Систем **памти** оцену и коментар о филму. (СО)
13. Систем **приказује**кориснику/администраторузапамћену оцену и коментар о филму и поруку: “Систем је запамтио оцену и коментар о филму“. (ИА)

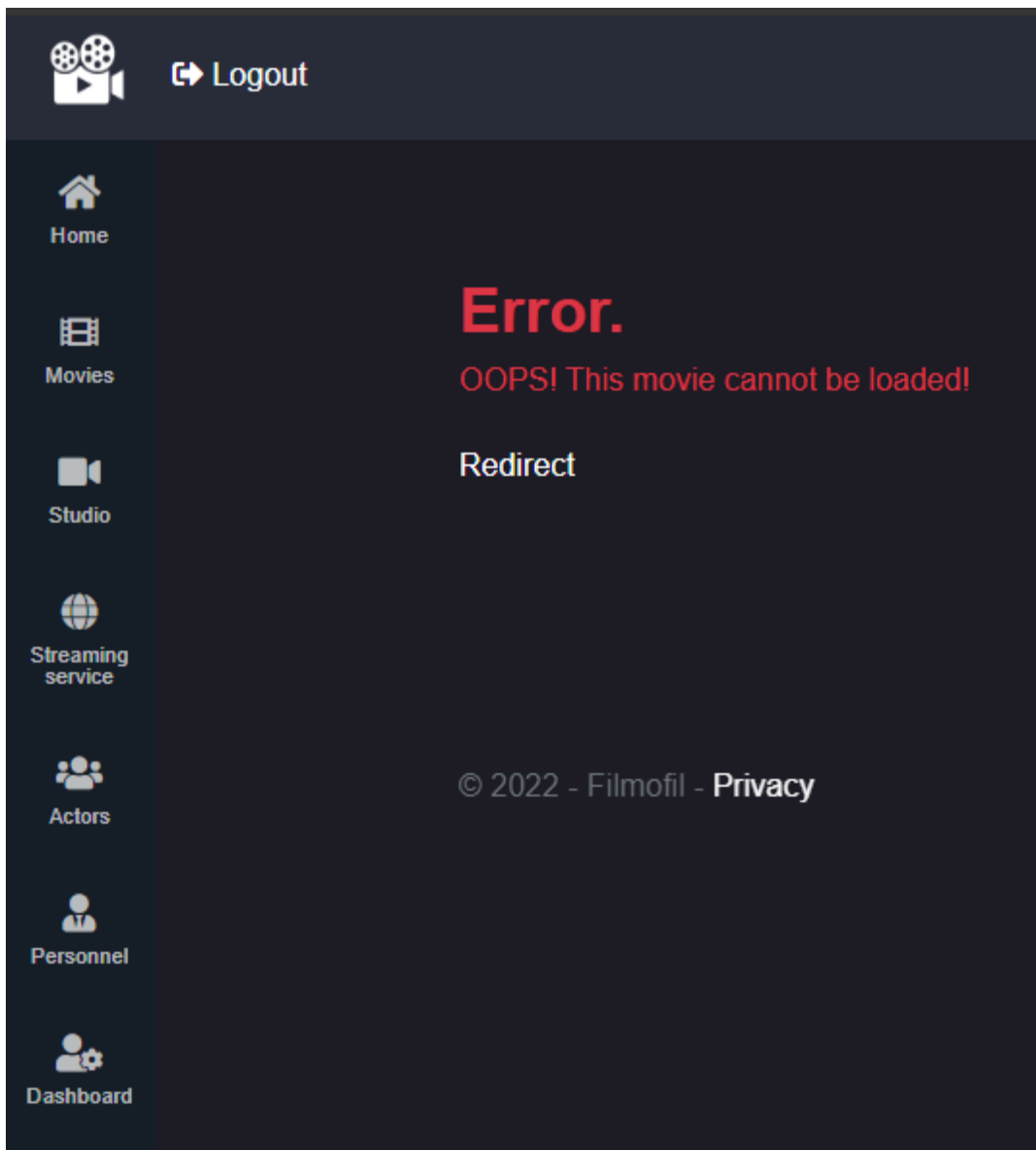


Слика 78. Форма за оцењивање филма

Алтернативна сценарија

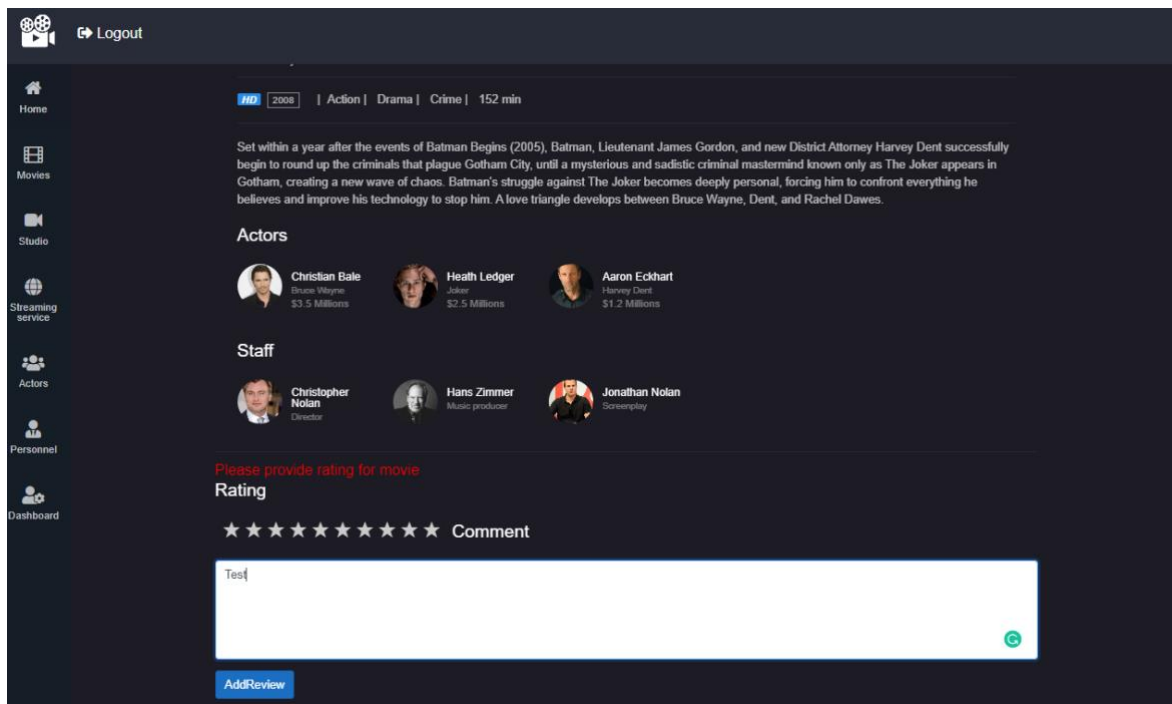
4.1. Уколико систем не може да нађе филмове он приказује кориснику/администратору поруку: “Систем не може да нађе филмове по задатој вредности”. Прекида се извршење сценарија. (ИА)

8.1. Уколико систем не може да учита филм он приказује кориснику/администратору поруку “Систем не може да учита филм”. Прекида се извршење сценарија. (ИА)



Слика 79. Неуспешно учитавање филма

13.1. Уколико систем не може да запамти оцену и коментар о филму он приказује кориснику/администратору поруку “Систем не може да запамти оцену и коментар о филму”. (ИА)



Слика 80. Неисправно попуњена форма

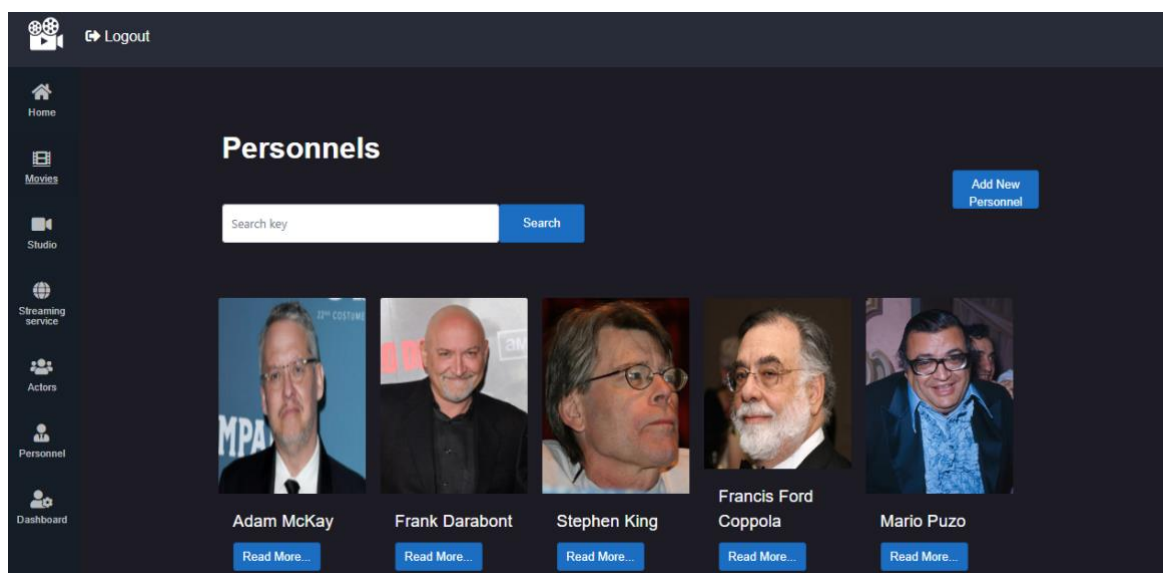
СК10: Случај коришћења – Претраживање особља

Назив СК: Претраживање особља

Актори СК: Корисник/Администратор

Учесници СК: Корисник/Администратор и систем (програм)

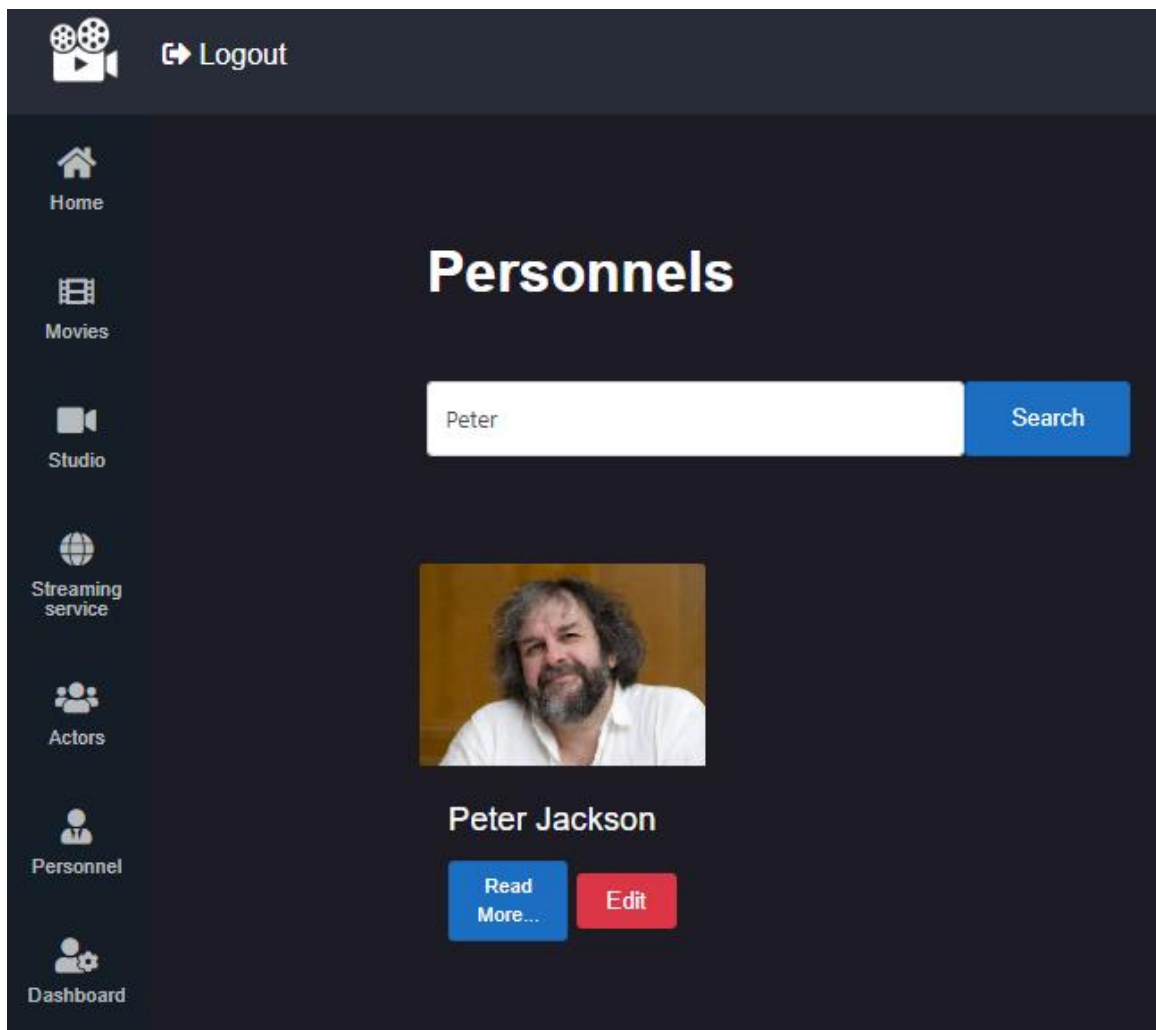
Предуслов: Систем је укључен и корисник/администратор је улогован под својом шифром. Систем приказује форму за рад са особљем. Учитана је листа особља.



Слика 81. Страница са особљем

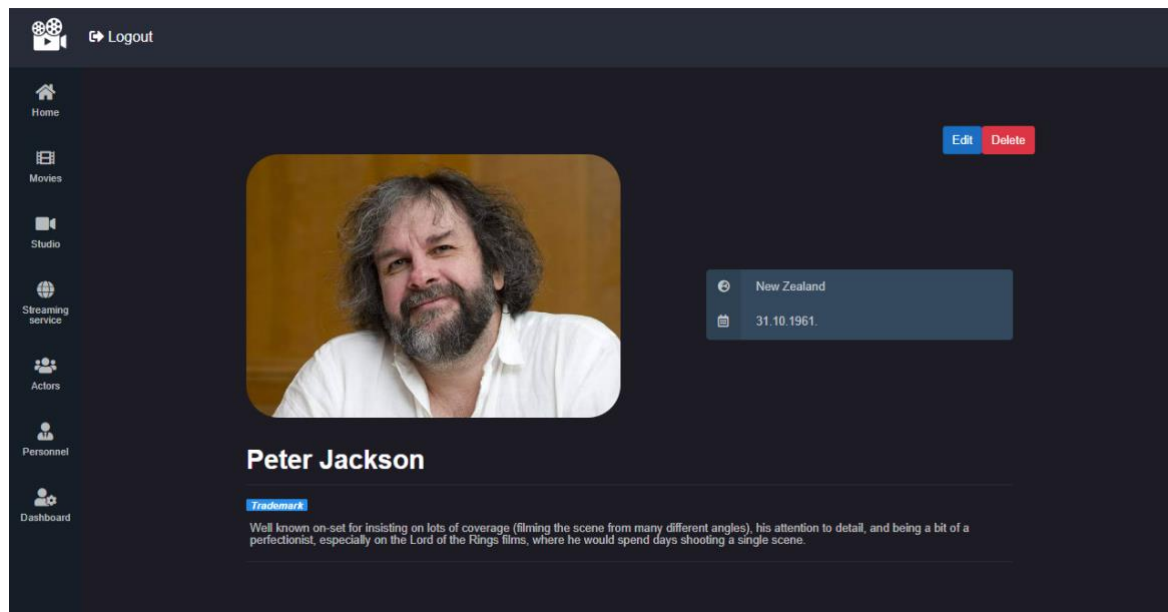
Основни сценарио СК

1. Корисник/Администратор **уноси** вредност по којој претражује особље. (АПУСО)
2. Корисник/Администратор **позива** систем да нађе припаднике особља по задатој вредности. (АПСО)
3. Систем **тражи** припаднике особља по задатој вредности. (СО)
4. Систем **приказује** кориснику/администратору податке о особљу и поруку: “Систем је нашао припаднике особља по задатој вредности”. (ИА)



Слика 82. Систем је нашао особље по задатој вредности

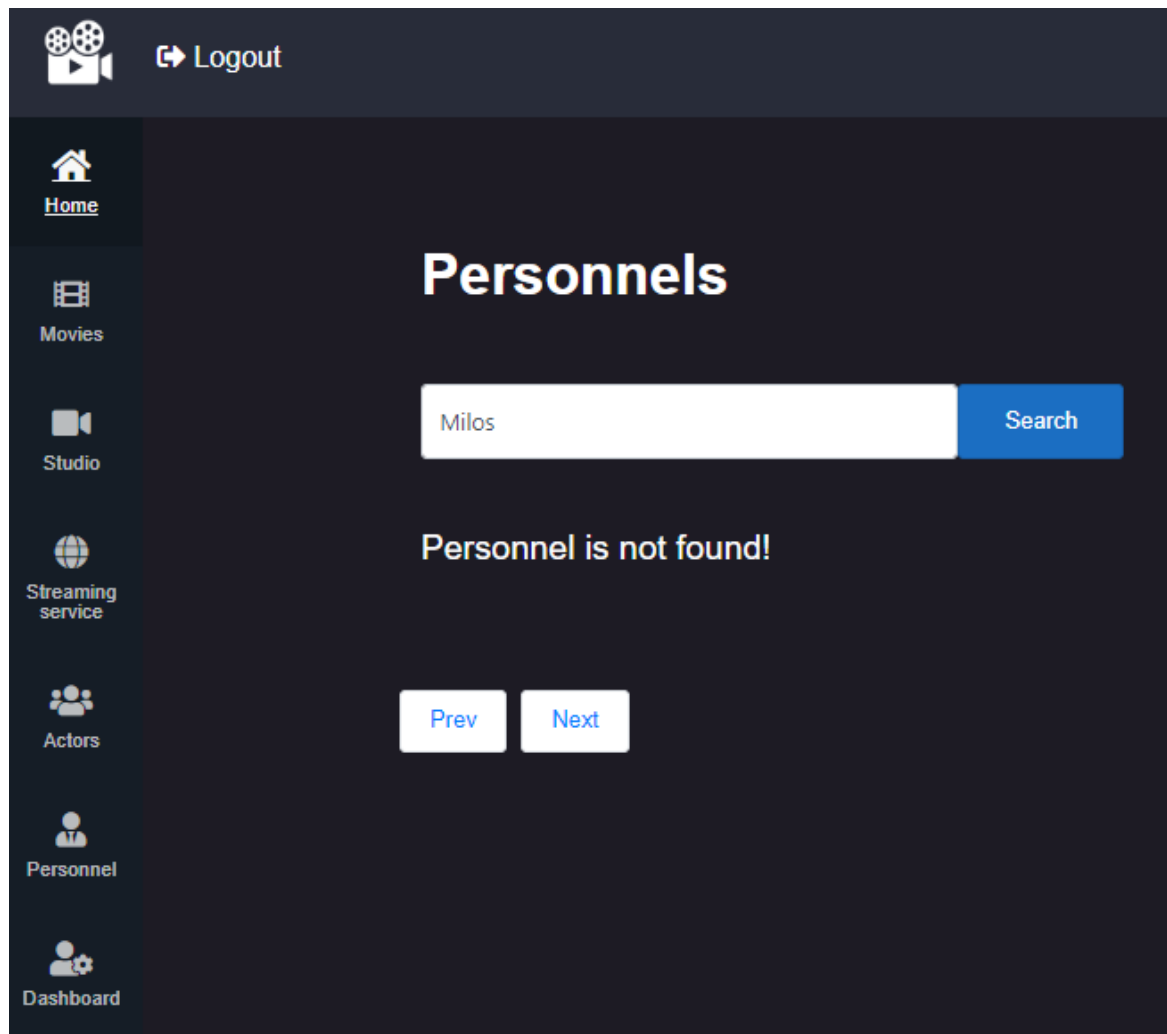
5. Корисник/Администратор **бира** припадника особља. (АПУСО)
6. Корисник/Администратор **позива** систем да учита припадника особља. (АПСО)
7. Систем **учитава** припадника особља. (СО)
8. Систем **приказује** кориснику/администратору податке о припаднику особља и поруку: “Систем је учитао припадника особља”. (ИА)



Слика 83. Страница нађеног особља

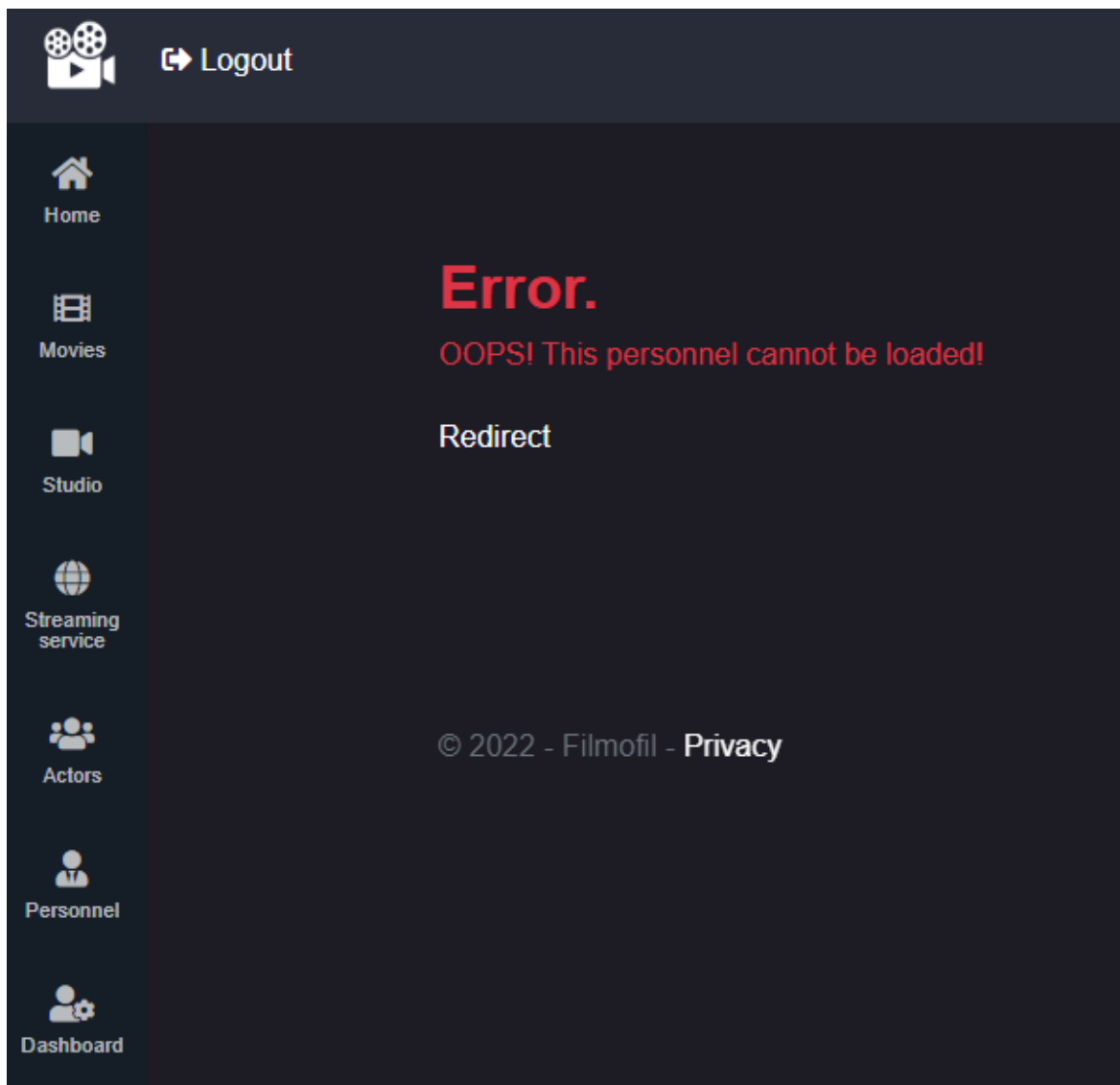
Алтернативна сценарија

4.1 Уколико систем не може да нађе припаднике особље он приказује кориснику/администратору поруку: “Систем не може да нађе припаднике особље по задатој вредности”. Прекида се извршење сценарија. (ИА)



Слика 84. Не постоји особа са задатим именом

8.1 Уколико систем не може да учита припадника особља он приказује кориснику/администратору поруку “Систем не може да учита припадника особља”. (ИА)



Слика 85. Неуспешно учитавање особља

6.2. ПРОЈЕКТОВАЊЕ АПЛИКАЦИОНЕ ЛОГИКЕ

Под пројектовањем апликационе логике се подразумева пројектовање контролера апликационе логике, пројектовање пословне логике и пројектовање слоја приступа подацима.

6.2.1. Пројектовање контролера апликационе логике

Контролер прихвата захтеве који пристижу са погледа, обрађује их и потом позива одговарајуће операције слоја приступа подацима. Након што добије резултат, контролер податке попуњава у одговарајући модел који се прослеђује заједно са погледом.



Слика 86. Приказ животног циклуса MVC захтева

6.2.2. Пројектовање пословне логике

Пословна логика је описана структуром (доменским класама) и понашањем (системским операцијама). За сваки од уговора системских операција дефинисаних у фази анализе пројектује се концептуално решење (Vlajić, 2020).

У наставку је за сваку идентификовану системску операцију у фази анализе формиран одговарајући дијаграм секвенци који приказује комуникацију контролера апликационе логике и слоја приступа подацима.

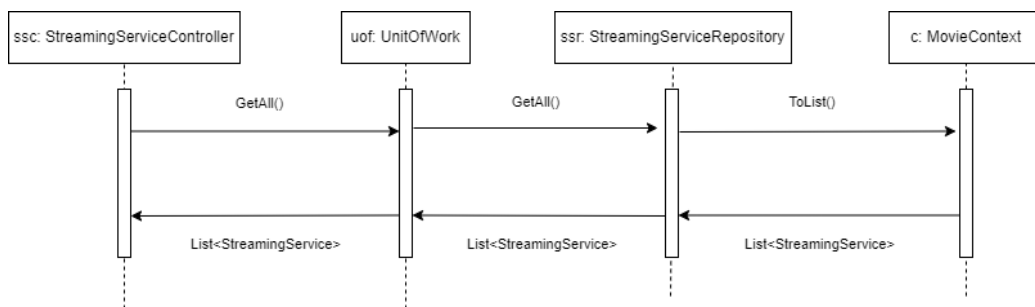
Уговор УГ2: GetAll

Операција: GetAll(List<StreamingService>): сигнал

Веза са СК: СК2, СК6

Предуслови:

Постуслови:



Слика 87. ДС – Уговор: GetAll(List<StreamingService>)

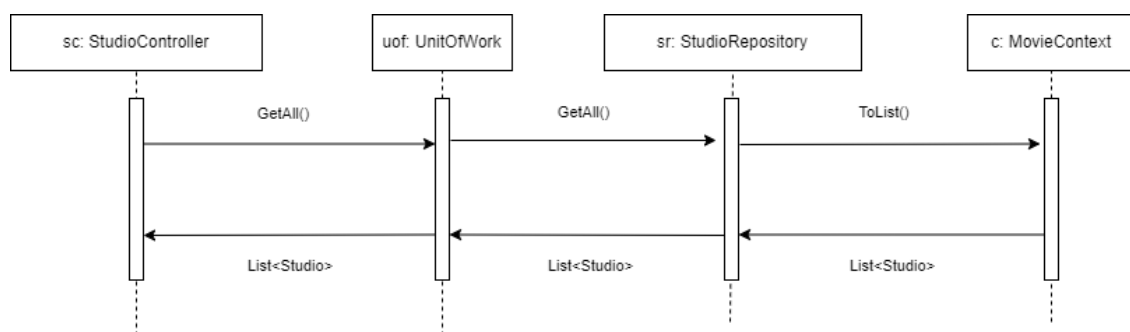
Уговор УГ3: GetAll

Операција: GetAll(List<Studio>): сигнал

Веза са СК: СК2

Предуслови:

Постуслови:



Слика 88. ДС – Уговор: GetAll(List<Studio>)

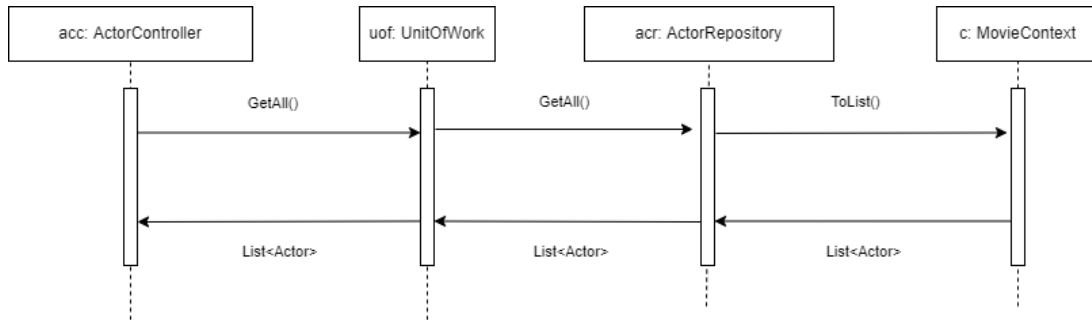
Уговор УГ4: GetAll

Операција: GetAll(List<Actor>): сигнал

Веза са СК: СК2

Предуслови:

Постуслови:



Слика 89. ДС – Уговор: GetAll(List<Actor>)

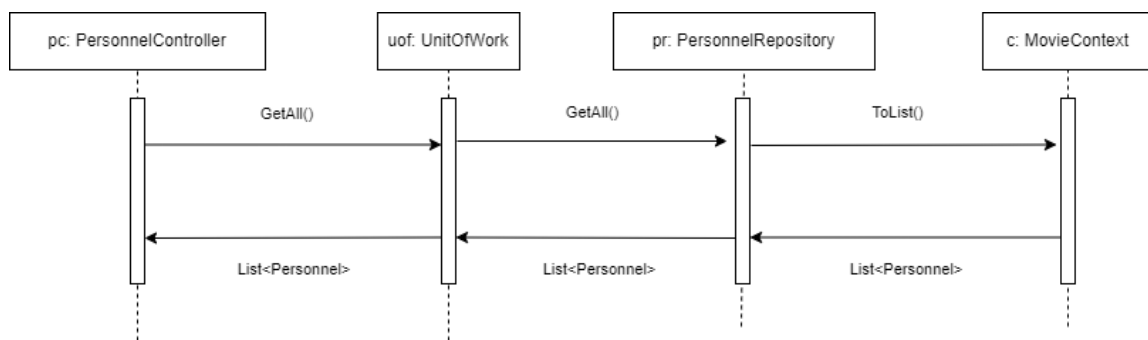
Уговор УГ5: GetAll

Операција: GetAll(List<Personnel>): сигнал

Веза са СК: СК2, СК7, СК10

Предуслови:

Постуслови:



Слика 90. ДС – Уговор: GetAll(List<Personnel>)

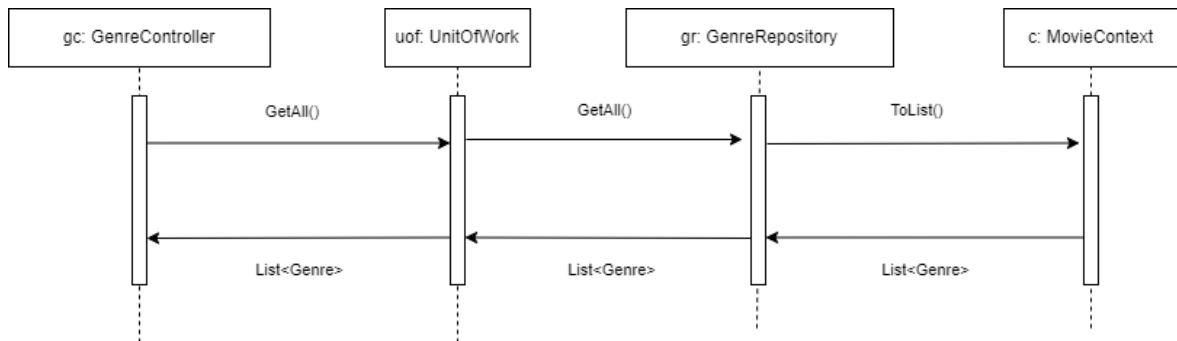
Уговор УГ6: GetAll

Операција: GetAll(List<Genre>): сигнал

Веза са СК: СК2

Предуслови:

Постуслови:



Слика 91. ДС – Уговор: GetAll(List<Genre>)

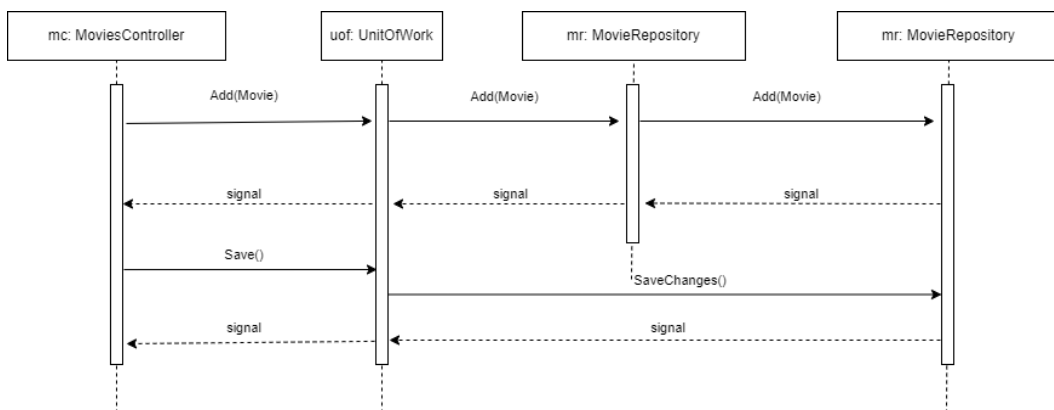
Уговор УГ7: Add

Операција: Add(List<Movie>): сигнал

Веа са СК: СК2

Предуслови: Вредносна и структурна ограничења над објектом Movie морају бити задовољена.

Постуслови: Филм је сачуван.



Слика 92. ДС – Уговор: Add(Movie)

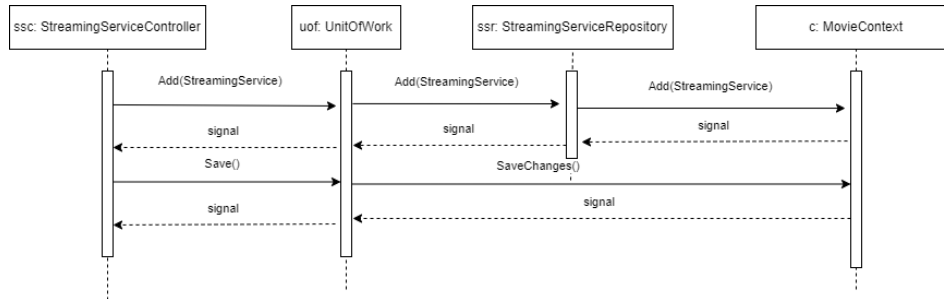
Уговор УГ8: Add

Операција: Add(StreamingService): сигнал

Веа са СК: СК3

Предуслови: Вредносна и структурна ограничења над објектом StreamingService морају бити задовољена.

Постуслови: Стриминг сервис је сачуван.



Слика 93. ДС – Уговор: Add(StreamingService)

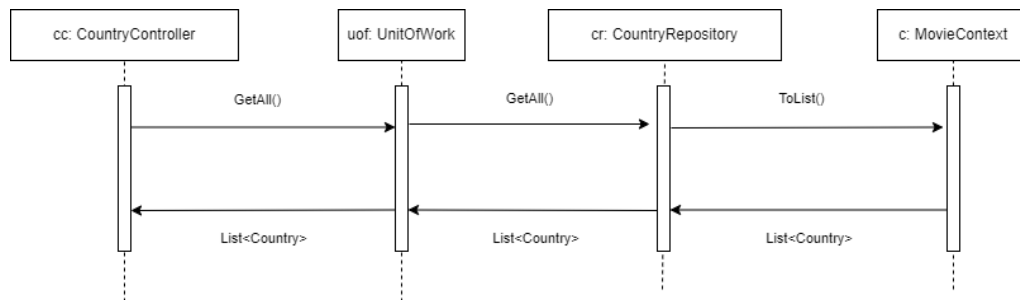
Уговор УГ9: GetAll

Операција: GetAll(List<Country>): сигнал

Веза са СК: СК4

Предуслови:

Постуслови:



Слика 94. ДС – Уговор: GetAll(Country)

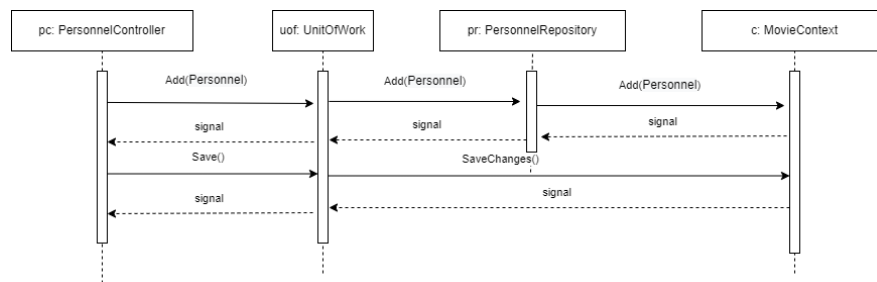
Уговор УГ10: Add

Операција: Add(Personnel): сигнал

Веза са СК: СК4

Предуслови: Вредносна и структурна ограничења над објектом Personnel морају бити задовољена.

Постуслови: Особље је сачувано.



Слика 95. ДС – Уговор: Add(Personnel)

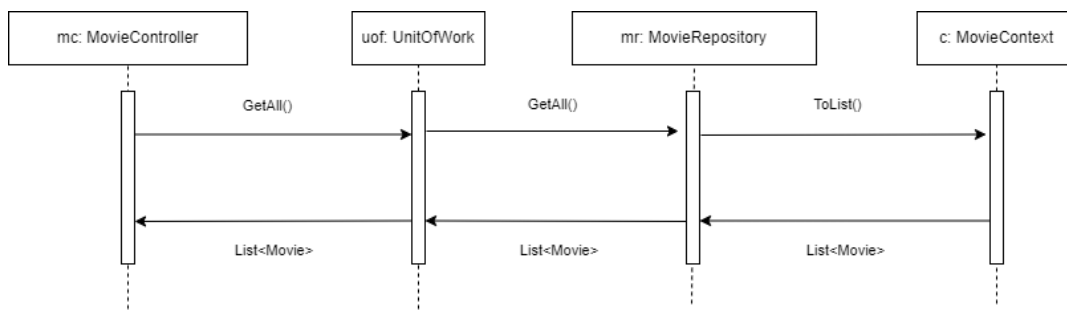
Уговор УГ11: GetAll

Операција: GetAll(List<Movie>): сигнал

Вежа са СК: СК5, СК8, СК9

Предуслови:

Постуслови:



Слика 96. ДС – Уговор: GetAll(List<Movie>)

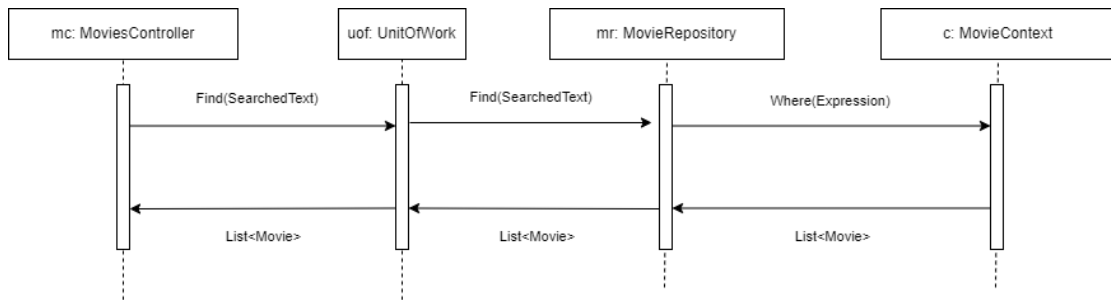
Уговор УГ12: Find

Операција: Find(Criteria, List<Movie>): сигнал

Вежа са СК: СК5, СК8, СК9

Предуслови:

Постуслови:



Слика 97. ДС – Уговор: Find(Criteria, List<Movie>)

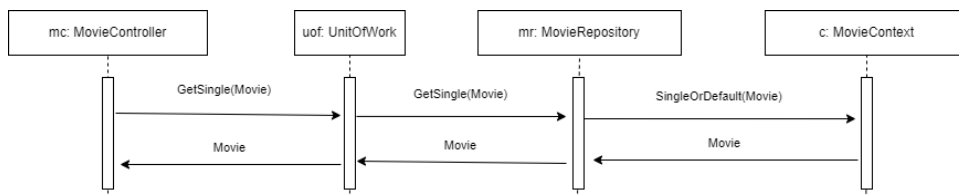
Уговор УГ13: GetSingle

Операција: GetSingle(Movie): сигнал

Веа са СК: СК5, СК8, СК9

Предуслови:

Постуслови:



Слика 98. ДС – Уговор: GetSingle(Movie)

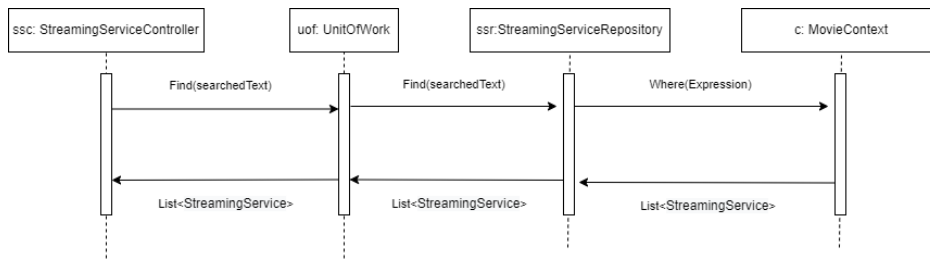
Уговор УГ14: Find

Операција: Find(Criteria, List<StreamingService>): сигнал

Веа са СК: СК6

Предуслови:

Постуслови:



Слика 99. ДС – Уговор: Find(Criteria, List<StreamingService>)

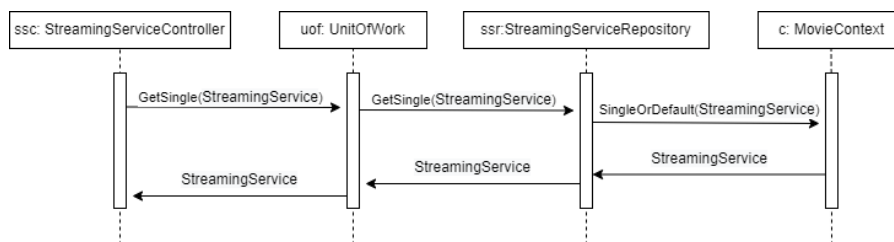
Уговор УГ15: GetSingle

Операција: GetSingle(StreamingService): сигнал

Веза са СК: СК6

Предуслови:

Постуслови:



Слика 100. ДС – Уговор: GetSingle(StreamingService)

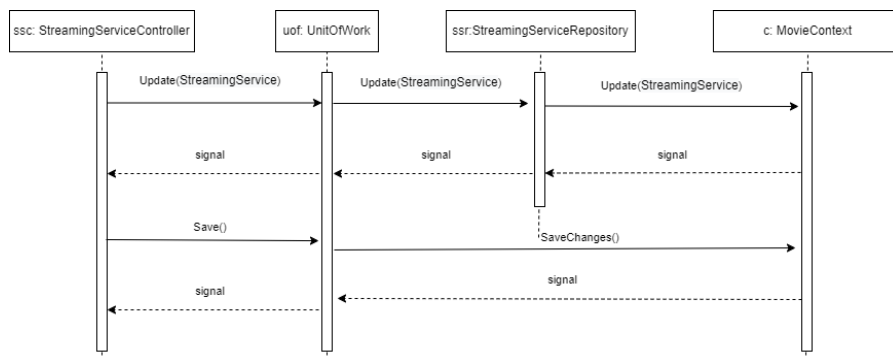
Уговор УГ16: Update

Операција: Update(StreamingService): сигнал

Веза са СК: СК6

Предуслови: Вредносна и структурна ограничења над објектом StreamingService морају бити задовољена.

Постуслови: Стриминг сервис је сачуван



Слика 101. ДС – Уговор: Update(StreamingService)

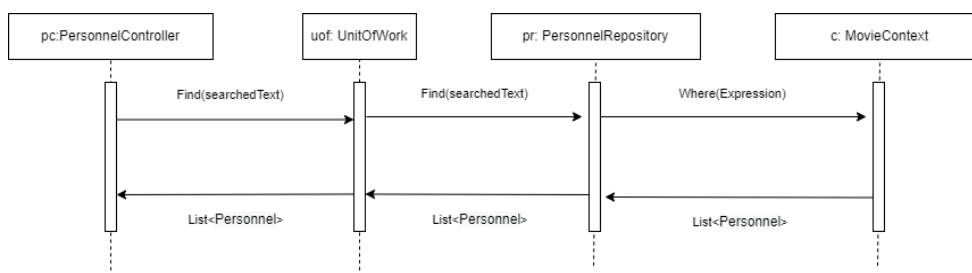
Уговор УГ17: Find

Операција: Find(Criteria, List<Personnel>): сигнал

Вежа са СК: СК7, СК10

Предуслови:

Постуслови:



Слика 102. ДС – Уговор: Find(Criteria, List<Personnel>)

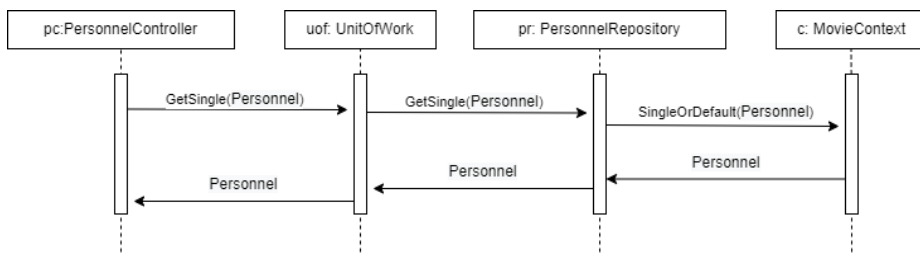
Уговор УГ18: GetSingle

Операција: GetSingle(Personnel): сигнал

Вежа са СК: СК7, СК10

Предуслови:

Постуслови:



Слика 103. ДС – Уговор: GetSingle(Personnel)

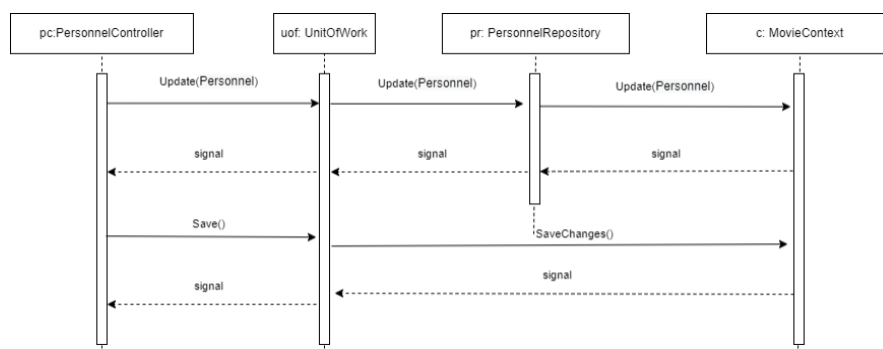
Уговор УГ19: Update

Операција: Update(Personnel) : сигнал

Веа са СК: СК7

Предуслови: Вредносна и структурна ограничења над објектом Actor морају бити задовољена.

Постуслови: Особље је сачувано.



Слика 104. ДС – Уговор: Update(Personnel)

Уговор УГ20: Delete

Операција: Delete(Movie) : сигнал

Веа са СК: СК8

Предуслови: Вредносна и структурна ограничења над објектом SiteUser морају бити задовољена.

Постуслови: Филм је обрисан.

Уговор УГ21: Add

Операција: Add(Review) : сигнал

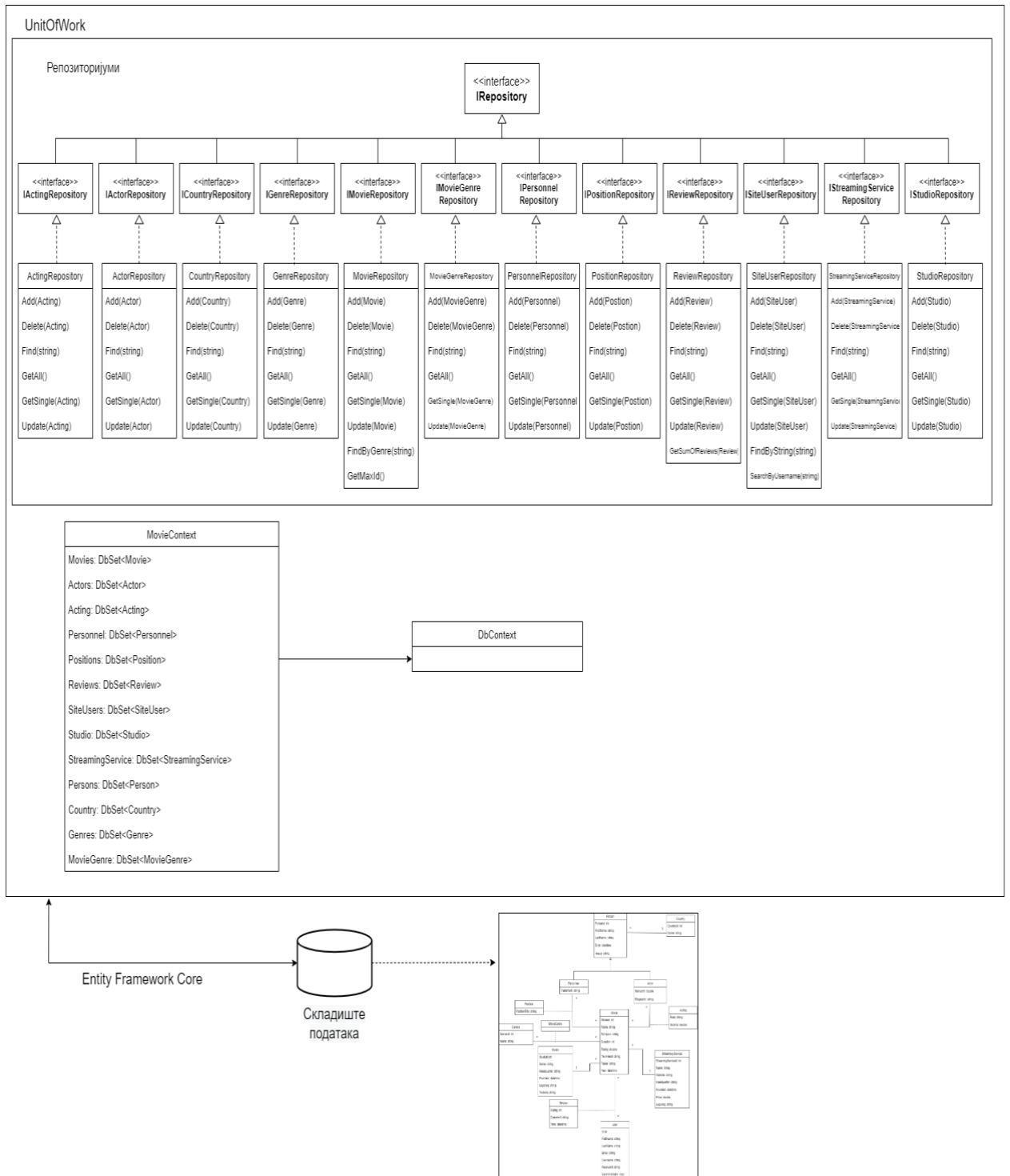
Веза са СК: СК9

Предуслови: Вредносна и структурна ограничења над објектом Review морају бити задовољена.

Постуслови: Рецензија је сачувана.

6.2.3. Пројектовање слоја приступа подацима

Контролери шаљу захтеве до слоја приступа подацима позивајући одговарајуће методе *Repository* класа. Свака од *Repository* класа има приступ контексту који омогућава комуникацију са складиштем података. Реализациојом *UnitOfWork* патерна избегава се стварање конфликта над базом података и формира се јединствена тачка приступа подацима и јединствена сесија над базом података.



Слика 105. Пројектовање слоја приступа података

6.3. ПРОЈЕКТОВАЊЕ СКЛАДИШТА ПОДАТАКА

На основу претходно креираног доменског модела, формиран је следећи релациони модел:

Persons (PersonId, FirstName, LastName, Born, Image, *CountryId*)

Country (CountryId, Name)

Personnel (PersonId, Trademark)

Actor (PersonId, Networth, Biography)

Acting (*ActorId*, *MovieId*, Role, Income)

Positions (*PersonnelId*, *MovieId*, PositionTitle)

Movies (MovieId, Name, Duration, Rating, Thumbnail, Trailer, Synopsis, Year, *StudioId*, *StreamingServiceId*)

Genres (GenreId, Name)

MovieGenre (*MovieId*, *GenreId*)

StreamingService (StreamingServiceId, Name, Website, Headquarter, Founded, Price, LogoImg)

Studio (StudioId, Name, Headquarter, Founded, LogoImg, Website)

Reviews(*MovieId*, *UserId*, Rating, Comment, Time)

AspNetUsers (Id, FirstName, LastName, IsAdministrator, UserName, Email, Password)

Табела 1. Persons

Табела Persons		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип Атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT RESTRICTED Country
	PersonId	Integer	Not null>0			UPDATE CASCADE
	FirstName	String	Not null			Actor, Personnel
	LastName	String	Not null			UPDATE RESTRICTED
	Born	Date	Not null			Country
	Image	String	Not null			DELETE CASCADE
	CountryId	Integer	Not null>0			Actor, Personnel

Табела 2. Country

Табела Country		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип Атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT / UPDATE CASCADE
	CountryId	Integer	Not null>0			Persons
	Name	String	Not null			DELETE RESTRICTED Persons

Табела 3. Personnel

Табела Personnel		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип Атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT RESTRICTED Persons
	PersonId	Integer	Not null>0			UPDATE RESTRICTED Persons
	Trademark	String	Not null			DELETE CASCADE Persons

Табела 4. Actor

Табела Actor		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип Атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT RESTRICTED Persons
	PersonId	Integer	Not null>0			UPDATE RESTRICTED Persons
	Networth	Float	Not null			DELETE CASCADE Persons
	Biography	String	Not null			DELETE CASCADE Persons

Табела 5. Acting

Табела Acting		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип Атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT RESTRICTED Actor, Movie
	ActorId	Integer	Not null>0			UPDATE RESTRICTED Actor, Movie
	MovieId	Integer	Not null>0			DELETE /
	Role	String	Not null			
	Income	Float	Not null			

Табела 6. Positions

Табела Positions		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип Атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT RESTRICTED Personnel, Movie
	PersonnelId	Integer	Not null>0			UPDATE RESTRICTED Personnel, Movie
	MovieId	Integer	Not null>0			DELETE /
	PositionTitle	String	Not null			

Табела 7. Movies

Табела Movies		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип Атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT RESTRICTED Studio, StreamingService
	MovieId	Integer	Not null>0			UPDATE RESTRICTED
	Name	String	Not null			Studio, StreamingService
	Duration	Integer	Not null			
	Rating	Float	Not null			UPDATE CASCADES MovieGenre
	Thumbnail	String	Not null			DELETE
	Trailer	String	Not null			CASCADES MovieGenre
	Synopsis	String	Not null			
	Year	Date	Not null			
	StudioId	Integer	Not null>0			
	StreamingServiceId	Integer	Not null>0			

Табела 8. Genres

Табела Genres		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип Атрибу та	Вреднос т атрибута	Међузависнос т атрибута једне табеле	Међузависнос т атрибута више табела	INSERT / UPDATE CASCADES MovieGenre
	GenreId	Integer	Not null>0			DELETE CASCADES MovieGenre
	Name	String	Not null			

Табела 9. MovieGenre

Табела MovieGenre		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип Атрибу та	Вреднос т атрибута	Међузависнос т атрибута једне табеле	Међузависнос т атрибута више табела	INSERT RESTRICTED Movies, Genres
	MovieId	Integer	Not null>0			UPDATE RESTRICTED Movies, Genres
	GenreId	Integer	Not null			DELETE /

Табела 10. StreamingService

Табела StreamingService		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип Атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT / UPDATE CASCADES Movies DELETE RESTRICTED Movies
	StreamingServiceId	Integer	Not null>0			
	Name	String	Not null			
	Website	String	Not null			
	Headquarter	String	Not null			
	Founded	Date	Not null			
	Price	Float	Not null			
	LogoImg	String	Not null			

Табела 11. Studio

Табела Studio		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип Атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT / UPDATE CASCADES Movies DELETE RESTRICTED Movies
	StudioId	Integer	Not null>0			
	Name	String	Not null			
	Headquarter	String	Not null			
	Founded	Date	Not null			
	LogoImg	String	Not null			
	Website	String	Not null			

Табела 12. Reviews

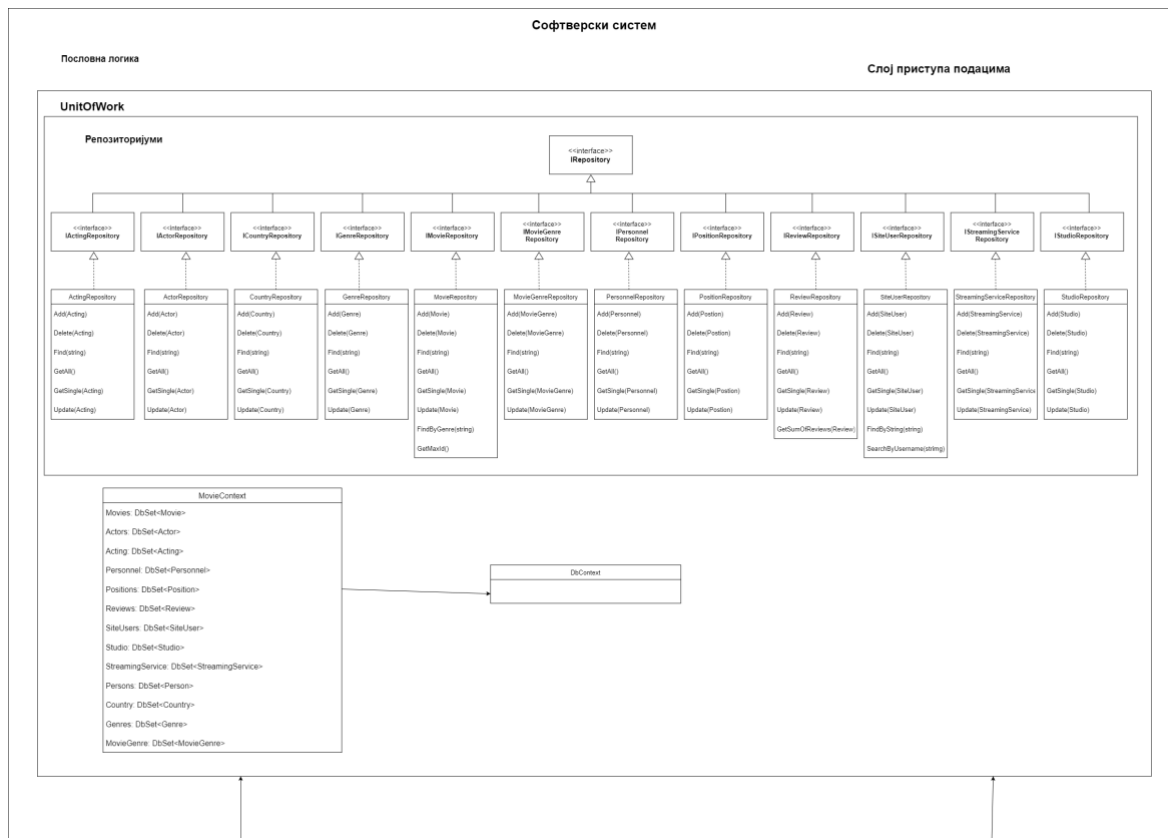
Табела Reviews		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип Атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT RESTRICTED AspNetUsers, Movies
	MovieId	Integer	Not null>0			UPDATE
	UserId	Integer	Not null			RESTRICTED AspNetUsers, Movies
	Rating	Integer	Not null			DELETE /
	Comment	String	Not null			
	Time	Date	Not null			

Табела 13. AspNetUsers

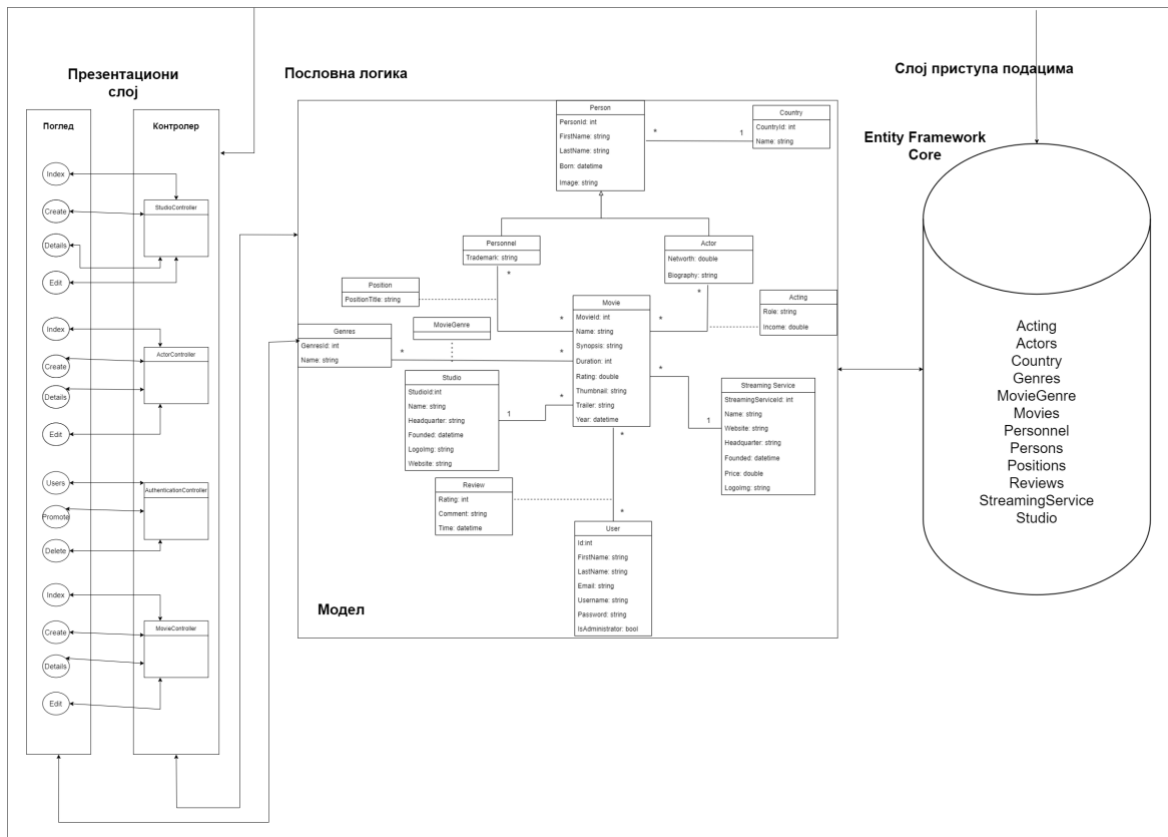
Табела AspNetUsers		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип Атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT /
	Id	Integer	Not null>0			UPDATE CASCADES Reviews
	FirstName	String	Not null			DELETE CASCADES Reviews
	IsAdministrator	Boolean	Not null			
	UserName	String	Not null			
	Email	String	Not null			
	Password	String	Not null			

6.4. КОНАЧАН ИЗГЛЕД АРХИТЕКТУРЕ СОФТВЕРСКОГ СИСТЕМА

На основу претходно пројектованих целина, добијена је коначна архитектура софтверског система као резултат фазе пројектовања.



Слика 106. Коначна архитектура софтверског система – први део



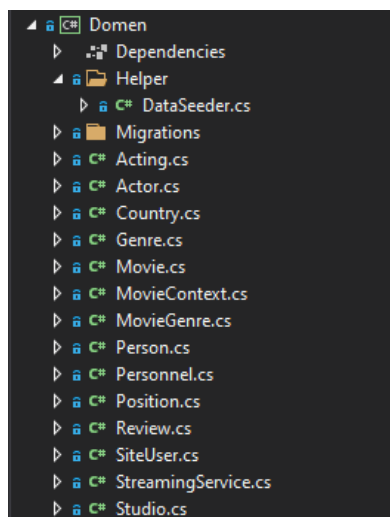
Слика 107. Коначна архитектура софтверског система – други део

7. ИМПЛЕМЕНТАЦИЈА

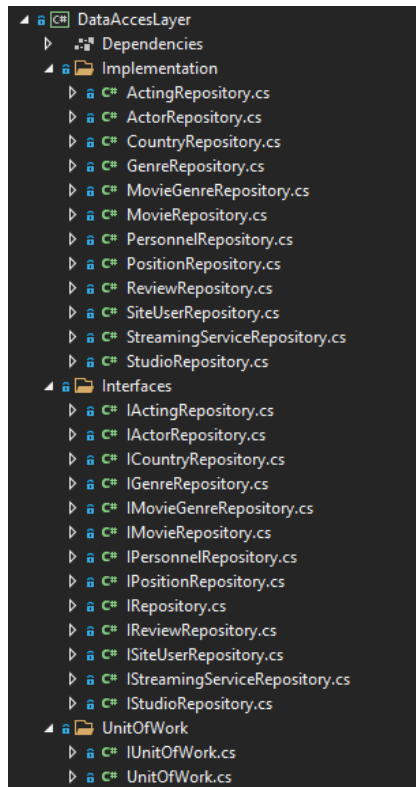
Веб апликација развијена је у програмском језику C#, употребом MVC оквира и ASP .NET Core. Апликација се извршава на IIS серверу, а за развојно окружење је коришћен Microsoft Visual Studio 2019. Кориснички интерфејс је представљен Razor страницама, а за управљање базом података коришћен је Microsoft SQL Server.

7.1. СТРУКТУРА СОФТВЕРСКОГ РЕШЕЊА

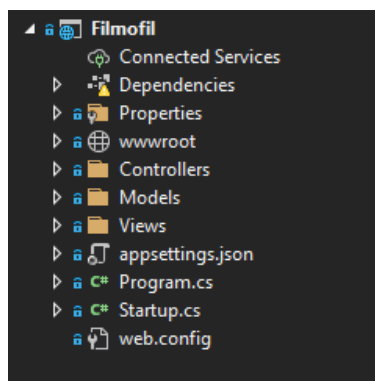
Имплементриране су следеће класе:



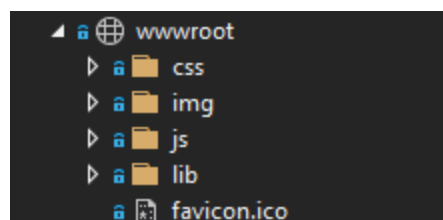
Слика 108. Библиотека доменских класа



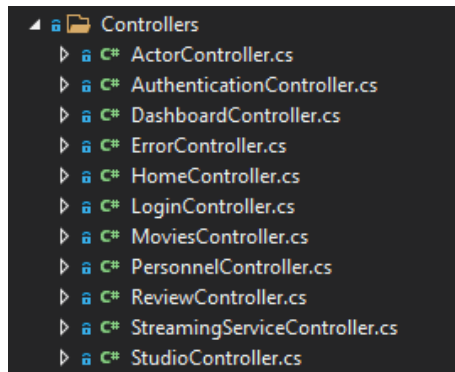
Слика 109. Библиотека класа DataAccesLayer



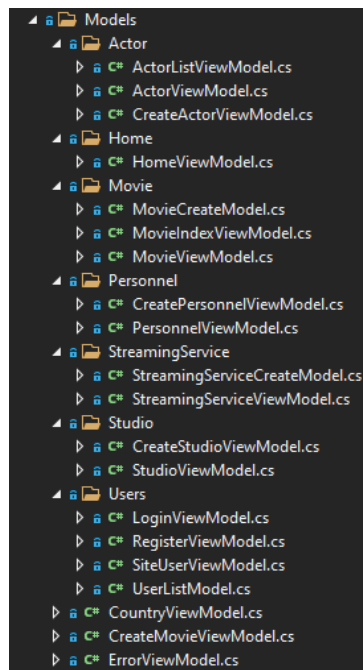
Слика 110. Пројекат Filmofil



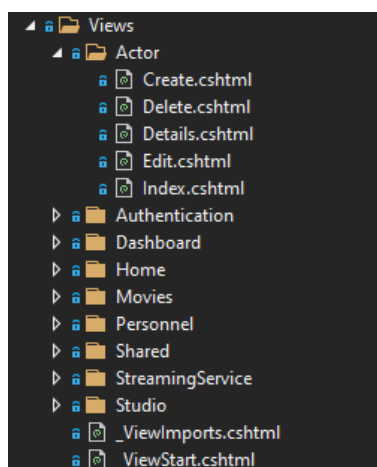
Слика 111. Фолдер wwwroot



Слика 112. Фолдер Controllers



Слика 113. Фолдер Models



Слика 114. Фолдер Views

7.2. ИМПЛЕМЕНТАЦИЈА ПОСЛОВНЕ ЛОГИКЕ

Пословна логика је описана структуром коју чине доменске класе и понашањем које се састоји из системских операција (Vlajić, 2020).

На основу дефинисаних класа и релација у доменском моделу, неопходно је креирати класе и у самом програму. Свака класа се састоји од приватних атрибута и за сваки од њих се креира *Property*, односно јавне методе које омогућавају приступ и измену тог атрибута. Доменске класе се налазе у оквиру билбиотеке класа *Domen*. У наставку је дат пример доменске класе особа (*Person*).

```
public class Person
{
    public int PersonId { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public DateTime Born { get; set; }
    public string Image { get; set; }
    public int CountryId { get; set; }
    public Country Country { get; set; }
    public override string ToString()
    {
        return FirstName + " " + LastName;
    }
}
```

Системске операције имплементирани су у оквиру одговарајућих *Repository* класа унутар библиотеке класа *DataAccessLayer*.

Уговор УГ2: GetAll

Операција: GetAll(List<StreamingService>): сигнал

Веза са СК: СК2, СК6

Предуслови:

Постуслови:

```
public List<StreamingService> GetAll()
{
    return context.StreamingService.ToList();
}
```

```
}
```

Уговор УГ3: GetAll

Операција: GetAll(List<Studio>): сигнал

Веза са СК: СК2

Предуслови:

Постуслови:

```
public List<Studio> GetAll()  
{  
    return context.Studio.ToList();  
}
```

Уговор УГ4: GetAll

Операција: GetAll(List<Actor>): сигнал

Веза са СК: СК2

Предуслови:

Постуслови:

```
public List<Actor> GetAll()  
{  
    return context.actors.Include(a=>a.Country).ToList();  
}
```

Уговор УГ5: GetAll

Операција: GetAll(List<Personnel>): сигнал

Веза са СК: СК2, СК7, СК10

Предуслови:

Постуслови:

```
public List<Personnel> GetAll()  
    {  
        return context.Personnel.Include(p=>p.Country).ToList();  
    }
```

Уговор УГ6: GetAll

Операција: GetAll(List<Genre>): сигнал

Веза са СК: СК2

Предуслови:

Постуслови:

```
public List<Genre> GetAll()  
    {  
        return context.Genres.ToList().OfType<Genre>().ToList();  
    }
```

Уговор УГ7: Add

Операција: Add(List<Movie>): сигнал

Веза са СК: СК2

Предуслови: Вредносна и структурна ограничења над објектом Movie морају бити задовољена.

Постуслови: Филм је сачуван.

```
public void Add(Movie entity)
    {
        context.Add(entity);
    }
```

Уговор УГ8: Add

Операција: Add(StreamingService): сигнал

Веза са СК: СК3

Предуслови: Вредносна и структурна ограничења над објектом StreamingService морају бити задовољена.

Постуслови: Стриминг сервис је сачуван.

```
public void Add(StreamingService entity)
    {
        context.Add(entity);
    }
```

Уговор УГ9: GetAll

Операција: GetAll(List<Country>): сигнал

Веза са СК: СК4

Предуслови:

Постуслови:

```
public List<Country> GetAll()
{
    return context.Country.ToList().OfType<Country>().ToList();
}
```

Уговор УГ10: Add

Операција: Add(Personnel): сигнал

Веза са СК: СК4

Предуслови: Вредносна и структурна ограничења над објектом Personnel морају бити задовољена.

Постуслови: Особље је сачувано.

```
public void Add(Personnel entity)
{
    context.Add(entity);
}
```

Уговор УГ11: GetAll

Операција: GetAll(List<Movie>): сигнал

Веза са СК: СК5, СК8, СК9

Предуслови:

Постуслови:

```
public List<Movie> GetAll()
{
    return context.Movies.
        Include(m => m.Studio).
        Include(m => m.StreamingService).
        Include(m => m.Genres).
        ToList();
}
```

Уговор УГ12: Find

Операција: Find(Criteria, List<Movie>): сигнал

Веза са СК: СК5, СК8, СК9

Предуслови:

Постуслови:

```
public List<Movie> Find(string text)
{
    return context.Movies.Include(m => m.Studio).
        Include(m => m.StreamingService).
        Include(m => m.Genres).
        Where(m => m.Name.ToLower().Contains(text.ToLower()))
        .ToList();
}
```

Уговор УГ13: GetSingle

Операција: GetSingle(Movie): сигнал

Веза са СК: СК5, СК8, СК9

Предуслови:

Постуслови:

```

public Movie GetSingle(Movie entity)
{
    var movie = context.Movies.
        Include(m => m.Studio).
        Include(m => m.StreamingService).
        Include(m => m.Actings).
        Include(m => m.Positions).
        Include(m => m.Reviews).
        Include(m => m.Genres).
        SingleOrDefault(m => m.MovieId == entity.MovieId);
    return movie;
}

```

Уговор УГ14: Find

Операција: Find(Criteria, List<StreamingService>): сигнал

Веза са СК: СК6

Предуслови:

Постуслови:

```

public List<StreamingService> Find(string text)
{
    return context.StreamingService.Where(s =>
s.Name.ToLower().Contains(text.ToLower())).ToList();
}

```

Уговор УГ15: GetSingle

Операција: GetSingle(StreamingService): сигнал

Веза са СК: СК6

Предуслови:

Постуслови:

```
public StreamingService GetSingle(StreamingService entity)
{
    return context.StreamingService.Find(entity.StreamingServiceId);
}
```

Уговор УГ16: Update

Операција: Update(StreamingService): сигнал

Веза са СК: СК6

Предуслови: Вредносна и структурна ограничења над објектом StreamingService морају бити задовољена.

Постуслови: Стриминг сервис је сачуван

```
public void Update(StreamingService entity)
{
    context.Update(entity);
}
```

Уговор УГ17: Find

Операција: Find(Criteria, List<Personnel>): сигнал

Веза са СК: СК7, СК10

Предуслови:

Постуслови:

```
public List<Personnel> Find(Personnel entity)
{
    return context.Personnel.ToList().FindAll(p => p.FirstName.ToLower().Contains(entity.FirstName)
|| p.LastName.ToLower().Contains(entity.LastName));
}
```


Уговор УГ18: GetSingle

Операција: GetSingle(Personnel): сигнал

Веза са СК: СК7, СК10

Предуслови:

Постуслови:

```
public Personnel GetSingle(Personnel entity)
{
    var list = context.Personnel.
        Include(p => p.Country).
        ToList();
    return list.Find(p => p.PersonId == entity.PersonId);
}
```

Уговор УГ19: Update

Операција: Update(Personnel) : сигнал

Веза са СК: СК7

Предуслови: Вредносна и структурна ограничења над објектом Actor морају бити задовољена.

Постуслови: Особље је сачувано.

```
public void Update(Personnel entity)
{
    context.Update(entity);
}
```

Уговор УГ20: Delete

Операција: Delete(Movie) : сигнал

Веза са СК: СК8

Предуслови: Вредносна и структурна ограничења над објектом SiteUser морају бити задовољена.

Постуслови: Филм је обрисан.

```
public void Delete(Movie entity)
{
    context.Remove(entity);
}
```

Уговор УГ21: Add

Операција: Add(Review) : сигнал

Веза са СК: СК9

Предуслови: Вредносна и структурна ограничења над објектом Review морају бити задовољена.

Постуслови: Рецензија је сачувана.

```
public void Add(Review entity)
{
    context.Add(entity);
}
```

7.3. ИМПЛЕМЕНТАЦИЈА СЛОЈА ПРИСТУПА ПОДАЦИМА

	Column Name	Data Type	Allow Nulls
▶	MovieId	int	<input type="checkbox"/>
▼	ActorId	int	<input type="checkbox"/>
	Role	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Income	float	<input type="checkbox"/>
			<input type="checkbox"/>

Слика 115. Acting

	Column Name	Data Type	Allow Nulls
▶	PersonId	int	<input type="checkbox"/>
	Networth	float	<input type="checkbox"/>
	Biography	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Слика 116. Actor

	Column Name	Data Type	Allow Nulls
▶	CountryId	int	<input type="checkbox"/>
	Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Слика 117. Country

	Column Name	Data Type	Allow Nulls
▶	GenreId	int	<input type="checkbox"/>
	Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Слика 118. Genre

	Column Name	Data Type	Allow Nulls
▶	MovieId	int	<input type="checkbox"/>
▼	GenreId	int	<input type="checkbox"/>
			<input type="checkbox"/>

Слика 119. MovieGenre

	Column Name	Data Type	Allow Nulls
▶	MovieId	int	<input type="checkbox"/>
	Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Duration	int	<input type="checkbox"/>
	Rating	float	<input type="checkbox"/>
	Thumbnail	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Trailer	nvarchar(MAX)	<input checked="" type="checkbox"/>
	StudioId	int	<input type="checkbox"/>
	StreamingServiceId	int	<input type="checkbox"/>
	Synopsis	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Year	datetime2(7)	<input type="checkbox"/>
			<input type="checkbox"/>

Слика 120. *Movie*

	Column Name	Data Type	Allow Nulls
▶	PersonId	int	<input type="checkbox"/>
	FirstName	nvarchar(MAX)	<input checked="" type="checkbox"/>
	LastName	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Born	datetime2(7)	<input type="checkbox"/>
	CountryId	int	<input type="checkbox"/>
	Image	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Слика 121. *Person*

	Column Name	Data Type	Allow Nulls
▶	Trademark	nvarchar(MAX)	<input checked="" type="checkbox"/>
⚠	PersonId	int	<input type="checkbox"/>
			<input type="checkbox"/>

Слика 122. *Personnel*

	Column Name	Data Type	Allow Nulls
▶	MovieId	int	<input type="checkbox"/>
⚠	PersonnelId	int	<input type="checkbox"/>
	PositionTitle	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Слика 123. *Position*

	Column Name	Data Type	Allow Nulls
▶	MovieId	int	<input type="checkbox"/>
⚠	UserId	int	<input type="checkbox"/>
	Rating	int	<input type="checkbox"/>
	Comment	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Time	datetime2(7)	<input type="checkbox"/>
			<input type="checkbox"/>

Слика 124. *Review*

	Column Name	Data Type	Allow Nulls
▶	StreamingServiceId	int	<input type="checkbox"/>
	Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Website	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Headquarter	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Founded	datetime2(7)	<input type="checkbox"/>
	Price	float	<input type="checkbox"/>
	LogoImg	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Слика 125. *StreamingService*

	Column Name	Data Type	Allow Nulls
▶	StudioId	int	<input type="checkbox"/>
	Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Headquarter	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Founded	datetime2(7)	<input type="checkbox"/>
	LogoImg	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Website	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Слика 126. *Studio*

7.4. ИМПЛЕМЕНТАЦИЈА ПРЕЗЕНТАЦИОНОГ СЛОЈА

СК1: Случај коришћења – Пријава на систем

Назив СК: Пријава на систем

Актори СК: Корисник/Администратор

Учесници СК: Корисник/Администратор и систем (програм)

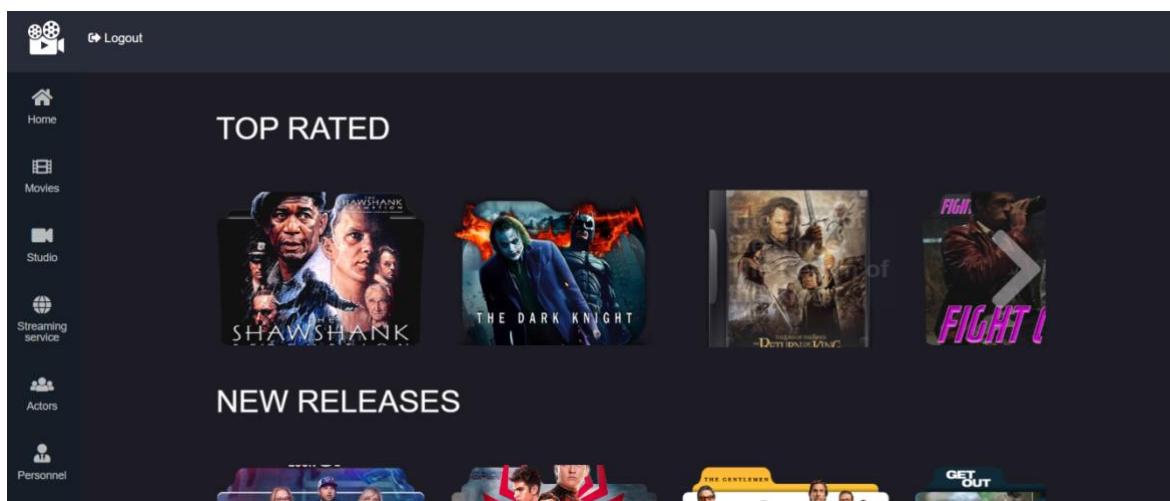
Предуслов: Систем је укључен у приказана је форма за пријаву на систем.

The image shows a dark-themed user interface for a system. At the top, there is a navigation bar with a film camera icon, a 'Login' button with a right-pointing arrow, and a 'Sign up' button with a person icon and a plus sign. Below this is a vertical navigation menu on the left side with icons and labels: Home (house icon), Movies (film strip icon), Studio (video camera icon), Streaming service (globe icon), Actors (group of people icon), and Personnel (single person icon). The main content area on the right contains a 'Username' label above a white input field, a 'Password' label above another white input field, and a blue 'Login' button. Below the login button is a 'Registration' section. At the bottom of the page, there is a footer with the text '© 2022 - Filmofil - Privacy'.

Слика 127. Форма за пријаву на систем

Основни сценарио СК

6. Корисник/Администратор **уноси** податке за пријаву на систем. (АПУСО)
7. Корисник/Администратор **контролише** да ли је коректно унео податке.(АНСО)
8. Корисник/Администратор **позива** систем да нађе пријављеног корисника/администратора. (АПСО)
Опис акције: Кликом на дугме „Login” корисник/администратор позива системску операцију *Login(Username, Password)*
9. Систем **тражи** пријављеног корисника/администратора. (СО)
10. Систем **приказује** кориснику/администратору поруку: “Успешна пријава”.(ИА)



Слика 128. Успешна пријава на систем (преусмеравање на почетну страницу)

Алтернативна сценарија

- 10.1. Уколико систем није успео да нађе пријављеног корисника/администратора, он **приказује** поруку: „Нисте унели одговарајуће податке“. (ИА)

Username
PetarPe

Password

Login

Wrong username or password!

Registration

Слика 129, Неуспешна пријава на систем

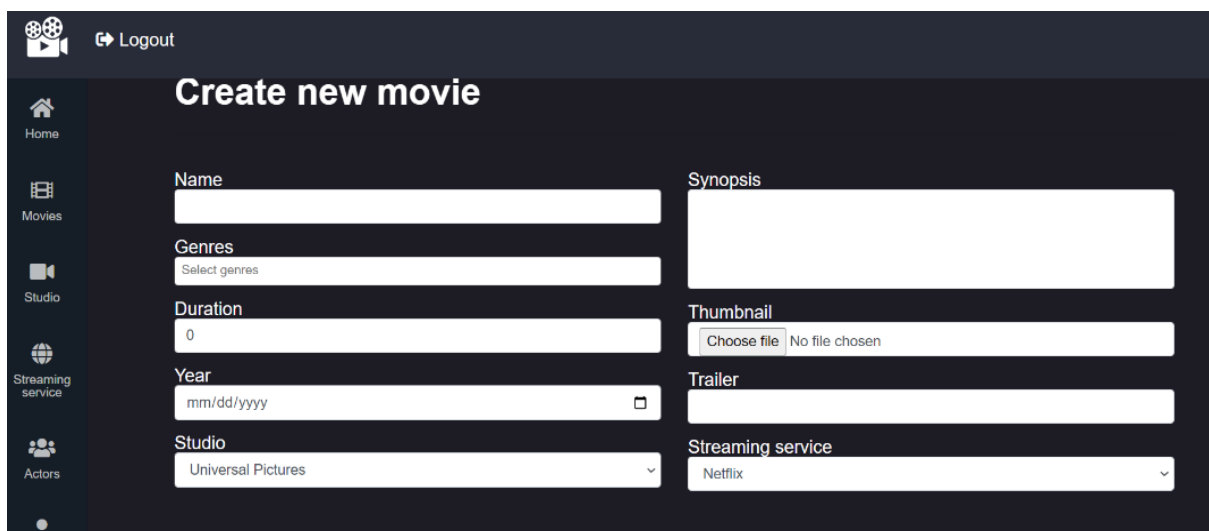
СК2: Случај коришћења – Креирање филма

Назив СК: Креирање филма

Актори СК: Администратор

Учесници СК: Администратор и систем (програм)

Предуслов: Систем је укључен и администратор је улогован под својом шифром. Систем приказује форму за рад са филмом. Учитана је листа студија, стриминг сервиса, глумаца, жанрова и особља.



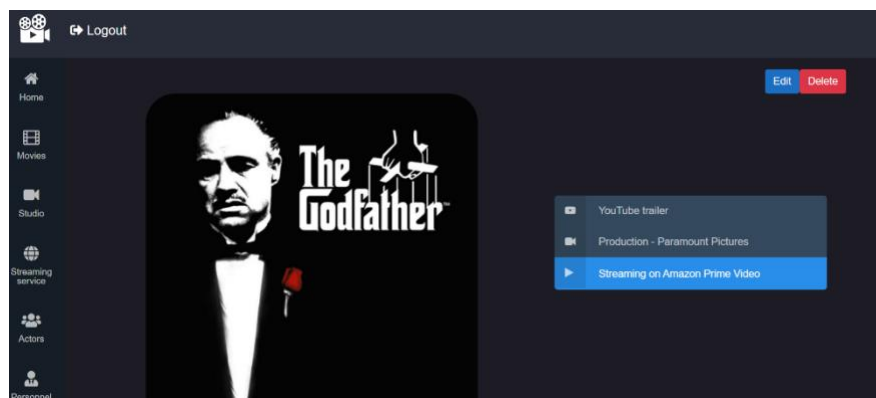
Слика 130. Форма за креирање новог филма

Основни сценарио СК

- Администратор **уноси** податке о филму. (АПУСО)
- Администратор **контролише** да ли је коректно унео податке о филму. (АНСО)
- Администратор **позива** систем да запамти филм. (АПСО)

Опис акције: Администратор притиском на дугме “*Create*” позива системску операцију *Add(Movie)*

- Систем **памти** филм. (СО)
- Систем **приказује** администратору запамћени филм и поруку: “Систем је запамтио филм”.(ИА)



Слика 131. Страница успешно креираног филма

Алтернативна сценарија

- 10.1. Уколико систем не може да креира филм он приказује администратору поруку “Систем не може да запамти филм”. (ИА)

A screenshot of a web application interface for creating a new movie. The form is titled "Create new movie" and contains several input fields with validation errors. The "Name" field is empty and has a red error message: "The Name field is required." The "Genres" field is empty and has a red error message: "The Movie must have at least one genre." The "Year" field is empty and has a red error message: "The Year field is required." The "Studio" field is set to "Universal Pictures". Other fields include "Synopsis", "Thumbnail" (with a "Choose file" button and "No file chosen" text), "Trailer", and "Streaming service" (set to "Netflix").

Слика 132. Неуспешно креирање филма

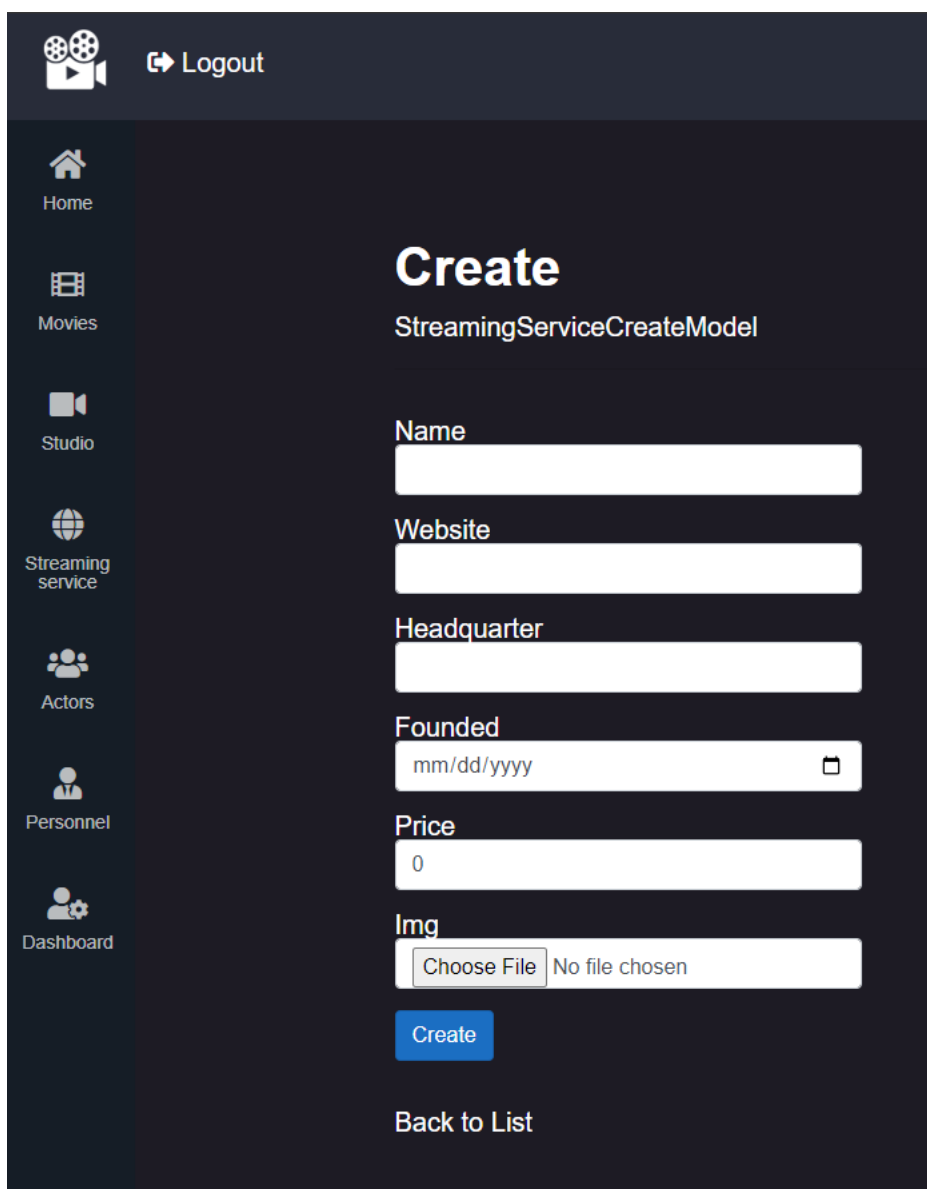
СКЗ: Случај коришћења – Креирање стриминг сервиса

Назив СК: Креирање стриминг сервиса

Актери СК: Администратор

Учесници СК: Администратор и систем (програм)

Предуслов: Систем је укључен и администратор је улогован под својом шифром.
Систем приказује форму за рад са стриминг сервисом.

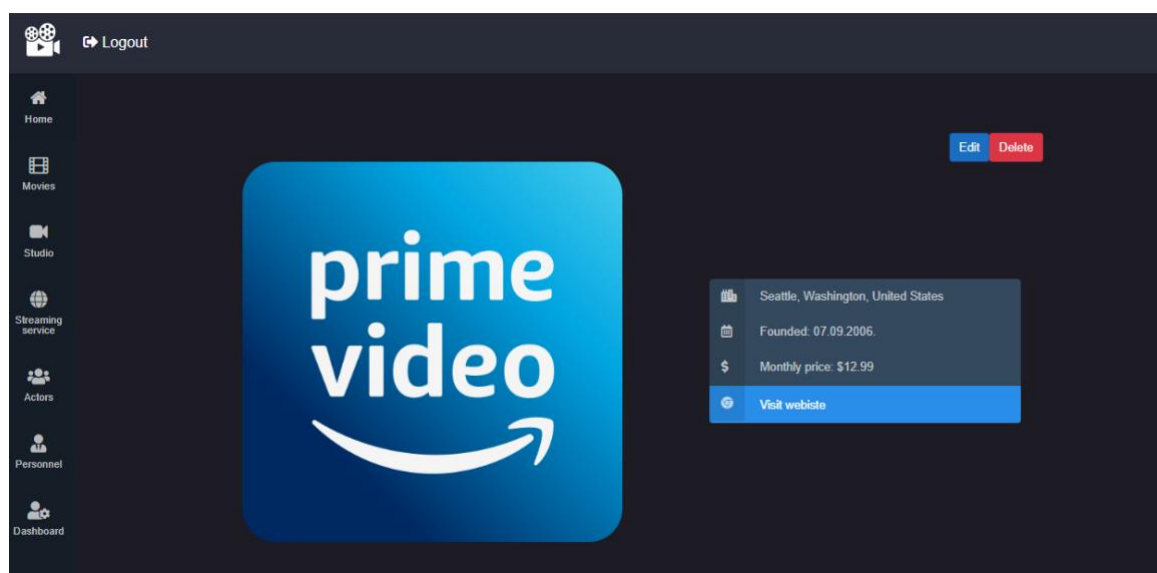


The screenshot shows a web application interface for creating a streaming service. The interface is dark-themed with a sidebar on the left containing navigation icons for Home, Movies, Studio, Streaming service, Actors, Personnel, and Dashboard. The main content area is titled "Create StreamingServiceCreateModel" and contains several input fields: Name, Website, Headquarter, Founded (with a date picker), Price (with the value 0), and Img (with a "Choose File" button and "No file chosen" text). A blue "Create" button is at the bottom, and a "Back to List" link is below it.

Слика 133. Форма за креирање стриминг сервиса

Основни сценарио СК

6. Администратор **уноси** податке о стриминг сервису. (АПУСО)
7. Администратор **контролише** да ли је коректно унео податке о стриминг сервису. (АНСО)
8. Администратор **позива** систем да запамти стриминг сервис. (АПСО)
Опис акције: Администратор притиском на дугме “*Create*” позива системску операцију *Add(StreamingService)*
9. Систем **памти** стриминг сервис. (СО)
10. Систем **приказује** администратору запамћени студио и поруку: “Систем је запамтио стриминг сервис”.(ИА)



Слика 134. Успешно креирање стриминг сервис (преусмеравање на страницу новокреираног стриминг сервиса)

Алтернативна сценарија

- 10.1. Уколико систем не може да креира стриминг сервис он приказује администратору поруку “Систем не може да запамти стриминг сервис”. (ИА)


Create

StreamingServiceCreateModel

Name

Website

Headquarter

Founded
 

The value " " is invalid.

Price

Img
 No file chosen

[Back to List](#)

Слика 135. Неуспешно креирање стриминг сервиса

СК4: Случај коришћења – Креирање особља

Назив СК: Креирање особља

Актери СК: Администратор

Учесници СК: Администратор и систем (програм)

Предуслов: Систем је укључен и администратор је улогован под својом шифром.
Систем приказује форму за рад са особљем.

Logout

Home

Movies

Studio

Streaming service

Actors

Personnel

Dashboard

Create new Personnel

Personnel

First name

Last name

Born

mm/dd/yyyy

Country

United States

Trademark

Image

Choose File No file chosen

Create

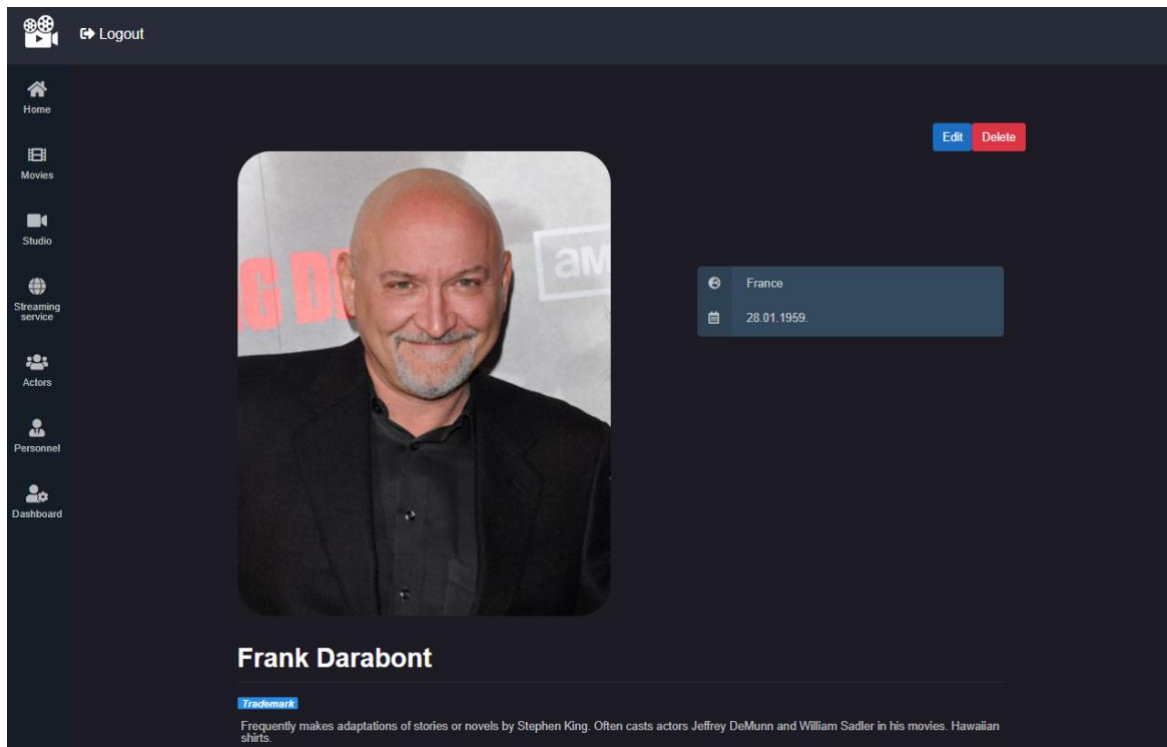
Back to List

Слика 136. Форма за креирање особља

Основни сценарио СК

6. Администратор **уноси** податке о особљу. (АПУСО)
7. Администратор **контролише** да ли је коректно унео податке о особљу. (АНСО)
8. Администратор **позива** систем да запамти особље. (АПСО)
Опис акције: Администратор притиском на дугме “*Create*” позива системску операцију *Add(Personnel)*
9. Систем **памти** особље. (СО)

10. Систем **приказује** администратору запамћено особље и поруку: “Систем је запамтио особље”. (ИА)



Слика 137. Успешно креирано особље (преусмеравање на страницу особља)

Алтернативна сценарија

10.1. Уколико систем не може да креира особље он приказује администратору поруку “Систем не може да запамти особље”. (ИА)

Create new Personnel

Personnel

First name

The FirstName field is required.

Last name

The LastName field is required.

Born

The value " is invalid.

Country

Trademark

Image

Create

Слика 138. Неуспешно креирање особља

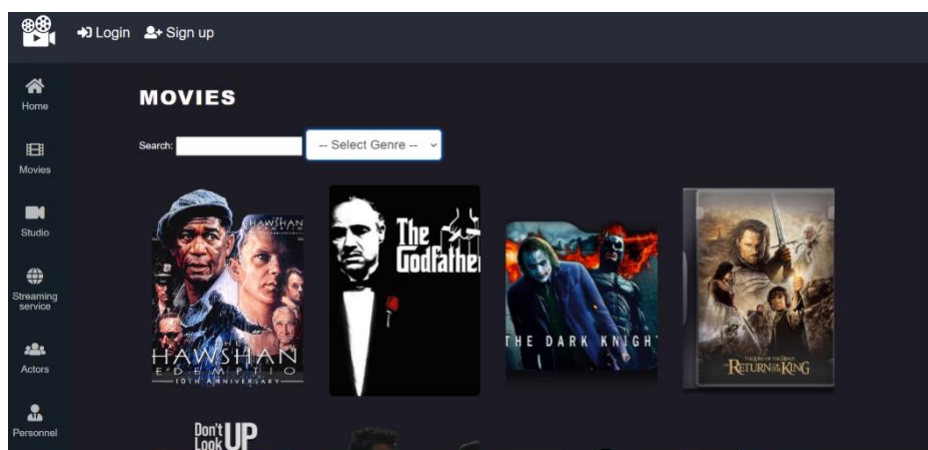
СК5: Случај коришћења – Претраживање филмова

Назив СК: Претраживање филмова

Актори СК: Корисник/Администратор

Учесници СК: Корисник/Администратор и систем (програм)

Предуслов: Систем је укључен и корисник/администратор је улогован под својом шифром. Систем приказује форму за рад са филмовима. Учитана је листа филмова.



Слика 139. Страница са филмовима

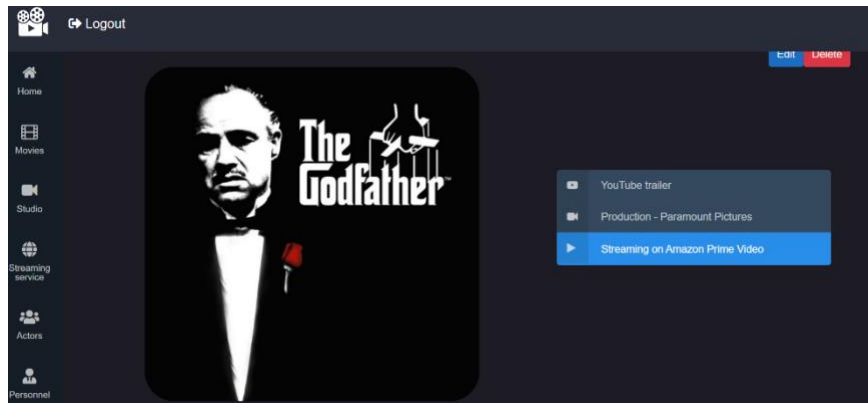
Основни сценарио СК

11. Корисник/Администратор **уноси** вредност по којој претражује филмове.
(АПУСО)
12. Корисник/Администратор **позива** систем да нађе филмове по задатој вредности.
(АПСО)
Опис акције: Корисник/Администратор уносом текста у “Textbox” позива системску операцију *Find(Movie)*
13. Систем **тражи** филмове по задатој вредности. (СО)
14. Систем **приказује** кориснику/администратору податке о филмовима и поруку:
“Систем је нашао филмове по задатој вредности”. (ИА)
15. Корисник/Администратор **бира** филм. (АПУСО)
16. Корисник/Администратор **позива** систем да прочита филм. (АПСО)

Опис акције: Корисник/Администратор притиском на филм позива системску операцију *GetSingle(Movie)*

17. Систем **учитава** филм. (СО)

18. Систем **приказује** кориснику/администратору податке о филму и поруку: “Систем је прочитао филм”. (ИА)

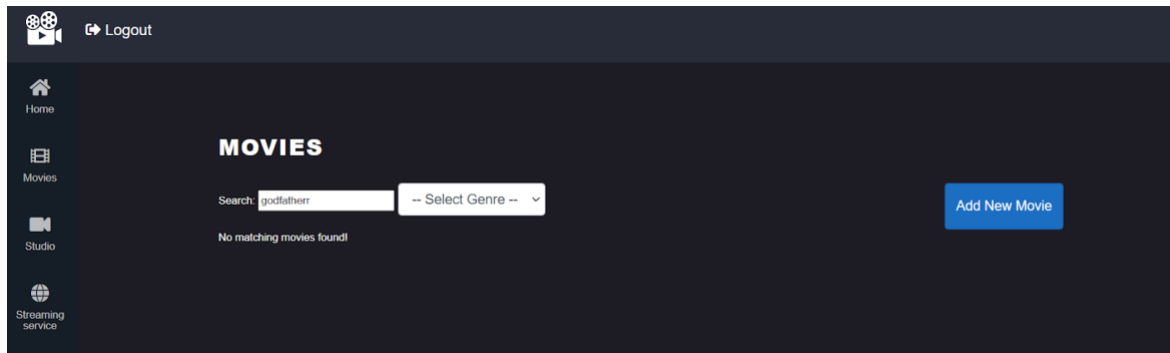


19.

Слика 140. Систем је успешно прочитао филм

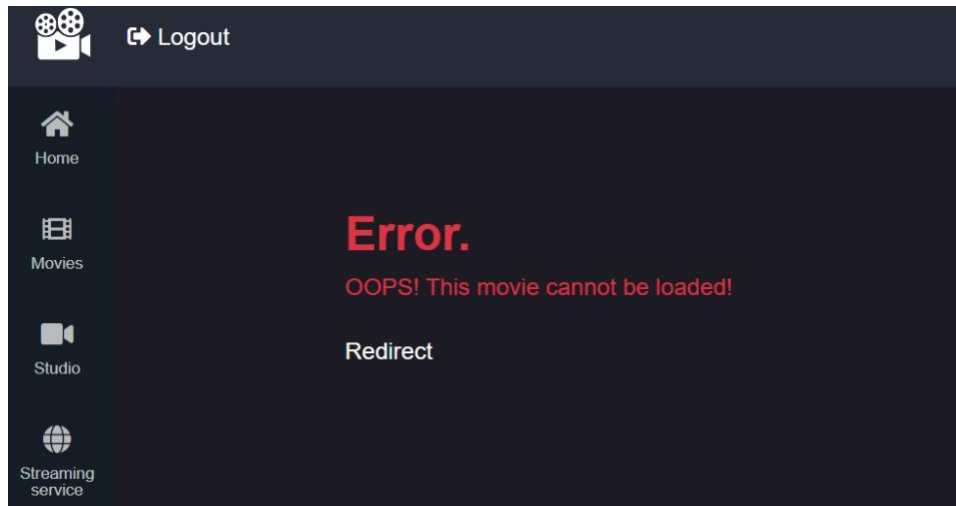
20. Алтернативна сценарија

4.1 Уколико систем не може да нађе филмове он приказује кориснику/администратору поруку: “Систем не може да нађе филмове по задатој вредности”. Прекида се извршење сценарија. (ИА)



Слика 141. Систем није пронашао филмове по задатој вредности

8.1 Уколико систем не може да прочита филм он приказује кориснику/администратору поруку “Систем не може да прочита филм”. (ИА)



Слика 142. Систем не може да учита филм

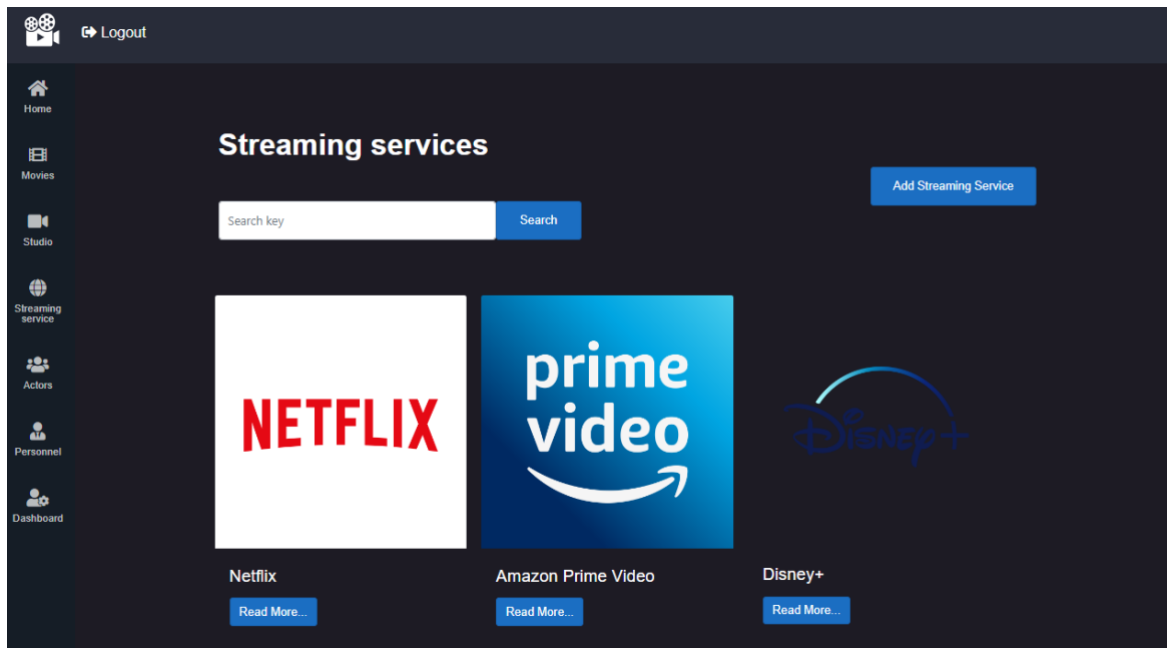
СК6: Случај коришћења – Промена података стриминг сервиса

Назив СК: Промена података стриминг сервиса

Актери СК: Администратор

Учесници СК: Администратор и систем (програм)

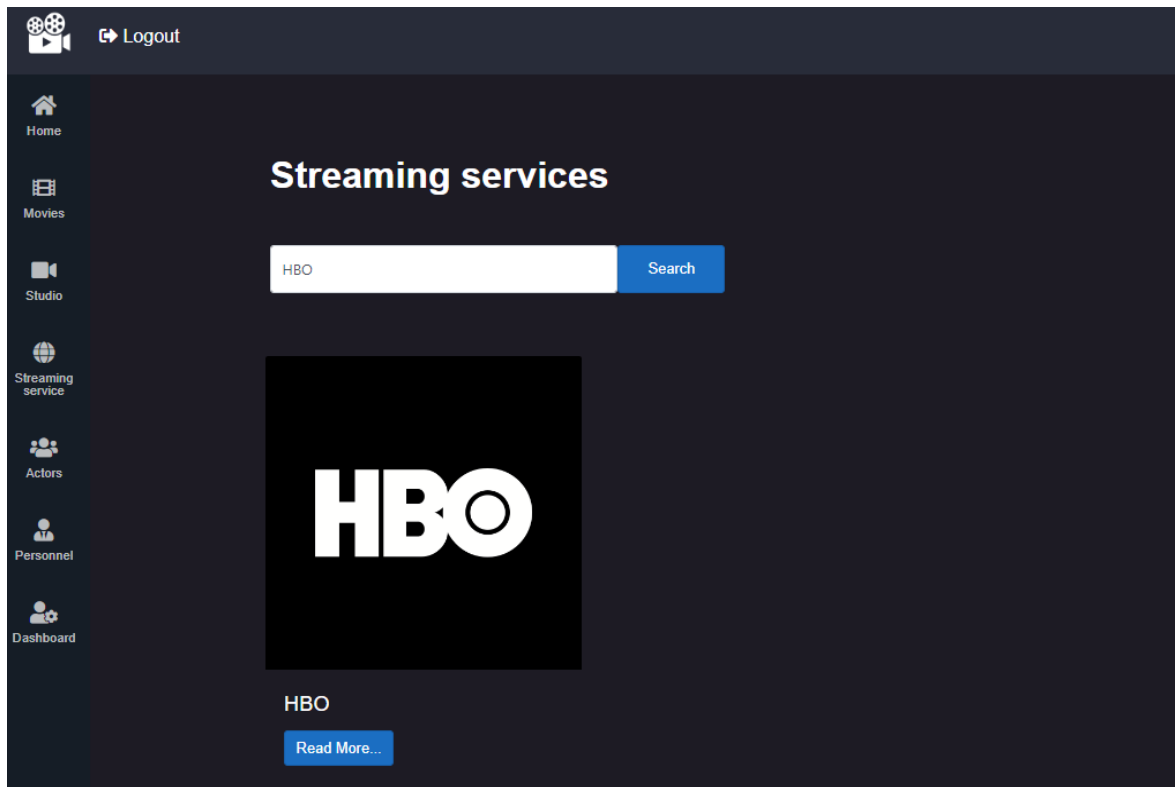
Предуслов: Систем је укључен и администратор је улогован под својом шифром. Систем приказује форму за рад са стриминг сервисом. Учитана је листа стриминг сервиса.



Слика 143. Страница са стриминг сервисима

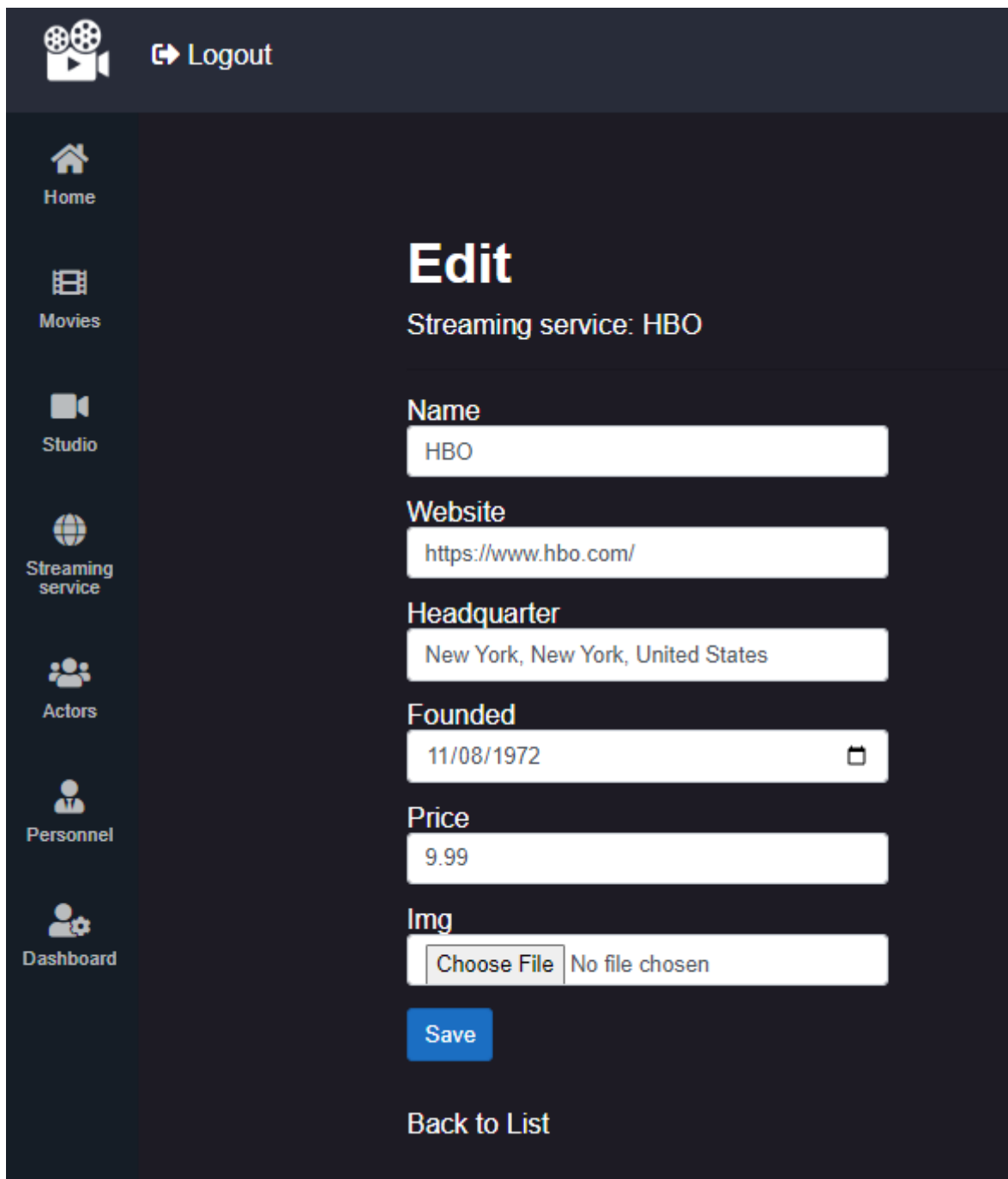
Основни сценарио СК

14. Администратор уноси вредност по којој претражује стриминг сервисе. (АПУСО)
15. Администратор позива систем да нађе стриминг сервис по задатој вредности.
(АПСО)
Опис акције: Администратор притиском на дугме “*Search*” позива системску операцију *Find(StreamingService)*
16. Систем тражи стриминг сервисе по задатој вредности. (СО)
17. Систем приказује студије и поруку: “Систем је нашао стриминг сервисе по задатој вредности”. (ИА)



Слика 144. Приказ претражених стриминг сервиса

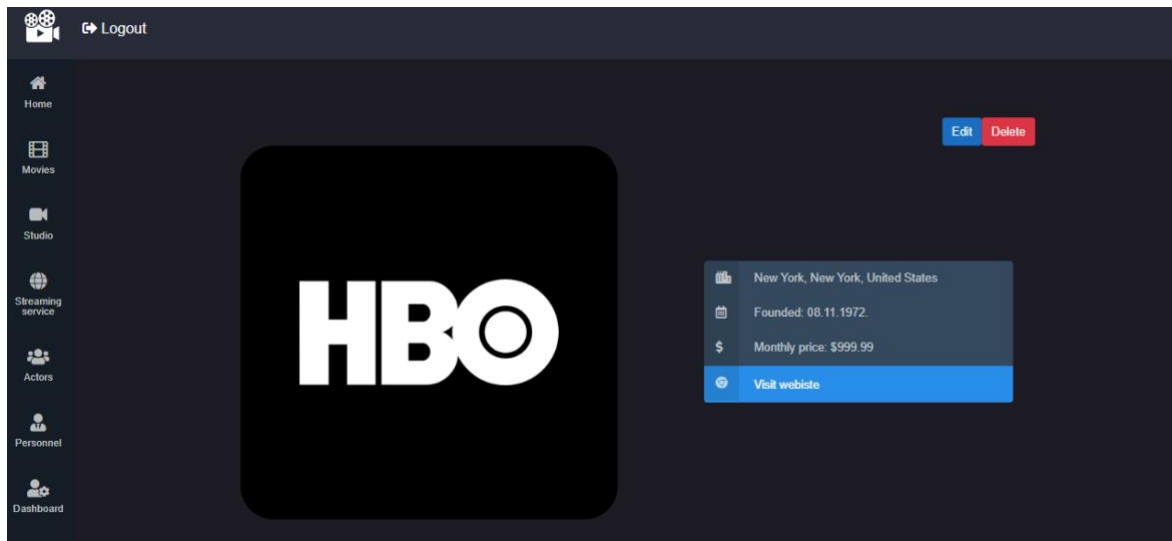
18. Администратор бира стриминг сервис. (АПУСО)
19. Администратор позива систем да учита стриминг сервис.(АПСО)
Опис акције: Администратор притиском на дугме “*Edit*” позива системску операцију *GetSingle(StreamingService)*
20. Систем учитава стриминг сервис. (СО)
21. Систем показује администратору податке о стриминг сервису и поруку “Систем је учитао стриминг сервис“. (ИА)



Слика 145. Успешно учитан стриминг сервис

22. Администратор **уноси(мења)** податке о стриминг сервису. (АПУСО)
23. Администратор **контролише** да ли је коректно унео податке о стриминг сервису. (АНСО)
24. Администратор **позива** систем да запамти стриминг сервис. (АПСО)
Опис акције: Администратор притиском на дугме “Save” позива системску операцију *Update(StreamingService)*
25. Систем **памти** стриминг сервис. (СО)

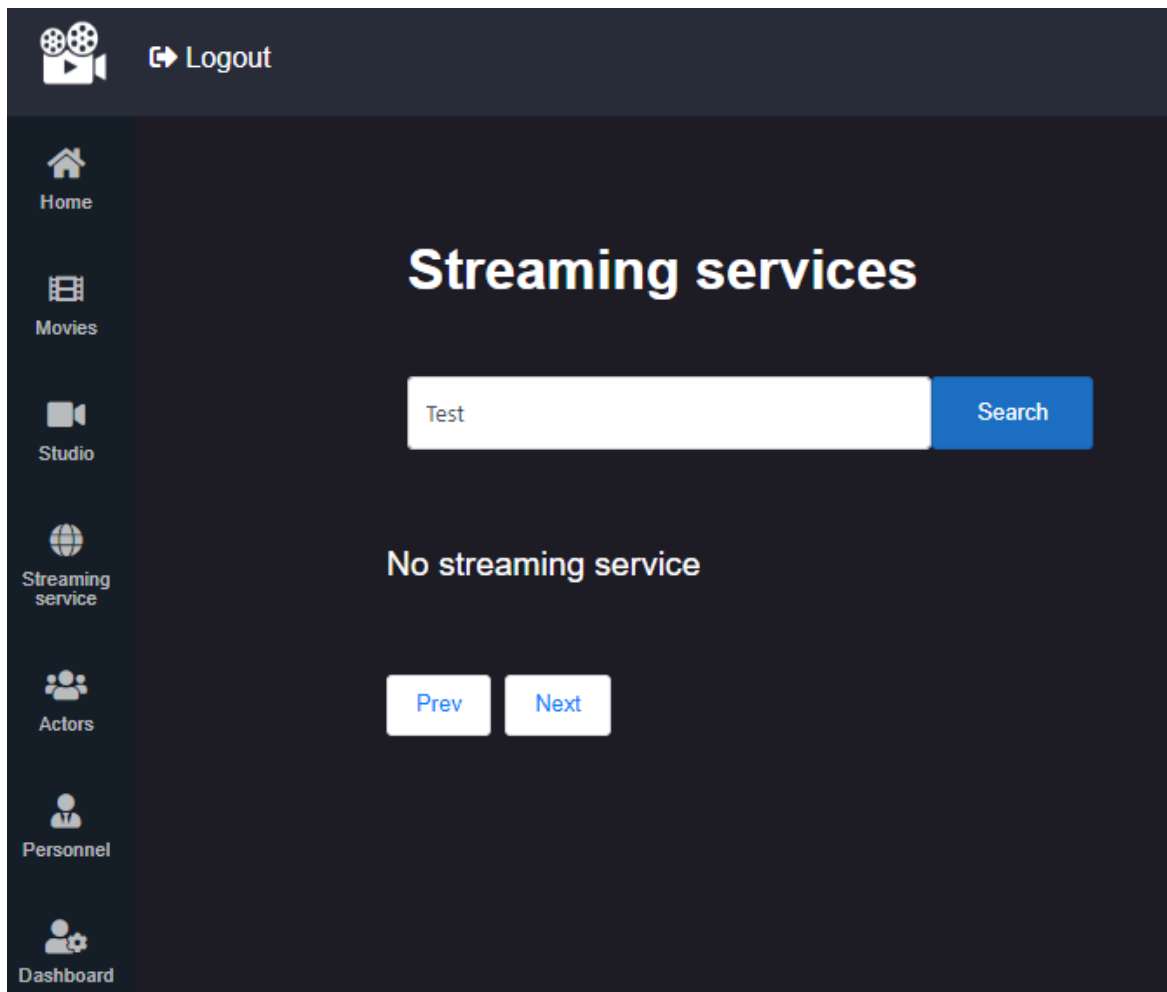
26. Систем **приказује** администратору запамћени стриминг сервис и поруку:
“Систем је запамтио стриминг сервис”.(ИА)



Слика 146. Успешно измењен стриминг сервис

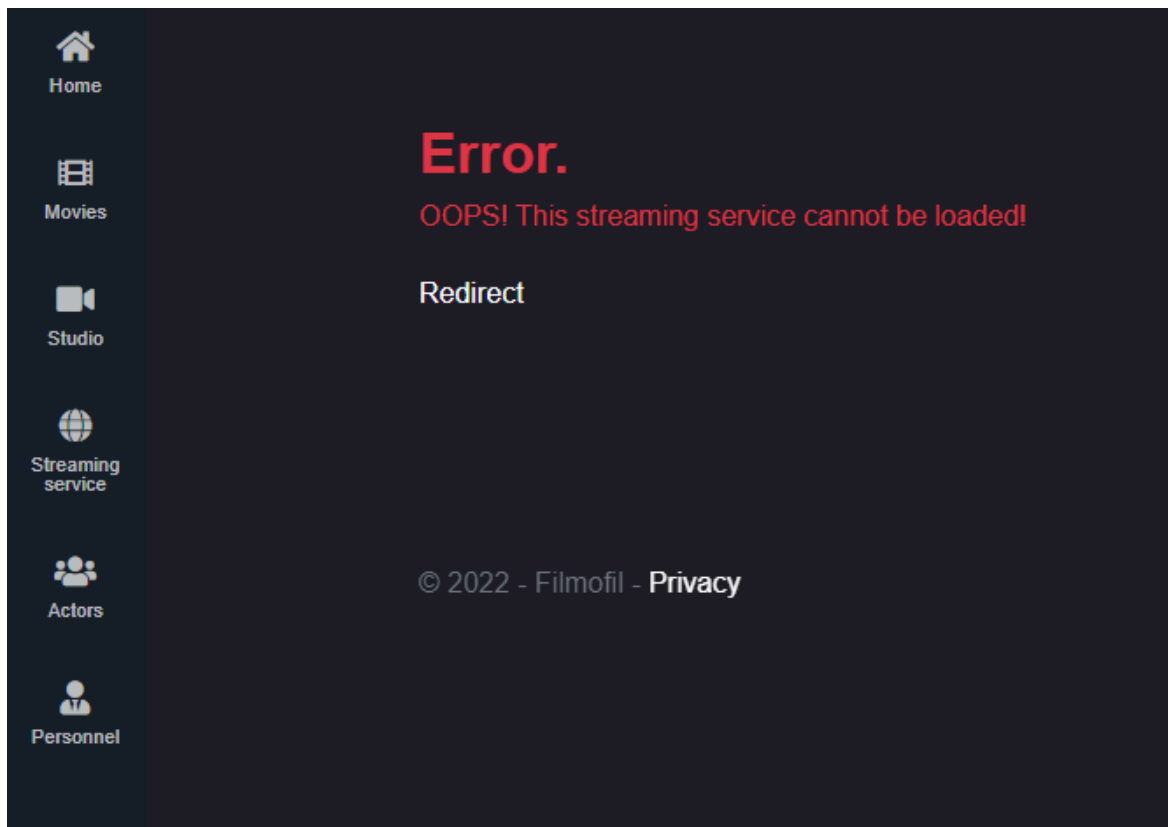
Алтернативна сценарија

4.1. Уколико систем не може да нађе стриминг сервис он приказује администратору поруку: “Систем не може да нађе стриминг сервис по задатој вредности”. Прекида се извршење сценарија. (ИА)



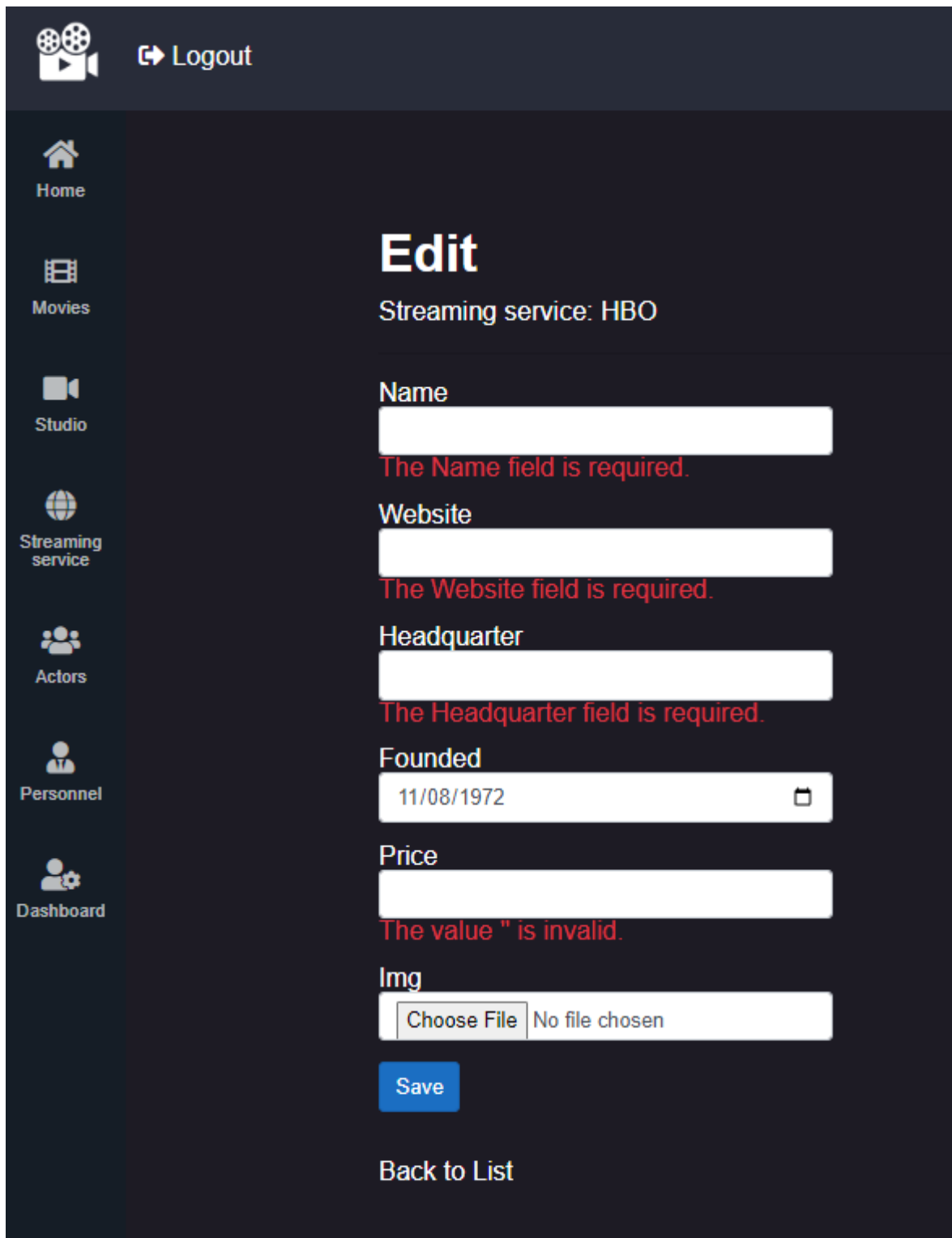
Слика 147. Систем није пронашао стриминг сервис по задатој вредности

8.1. Уколико систем не може да учита стриминг сервис он приказује администратору поруку “Систем не може да учита стриминг сервис”. Прекида се извршење сценариа. (ИА)



Слика 148. Неуспешно учитавање стриминг сервиса

13.1. Уколико систем не може да запамти податке о стриминг сервису он приказује администратору поруку “Систем не може да запамти стриминг сервис”. (ИА)



Слика 149. Неуспешна измена стриминг сервиса

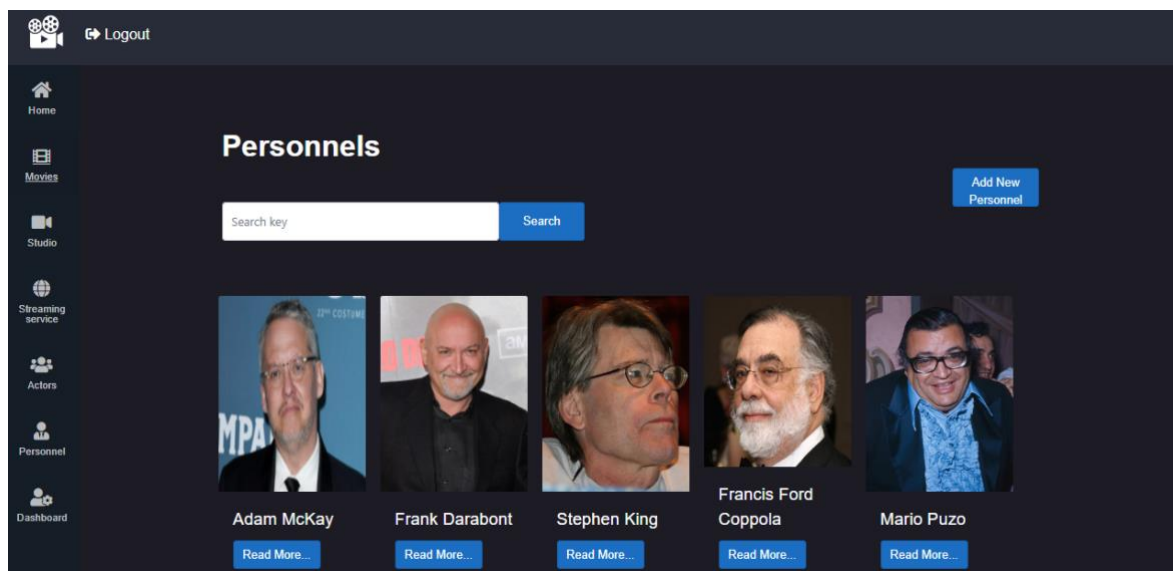
СК7: Случај коришћења – Промена података особља

Назив СК: Промена података особља

Актери СК: Администратор

Учесници СК: Администратор и систем (програм)

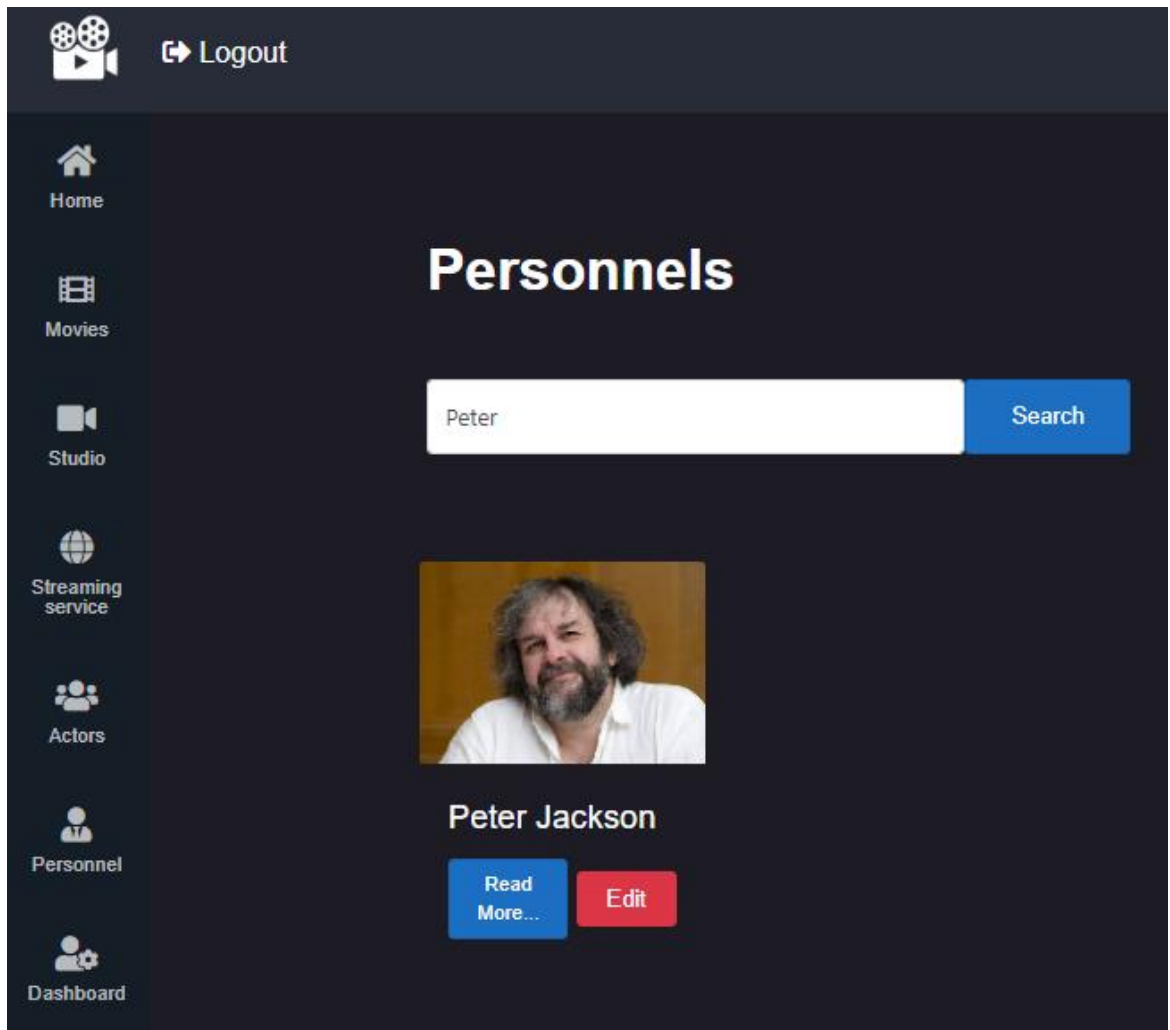
Предуслов: Систем је укључен и администратор је улогован под својом шифром. Систем приказује форму за рад са особљем. Учитана је листа особља.



Слика 150. Страница са особљем

Основни сценарио СК

14. Администратор **уноси** вредност по којој претражује особље. (АПУСО)
15. Администратор **позива** систем да нађе особље по задатој вредности. (АПСО)
Опис акције: Администратор притиском на дугме “*Search*” позива системску операцију *Find(Personnel)*
16. Систем **тражи** особље по задатој вредности. (СО)
17. Систем **приказује** глумце и поруку: “Систем је нашао особље по задатој вредности”. (ИА)



Слика 151. Систем је нашао особље по задатој вредности

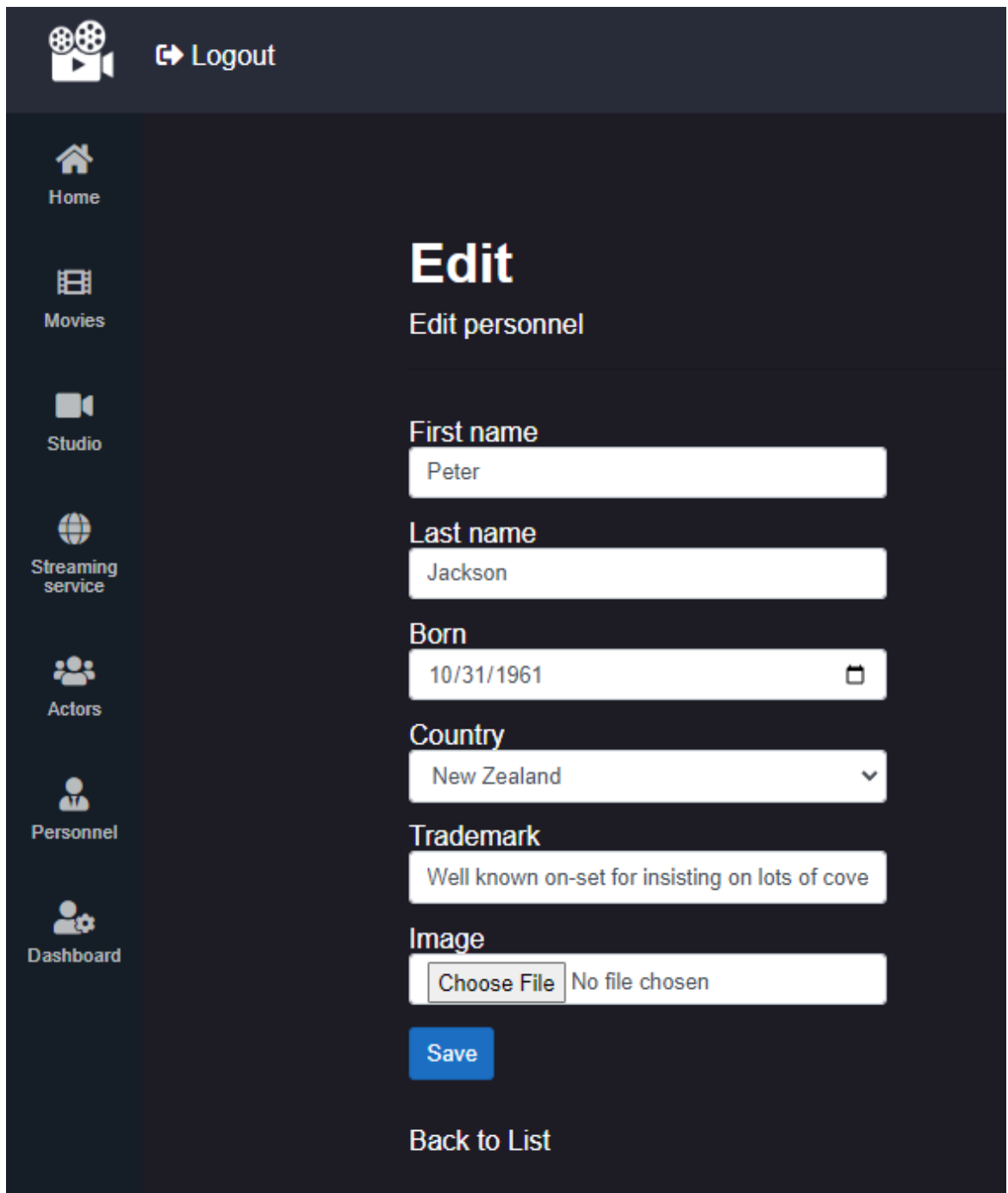
18. Администратор **бира** особље. (АПУСО)

19. Администратор **позива** систем да учита особље.(АПСО)

Опис акције: Администратор притиском на дугме “*Edit*” позива системску операцију *GetSingle(Personnel)*

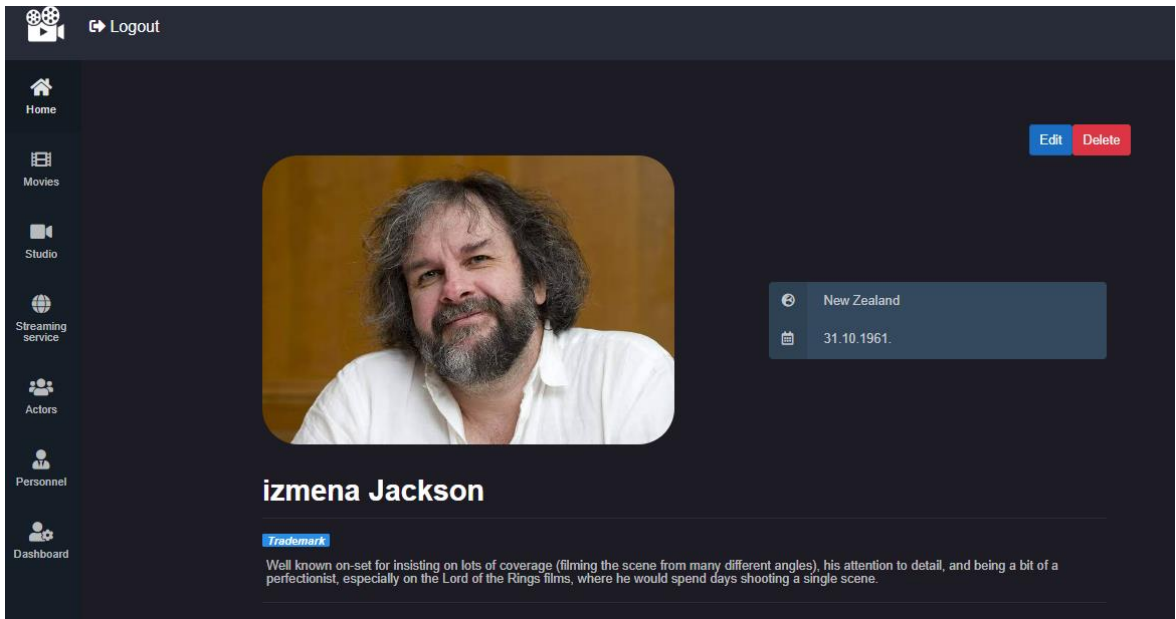
20. Систем **учитава** особље. (СО)

21. Систем **показује** администратору податке о особљу и поруку “Систем је учитао особље“. (ИА)



Слика 152. Успешно учитано особље

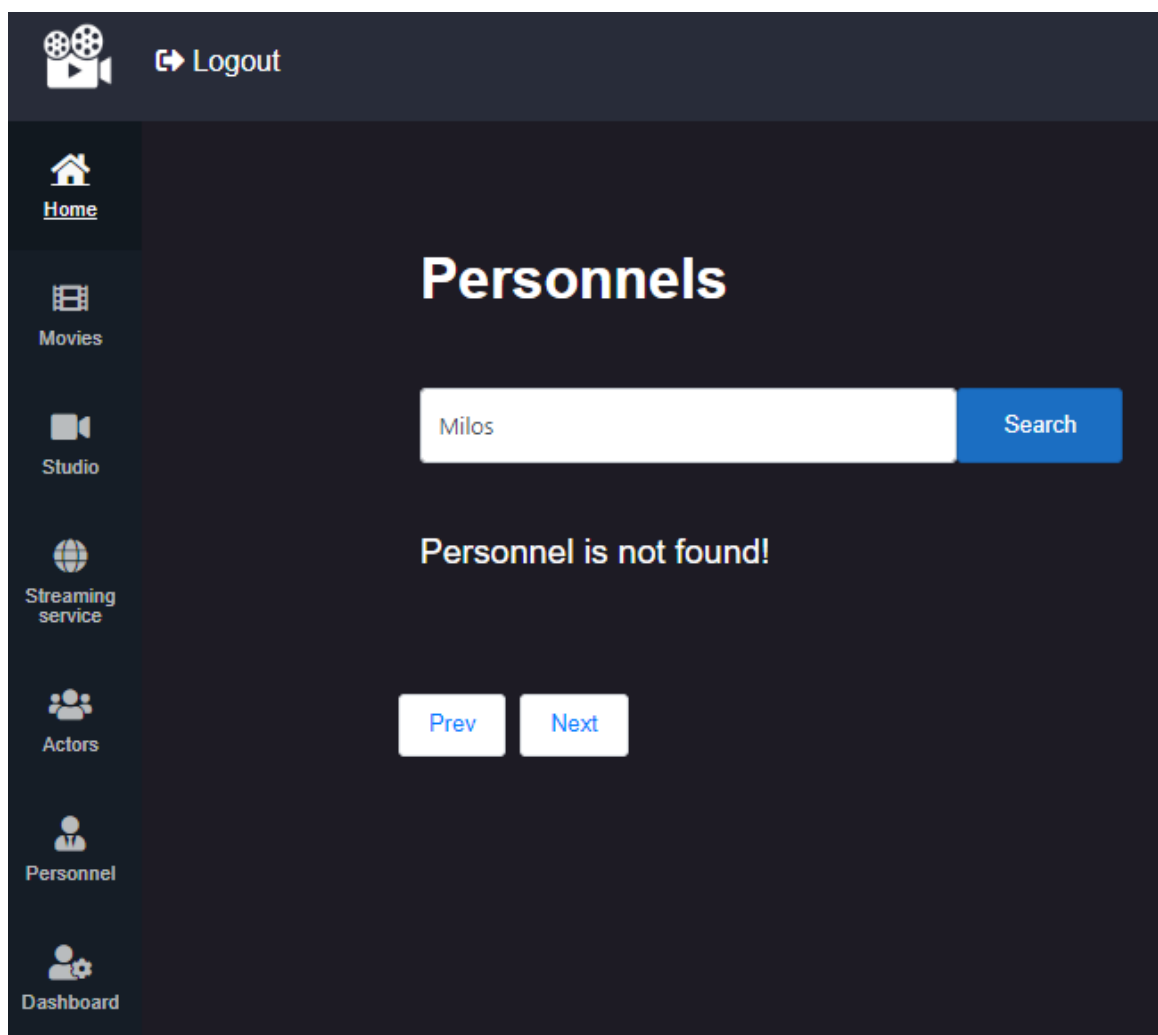
22. Администратор **уноси(мења)** податке о особљу. (АПУСО)
23. Администратор **контролише** да ли је коректно унео податке о особљу. (АНСО)
24. Администратор **позива** систем да запамти особље. (АПСО)
25. Систем **памти** особље. (СО)
26. Систем **приказује** администратору запамћеног припадника особља и поруку:
“Систем је запамтио особље”. (ИА)



Слика 153. Успешно измењено особље

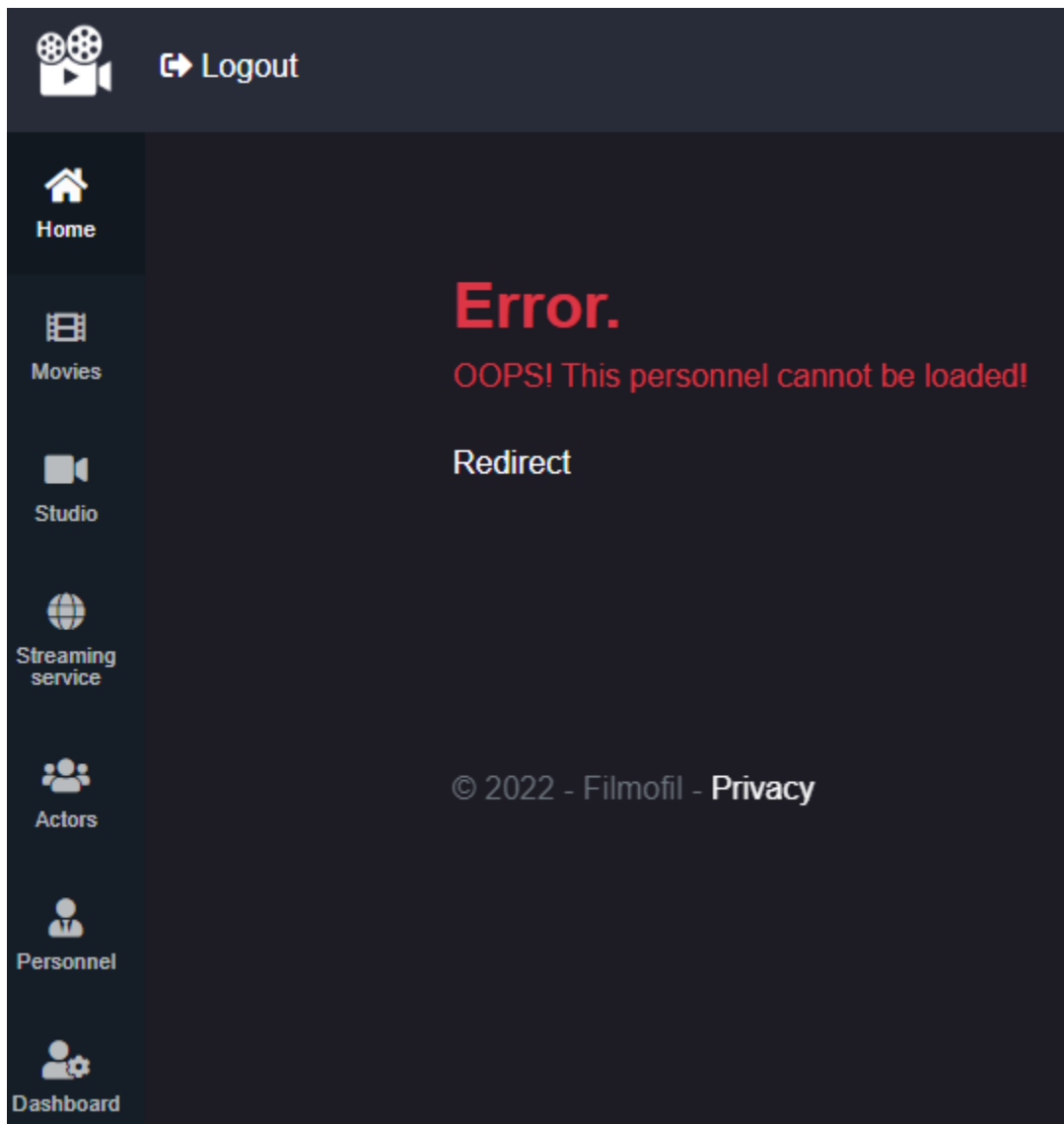
Алтернативна сценарија

4.1. Уколико систем не може да нађе особље он приказује администратору поруку: “Систем не може да нађе особље по задатој вредности”. Прекида се извршење сценарија. (ИА)



Слика 154. Систем није пронашао особље по задатој вредности

8.1. Уколико систем не може да учита припадника особља он приказује администратору поруку “Систем не може да учита припадника особља”. Прекида се извршење сценарија. (ИА)



Слика 155. Неуспешно учитавање особља

13.1. Уколико систем не може да запамти податке о припаднику особља он приказује администратору поруку “Систем не може да запамти припадника

особља”. (ИА)

Logout

Home

Movies

Studio

Streaming service

Actors

Personnel

Dashboard

Edit

Edit personnel

First name

The FirstName field is required.

Last name

The LastName field is required.

Born

10/31/1961

Country

New Zealand

Trademark

Well known on-set for insisting on lots of cove

Image

Choose File No file chosen

Save

Back to List

Слика 156. Систем не може да запамти особље

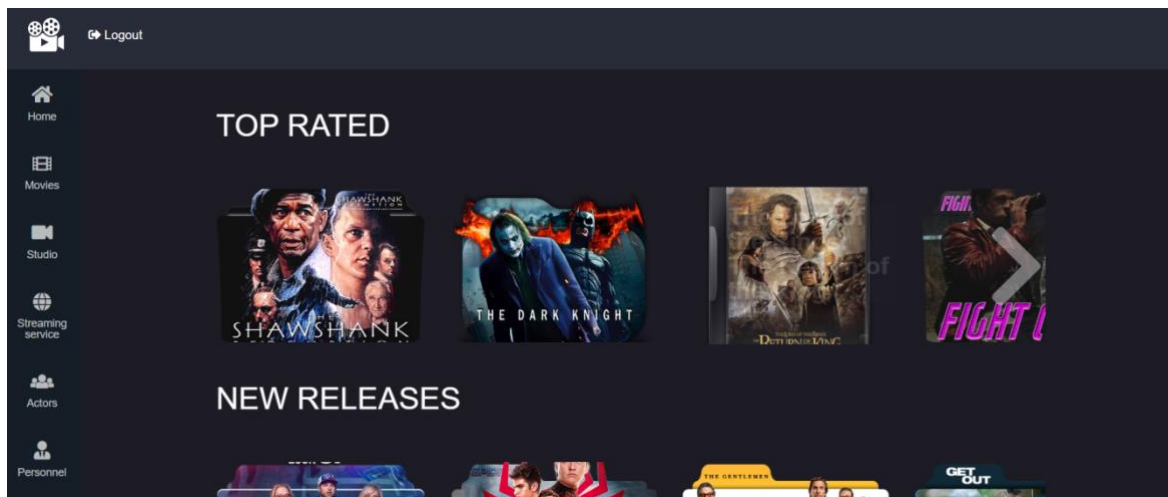
СК8: Случај коришћења – Брисање филма

Назив СК: Брисање филма

Актери СК: Администратор

Учесници СК: Администратор и систем (програм)

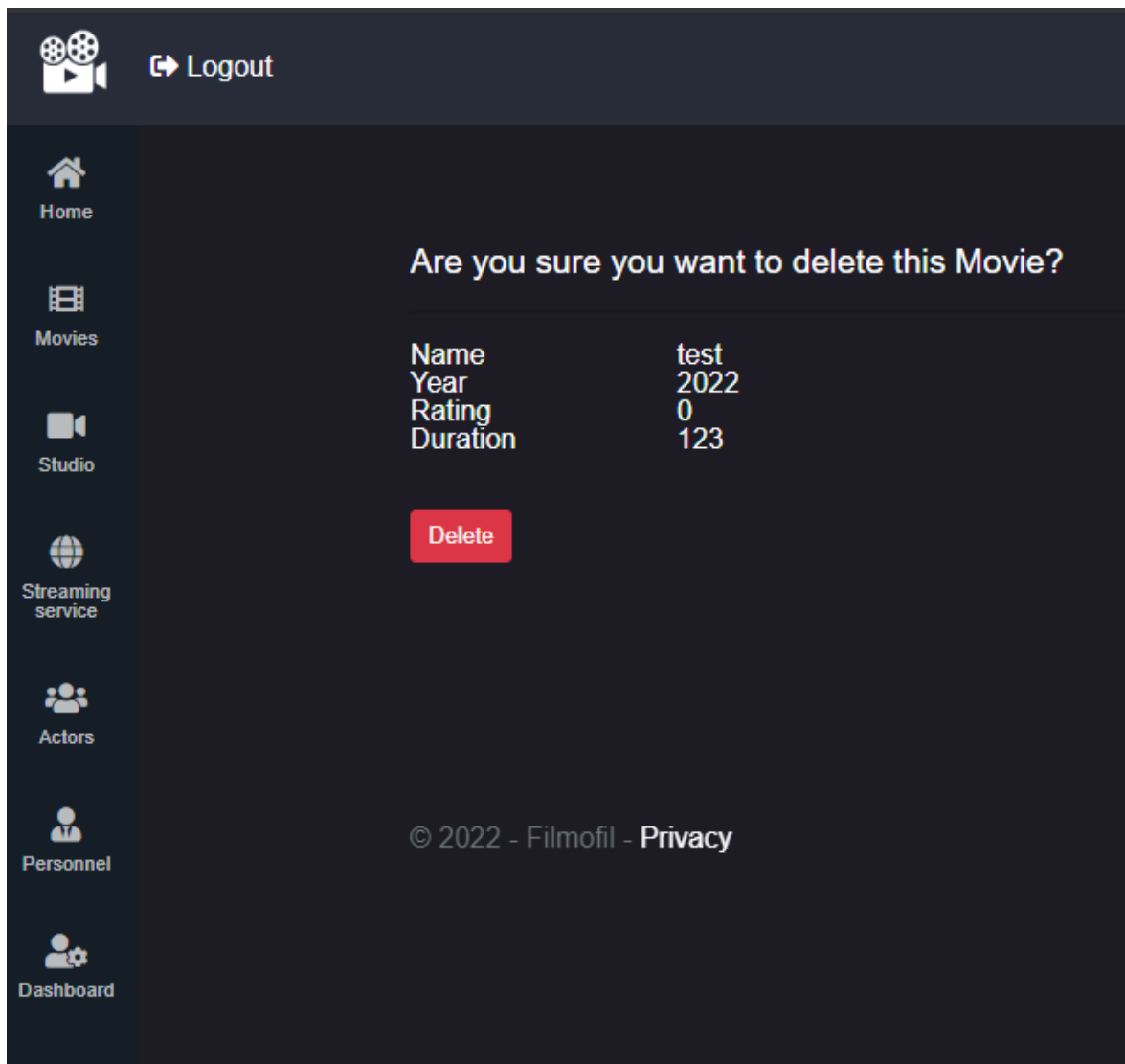
Предуслов: Систем је укључен и администратор је улогован под својом шифром. Систем приказује форму за рад са филмовима. Учитана је листа филмова.



Слика 157. Страница са филмовима

Основни сценарио СК

9. Администратор **уноси** вредност по којој претражује филм. (АПУСО)
10. Администратор **позива** систем да нађе филм по задатој вредности. (АПСО)
11. Систем **тражи** филм по задатој вредности. (СО)
12. Систем **приказује** кориснике и поруку: “Систем је нашао филм по задатој вредности”. (ИА)
13. Администратор **бира** филм. (АПУСО)
14. Администратор **позива** систем да обрише филм. (АПСО)



Слика 158. Позив система да обрише филм

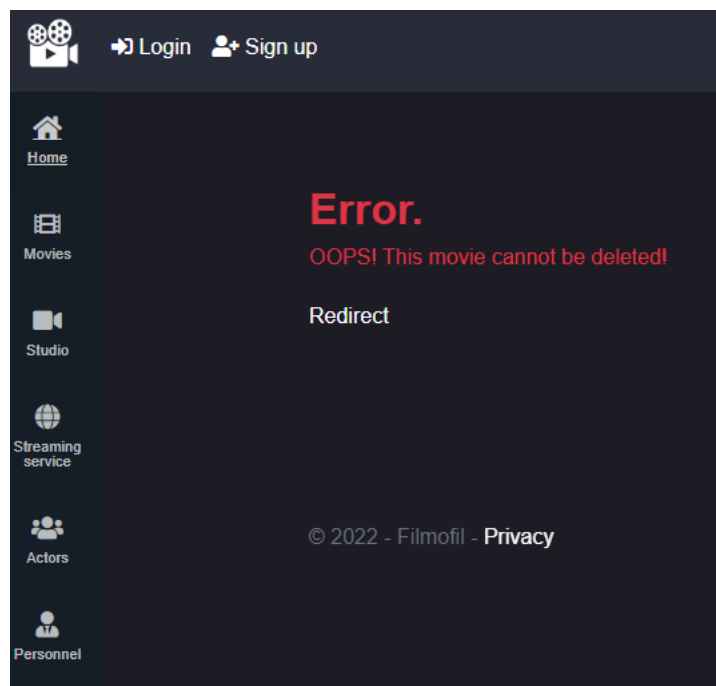
15. Систем **брише** филм. (СО)

16. Систем **приказује** администратору поруку: “Систем је обрисао филм”. (ИА)

Алтернативна сценарија

4.1. Уколико систем не може да нађе филм он приказује администратору поруку: “Систем не може да нађе филм по задатој вредности”. Прекида се извршење сценарија. (ИА)

8.1. Уколико систем не може да обрише филм он приказује администратору поруку “Систем не може да обрише филм”. (ИА)



Слика 159. Неуспешно брисање филма

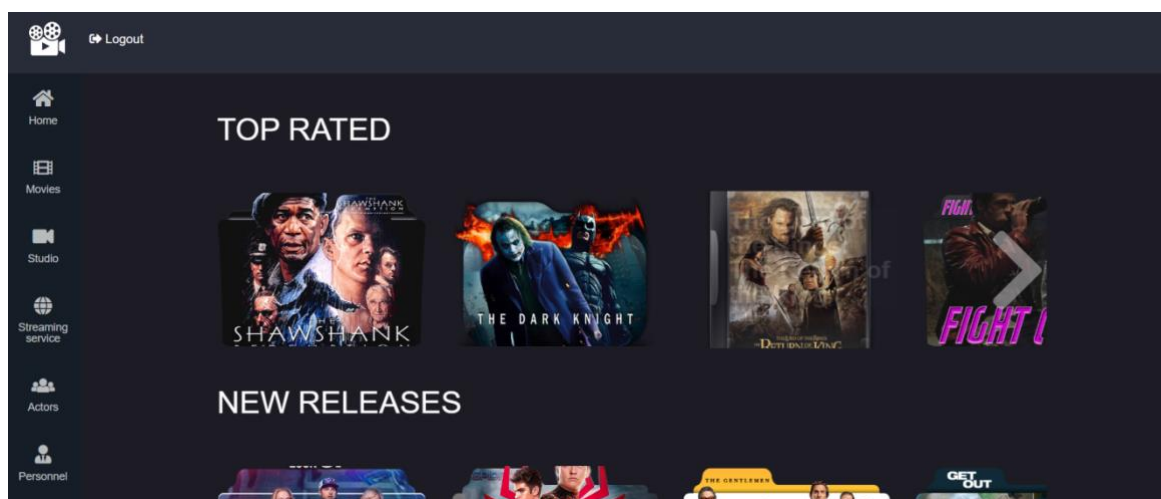
СК9: Случај коришћења – Оцењивање филма

Назив СК: Оцењивање филма

Актери СК: Корисник / Администратор

Учесници СК: Корисник / Администратор и систем (програм)

Предуслов: Систем је укључен и корисник / администратор је улогован под својом шифром. Систем приказује форму за рад са филмовима. Учитана је листа филмова.



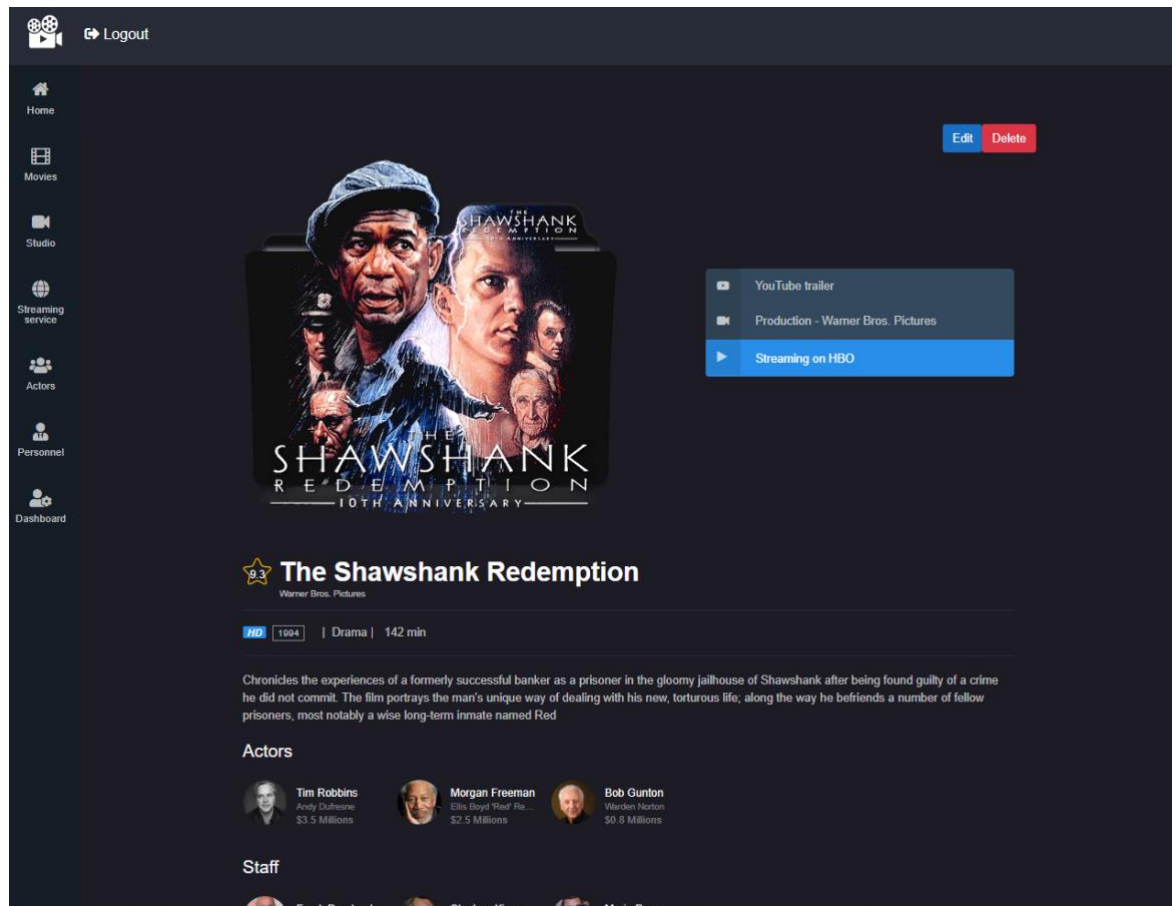
Слика 160. Страница са филмовима

Основни сценарио СК

14. Корисник/Администратор **уноси** вредност по којој претражује филмове. (АПУСО)
15. Корисник/Администратор **позива** систем да нађе филмове по задатој вредности. (АПСО)
Опис акције: Администратор уносом текста у Textbox позива системску операцију Find(Movie)
16. Систем **тражи** филмове по задатој вредности. (СО)
17. Систем **приказује** филмове и поруку: “Систем је нашао филмове по задатој вредности”. (ИА)
18. Корисник/Администратор **бира** филм. (АПУСО)
19. Корисник/Администратор **позива** систем да учита филм.(АПСО)

20. Систем **учитава** филм. (СО)

21. Систем **показује** кориснику/администратору податке о филму и поруку “Систем је прочитао филм“. (ИА)



Слика 161. Филм

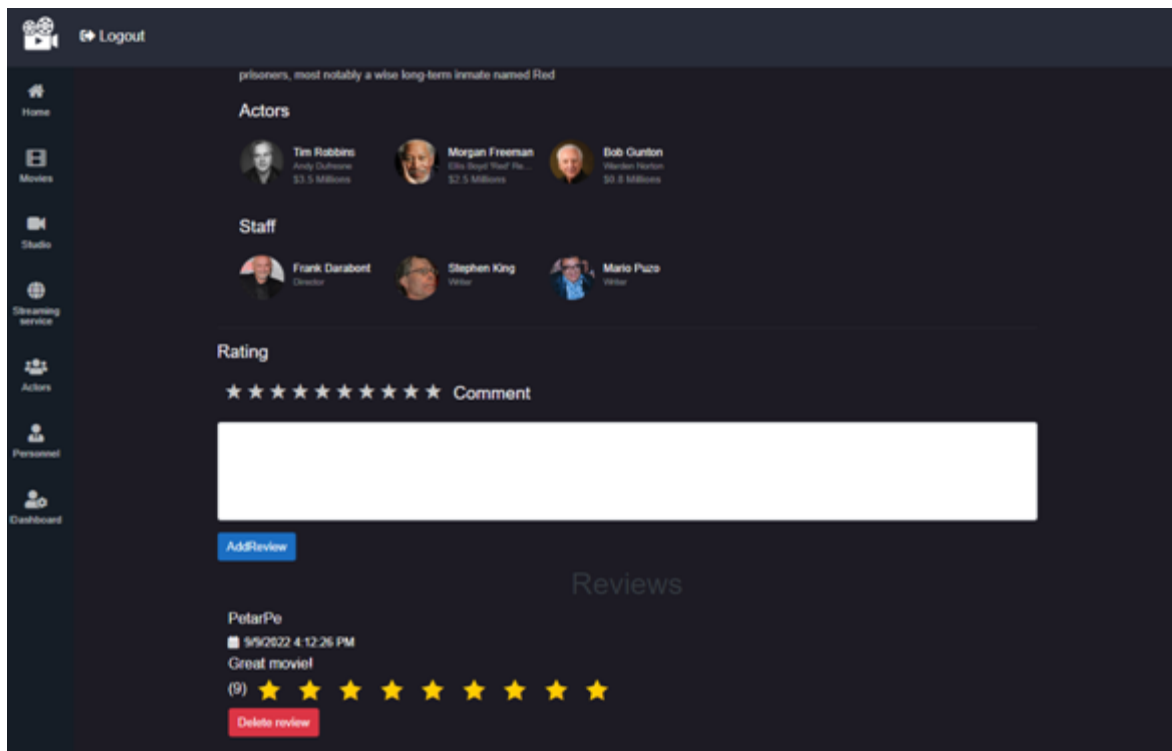
22. Корисник/Администратор **уноси(мења)**оцену и коментар о филму. (АПУСО)

23. Корисник/Администратор **контролише** да ли је коректно унео оцену и коментар о филму. (АНСО)

24. Корисник/Администратор **позива** систем да запамти оцену и коментар о филму. (АПСО)

25. Систем **памти** оцену и коментар о филму. (СО)

26. Систем **приказује**кориснику/администраторузапамћену оцену и коментар о филму и поруку: “Систем је запамтио оцену и коментар о филму“. (ИА)

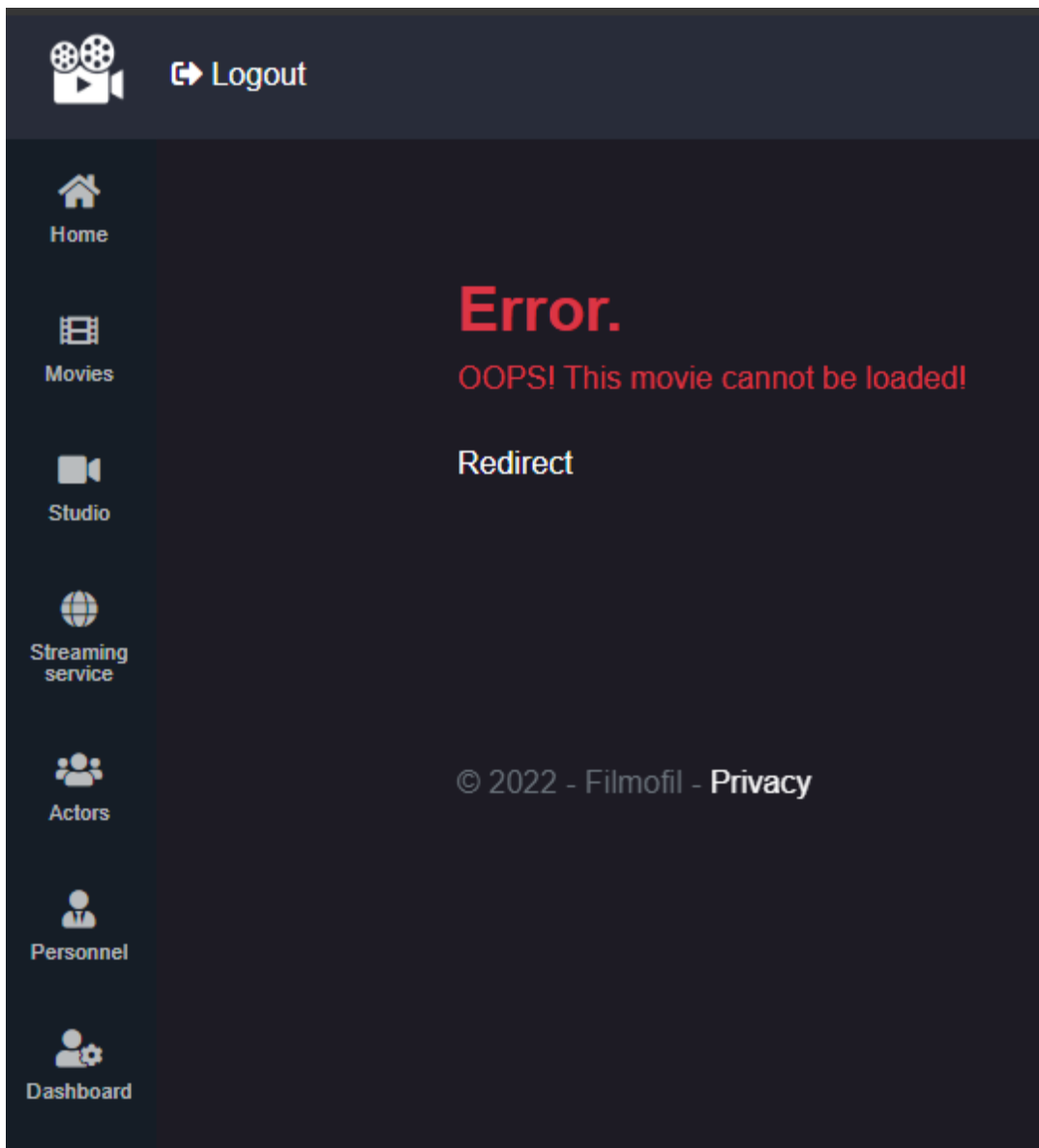


Слика 162. Форма за оцењивање филма

Алтернативна сценарија

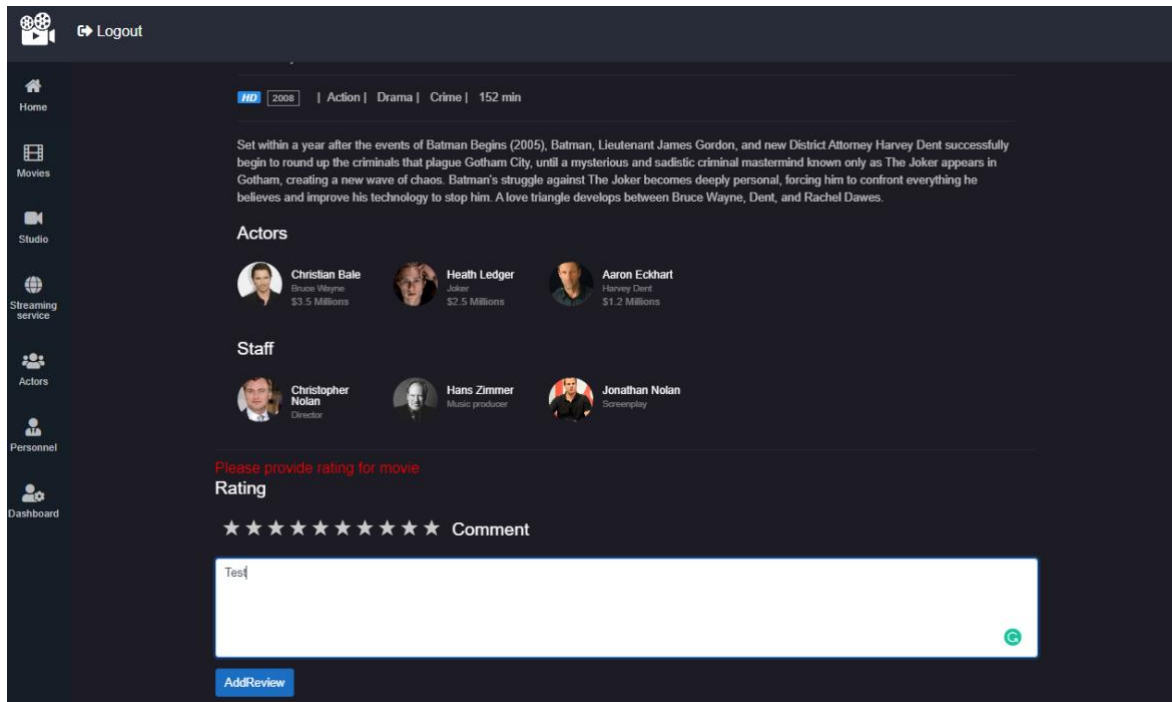
4.1. Уколико систем не може да нађе филмове он приказује кориснику/администратору поруку: “Систем не може да нађе филмове по задатој вредности”. Прекида се извршење сценарија. (ИА)

8.1. Уколико систем не може да учита филм он приказује кориснику/администратору поруку “Систем не може да учита филм”. Прекида се извршење сценарија. (ИА)



Слика 163. Неуспешно учитавање филма

13.1. Уколико систем не може да запамти оцену и коментар о филму он приказује кориснику/администратору поруку “Систем не може да запамти оцену и коментар о филму”. (ИА)



Слика 164. Неисправно попуњена форма

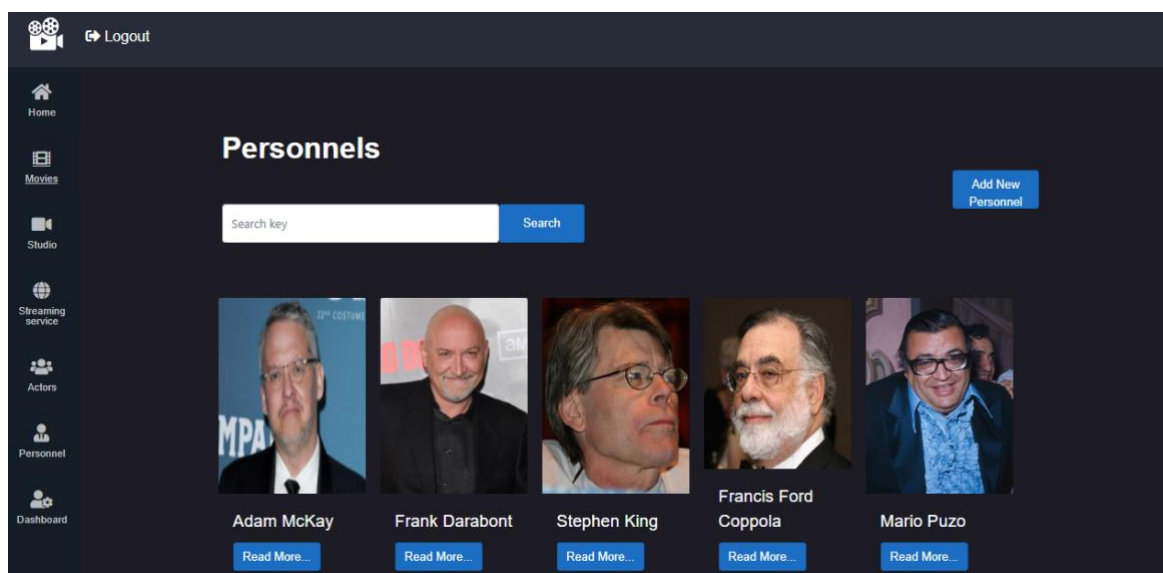
СК10: Случај коришћења – Претраживање особља

Назив СК: Претраживање особља

Актори СК: Корисник/Администратор

Учесници СК: Корисник/Администратор и систем (програм)

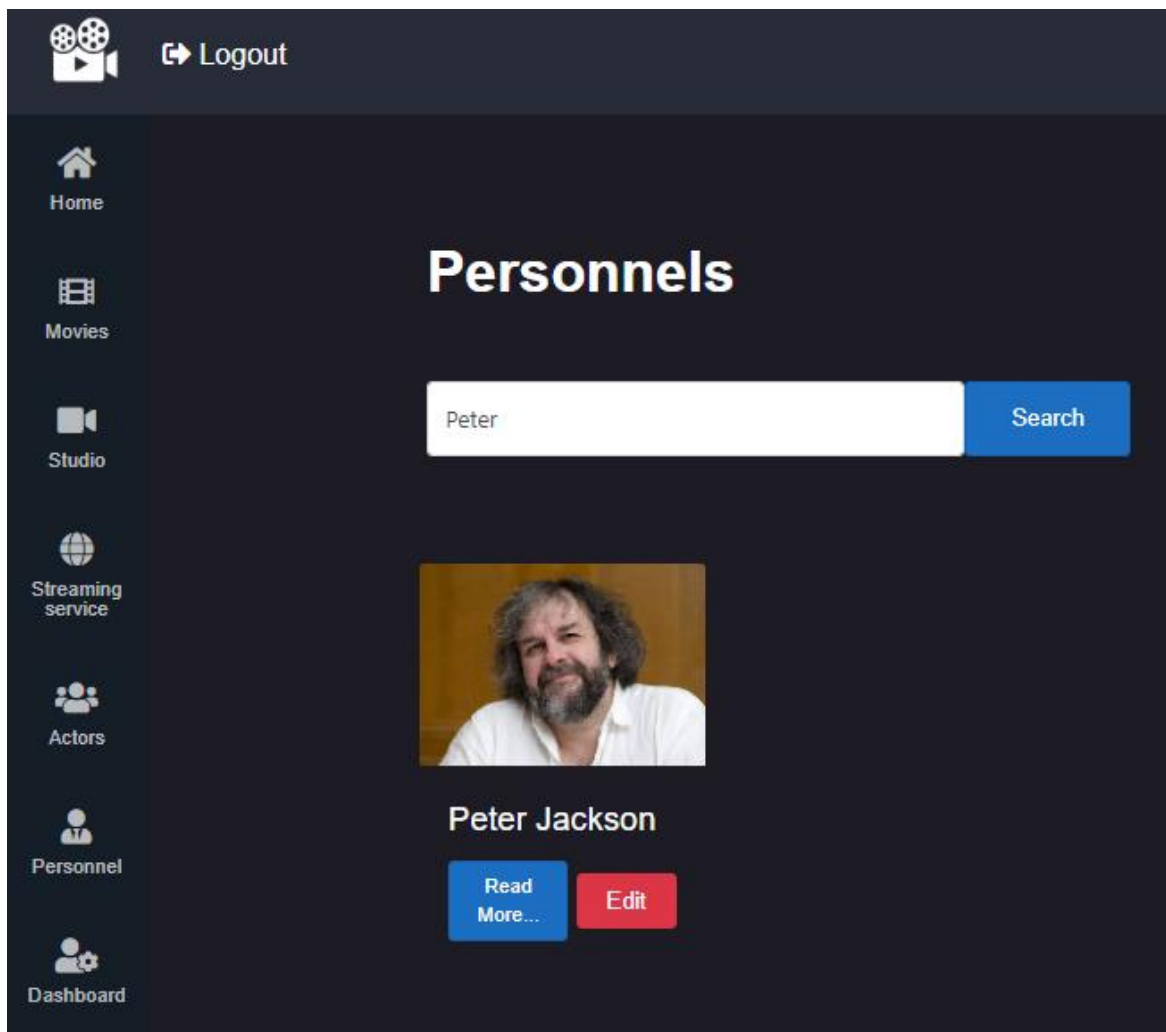
Предуслов: Систем је укључен и корисник/администратор је улогован под својом шифром. Систем приказује форму за рад са особљем. Учитана је листа особља.



Слика 165. Страница са особљем

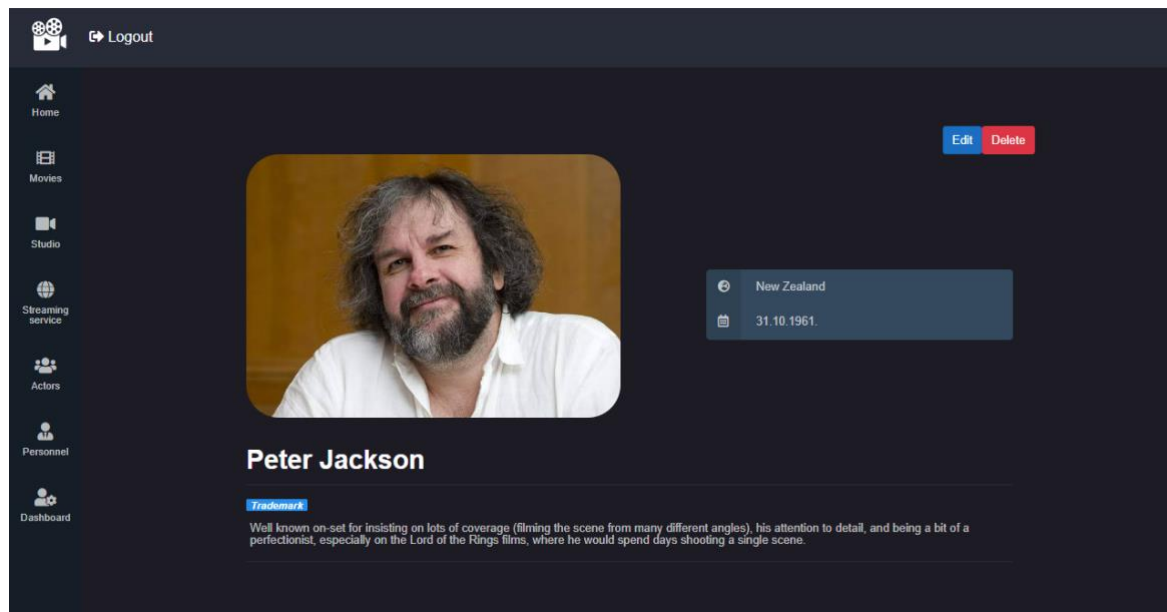
Основни сценарио СК

9. Корисник/Администратор **уноси** вредност по којој претражује особље. (АПУСО)
10. Корисник/Администратор **позива** систем да нађе припаднике особља по задатој вредности. (АПСО)
11. Систем **тражи** припаднике особља по задатој вредности. (СО)
12. Систем **приказује** кориснику/администратору податке о особљу и поруку: “Систем је нашао припаднике особља по задатој вредности”. (ИА)



Слика 166. Систем је нашао особље по задатој вредности

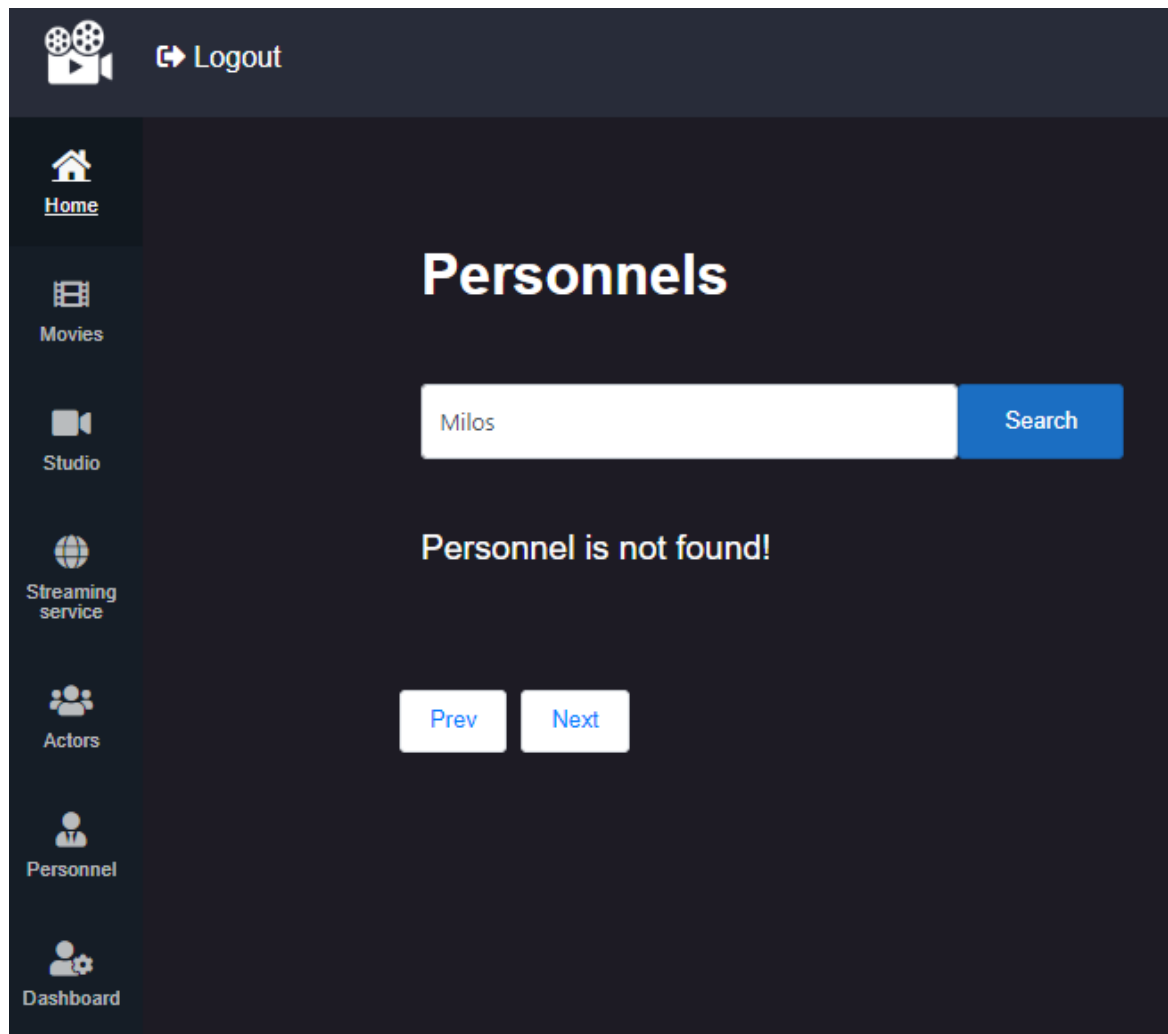
13. Корисник/Администратор **бира** припадника особља. (АПУСО)
14. Корисник/Администратор **позива** систем да учита припадника особља. (АПСО)
15. Систем **учитава** припадника особља. (СО)
16. Систем **приказује** кориснику/администратору податке о припаднику особља и поруку: “Систем је учитао припадника особља”. (ИА)



Слика 167. Страница нађеног особља

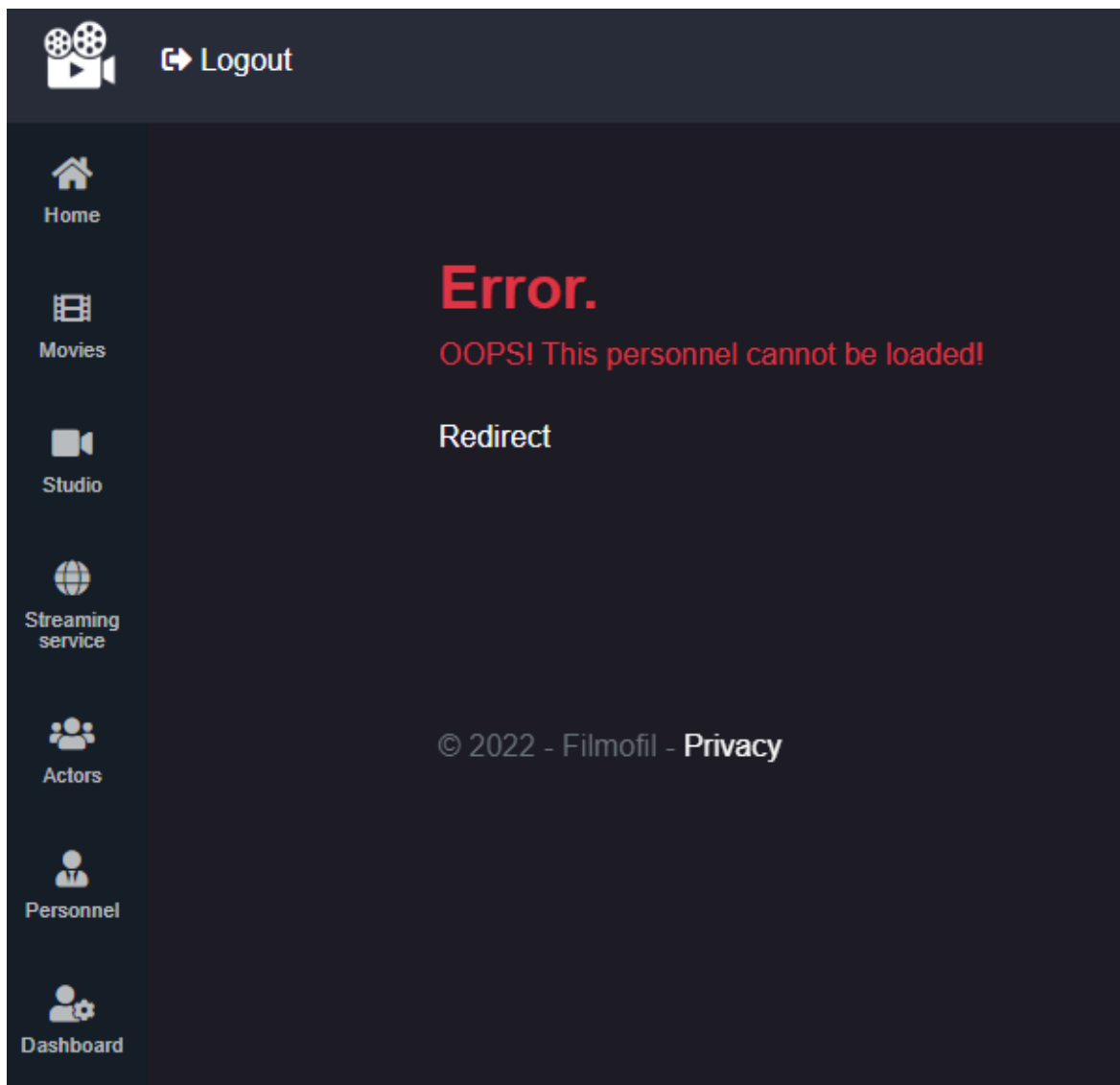
Алтернативна сценарија

4.1 Уколико систем не може да нађе припаднике особље он приказује кориснику/администратору поруку: “Систем не може да нађе припаднике особље по задатој вредности”. Прекида се извршење сценарија. (ИА)



Слика 168. Не постоји особа са задатим именом

8.1 Уколико систем не може да учита припадника особља он приказује кориснику/администратору поруку “Систем не може да учита припадника особља”. (ИА)



Слика 169. Неуспешно учитавање особља

8. ТЕСТИРАЊЕ

Фаза тестирања ове апликације морала је обухватити испитивање рада приликом уношења исправних, али и неисправних података како би се утврдио њен рад. Уношењем исправних података, проверава се да ли је у одговарајуће базе уписан одговарајући податак, као и исправност наставка коришћења апликације. Уношењем неисправних података проверава се испис одговарајућег типа грешке.

9. ЗАКЉУЧАК

Убрзани развој веб технологија отворио је могућност развоја најразличитијих веб апликација. Овим дипломским радом обухваћен је развој веб апликације која је намењена за рецензије филмова и њихову евиденцију од стране пријављених корисника – од саме спецификације захтева и анализе семантике рада, па све до креирања софтвера. Посебно је дат нагласак на елементима вишеслојне архитектуре софтвера. Овакав вид софтверске архитектуре у значајној мери олакшава управљање и одржавање сваког слоја софтвера.

Што се тиче функционалности, развој једне веб апликације никада није у потпуности готов процес. Функционалности се мењају у зависности од захтева корисника и новим технологијама које се свакодневно развијају. Ова апликација ради као један функционални систем за управљање подацима али се функционалности исте итекако могу проширити.

Оваквом типу апликације нека будућа унапређења могла би бити: Пријављивање на сервис путем неке платформи друштвених мрежа као што су *Facebook* и *Twitter*, или директно путем *Google* налога; додавање препоруке филмова засноване на алгоритмима вештачке интелигенције; најаве нових филмова у биоскопима са додатим трејлерима са глумцима и информацијама о датуму изласка; дељење своје оцене са пријатељима на повезаним друштвеним мрежама; опција за сертификоване критичаре; управљање листом филмова које корисник жели да гледа помоћу функције *Листа за гледање*. Такође, апликација би могла у многоме да помогне и самим глумцима, јер представља једноставан увид у кинематографију и достигнућа глумаца, па би је режисери и продуценти могли користити приликом избора будућих звезда са своје филмове.

10. ЛИТЕРАТУРА

1. Anderson, R., Mullen, T., & Vicarel, D. (2022, 07 27). *Razor syntax reference for ASP.NET Core*. Retrieved 09 04, 2022, from Microsoft technical documentation.
2. Chiarelli, A. (2021, 10 15). *What is .NET? An Overview of the Platform*. Retrieved 9 3, 2022, from Auth0: <https://auth0.com/blog/what-is-dotnet-platform-overview/>
3. Dykstra, T. (2022, 07 01). *Implementing the Repository and Unit of Work Patterns in an ASP.NET MVC Application*. Retrieved 09 06, 2022, from Microsoft technical documentation: <https://docs.microsoft.com/en-us/aspnet/mvc/overview/older-versions/getting-started-with-ef-5-using-mvc-4/implementing-the-repository-and-unit-of-work-patterns-in-an-asp-net-mvc-application>
4. Freeman, A. (2022). *Pro ASP.NET Core MVC* (9th ed.). New York, New York, United States: Apress.
5. Freeman, A. (2018). *Pro Entity Framework Core 2 for ASP.NET Core MVC*. New York, New York, United States: Apress.
6. Lock, A. (2021). *ASP.NET Core in Action*. Shelter Island, New York, United States: Manning.
7. Murugan, M. (2021, 4 24). *Repository Pattern in ASP.NET Core – Ultimate Guide*. Retrieved 09 03, 2022, from Code with Mukesh: <https://codewithmukesh.com/blog/repository-pattern-in-aspnet-core/#:~:text=A%20Repository%20pattern%20is%20a,to%20store%20or%20retrieve%20data>.
8. Nixon, R. (2014). *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*. (4th, Ed.) Sebastopol, California, United States: O'Reilly Media.
9. Price, M. J. (2019). *C# 8.0 and .NET Core 3.0* (4th ed.). Birmingham, United Kingdom: Packt Publishing.

10. Smith, S. (2022, 06 27). *Overview of ASP.NET Core MVC*. Retrieved 09 01, 2022, from Microsoft technical documentation: <https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-6.0>
11. Svyryd, A., Alankuş, E., Lambson, B., & Warren, G. (2022, 11 05). *Data Seeding*. Retrieved 09 07, 2022, from Microsoft technical documentation: <https://docs.microsoft.com/en-us/ef/core/modeling/data-seeding>
12. Taher, R. (2019). *Hands-On Object-Oriented Programming with C#*. Birmingham, United Kingdom: Packt Publishing.
13. Vickers, A., Cincura, J., & Rojansky, S. (2022, 11 05). *Database Providers*. Retrieved 09 08, 2022, from Microsoft technical documentation: <https://docs.microsoft.com/en-us/ef/core/providers/?tabs=dotnet-core-cli>
14. Vickers, A., Rojansky, S., & Anderson, R. (2021, 05 25). *Entity Framework Core*. Retrieved 9 8, 2022, from Microsoft technical documentation: <https://docs.microsoft.com/en-us/ef/core/>
15. Vidović, A. (2017, 04 04). *Uvod u MVC dizajn patern*. Retrieved 09 05, 2022, from startit: <https://startit.rs/dizajn-paterni-mvc/>
16. Zakas, N. C. (2014). *The Principles of Object-Oriented JavaScript*. San Francisco, California, United States: No Starch Press.