

**УНИВЕРЗИТЕТ У БЕОГРАДУ  
ФАКУЛТЕТ ОРГАНИЗАЦИОНИХ НАУКА**

**ЗАВРШНИ РАД  
Развој софтверског система за управљање  
уређајима у паметној кући употребом .NET  
технологија**

**Ментор**

**Проф. др Саша Лазаревић**

**Редовни професор**

**Студент:**

**Бојан Стојев 199/15**

**Београд, 2022. године**

## **Развој софтверског система за управљање уређајима у паметној кући употребом .NET технологија**

У складу са научно-технолошким развојем и свеопштом дигитализацијом живота савременог човека, развијају се нове веб апликације са циљем да се свакодневница унапреди, прилагоди презапосленом друштву и што ефикасније искористи сваки минут у стресним данима. Ове апликације се константно надограђују и постају све приступачније, сигурније, лакше и практичније за све профиле људи, чак и оне који се веома слабо разумеју у технологију и коришћење исте. Апликација креирана за потребе овог рада прилагођена је коришћењу у свакодневном животу обичног човека и предвиђена за употребу преко уређаја који већина људи данас увек има у близини - мобилног телефона.

У овом раду описан је софтверски систем у виду апликације креиране употребом .NET технологија. Софтверски систем је развијен помоћу упрошћене Ларманове методе, па је развој апликације испраћен документацијом кроз све фазе развоја: прикупљање захтева, анализа, пројектовање, имплементација и тестирање.

*Кључне речи: софтверски систем, веб апликација, мобилна апликација, Ларманова метода, .NET, ASP.NET, WebAPI, EntityFramework Core, Xamarin*

## САДРЖАЈ

1. Увод.....	10
2. Преглед коришћених технологија.....	11
2.1 C# програмски језик.....	11
2.1.1 Енкапсулација.....	11
2.1.2 Наслеђивање.....	12
2.1.3 Полиморфизам.....	14
2.1.4 Асинхроно програмирање.....	14
2.2 .NET платформа.....	14
2.2.1 ASP.NET.....	15
2.2.2 Entity Framework Core.....	16
2.3 Аутентификација и ауторизација.....	17
2.3.1 Токени.....	17
2.3.2 ASP.NET Identity.....	18
2.4 WebSocket.....	19
2.5 Xamarin.....	20
2.5.1 MVVM патерн.....	22
2.5.2 Везивање података.....	22
2.5.3 Shell.....	22
2.6 Паметни уређаји.....	23
3. Студијски пример.....	24
3.1 Ларманова метода.....	24
3.1.1 Фаза прикупљања захтева.....	24
3.1.2 Фаза анализе.....	24
3.1.3 Фаза пројектовања.....	24
3.1.4 Фаза имплементације.....	24
3.1.4 Фаза тестирања.....	24
4. Кориснички захтеви.....	25
4.1 Вербални опис.....	25
4.2 Опис захтева помоћу модела случајева коришћења.....	25
4.2.1 СК1: Случај коришћења – Пријава на систем.....	28
4.2.2 СК2: Случај коришћења – Промена стања уређаја.....	29
4.2.3 СК3: Случај коришћења – Измена података о налогу.....	30

4.2.4 СК4: Случај коришћења – Унос уређаја.....	31
4.2.5 СК5: Случај коришћења – Измена података о уређају .....	32
4.2.6 СК6: Случај коришћења – Брисање уређаја.....	33
4.2.7 СК7: Случај коришћења – Унос собе .....	34
4.2.8 СК8: Случај коришћења – Брисање собе .....	35
4.2.9 СК9: Случај коришћења – Унос типа уређаја .....	36
4.2.10 СК10: Случај коришћења – Брисање типа уређаја .....	37
4.2.11 СК11: Случај коришћења – Измена података о налогу корисника .....	38
4.2.12 СК12: Случај коришћења – Додавање дозволе за коришћење уређаја	39
4.2.13 СК13: Случај коришћења – Брисање дозволе за коришћење уређаја ..	41
5. Фаза анализе .....	43
5.1 Понашање софтверског система – системски дијаграм секвенци .....	43
5.1.1 ДС1: Дијаграм секвенци случаја коришћења – Пријава на систем.....	43
5.1.2 ДС2: Дијаграм секвенци случаја коришћења – Промена стања уређаја.	45
5.1.3 ДС3: Дијаграм секвенци случаја коришћења – Измена података о налогу .....	47
5.1.4 ДС4: Дијаграм секвенци случаја коришћења – Унос уређаја .....	49
5.1.5 ДС5: Дијаграм секвенци случаја коришћења – Измена података о уређају .....	50
5.1.6 ДС6: Дијаграм секвенци случаја коришћења – Брисање уређаја.....	53
5.1.7 ДС7: Дијаграм секвенци случаја коришћења – Унос собе.....	56
5.1.8 ДС8: Дијаграм секвенци случаја коришћења – Брисање собе.....	57
5.1.9 ДС9: Дијаграм секвенци случаја коришћења – Унос типа уређаја .....	59
5.1.10 ДС10: Дијаграм секвенци случаја коришћења – Брисање типа уређаја	60
5.1.11 ДС11: Дијаграм секвенци случаја коришћења – Измена података о налогу корисника .....	62
5.1.12 ДС12: Дијаграм секвенци случаја коришћења – Додавање дозволе за коришћење уређаја .....	65
5.1.13 ДС13: Дијаграм секвенци случаја коришћења – Брисање дозволе за коришћење уређаја .....	68
5.2 Понашање софтверског система – Дефинисање уговора системским операцијама .....	71
5.3 Структура софтверског система – Концептуални модел .....	76
6. Фаза пројектовања .....	78

6.1 Архитектура софтверског система .....	78
6.2 Пројектовање корисничког интерфејса .....	79
6.3 Пројектовање апликационе логике .....	80
6.3.1 Пројектовање контролера апликационе логике .....	80
6.3.2 Пословна логика .....	115
6.3.3 Пројектовање базе података .....	125
6.4 Коначан изглед архитектуре софтверског система .....	130
7. Фаза имплементације .....	131
7.1 Структура софтверског система .....	131
7.2 Имплементација апликационе логике .....	133
7.2.1 Комуникација са клијентом .....	134
7.2.2 Пословна логика .....	138
7.2.3 Складиште података .....	146
7.3 Имплементација презентационог слоја .....	148
8. Фаза тестирања .....	182
9. Закључак .....	183
10. Литература .....	184

## Списак слика

Слика 1 Структура JWT-а .....	17
Слика 2 Принцип функционисања Хамагін платформе .....	20
Слика 3 Архитектура Хамагін апликација .....	21
Слика 4 ESP8266 Микрочип .....	23
Слика 5 Модел случајева коришћења - Станар .....	26
Слика 6 Модел случајева коришћења - Власник .....	27
Слика 7 Модел случајева коришћења - Администратор .....	27
Слика 8 Пријава на систем .....	43
Слика 9 Неуспешна пријава на систем .....	44
Слика 10 Промена стања уређаја .....	45
Слика 11 Грешка приликом враћања уређаја .....	45
Слика 12 Грешка са дозволом за приступ уређају .....	46
Слика 13 Грешка при промени стања .....	46
Слика 14 Измена података о налогу .....	47
Слика 15 Грешка при враћању података о налогу .....	47
Слика 16 Грешка при чувању података о налогу .....	48
Слика 17 Унос уређаја .....	49

Слика 18	Грешка при уносу уређаја .....	49
Слика 19	Измена података о уређају .....	50
Слика 20	Грешка при враћању уређаја .....	51
Слика 21	Грешка при измени података о уређају .....	51
Слика 22	Брисање уређаја .....	53
Слика 23	Грешка при враћању уређаја .....	54
Слика 24	Грешка при брисању уређаја .....	54
Слика 25	Унос собе .....	56
Слика 26	Грешка при уносу собе .....	56
Слика 27	Брисање собе .....	57
Слика 28	Грешка при брисању собе .....	57
Слика 29	Додавање типа уређаја .....	59
Слика 30	Грешка при додавању типа уређаја .....	59
Слика 31	Брисање типа уређаја .....	60
Слика 32	Грешка при брисању типа уређаја .....	60
Слика 33	Измена података налога о кориснику .....	62
Слика 34	Грешка при враћању налога .....	63
Слика 35	Грешка при измени података о налогу .....	63
Слика 36	Додавање дозволе за коришћење уређаја .....	65
Слика 37	Грешка при враћању корисника .....	66
Слика 38	Грешка при додавању дозволе .....	66
Слика 39	Брисање дозволе за коришћење уређаја .....	68
Слика 40	Грешка при враћању корисника .....	69
Слика 41	Грешка при брисању дозволе .....	69
Слика 42	Концептуални модел .....	76
Слика 43	Концептуални модел и системске операције .....	77
Слика 44	Тронивојска архитектура софтверског система .....	78
Слика 45	Архитектура клијентске апликација .....	79
Слика 46	Детаљна комуникација клијента са сервером .....	79
Слика 47	Приказ апликационе логике серверске апликације .....	80
Слика 48	Страница за пријаву на систем .....	81
Слика 49	Почетна страница .....	82
Слика 50	Грешка при пријави .....	83
Слика 51	Почетна страница .....	84
Слика 52	Страница за контролу ролетни .....	84
Слика 53	Измењено стање уређаја .....	85
Слика 54	Грешка при враћању уређаја .....	86
Слика 55	Грешка са дозволом за приступ уређају .....	86
Слика 56	Грешка при промени стања .....	86
Слика 57	Страница за измену података о налогу .....	87
Слика 58	Страница са информацијама о налогу .....	87
Слика 59	Успешно измењени подаци о налогу .....	88
Слика 60	Грешка приликом враћања налога .....	89
Слика 61	Грешка приликом измене података о налогу .....	89

Слика 62	Страница за унос уређаја .....	90
Слика 63	Грешка приликом уноса уређаја.....	91
Слика 64	Успешан унос уређаја .....	91
Слика 65	Страница за управљање уређајима .....	92
Слика 66	Подаци о уређају су измењени.....	93
Слика 67	Грешка при враћању уређаја.....	94
Слика 68	Грешка при измени података о уређају .....	94
Слика 69	Страница за управљање уређајима .....	95
Слика 70	Успешно брисање уређаја .....	96
Слика 71	Грешка при враћању уређаја.....	97
Слика 72	Грешка при брисању уређаја.....	97
Слика 73	Страница за управљање уређајима .....	98
Слика 74	Прозор за унос имена нове собе.....	98
Слика 75	Успешан унос собе.....	99
Слика 76	Грешка при уносу собе.....	99
Слика 77	Прозор за брисање собе.....	100
Слика 78	Успешно брисање собе.....	101
Слика 79	Грешка при брисању собе .....	101
Слика 80	Прозор за унос типа уређаја.....	102
Слика 81	Успешан унос типа уређаја.....	103
Слика 82	Грешка при уносу типа.....	103
Слика 83	Прозор за брисање типа уређаја .....	104
Слика 84	Успешно брисање типа уређаја .....	105
Слика 85	Грешка при брисању типа.....	105
Слика 86	Страница за управљање налозима .....	106
Слика 87	Успешна измена података о налогу .....	107
Слика 88	Грешка при враћању налога.....	108
Слика 89	Грешка при измени података о налогу .....	108
Слика 90	Страница за управљање корисницима.....	109
Слика 91	Прозор за одабир уређаја.....	110
Слика 92	Успешан унос дозволе .....	110
Слика 93	Грешка при враћању налога.....	111
Слика 94	Грешка при додавању дозволе .....	111
Слика 95	Страница за управљање налозима .....	112
Слика 96	Страница за управљање подацима о налогу .....	113
Слика 97	Успешно брисање дозволе .....	113
Слика 98	Грешка при враћању налога .....	114
Слика 99	Грешка при брисању дозволе.....	114
Слика 100	ДС Пријава корисника .....	115
Слика 101	ДС Измена стања уређаја.....	116
Слика 102	ДС Врати податке о налогу.....	116
Слика 103	ДС Измена корисника.....	117
Слика 104	ДС Додавање уређаја .....	117
Слика 105	ДС Врати све уређаје.....	118

Слика 106 ДС Врати уређај .....	118
Слика 107 ДС Измена уређаја .....	119
Слика 108 ДС Брисање уређаја .....	119
Слика 109 ДС Врати све собе.....	120
Слика 110 ДС Додавање собе .....	120
Слика 111 ДС Брисање собе .....	121
Слика 112 ДС Врати све типове уређаја .....	121
Слика 113 ДС Додавање типа уређаја.....	122
Слика 114 ДС Брисање типа уређаја.....	122
Слика 115 ДС Врати све кориснике .....	123
Слика 116 ДС Врати корисника .....	123
Слика 117 ДС Додавање приступа уређају .....	124
Слика 118 Брисање приступа уређају.....	124
Слика 119 Коначна архитектура.....	130
Слика 120 Табела AspNetRoles.....	146
Слика 121 Табела AspNetUserRoles .....	146
Слика 122 Табела AspNetUsers.....	146
Слика 123 Табела DeviceUser .....	146
Слика 124 Табела Room .....	147
Слика 125 Табела DeviceType.....	147
Слика 126 Табела Device.....	147
Слика 127 Страница за пријаву на систем .....	148
Слика 128 Почетна страница.....	149
Слика 129 Грешка при пријави .....	150
Слика 130 Почетна страница.....	151
Слика 131 Страница за контролу ролетни.....	151
Слика 132 Измењено стање уређаја.....	152
Слика 133 Грешка при враћању уређаја.....	153
Слика 134 Грешка са дозволом за приступ уређају.....	153
Слика 135 Грешка при промени стања .....	153
Слика 136 Страница са информацијама о налогу .....	154
Слика 137 Страница за измену података о налогу .....	154
Слика 138 Успешно измењени подаци о налогу .....	155
Слика 139 Грешка приликом враћања налога.....	156
Слика 140 Грешка приликом измене података о налогу .....	156
Слика 141 Страница за унос уређаја .....	157
Слика 142 Успешан унос уређаја .....	158
Слика 143 Грешка приликом уноса уређаја.....	158
Слика 144 Страница за управљање уређајима.....	159
Слика 145 Подаци о уређају су измењени.....	160
Слика 146 Грешка при враћању уређаја.....	161
Слика 147 Грешка при измени података о уређају .....	161
Слика 148 Страница за управљање уређајима.....	162
Слика 149 Успешно брисање уређаја.....	163



Слика 150	Грешка при враћању уређаја.....	164
Слика 151	Грешка при брисању уређаја.....	164
Слика 152	Прозор за унос имена нове собе.....	165
Слика 153	Страница за управљање уређајима.....	165
Слика 154	Успешан унос собе.....	166
Слика 155	Грешка при уносу собе.....	166
Слика 156	Прозор за брисање собе.....	167
Слика 157	Успешно брисање собе.....	168
Слика 158	Грешка при брисању собе.....	168
Слика 159	Прозор за унос типа уређаја.....	169
Слика 160	Успешан унос типа уређаја.....	170
Слика 161	Грешка при уносу типа.....	170
Слика 162	Прозор за брисање типа уређаја.....	171
Слика 163	Успешно брисање типа уређаја.....	172
Слика 164	Грешка при брисању типа.....	172
Слика 165	Страница за управљање налозима.....	173
Слика 166	Успешна измена података о налогу.....	174
Слика 167	Грешка при враћању налога.....	175
Слика 168	Грешка при измени података о налогу.....	175
Слика 169	Страница за управљање корисницима.....	176
Слика 170	Успешан унос дозволе.....	177
Слика 171	Прозор за одабир уређаја.....	177
Слика 172	Грешка при враћању налога.....	178
Слика 173	Грешка при додавању дозволе.....	178
Слика 174	Страница за управљање налозима.....	179
Слика 175	Успешно брисање дозволе.....	180
Слика 176	Страница за управљање подацима о налогу.....	180
Слика 177	Грешка при враћању налога.....	181
Слика 178	Грешка при брисању дозволе.....	181

## Списак табела

Табела 1	- Device.....	127
Табела 2	- DeviceType.....	127
Табела 4	DeviceUser.....	128
Табела 3	AspNetUser.....	128
Табела 5	AspNetUserRole.....	129
Табела 6	Room.....	129
Табела 7	AspNetRoles.....	129

## 1. Увод

Убрзан начин живота модерног доба ствара све већу поребу за коришћењем веб апликација у свакодневном животу. Људи су условљени динамиком рада, позицијом на којој се налазе или пак физичком ограниченошћу, због чега све више теже ка коришћењу система тзв. „памених кућа“; ови системи нам олакшавају и осавременењују свакодневницу помажући нам у задацима који на први поглед могу изгледати просто, али у горе наведеним ситуацијама, па чак и у тренуцима пуке лењости, су веома корисне. Ове апликације повезују целу нашу кућу са мобилним телефонима, тако да лако можемо имати приступ и увид у функционисање електричних уређаја, што на дуже стазе може имати и практичне и економске бенефите (пр. мања потрошња струје када заборавимо да угасимо светло или аутоматизовање укључивања бојлера у време „јефтине струје“).

Серверска страна апликације реализована је као *ASP.NET WebAPI* апликација, док је клијентска апликација креирана коришћењем *Xamarin* платформе, која представља .NET платформу за развој мобилних апликација које се могу извршавати на различитим оперативним системима мобилних телефона.

Циљ овог рада је креирање софтверског система који би чланови једног домаћинства користили за управљање паменим уређајима у том домаћинству. Овај систем би омогућио свим члановима домаћинства креирање налога, увид у расположиве уређаје, њихове тренутне вредности и стања, као и могућност промене стања ИоТ уређаја постављених у том домаћинству. Такође, власник система може ограничити приступ одређеним уређајима, тако да корисници са својих налога могу управљати само одобреним деловима домаћинства, што олакшава контролу самог система. Софтверски систем описан у даљем раду развијен је коришћењем Ларманове методе развоја софтвера. Приликом имплементације пројекта коришћен је лични стан у коме су били постављени употребљавани уређаји.

Друго поглавље овог рада односи се на теоријски опис коришћених технологија. Треће поглавље описује Ларманову методу која је коришћена за у читавом процесу развоја овог софтверског система. У оквиру овог рада идентификовано је 13 случајева коришћења и за њих се у фази анализе дефинише пословна логика тј. одговарајућа структура и понашање софтверског система. У следећој фази, фази пројектовања, описана је физичка структура и понашање (архитектура) софтверског система. У оквиру фазе имплементације, дат је приказ софтверског система имплементације помоћу програмског језика C# и .NET платформе. У оквиру слоја приступа подацима коришћен је Entity Framework Core заједно са SQLite базом података. На самом крају дати су закључак и коришћена литература.

## 2. Преглед коришћених технологија

Током развоја овог пројекта примарно су коришћене .NET технологије. У овом поглављу биће детаљно описан сваки аспект коришћених технологија.

### 2.1 C# програмски језик

C# представља објектно-оријентисан програмски језик који омогућава програмерима да направе различите типове сигурних и робусних апликација које раде у .NET-у. Корене има у C програмском језику и његове основе су одмах познате програмерима који користе програмске језике C, C++, Java и JavaScript. Током даљег развоја, C# је додао нове функције које подржавају нове праксе дизајна софтвера и нова радна оптерећења[2].

Многе су предности коришћења C# језика. Garbage collection вам аутоматски ослобађа меморију која је заузета од стране недоступних, неискоришћених ресурса. Nullable типови штите од могућности грешке код променљивих које не показују на алоциране објекте. Руковање изузецима пружа структуриран и проширив приступ откривању грешака и опоравку. Технике функционалног програмирања су подржане од стране лямбда израза. Синтакса Language Integrated Query-а (LINQ) омогућава рад са подацима из било ког извора. Синтакса за израду дистрибуираних система је омогућена помоћу језичке подршке за асинхроне операције[2].

Програмски језик C# поседује јединствен систем типова где сви типови се наслеђују од једног основног типа објекта при чему деле скуп заједничких операција. Вредности било ког типа се на доследан начин складиштити и транспортовати. Кориснички дефинисани референтни типови и типови вредности су такође подржани у C# програмском језику, као и динамичка алокација објеката, генеричке методе и типови. [2]

C# такође обезбеђује итераторе, који имплементаторима класа колекције омогућавају да дефинишу прилагођена понашања за клијентски код.

C# истиче верзионисање да би се на тај начин осигурало да се програми и библиотеке на компатибилан начин могу развијати током времена. Верзионисање кода је утицало да елементи дизајна укључују virtual и override модификаторе, правила са overload методе као и подршку за експлицитне декларације чланова интерфејса. [2]

#### 2.1.1 Енкапсулација

Енкапсулација је принцип објектно-оријентисаног програмирања који означава концепт да класа или структура одређују колико су њихови чланови приступачни

коду изван класе или структуре. Променљиве и методе које нису намењене за употребу ван класе могу бити скривене да би се ограничила могућност грешака. [3]

Чланови одређеног типа укључују сва поља, методе, константе, атрибуте и догађаје дефинисане за тај тип. У класи или структури могу бити дефинисани [3]:

- Поља
- Константе
- Атрибути
- Методе
- Конструктори
- Догађаји
- Финализатори
- Индексери
- Оператери
- Угњеждени типови

Доступност кода се може ограничити тако да само жељени клијентски код има приступ. Доступност се ограничава путем модификатора приступа [3]:

- public
- protected
- internal
- protected internal
- private
- private protected

Подразумевана доступност је private.

### 2.1.2 Наслеђивање

Концепт наслеђивања је подржан само од стране класе, а не и структуре. Класа која је изведена од друге класе, која се назива базна класа, садржи све public, protected, и internal чланове базне класе, осим конструктора и финализатора. Класе дефинисане као abstract морају садржати једну или више метода које немају имплементацију при чему се оне не могу директно инстанцирати, али служе као основне за друге класе које обезбеђују имплементацију. Класе дефинисане као sealed није могуће наследити. [4]

Класе и структуре могу имплементирати више различитих интерфејса при чему имплементација значи да тип мора да имплементира све методе дефинисане у интерфејсу.

C# подржава генеричке типове где се класе и структуре могу дефинисати са једним или више параметара типа. На пример, класа List<T> је дефинисана једним параметром типа. [4]

Класе се такође могу дефинисати као статичке (static) при чему статичка класа може садржати само статичке чланове и не може се инстанцирати са новом кључном речи. Постоји само једна копија класе која се учитава у меморију, а члановима се приступа преко имена класе. Међутим, класе и структуре могу садржати статичке чланове. [4]

Класе и структуре могу бити угнеждене унутар друге класе или структуре.

Путем парцијалних типова (partial types) могуће је један део класе, структуре или методе дефинисати у једној датотеци кода, а други део у другој.

Инстанцирати и иницијализовати објекте класе, структуре или колекције је могуће урадити путем додељивања вредности њиховим својствима.

Омогућено је проширити класу без креирања изведене класе помоћу extension метода при чему се методе могу позвати као да припадају оригиналном типу.

У класи и структури могуће је имплицитно дефинисати локалне променљиве где се компајлеру налаже да одреди тип променљиве у време компајлирања. То се може урадити тако што се променљива дефинише помоћу кључне речи var.

Дефиниција класе или структуре се посматра као нацрт који дефинише шта одређени тип може да уради. У својој основи, објекат представља блок меморије који је додељен и конфигуриран према том нацрту. Објекат или инстанца се чувају у именованој променљивој или у низу или у колекцији. Програм може направити много инстанци исте класе. Ове променљиве код користи за позивање метода или приступ атрибутима објекта. [4]

Класе су референтни типови где променљива објекта класе садржи референцу на адресу објекта на heap-у. Ако је друга променљива истог типа додељене првој, онда обе показују на објекат на тој адреси[4].

Оператор new се користи за креирање инстанце класе.

Наслеђивање омогућава креирање нове класе која се назива изведена класа која поново користи, проширује или мења понашање дефинисано у другој класи која се назива основна класа. Ограничење је што изведена класа може имати само једну директну основну класу, али наслеђивање је прелазно па ако, на пример, класа Ц је изведена из класе Б која је изведена из класе А, онда класа Ц наслеђује чланове класе Б и класе А. Изведена класа представља специјализацију основне класе где, на пример, ако имате основну класу Особа, можете имати једну изведену класу Запослени и другу Студент. Пошто су и запослени и студенти особе, изведене класе представљају специјализације основне. Изведена класа

имплицитно добија све чланове основне, осим конструктора и финализаторе при чему може користити код основне класе без да га поново имплементира[4].

### 2.1.3 Полиморфизам

Концепт полиморфизма има два аспекта [5]:

У време извођења, објекти изведене класе могу се третирали као објекти основне класе на местима као што су параметри метода и колекције или низови.

Основне класе могу да дефинишу и имплементирају виртуелне (virtual) методе, а изведене класе могу да их замене, што значи да дају сопствену дефиницију и имплементацију.

### 2.1.4 Асинхроно програмирање

Асинхроно програмирање је једна од битних функционалности која је у C# програмском језику имплементирана на нивоу самог језика и омогућава једноставно писање асинхроног кода. Прати тзв. Task-based Asynchronous Pattern (TAP).

Асинхроно програмирање омогућава да дугачке операције као што су приступање бази података, читавање ресурса са удаљених сервера или читање и писање у датотеке, не блокирају извршавање апликације до њиховог завршетка, већ се коришћењем кључних речи `async` и `await` постиже да апликација када дође до такве операције, она започне са извршавањем у позадини, а кориснички интерфејс остаје респонзиван. [6]

## 2.2 .NET платформа

.NET је бесплатна open source платформа за израду различитих врста апликација при чему је је изграђена на runtime окружењу високих перформанси.

.NET 6 је последња верзија .NET-а избачена новембра 2021. године која представља побољшану верзију и уводи нове функције у платформу. Пре .NET 6 постојале су .NET Framework и .NET Core који су били одвојени производи. .NET Framework је био оригинални .NET који није био open source, док је .NET Core верзија која јесте open source и ради на свим платформама, Windows, Linux и Mac. Са .NET 6 и свим новијим, постоји само једна верзија .NET-а која се зове само .NET. [7]

.NET 6 представља обједињену развојну платформу који програмерима пружа могућност да праве апликације за cloud, web, desktop, gaming, mobile, IoT и AI. Наведене подплатформе деле библиотеке основних класа, инфраструктуру која укључује језик и компајлере и API-је.

.NET апликације се могу развијати у Visual Studio-у, Visual Studio Code-у и CDI алатима.

Такође, .NET 6 је бржа верзија .NET-а при чему се могу користити и последње верзије C# као што је C# 10 која нуди многа побољшања и једноставнија је за коришћење.

### 2.2.1 ASP.NET

ASP.NET проширује .NET платформу различитим алатима и библиотекама за израду web апликација.

Појединости које ASP.NET додаје .NET платформи су [10]:

- Основни оквир за обраду веб захтева у C# или F#-у
- Синтакса шаблона веб страница, познатија као Razor, која се користи за израду динамичких веб страница коришћењем C#-а
- Библиотеке за уобичајене веб обрасце као што је Model View Controller
- Систем за аутентификацију који укључује базу података, библиотеке и шаблонске стране за управљање пријављивањем
- Екстензије које пружају истицање синтаксе, довршавање кода и разне друге функционалности

ASP.NET омогућава развијање за macOS, Windows, Linux и Docker при чему је могуће направити различите типове веб апликација што укључује MVC веб апликације, REST API-је, микросервисе и апликације за комуникацију у реалном времену коришћењем SignalR-а.

ASP.NET функционише брже од било ког популарног оквира у независним benchmark-овима при чему подржава стандардне индустријске протоколе за аутентификацију. [10]

Апликације су заштићене од унакрсног скриптовања (XSS) и фалсификовања захтева (CSRF) помоћу уграђених функција.

Крајње тачке (endpoints) аутоматски серијализију ваше класе у правилно фоматиран JSON без употребе посебне конфигурације при чему се серијализација може прилагодити за крајње тачке (endpoints) које имају јединствене захтеве.

Имате могућност дефинисања рута у свом коду помоћу атрибута. Подаци из путање захтева, низа упита и тела захтева се аутоматски везују за параметре метода.

Коришћењем ASP.NET-а можете аутоматски генерисати тест сертификат и лако га увести да би сте покренули локални HTTPS да бисте покренули и отклонили грешке у својим апликацијама које треба да буду заштићене.

## 2.2.2 Entity Framework Core

*Entity Framework Core* представља најпопуларнији и најнапреднији објектно-релациони мапер алат у оквиру *.NET* платформе. Он пружа *.NET* програмерима да управљају базом података коришћењем *.NET* објеката, без потребе за конкретним кодом за директан приступ бази података. [11]

Омогућава мапирање објекта ентитета у табеле базе података, аутоматски креира и одржава шему базе података на основу промена у ентитетским класама или генерисање ентитета на основу постојеће базе података, генерисање *sql* кода за упите, као и управљање трансакцијама приликом упита.

Табеле се креирају на основу инстанци *DbSet* класе у оквиру контекста базе података. Самом контекстз базе података може се у оквиру сервиса апликације приступити помоћу *Dependency Injection*-а и на једноставан начин вршити упите над базом.

Као што је већ поменуто, постоје два приступа којима се *Entity Framework Core* имплементира [11]:

- *Code-First* – у овом приступу прво се креирају класе ентитети, са свим атрибутима дефинисаним у њима, а затим се на основу њих генерише база података.
- *Database-First* – у овом приступу потребно је креирати базу података, а затим се на основу ње генеришу класе ентитети.

Одабри приступа искључиво зависи од типа апликације која се развија.

У току развоја апликације често долази до измена у ентитетским класама апликације, што мора да буде испраћено и у бази података. *Entity Framework Core* омогућава једноставно решење овог проблема коришћењем миграција. Миграције представљају инкременталне промене на бази, што је веома корисно у тимовима програмера, јер се све миграције чувају на једном месту и уколико неко има на локалном нивоу старију верзију базе података, једноставно је може ажурирати, где ће се све новије миграције применити на ту локалну базу података.

Упити ка бази врше се коришћењем *LINQ* библиотеке. Стандардни *LINQ* упити подсећају на *SQL* упите, али постоји и могућност коришћења лямбда израза, што значајно повећава читљивост кода. Као резултат упита добијају се објекти ентитетских класа. Након сваке измене над добијеним објектима, потребно је сачувати промене над контекстом базе података, што ће потврдити трансакцију.

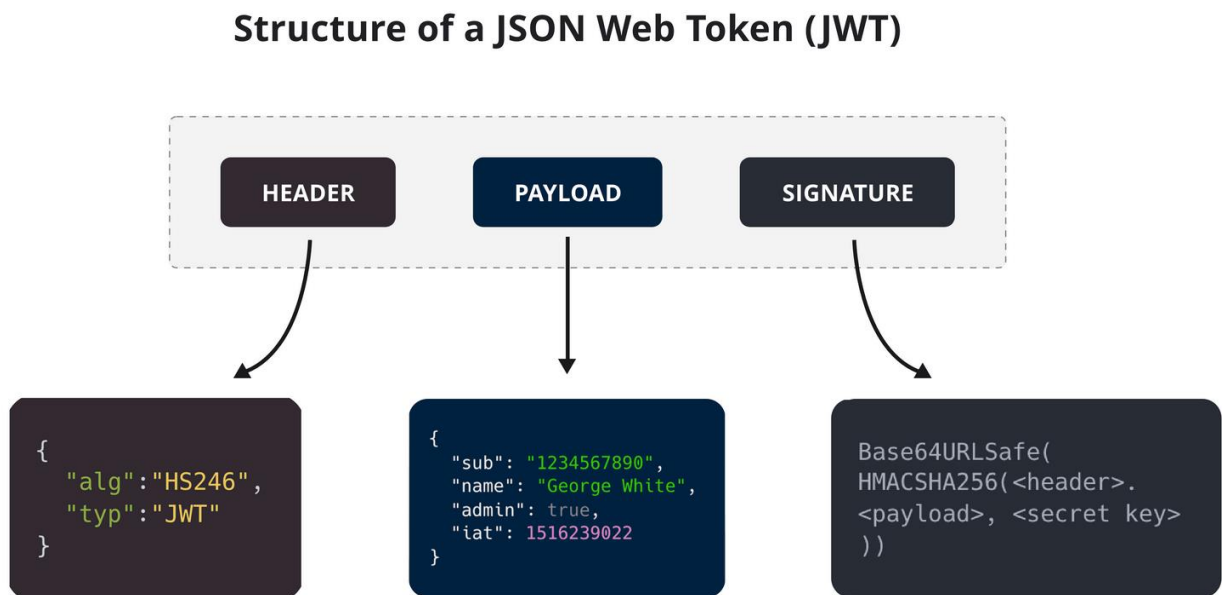


## 2.3 Аутентификација и ауторизација

У оквиру овог пројекта коришћени су *JWT* токени за ауторизацију захтева корисника, у комбинацији са *ASP.NET Identity* системом у оквиру *.NET* платформе.

### 2.3.1 Токени

*JSON Web Token* представља једноставан и компактан интернет стандард за ауторизацију приступа серверским функционалностима. Они садрже основне информације о аутентификованом кориснику и налазе се у заглављу сваког *HTTP* захтева који клијент шаље серверу. [12]



Слика 1 Структура JWT-а

Сервер може верификовати и веровати информацијама које се налазе у токenu, због тога што токен садржи дигитални потпис, а затим је и енкриптован тајним кључем који је познат само серверској апликацији. За дигитални потпис користи се *HMAC* алгоритам који користи криптографске *Hash* функције, чиме се гарантује и интегритет и аутентичност података у токenu.

Токенима се омогућава ауторизација без чувања стања о повезаном клијенту (*Stateless Authorization*), где се приликом сваког приступа крајњим тачкама контролера серверске апликације проверава да ли *HTTP* захтев у свом хедеру за ауторизацију садржи валидан токен. [13]

### 2.3.2 ASP.NET Identity

*ASP.NET Identity* је систем у оквиру .NET платформе који служи за аутентификацију и ауторизацију. Омогућава једноставно креирање прилагођених функционалности за пријаву и одјаву са серверске апликације, као и једноставно прилагођавање профила карактеристика корисника ради олакшања прилагођавања података о пријављеним корисницима. [14]

Омогућава како традиционалан начун пријаве систему путем корисничког имена и лозинке, тако и модерније начине пријављивања путем ексерних пружаоца услуга пријављивања, као што су Facebook, Twitter, Google, Microsoft, итд.

По подразумеваном шаблону, *ASP.NET Identity* креира табелу у бази са подацима о кориснику, али се наслеђивањем класе корисника може лако проширити количина информација која се чува о сваком кориснику.

*ASP.NET Identity* такође управља креирањем рола, те је веома лако омогућити ауторизацију у контролеру на основу рола пошиљаоца захтева.

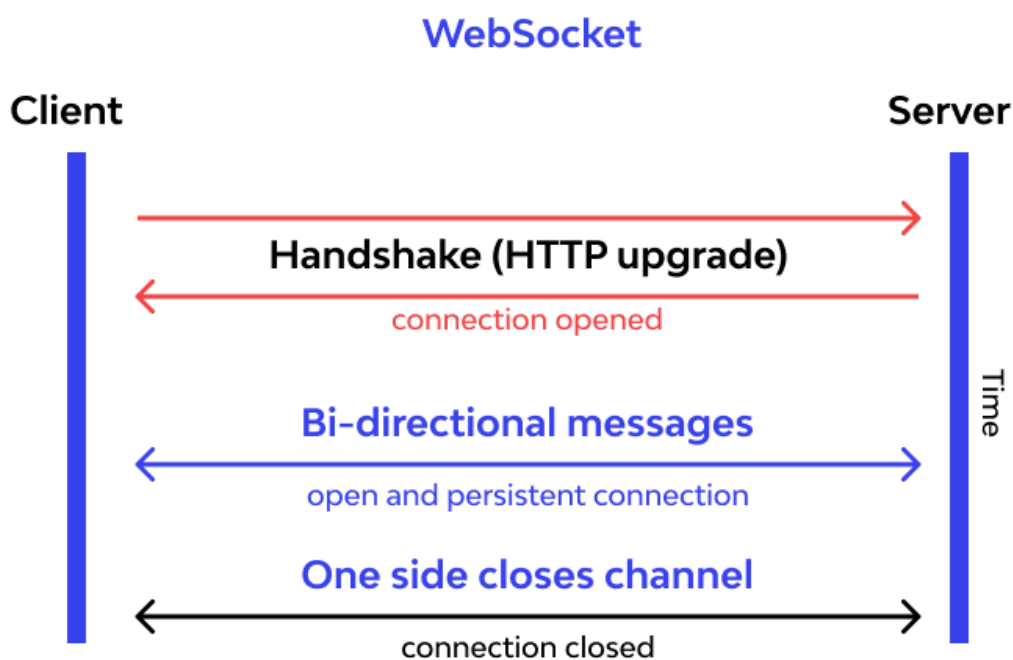
Неке од додатних функционалности које *ASP.NET Identity* пружа су [15]:

- Двофакторска аутентификација корисника, која уводи додатни ниво заштите апликацији. Двофакторска аутентификација поред корисничког имена и лозинке, при пријави на систем захтева и једнократну лозинку, која се обично генерише путем апликације на мобилном телефону корисника, путем смс поруке или слањем пин-а на емаил корисника. Корисник мора проћи кроз двофакторску аутентификацију при сваком пријављивању на систем, при чему је једнократна лозинка, односно пин, увек насумично генерисан и важиће само одређени краћи временски период.
- Закључавање налога, које се може аутоматски активирати уколико корисник унесе погрешну лозинку или једнократну лозинку више од дозвољеног броја пута, али је такође могуће администраторима да ручно закључају налог корисника уколико сматрају да је то потребно.
- Потврђивање налога, које се може обавити кликом на линк који корисник добија на своју емаил адресу. Ова опција може да смањи број лажних налога који се креирају. Ако је ова опција омогућена, корисник неће имати приступ налогу пре него што га потврди.
- Контрола параметрара неопходне комплексности лозинке
- Ресетовање лозинке
- Брисање налога корисника

## 2.4 WebSocket

*WebSocket* је комуникациони протокол који се разликује од обичних *HTTP* захтева у томе што пружа *full-duplex* комуникацију путем *TCP* протокола, што омогућава комуникацију у реалном времену између клијента и сервера. Алтернатива су тзв. *Polling*-у. [15]

Животни циклус *WebSocket*-а започиње обичним *HTTP* захтевом којем се поставља хедер за надоградњу везе (*Upgrade Header*) чиме се у току *Handshake*-а мења обрада захтева са серверске стране из обичног *HTTP* захтева у *WebSocket* захтев.



*WebSocket*-и користе уједињену *TCP* конекцију и да би се конекција прекинула неопходно је да један од учесника пошаље захтев за прекид конекције, док се то не деси конекција остаје отворена, за разлику од стандардних *HTTP* захтева који након послатог одговора аутоматски затварају конекцију и захтевају посебне конекције за сваки захтев.

## 2.5 Xamarin

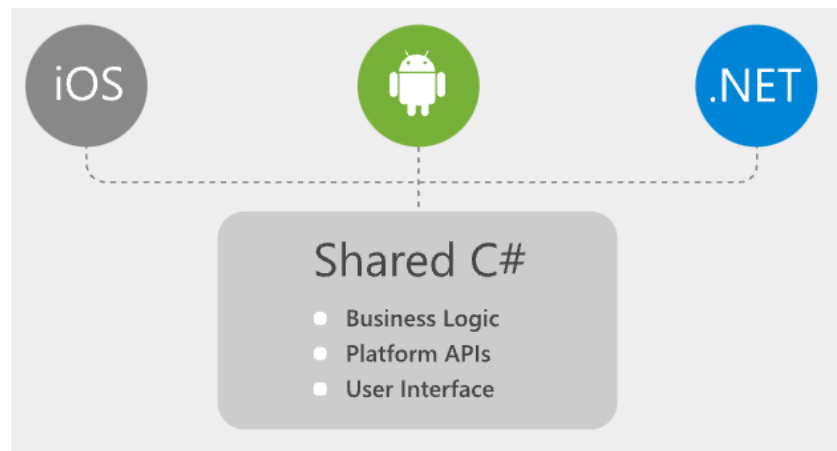
Xamarin је платформа која такође проширује .NET платформу многим алатима и библиотекама за израду апликација на Android, iOS, итд.

Неке од појединости које Xamarin додаје .NET платформи су [8]:

- Основни оквир за приступ изворним функцијама
- Прошириви Markup Language XAML за израду мобилних динамичких апликација коришћењем C# библиотека за уобичајене обрасце као што је Model View ViewModel (MVVM)
- Специфичне библиотеке за платформу које укључују приступ API-јима са Google-а, Facebook-а и других.
- Екстензије које пружају истицање синтаксе, довршавање кода и разне друге функционалности

Xamarin и .NET имају широку употребу кроз различите индустрије за израду изворних мобилних апликација. Неке од апликација које су изграђене коришћењем ових технологија су UPS, BBC Good Food, Novarum DX, Seeing AI и многе друге.

Помоћу Xamarin-а, могуће је написати целокупну апликацију помоћу C# програмског језика, од back end кода као што је приступ подацима и пословна логика до изворног приступа API-ју. Као што смо већ поменули Xamarin проширује .NET па се може користити велики екосистем пакета и библиотека, али се могу креирати и сопствене библиотеке.



Слика 2 Принцип функционисања Xamarin платформе

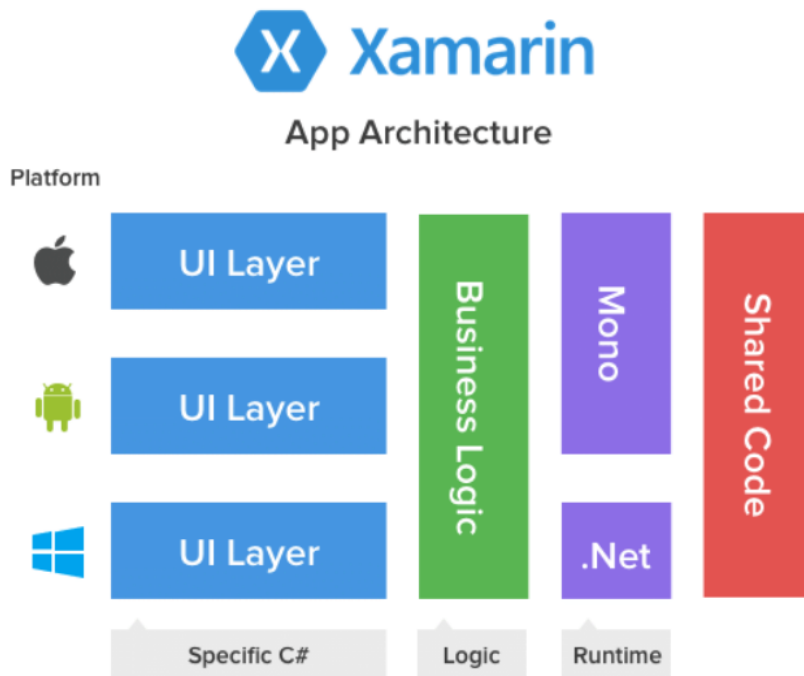
Апликације изграђене помоћу Xamarin-а користе изворне функционалности платформи за које се развијају. Могуће је извршити ahead-of-time (AOT) компилацију да би се смањило време покретања, побољшале перформансе и повећало дељење меморије.

Хамарин апликације имају приступ спектру функционалности које укључују и специфичности везане за платформе као што су ARKit, CoreML, Fingerprint, Bluetooth, NFC итд.

Хамарин.Forms облик Хамарин-а који представља open source cross-platform UI оквир за израду iOS, Android, и Windows апликација са .NET-ом из једне, дељене базе кода.

Можете користити Хамарин.Forms уграђене странице, контроле и layout-е да бисте дизајнирали мобилну апликацију из једног API-ја при чему имате могућност израде подкласа било које контроле да бисте прилагодили њихово понашање или дефинисали контроле, layout-е или странице. Хамарин.Forms укључује и cross-platform навигацију, API-је за анимацију, центар за размену порука итд.

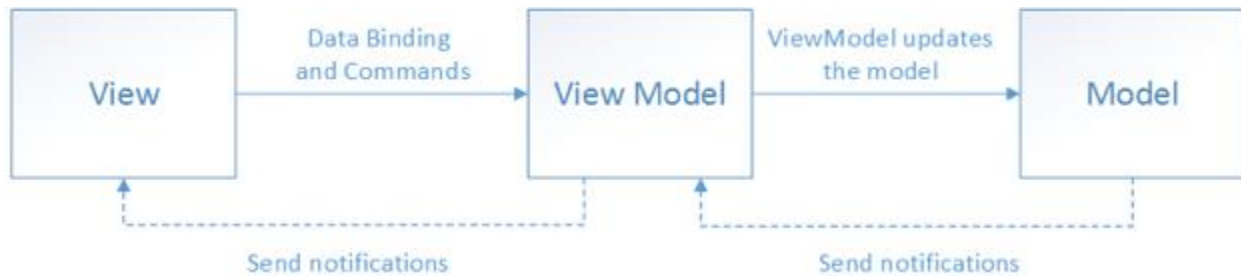
Хамарин омогућава уграђену подршку за MVVM образац што укључује data binding података што даље олакшава праћење дизајна шаблона.



Слика 3 Архитектура Хамарин апликација

### 2.5.1 MVVM патерн

При развоју Xamarin апликације, пожељно је коришћење MVVM патерна, којим се кориснички интерфејс креиран у XAML-у одваја од логике, а која се смешта у тзв. ViewModel-е. Применом овог патерна постиже се чистији код и јасна граница између пословне логике и презентационе логике у мобилној апликацији. Такође се омогућава лакше тестирање, одржавање, даљи развој, као и поновно коришћење кода (*Code Reusability*). [16]



### 2.5.2 Везивање података

Везивање података (*Data Binding*) је важан сегмент развоја Xamarin апликације, јер аутоматизује задатак освежавања вредности на страници уколико дође до промене променљивих у ViewModel-у, као и промену променљивих у ViewModel-у када корисник унесе измену у некој визуелној компоненти на страници.

Као резултат, знатно се смањује комплексност и количина кода потребна да испуни ове функционалности, а сам код постаје чистији и читљивији.

Омогућене су и додатне функционалности као што су прилагођени конвертери, одабир мода везивања, везивање за другачији *Binding Context* од остатка странице и слично. [18]

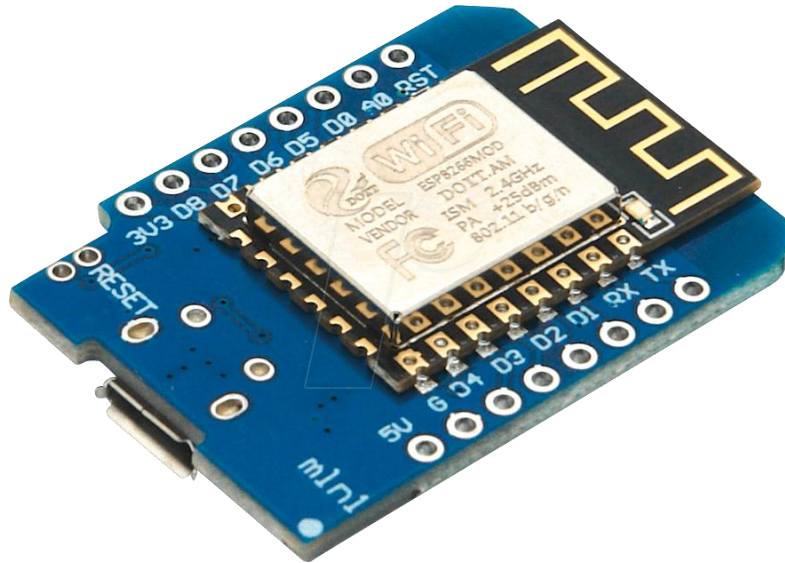
### 2.5.3 Shell

Shell представља алтернативу стандардном начину навигације кроз апликацију. Нуди навигацију базирану на URL навигацији, дефинисање хијерархије апликације на једном месту, једноставно креирање траке са картицама (*TabBar*) или бочног менија (*FlyoutItem*), као и једноставно прослеђивање вредности приликом навигације између страница. [17]

## 2.6 Паметни уређаји

У току израде пројекта, сви коришћени уређаји су били базирани на микрочипу ознаке ESP8266.

Овај микрочип функционише налик на Arduino, најбитнија разлика је што овај чип на себи има уграђен WIFI модул, тако да се врло једноставно може повезати на локалну мрежу у стану и слати или примати стандардне HTTP захтеве. Има изузетно ниску цену, веома мале димензије, и малу потрошњу. [19]



Слика 4 ESP8266 Микрочип

Налик на Arduino, програмира се у Arduino развојном окружењу, користећи C++ програмски језик. Сваки фајл (*Sketch*) мора да садржи две методе, `setup()` и `loop()`, где се прва позива само при покретању уређаја, а друга се непрестано извршава.

## 3. Студијски пример

Развој овог софтверског система одвија се у фазама које се везују за Ларманову методу, која је детаљније описана у наставку овог поглавља.

### 3.1 Ларманова метода

Ова метода посматра развој софтверског система кроз 5 фаза, а те фазе су:

1. Прикупљање захтева
2. Фаза анализе
3. Фаза пројектовања
4. Фаза имплементације
5. Фаза тестирања

#### 3.1.1 Фаза прикупљања захтева

Ова фаза подразумева дефинисање особина и функционалности које систем мора да поседује да би задовољио потребе крајњих корисника. Обавља се у сарадњи пројектанта и експерта из области на коју се пројекат односи. Захтеви корисника дефинишу се моделом случајева коришћења.

#### 3.1.2 Фаза анализе

Фаза анализе заснива се на случајевима коришћења добијених из претходне фазе. Она треба да опише структуру и понашање система, а то се постиже креирањем дијаграма секвенци, уговора као и концептуалног модела.

#### 3.1.3 Фаза пројектовања

У фази пројектовања потребно је дефинисати архитектуру целокупног софтверског система. Архитектура је тронивојска и састоји се од нивоа корисничког интерфејса, нивоа апликационе логике, као и нивоа складишта података.

#### 3.1.4 Фаза имплементације

Ова фаза односи се на реализацију самог система у неком од објектно оријентисаних језика, што је у овом случају језик C#.

#### 3.1.4 Фаза тестирања

Представља фазу у оквиру које се жељеном методом проверава функционалност и стабилност развијеног система.



## 4. Кориснички захтеви

### 4.1 Вербални опис

Потребно је креирати софтверски систем који ће користити чланови једне куће или стана како би управљали паметним уређајима у том стану.

Циљ је креирати такав софтверски систем који ће омогућити станарима креирање налога, увид у расположиве уређаје, њихове тренутне вредности и стања, као и могућност промене стања стања ИоТ уређаја који су у том стану постављени.

Постоје 3 групе корисника, власник, администратори, као и обични корисници, односно станари.

Обичан корисник има приступ само уређајима за које је добио експлицитну дозволу да их користи, такође има могућност измене података о свом налогу.

Администратори неограничен приступ свим уређајима у стану, као и приступ управљању уређајима, тј. додавању, измени и брисању уређаја.

Власник система има додатно приступ управљању свим корисницима, измени података на њиховим налозима, као и додавању и одузимању дозвола за коришћење уређаја обичним корисницима.

Сви корисници се морају пријавити на свој налог на систему, који карактеришу корисничко име и лозинка, како би могао да врши измене на самом систему. Након пријаве, корисници добијају токен за ауторизацију који се мора садржати у сваком њиховом захтеву ка серверу.

### 4.2 Опис захтева помоћу модела случајева коришћења

Власнику треба омогућити следеће функционалности:

- Пријава на систем
- Промена стања уређаја
- Измена података о налогу
- Унос уређаја
- Измена података о уређају
- Брисање уређаја
- Унос собе
- Брисање собе
- Унос типа уређаја
- Брисање типа уређаја
- Измена података о налогу корисника
- Додавање дозволе за коришћење уређаја
- Брисање дозволе за коришћење уређаја

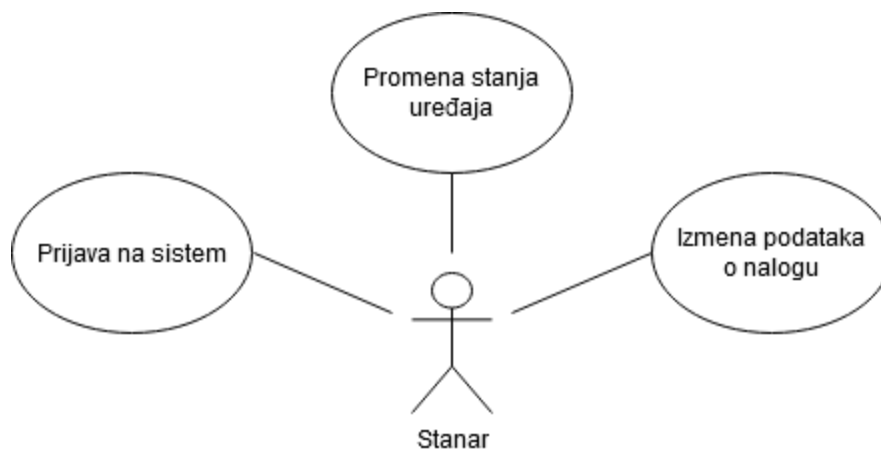
Администратору треба омогућити следеће функционалности:

- Пријава на систем
- Промена стања уређаја
- Измена података о налогу
- Унос уређаја
- Измена података о уређају
- Брисање уређаја
- Унос собе
- Брисање собе
- Унос типа уређаја
- Брисање типа уређаја

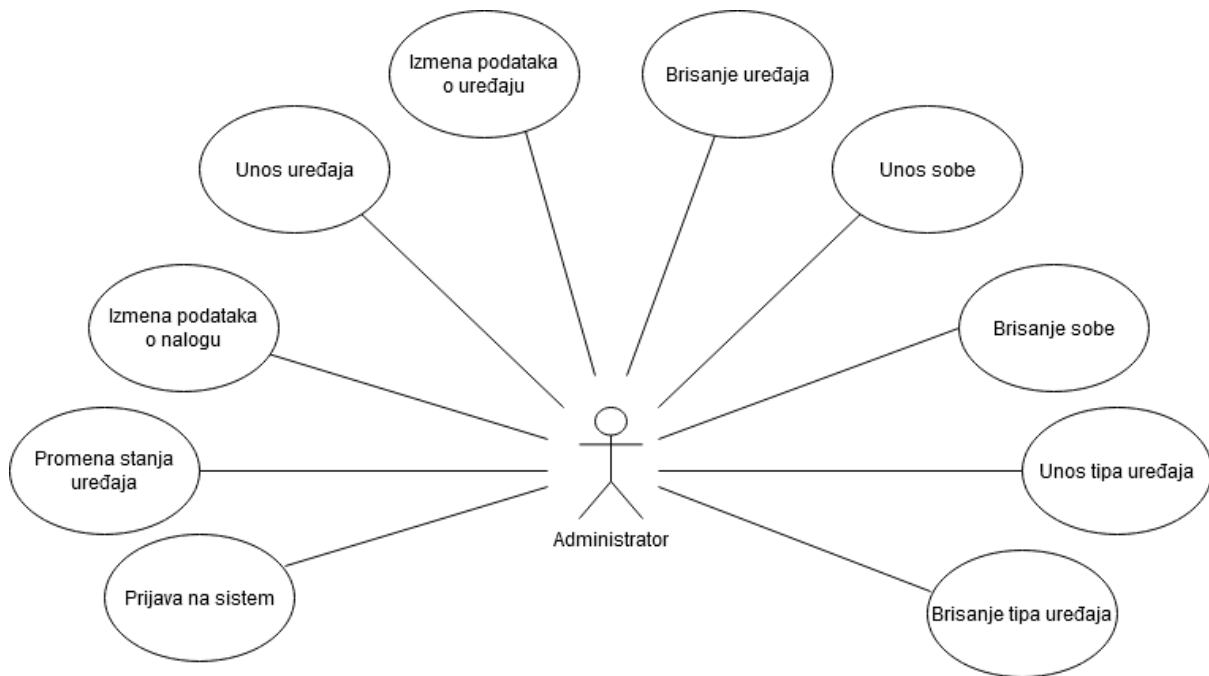
Станарима треба омогућити функционалности:

- Пријава на систем
- Промена стања уређаја
- Измена података о налогу

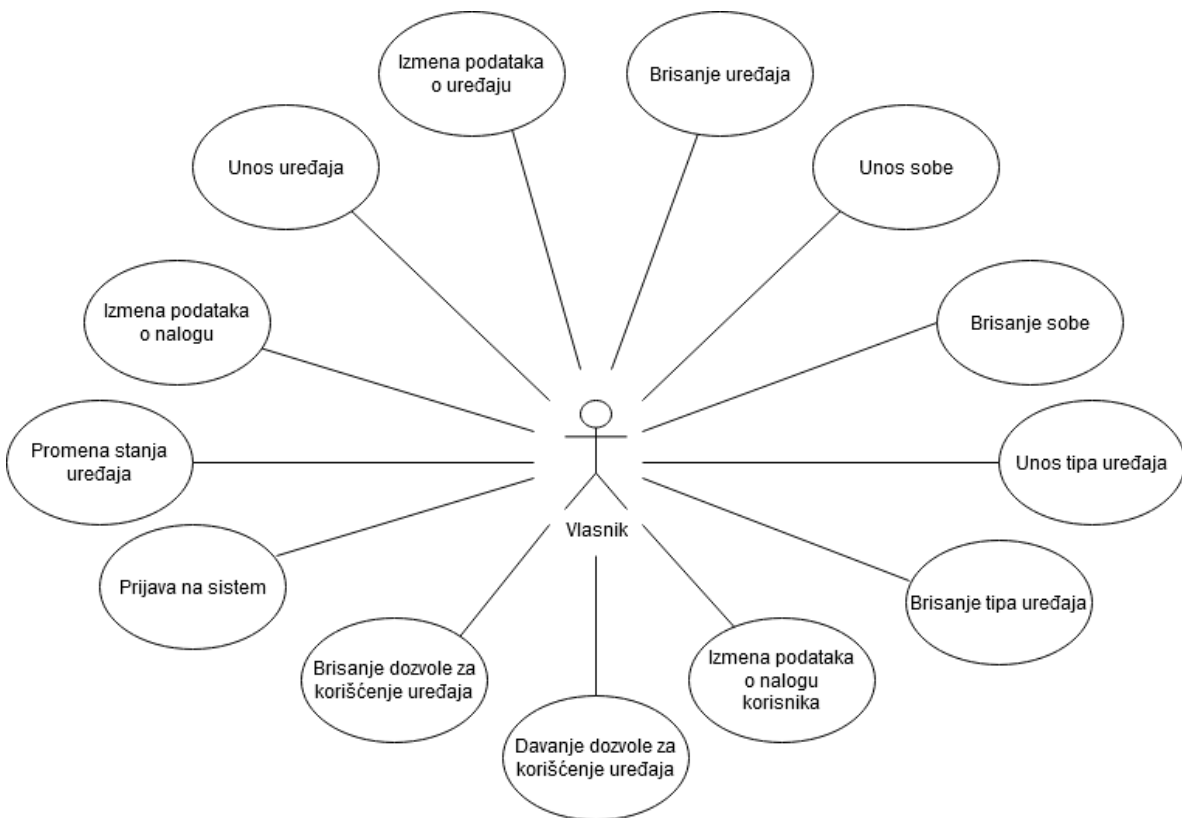
Случајеви коришћења за власника, администратора и станара су представљени следећим сликама.



Слика 5 Модел случајева коришћења - Станар



Слика 7 Модел случајева коришћења - Администратор



Слика 6 Модел случајева коришћења - Власник

#### 4.2.1 СК1: Случај коришћења – Пријава на систем

##### **Назив СК**

Пријава на систем

##### **Актер СК**

Корисник

##### **Учесници СК**

Корисник и систем (програм)

##### **Предуслов**

Систем је укључен и приказује страницу за пријаву на систем.

##### **Основни сценарио**

1. Корисник уноси корисничко име и лозинку (АПУСО)
2. Корисник контролише да ли је коректно унео корисничко име и лозинку (АНСО)
3. Корисник позива систем да се пријави (АПСО)
4. Систем проверава податке о кориснику (СО)
5. Систем кориснику враћа податке о налогу и токен за ауторизацију и приказује му почетну страну (ИА)

##### **Алтернативна сценарија**

- 5.1. Уколико систем не може да нађе корисника са унетим корисничким именом и лозинком, приказује кориснику поруку: „Погрешно корисничко име или лозинка!“ (ИА)

## 4.2.2 СК2: Случај коришћења – Промена стања уређаја

### **Назив СК**

Промена стања уређаја

### **Актер СК**

Корисник

### **Учесници СК**

Корисник и систем (програм)

### **Предуслов**

Систем је укључен и корисник је пријављен на своја налог. Систем приказује почетну страницу или страницу специфичног уређаја.

### **Основни сценарио**

1. Корисник уноси ново стање уређаја (АПУСО)
2. Корисник позива систем да изврши промену стања уређаја (АПСО)
3. Систем проналази уређај, проверава да ли корисник има приступ уређају, а затим врши измену стања уређаја (СО)
4. Систем враћа кориснику информације о уређају са измењеним стањем и приказује ново стање уређаја (ИА)

### **Алтернативна сценарија**

- 4.1. Уколико систем не може да нађе уређај, приказује кориснику поруку: „Уређај није пронађен!“. Прекида се извршење сценарија (ИА)
- 4.2. Уколико корисник нема приступ уређају, приказује поруку: „Немате приступ овом уређају!“. Прекида се извршење сценарија (ИА)
- 4.3. Уколико систем није у могућности да изврши тражену измену, приказује прокук: „Дошло је до грешке приликом измене стања уређаја!“ (ИА)

### 4.2.3 СКЗ: Случај коришћења – Измена података о налогу

#### **Назив СК**

Измена података о налогу

#### **Актер СК**

Корисник

#### **Учесници СК**

Корисник и систем (програм)

#### **Предуслов**

Систем је укључен и корисник је пријављен на своја налог. Систем приказује страницу за рад са подацима о налогу.

#### **Основни сценарио**

1. Корисник позива систем да прикаже податке о његовом налогу (АПСО)
2. Систем тражи податке о налогу корисника (СО)
3. Систем приказује податке о налогу корисника (ИА)
4. Корисник уноси измене података о налогу (АПУСО)
5. Корисник проверава да ли је исправно унео нове податке (АНСО)
6. Корисник позива систем да запамти податке о налогу (АПСО)
7. Систем памти измењене податке о налогу (СО)
8. Систем приказује кориснику поруку: „Подаци о налогу су успешно сачувани.“ (ИА)

#### **Алтернативна сценарија**

- 4.1. Уколико систем не може да пронађе податке о налогу корисника, приказује кориснику поруку: „Подаци о налогу нису пронађени!“. Прекида се извршење сценарија (ИА)
- 8.1. Уколико систем не може да сачува податке о налогу, приказује кориснику поруку: „Није могуће сачувати податке о налогу!“ (ИА)

#### 4.2.4 СК4: Случај коришћења – Унос уређаја

##### **Назив СК**

Унос уређаја

##### **Актер СК**

Корисник (Администратор или Власник)

##### **Учесници СК**

Корисник и систем (програм)

##### **Предуслов**

Систем је укључен и корисник је пријављен на своја налог. Систем приказује страницу за унос новог уређаја.

##### **Основни сценарио**

1. Корисник уноси податке о новом уређају (АПУСО)
2. Корисник проверава да ли је исправно унео податке (АНСО)
3. Корисник позива систем да запамти податке о новом уређају (АПСО)
4. Систем памти податке о уређају (СО)
5. Систем приказује кориснику поруку: „Уређај успешно сачуван.“ (ИА)

##### **Алтернативна сценарија**

- 5.1. Уколико систем не може да сачува податке о уређају, приказује кориснику поруку: „Неуспешно додавање уређаја!“ (ИА)

#### 4.2.5 СК5: Случај коришћења – Измена података о уређају

##### **Назив СК**

Измена података о уређају

##### **Актер СК**

Корисник (Администратор или Власник)

##### **Учесници СК**

Корисник и систем (програм)

##### **Предуслов**

Систем је укључен и корисник је пријављен на своја налог. Систем приказује страницу за управљање уређајума.

##### **Основни сценарио**

1. Корисник позива систем да прикаже листу уређаја (АПСО)
2. Систем проналази све постојеће уређаје (СО)
3. Систем приказује листу свих уређаја (ИА)
4. Корисник одабира уређај чије податке жели да измени (АПУСО)
5. Корисник позива систем да прикаже податке о изабраном уређају (АПСО)
6. Систем учитава податке о изабраном уређају (СО)
7. Систем приказује кориснику податке о изабраном уређају (ИА)
8. Корисник мења податке о уређају (АПУСО)
9. Корисник проверава да ли је добро унео податке (АНСО)
10. Корисник позива систем да запамти унете податке (АПСО)
11. Систем памти измењене податке о уређају (СО)
12. Систем приказује кориснику поруку: „Подаци о уређају су успешно измењени.“ (ИА)

##### **Алтернативна сценарија**

- 7.1. Уколико систем не може да пронађе податке о изабраном уређају, приказује кориснику поруку: „Подаци о уређају нису пронађени!“. Прекида се извршење сценарија (ИА)
- 12.1. Уколико систем не може да сачува измењене податке, приказује кориснику поруку: „Неуспешна измена података о уређају!“ (ИА)



#### 4.2.6 СК6: Случај коришћења – Брисање уређаја

##### **Назив СК**

Брисање уређаја

##### **Актер СК**

Корисник (Администратор или Власник)

##### **Учесници СК**

Корисник и систем (програм)

##### **Предуслов**

Систем је укључен и корисник је пријављен на своја налог. Систем приказује страницу за управљање уређајума.

##### **Основни сценарио**

1. Корисник позива систем да прикаже листу уређаја (АПСО)
2. Систем проналази све постојеће уређаје (СО)
3. Систем приказује листу свих уређаја (ИА)
4. Корисник одабира уређај који жели да обрише (АПУСО)
5. Корисник позива систем да прикаже податке о изабраном уређају (АПСО)
6. Систем учитава податке о изабраном уређају (СО)
7. Систем приказује кориснику податке о изабраном уређају (ИА)
8. Корисник позива систем да обрише изабрани уређај (АПСО)
9. Систем врши брисање уређаја (СО)
10. Систем приказује кориснику поруку: „Уређај успешно обрисан.“ (ИА)

##### **Алтернативна сценарија**

- 7.1. Уколико систем не може да пронађе податке о изабраном уређају, приказује кориснику поруку: „Подаци о уређају нису пронађени!“. Прекида се извршење сценарија (ИА)
- 10.1. Уколико систем не може да обрише уређај, приказује кориснику поруку: „Неуспешно брисање уређаја!“ (ИА)

#### 4.2.7 СК7: Случај коришћења – Унос собе

##### **Назив СК**

Унос собе

##### **Актер СК**

Корисник (Администратор или Власник)

##### **Учесници СК**

Корисник и систем (програм)

##### **Предуслов**

Систем је укључен и корисник је пријављен на своја налог. Систем приказује прозор за унос собе.

##### **Основни сценарио**

1. Корисник уноси назив нове собе (АПУСО)
2. Корисник проверава да ли је добро унео назив (АНСО)
3. Корисник позива систем да запамти податке о новој соби (АПСО)
4. Систем памти податке о соби (СО)
5. Систем приказује кориснику поруку: „Соба успешно унета.“ (ИА)

##### **Алтернативна сценарија**

- 5.1. Уколико систем не може да сачува податке о новој соби, приказује кориснику поруку: „Неуспешан унос собе!“ (ИА)

#### 4.2.8 СК8: Случај коришћења – Брисање собе

##### **Назив СК**

Брисање собе

##### **Актер СК**

Корисник (Администратор или Власник)

##### **Учесници СК**

Корисник и систем (програм)

##### **Предуслов**

Систем је укључен и корисник је пријављен на своја налог. Систем приказује прозор за брисање собе.

##### **Основни сценарио**

1. Корисник позива систем да прикаже листу соба (АПСО)
2. Систем проналази све постојеће собе (СО)
3. Систем приказује листу свих соба (ИА)
4. Корисник одабира собу коју жели да обрише (АПУСО)
5. Корисник позива систем да обрише изабрану собу (АПСО)
6. Систем врши брисање собе (СО)
7. Систем приказује кориснику поруку: „Соба успешно обрисана.“ (ИА)

##### **Алтернативна сценарија**

- 7.1. Уколико систем не може да обрише собу, приказује кориснику поруку: „Неуспешно брисање собе!“ (ИА)

#### 4.2.9 СК9: Случај коришћења – Унос типа уређаја

##### **Назив СК**

Унос типа уређаја

##### **Актер СК**

Корисник (Администратор или Власник)

##### **Учесници СК**

Корисник и систем (програм)

##### **Предуслов**

Систем је укључен и корисник је пријављен на своја налог. Систем приказује прозор за унос типа уређаја.

##### **Основни сценарио**

1. Корисник уноси назив новог типа уређаја (АПУСО)
2. Корисник проверава да ли је добро унео назив (АНСО)
3. Корисник позива систем да запамти податке о новом типу уређаја (АПСО)
4. Систем памти податке о типу уређаја (СО)
5. Систем приказује кориснику поруку: „Тип уређаја успешно унет.“

##### **Алтернативна сценарија**

- 5.1. Уколико систем не може да сачува податке о новом типу уређаја, приказује кориснику поруку: „Неуспешан унос типа уређаја!“ (ИА)

#### 4.2.10 СК10: Случај коришћења – Брисање типа уређаја

##### **Назив СК**

Брисање типа уређаја

##### **Актер СК**

Корисник (Администратор или Власник)

##### **Учесници СК**

Корисник и систем (програм)

##### **Предуслов**

Систем је укључен и корисник је пријављен на своја налог. Систем приказује прозор за брисање типа уређаја.

##### **Основни сценарио**

1. Корисник позива систем да прикаже листу типова уређаја (АПСО)
2. Систем проналази све постојеће типове уређаја (СО)
3. Систем приказује листу свих типова уређаја (ИА)
4. Корисник одабира тип уређаја који жели да обрише (АПУСО)
5. Корисник позива систем да обрише изабрани тип уређаја (АПСО)
6. Систем врши брисање типа уређаја (СО)
7. Систем приказује кориснику поруку: „Тип уређаја успешно обрисан.“ (ИА)

##### **Алтернативна сценарија**

- 7.1. Уколико систем не може да обрише тип уређаја, приказује кориснику поруку: „Неуспешно брисање типа уређаја!“ (ИА)

#### 4.2.11 SK11: Случај коришћења – Измена података о налогу корисника

##### **Назив СК**

Измена података о уређају

##### **Актер СК**

Власник

##### **Учесници СК**

Власник и систем (програм)

##### **Предуслов**

Систем је укључен и власник је пријављен на своја налог. Систем приказује страницу за управљање налозима.

##### **Основни сценарио**

1. Власник позива систем да прикаже листу налога (АПСО)
2. Систем проналази све постојеће налоге (СО)
3. Систем приказује листу свих налога (ИА)
4. Власник одабира налог чије податке жели да измени (АПУСО)
5. Власник позива систем да прикаже податке о изабраном налогу (АПСО)
6. Систем учитава податке о изабраном налогу (СО)
7. Систем приказује власнику податке о изабраном налогу (ИА)
8. Власник мења податке о налогу (АПУСО)
9. Власник проверава да ли је добро унео податке (АНСО)
10. Власник позива систем да запамти унете податке (АПСО)
11. Систем памти измењене податке о налогу (СО)
12. Систем приказује власнику поруку: „Подаци о налогу су успешно измењени.“ (ИА)

##### **Алтернативна сценарија**

- 7.1. Уколико систем не може да пронађе податке о изабраном налогу, приказује власнику поруку: „Подаци о налогу нису пронађени!“. Прекида се извршење сценарија (ИА)
- 12.1. Уколико систем не може да сачува измењене податке, приказује власнику поруку: „Неуспешна измена података о налогу!“ (ИА)

## 4.2.12 SK12: Случај коришћења – Додавање дозволе за коришћење уређаја

### Назив СК

Додавање дозволе за коришћење уређаја

### Актер СК

Власник

### Учесници СК

Власник и систем (програм)

### Предуслов

Систем је укључен и власник је пријављен на своја налог. Систем приказује страницу за управљање налозима.

### Основни сценарио

1. Власник позива систем да прикаже листу налога (АПСО)
2. Систем проналази све постојеће налоге (СО)
3. Систем приказује листу свих налога (ИА)
4. Власник одабира налог чије коме жели да дода дозволу за коришћење уређаја (АПУСО)
5. Власник позива систем да врати податке о изабраном кориснику (АПСО)
6. Систем проналази изабраног корисника (СО)
7. Систем приказује податке о изабраном кориснику (ИА)
8. Власник позива систем да прикаже листу уређаја (АПСО)
9. Систем проналази све постојеће уређаје (СО)
10. Систем приказује листу свих уређаја (ИА)
11. Власник одабира уређај за који даје дозволу коришћења (АПУСО)
12. Власник позива систем да запамти податке о дозволи коришћења изабраног уређаја за претходно изабраног корисника (АПСО)
13. Систем памти дозволу коришћења уређаја (СО)
14. Систем приказује власнику поруку: „Успешно додавање дозволе коришћења уређаја.“ (ИА)

### Алтернативна сценарија

- 7.1. Уколико систем не може да пронађе корисника, приказује власнику поруку: „Подаци о изабраном кориснику нису пронађени!“. Прекида се извршење сценарија (ИА)

14.1. Уколико систем не може да сачува дозволу коришћења, приказује власнику поруку: „Неуспешна додавање дозволе!“ (ИА)



#### 4.2.13 СК13: Случај коришћења – Брисање дозволе за коришћење уређаја

##### **Назив СК**

Брисање дозволе за коришћење уређаја

##### **Актер СК**

Власник

##### **Учесници СК**

Власник и систем (програм)

##### **Предуслов**

Систем је укључен и власник је пријављен на своја налог. Систем приказује страницу за управљање налозима.

##### **Основни сценарио**

1. Власник позива систем да прикаже листу налога (АПСО)
2. Систем проналази све постојеће налоге (СО)
3. Систем приказује листу свих налога (ИА)
4. Власник одабира налог чије коме жели да обрише дозволу за коришћење уређаја (АПУСО)
5. Власник позива систем да врати податке о изабраном кориснику (АПСО)
6. Систем проналази изабраног корисника (СО)
7. Систем приказује податке о изабраном кориснику (ИА)
8. Власник позива систем да прикаже листу уређаја којима изабрани корисник има приступ (АПСО)
9. Систем проналази све постојеће уређаје који одговарају критеријуму (СО)
10. Систем приказује листу свих уређаја који одговарају критеријуму (ИА)
11. Власник одабира уређај за који брише дозволу коришћења (АПУСО)
12. Власник позива систем да обрише дозволу коришћења изабраног уређаја за претходно изабраног корисника (АПСО)
13. Систем брише дозволу коришћења уређаја (СО)
14. Систем приказује власнику поруку: „Успешно брисање дозволе коришћења уређаја.“ (ИА)

##### **Алтернативна сценарија**

- 7.1. Уколико систем не може да пронађе корисника, приказује власнику поруку: „Подаци о изабраном кориснику нису пронађени!“. Прекида се извршење сценарија (ИА)

14.1. Уколико систем не може да обрише дозволу коришћења, приказује власнику поруку: „Неуспешно брисање дозволе!“ (ИА)

## 5. Фаза анализе

Фаза анализе описује пословну логику тј. понашање и структуру софтверског система. Структура софтверског система описује се уз помоћ концептуалног и релационог модела, док се понашање софтверског система може описати преко системских дијаграма секвенци и системских операција.

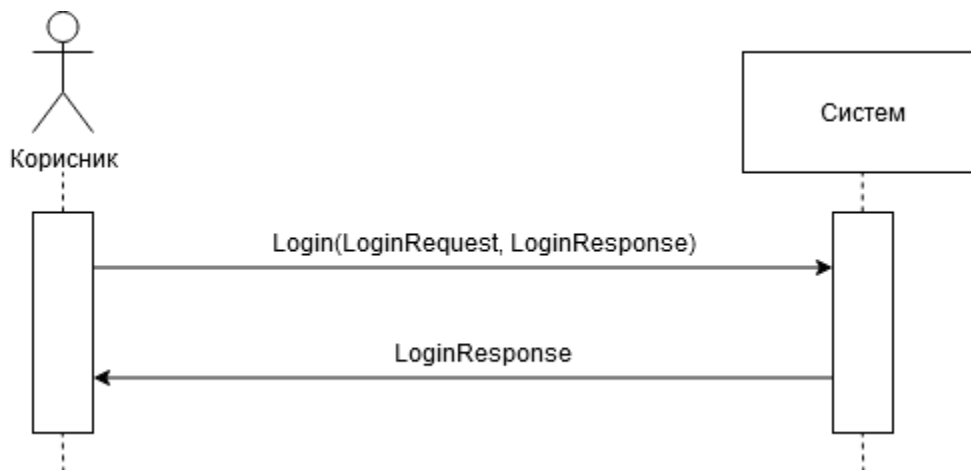
### 5.1 Понашање софтверског система – системски дијаграм секвенци

Ради бољег представљања понашања софтверског система, за сваки сценарио идентификован у оквиру случајева коришћења у фази прикупљања корисничких захтева креира се одговарајући системски дијаграм секвенци. Он успоставља интеракцију између актора и софтверског система приказом догађаја у одређеном редоследу. Актори покрећу догађаје преко посредника у оквиру APSO акција, прималац догађаја га прихвата и позива системску операцију на страни система која се извршава, након чега систем враћа резултат као одговор на догађај.

#### 5.1.1 ДС1: Дијаграм секвенци случаја коришћења – Пријава на систем

##### Основни сценарио СК

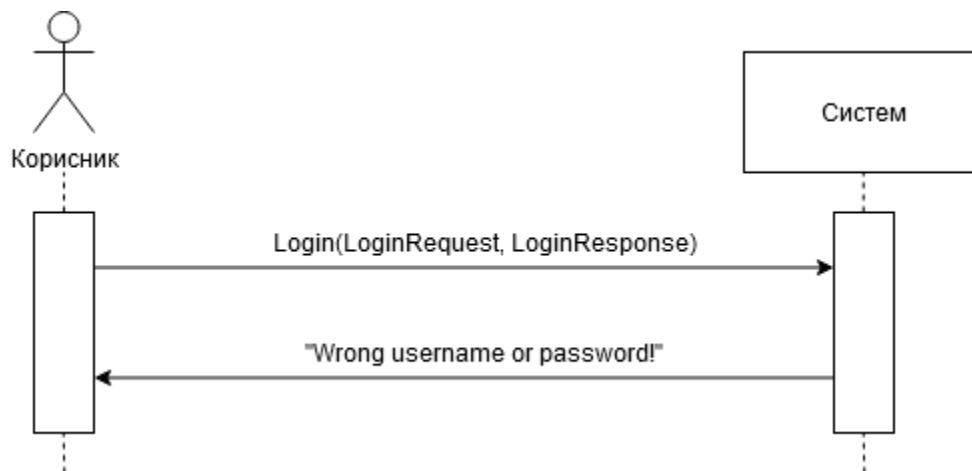
1. Корисник позива систем да се пријави
2. Систем враћа кориснику податке о налогу и токен за ауторизацију и приказује му почетну страну



Слика 8 Пријава на систем

## Алтернативна сценарија СК

- 2.1. Уколико систем не може да нађе корисника са унетим корисничким именом и лозинком, приказује кориснику поруку: „Погрешно корисничко име или лозинка!“ (ИА)



Слика 9 Неуспешна пријава на систем

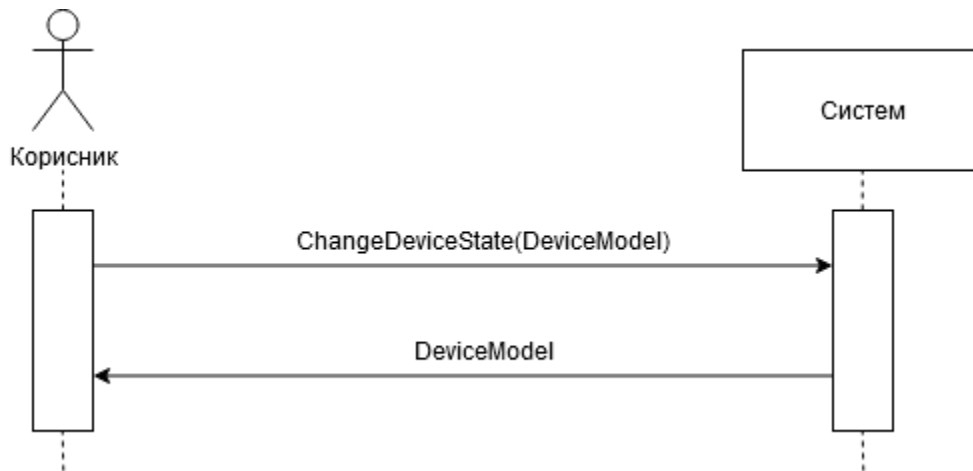
Идентификоване системске операције:

- signal Login(LoginRequest, LoginResult)

## 5.1.2 ДС2: Дијаграм секвенци случаја коришћења – Промена стања уређаја

### Основни сценарио СК

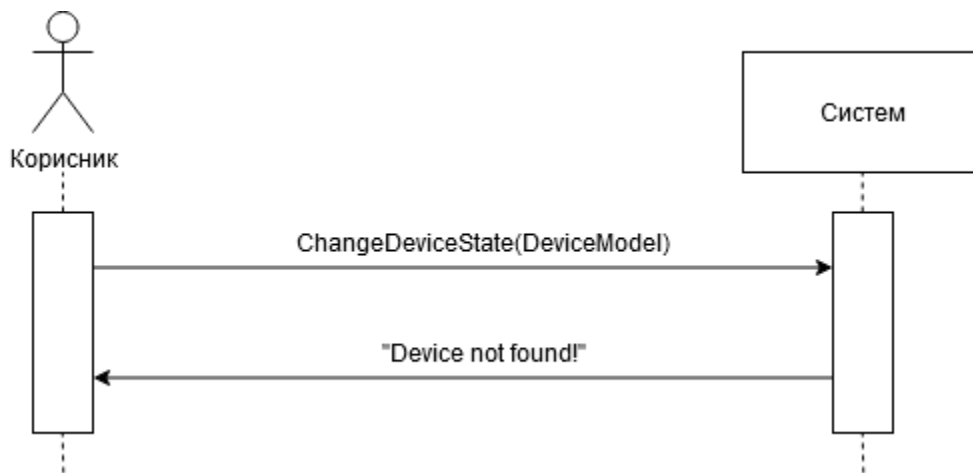
1. Корисник позива систем да изврши промену стања уређаја (АПСО)
2. Систем враћа кориснику информације о уређају са измењеним стањем и приказује ново стање уређаја (ИА)



Слика 10 Промена стања уређаја

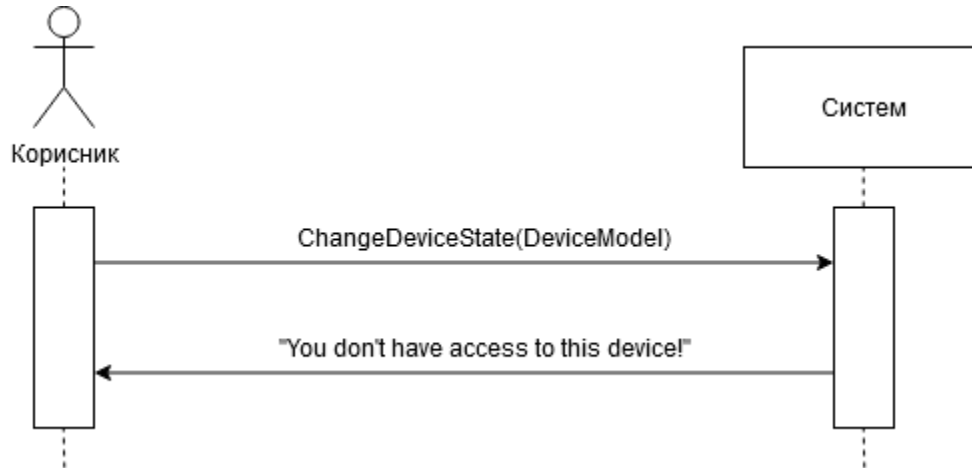
### Алтернативна сценарија

- 2.1. Уколико систем не може да нађе уређај, приказује кориснику поруку: „Уређај није пронађен!“. Прекида се извршење сценарија (ИА)



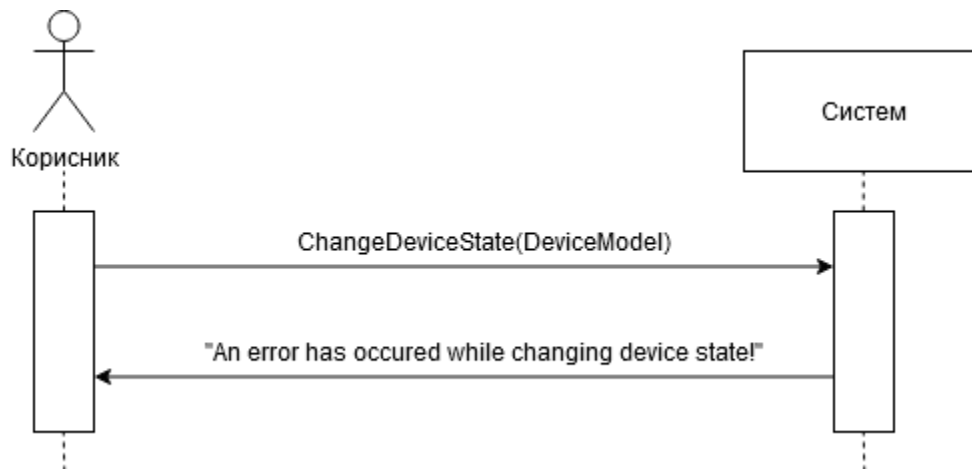
Слика 11 Грешка приликом враћања уређаја

- 2.2. Уколико корисник нема приступ уређају, приказује поруку: „Немате приступ овом уређају!“. Прекида се извршење сценарија (ИА)



Слика 12 Грешка са дозволом за приступ уређају

- 2.3. Уколико систем није у могућности да изврши тражену измену, приказује прокук: „Дошло је до грешке приликом измене стања уређаја!“ (ИА)



Слика 13 Грешка при промени стања

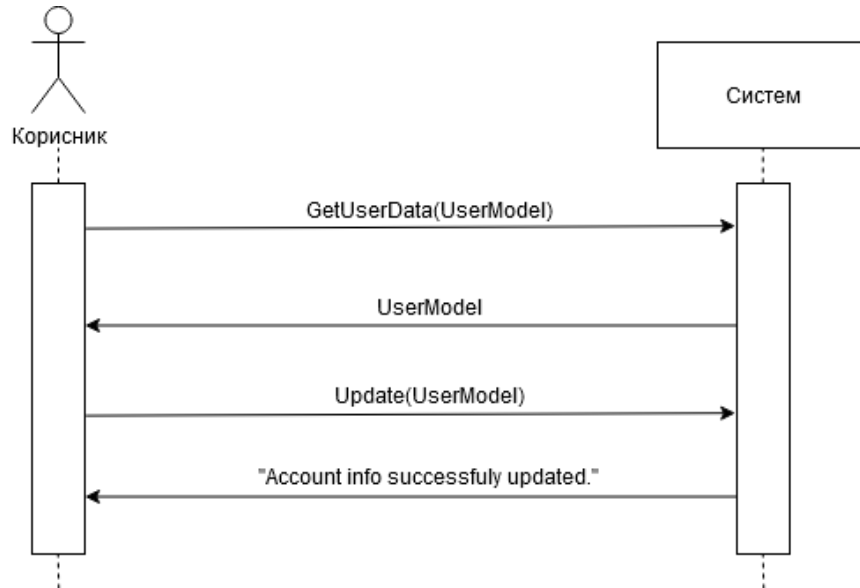
Идентификоване системске операције:

- signal `ChangeDeviceState(DeviceModel)`

### 5.1.3 ДС3: Дијаграм секвенци случаја коришћења – Измена података о налогу

#### Основни сценарио СК

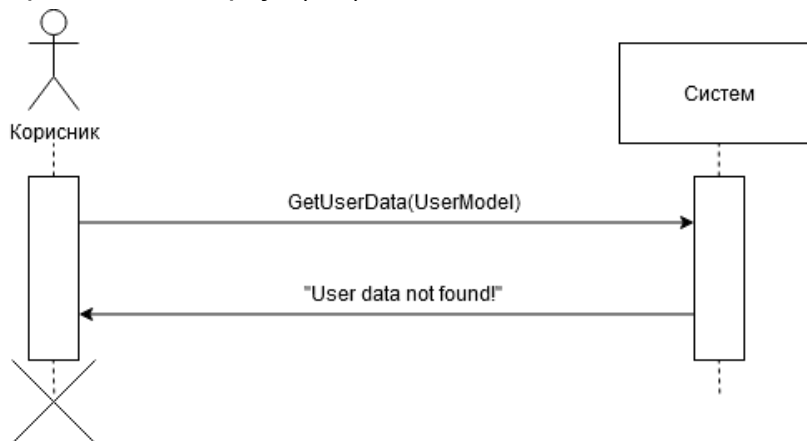
1. Корисник позива систем да прикаже податке о његовом налогу (АПСО)
2. Систем приказује податке о налогу корисника (ИА)
3. Корисник позива систем да запамти податке о налогу (АПСО)
4. Систем приказује кориснику поруку: „Подаци о налогу су успешно сачувани.“ (ИА)



Слика 14 Измена података о налогу

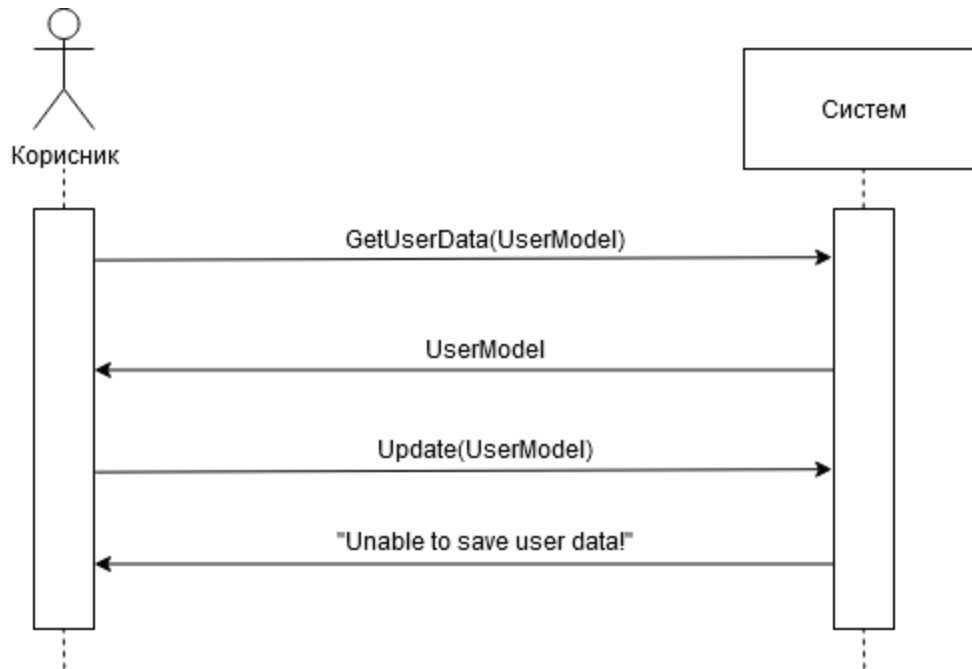
#### Алтернативна сценарија

- 2.1. Уколико систем не може да пронађе податке о налогу корисника, приказује кориснику поруку: „Подаци о налогу нису пронађени!“. Прекида се извршење сценарија (ИА)



Слика 15 Грешка при враћању података о налогу

- 4.1. Уколико систем не може да сачува податке о налогу, приказује кориснику поруку: „Није могуће сачувати податке о налогу!“ (ИА)



Слика 16 Грешка при чувању података о налогу

Идентификоване системске операције:

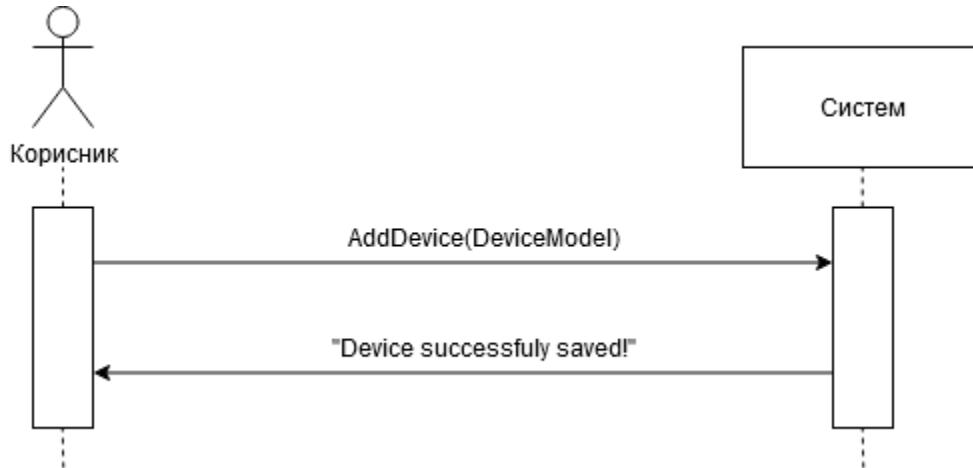
- signal `GetUserData(UserModel)`
- signal `UpdateUser(UserModel)`



#### 5.1.4 ДС4: Дијаграм секвенци случаја коришћења – Унос уређаја

##### Основни сценарио СК

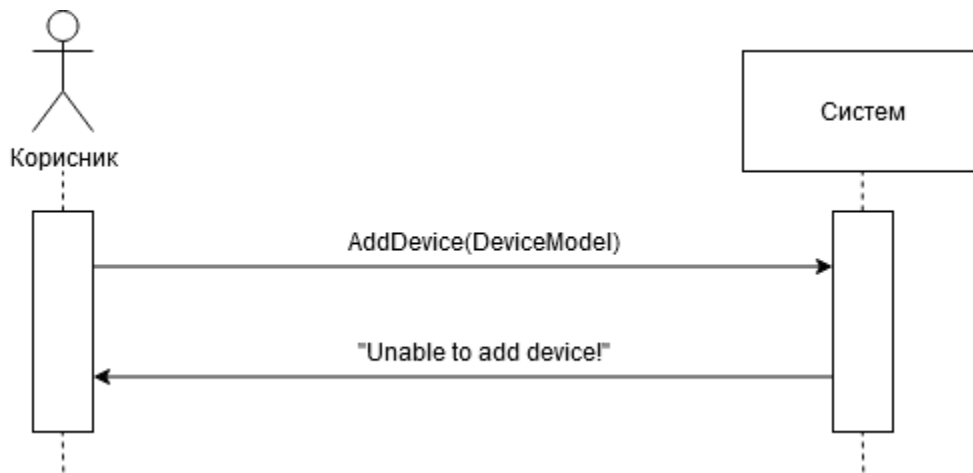
1. Корисник позива систем да запамти податке о новом уређају (АПСО)
2. Систем приказује кориснику поруку: „Уређај успешно сачуван.“ (ИА)



Слика 17 Унос уређаја

##### Алтернативна сценарија

- 2.1. Уколико систем не може да сачува податке о уређају, приказује кориснику поруку: „Неуспешно додавање уређаја!“ (ИА)



Слика 18 Грешка при уносу уређаја

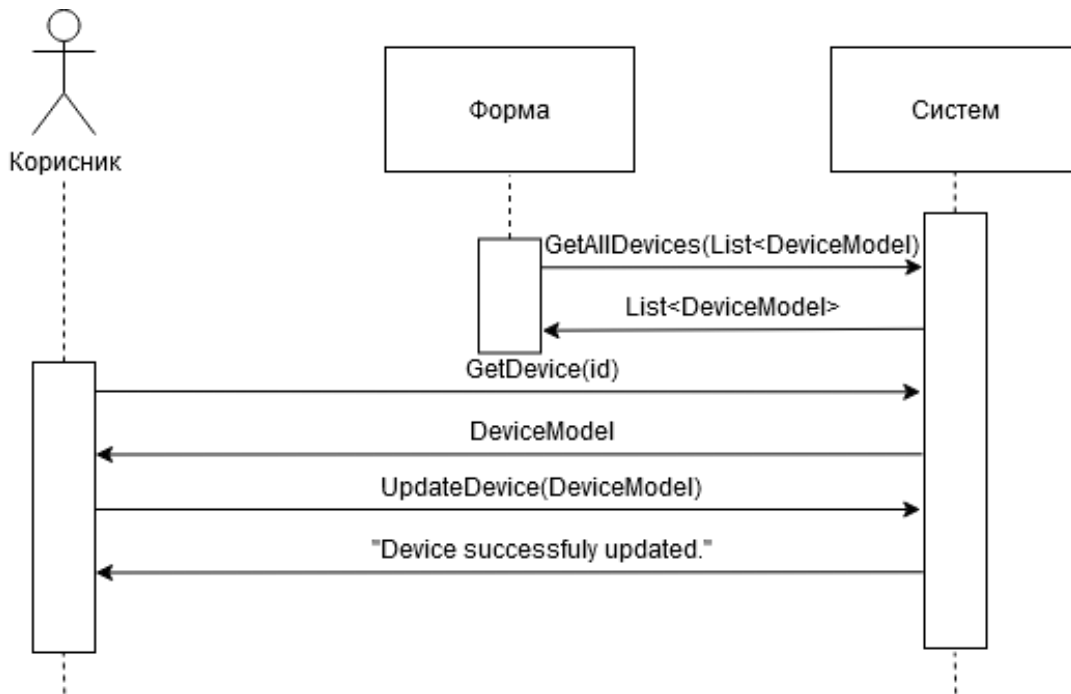
Идентификоване системске операције:

- signal AddDevice(DeviceModel)

## 5.1.5 ДС5: Дијаграм секвенци случаја коришћења – Измена података о уређају

### Основни сценарио СК

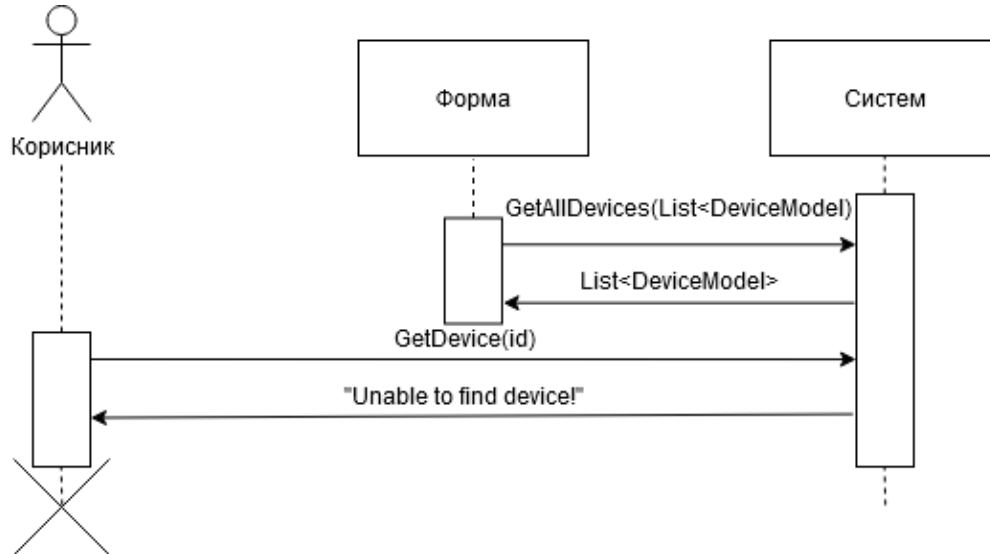
1. Форма позива систем да прикаже листу уређаја (АПСО)
2. Систем враћа форми листу свих уређаја (ИА)
3. Корисник позива систем да прикаже податке о изабраном уређају (АПСО)
4. Систем приказује кориснику податке о изабраном уређају (ИА)
5. Корисник позива систем да запамти унете податке (АПСО)
6. Систем приказује кориснику поруку: „Подаци о уређају су успешно измењени.“ (ИА)



Слика 19 Измена података о уређају

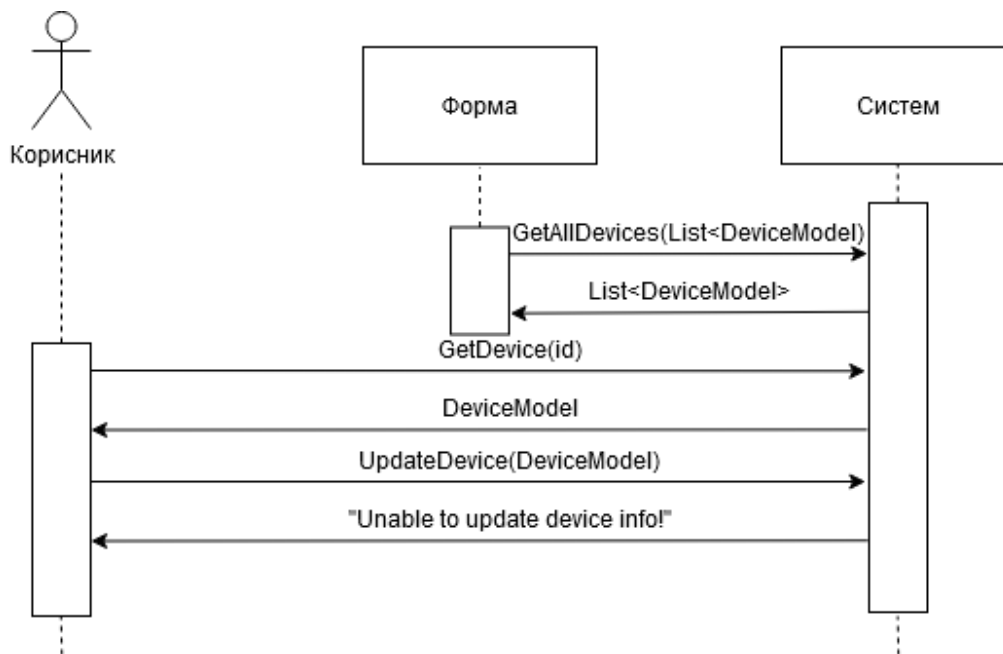
## Алтернативна сценарија

- 4.1. Уколико систем не може да пронађе податке о изабраном уређају, приказује кориснику поруку: „Подаци о уређају нису пронађени!“. Прекида се извршење сценарија (ИА)



Слика 20 Грешка при враћању уређаја

- 6.1. Уколико систем не може да сачува измењене податке, приказује кориснику поруку: „Неуспешна измена података о уређају!“ (ИА)



Слика 21 Грешка при измени података о уређају

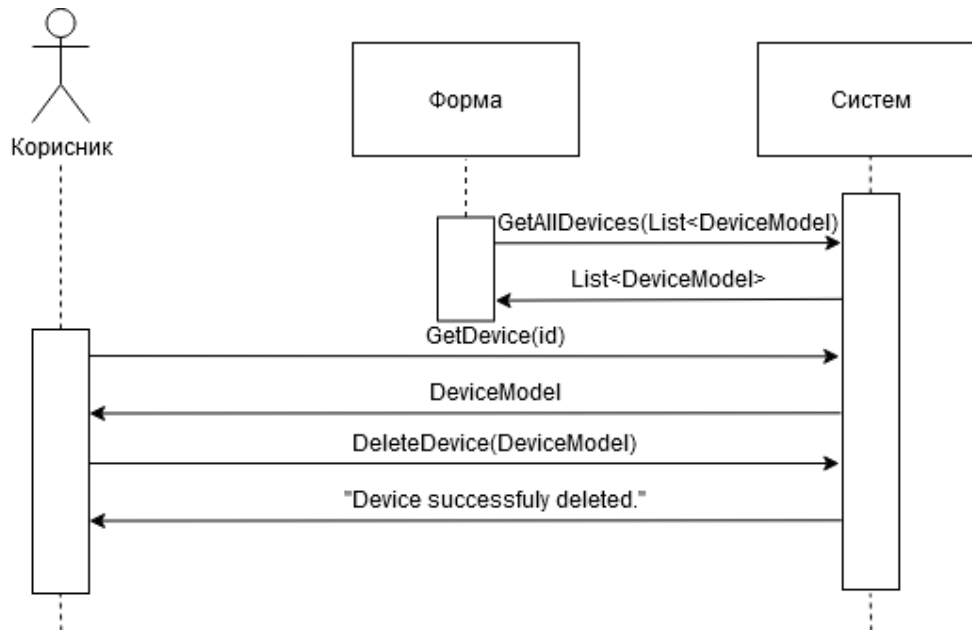
Идентификоване системске операције:

- signal GetAllDevices(List<DeviceModel>)
- signal GetDevice(id, DeviceModel)
- signal UpdateDevice(DeviceModel)

## 5.1.6 ДС6: Дијаграм секвенци случаја коришћења – Брисање уређаја

### Основни сценарио СК

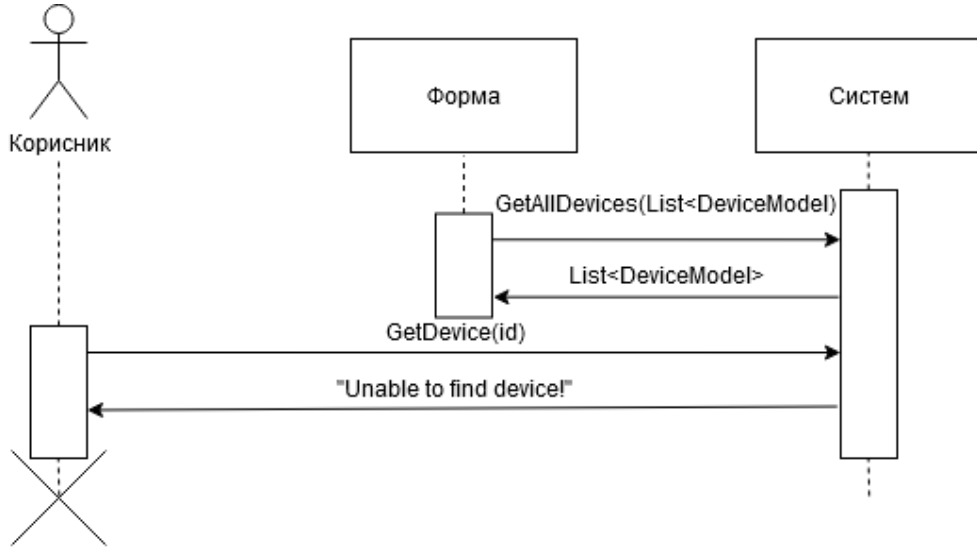
1. Форма позива систем да прикаже листу уређаја (АПСО)
2. Систем враћа форми листу свих уређаја (ИА)
3. Корисник позива систем да прикаже податке о изабраном уређају (АПСО)
4. Систем приказује кориснику податке о изабраном уређају (ИА)
5. Корисник позива систем да обрише изабрани уређај (АПСО)
6. Систем приказује кориснику поруку: „Уређај успешно обрисан.“ (ИА)



Слика 22 Брисање уређаја

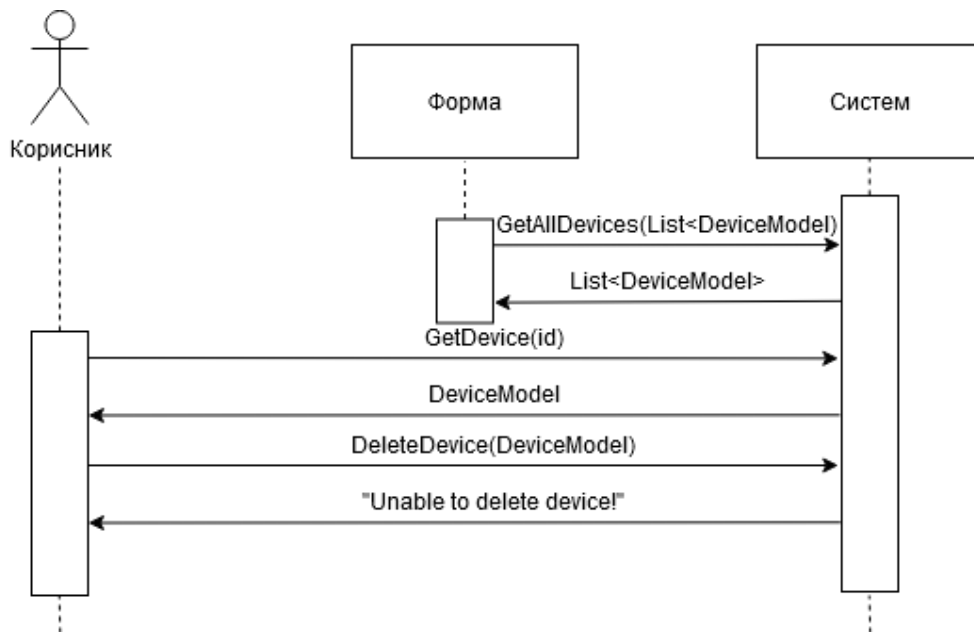
## Алтернативна сценарија

- 4.1. Уколико систем не може да пронађе податке о изабраном уређају, приказује кориснику поруку: „Подаци о уређају нису пронађени!“. Прекида се извршење сценарија (ИА)



Слика 23 Грешка при враћању уређаја

- 6.1. Уколико систем не може да обрише уређај, приказује кориснику поруку: „Неуспешно брисање уређаја!“ (ИА)



Слика 24 Грешка при брисању уређаја

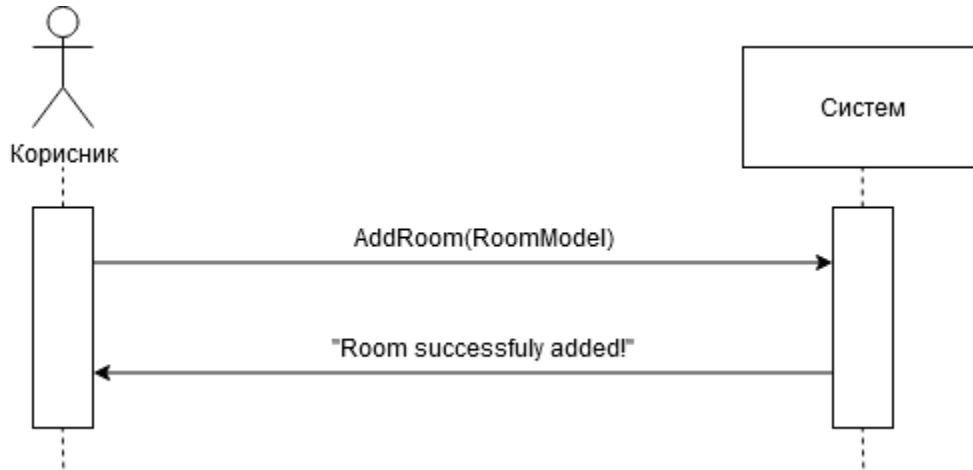
Идентификоване системске операције:

- signal GetAllDevices(List<DeviceModel>)
- signal GetDevice(id, DeviceModel)
- signal DeleteDevice(DeviceModel)

### 5.1.7 ДС7: Дијаграм секвенци случаја коришћења – Унос собе

#### Основни сценарио СК

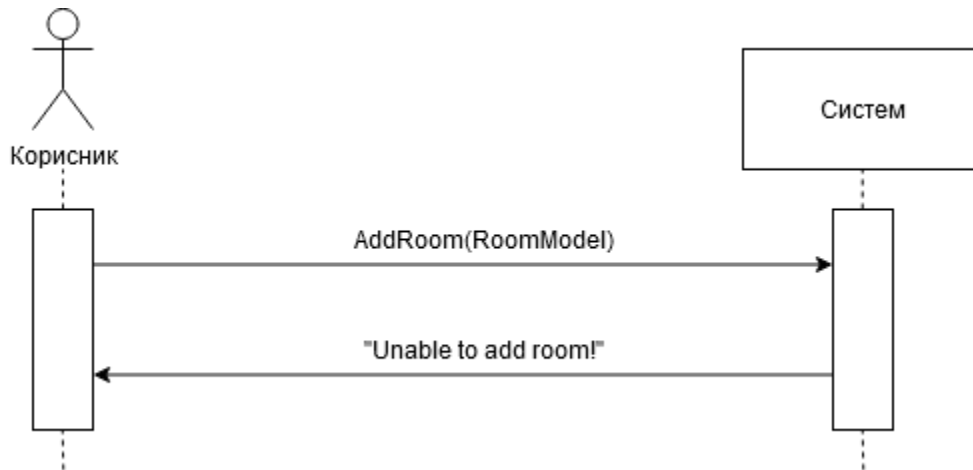
1. Корисник позива систем да запамти податке о новој соби (АПСО)
2. Систем приказује кориснику поруку: „Соба успешно унета.“ (ИА)



Слика 25 Унос собе

#### Алтернативна сценарија

- 2.1. Уколико систем не може да сачува податке о новој соби, приказује кориснику поруку: „Неуспешан унос собе!“ (ИА)



Слика 26 Грешка при уносу собе

Идентификоване системске операције:

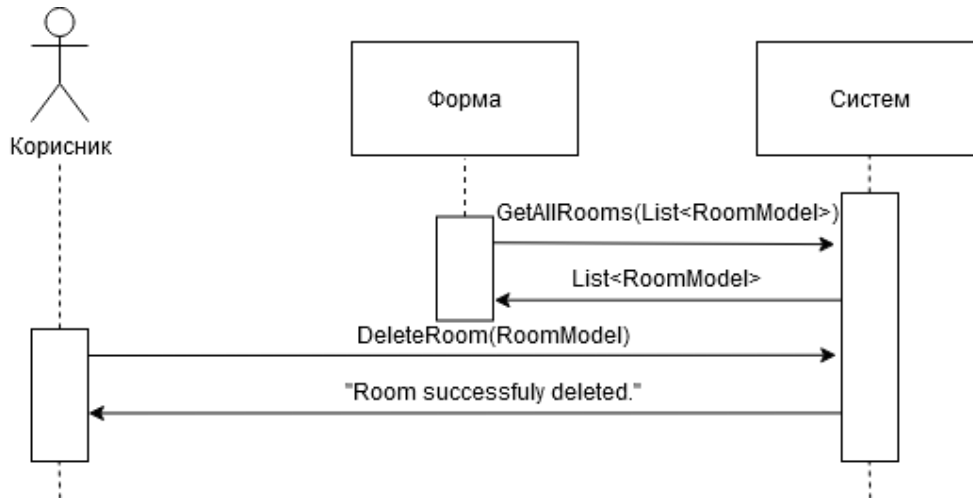
- signal AddRoom(RoomModel)



## 5.1.8 ДС8: Дијаграм секвенци случаја коришћења – Брисање собе

### Основни сценарио СК

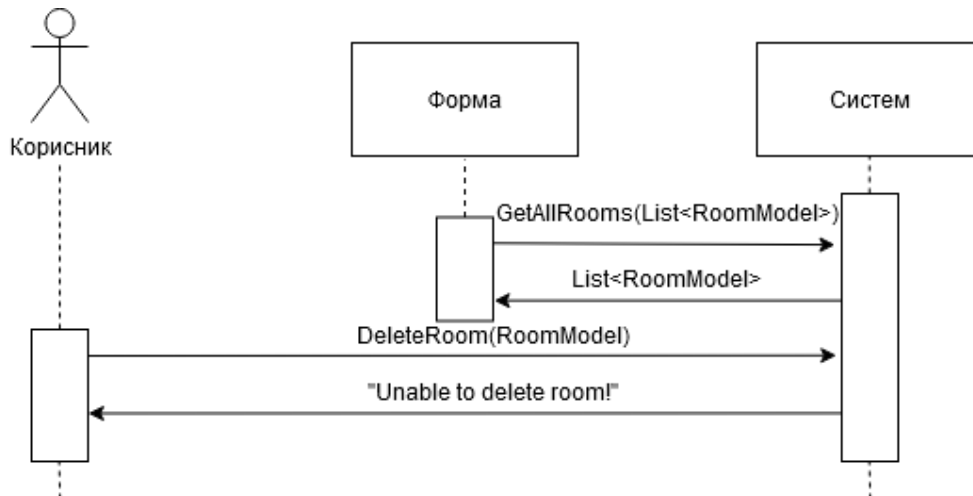
1. Форма позива систем да прикаже листу соба (АПСО)
2. Систем враћа форми листу свих соба (ИА)
3. Корисник позива систем да обрише изабрану собу (АПСО)
4. Систем приказује кориснику поруку: „Соба успешно обрисана.“ (ИА)



Слика 27 Брисање собе

### Алтернативна сценарија

- 4.1. Уколико систем не може да обрише собу, приказује кориснику поруку: „Неуспешно брисање собе!“ (ИА)



Слика 28 Грешка при брисању собе

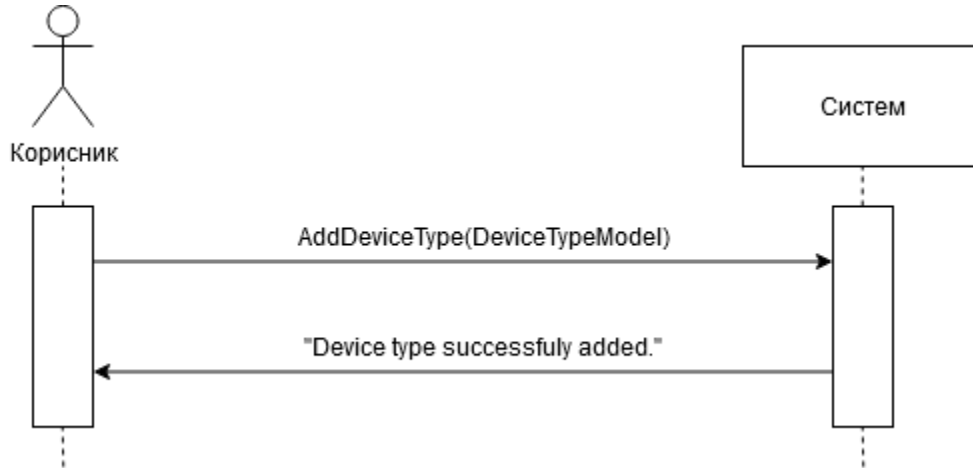
Идентификоване системске операције:

- signal GetAllRooms(List<RoomModel>)
- signal DeleteRoom(RoomModel)

## 5.1.9 ДС9: Дијаграм секвенци случаја коришћења – Унос типа уређаја

### Основни сценарио СК

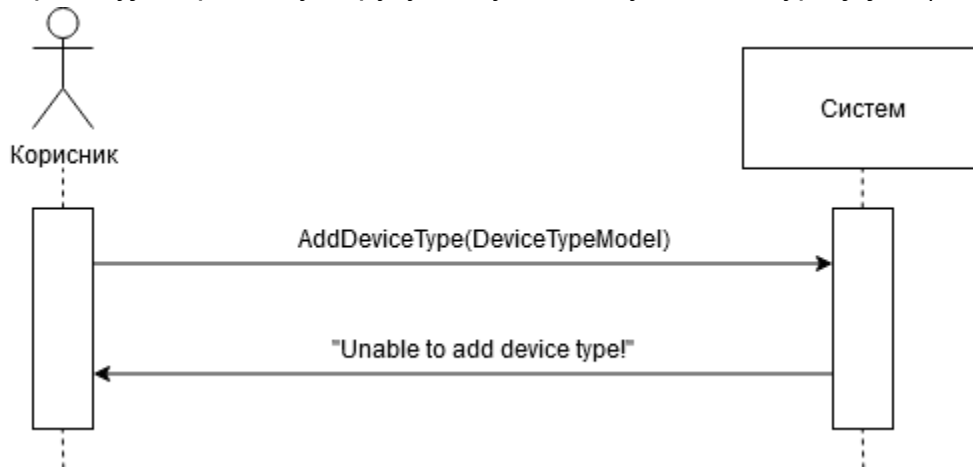
1. Корисник позива систем да запамти податке о новом типу уређаја (АПСО)
2. Систем приказује кориснику поруку: „Тип уређаја успешно унет.“



Слика 29 Додавање типа уређаја

### Алтернативна сценарија

- 2.1. Уколико систем не може да сачува податке о новом типу уређаја, приказује кориснику поруку: „Неуспешан унос типа уређаја!“ (ИА)



Слика 30 Грешка при додавању типа уређаја

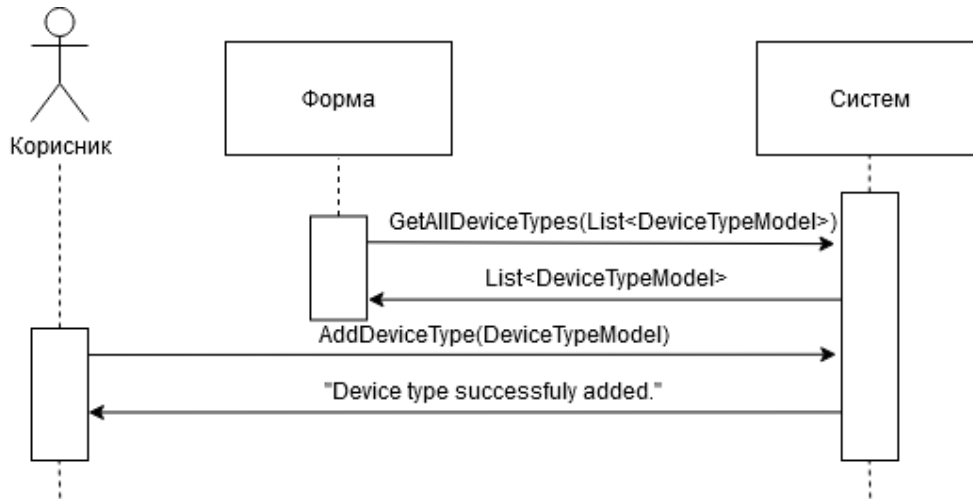
Идентификоване системске операције:

- signal AddDeviceType(DeviceTypeModel)

### 5.1.10 ДС10: Дијаграм секвенци случаја коришћења – Брисање типа уређаја

#### Основни сценарио СК

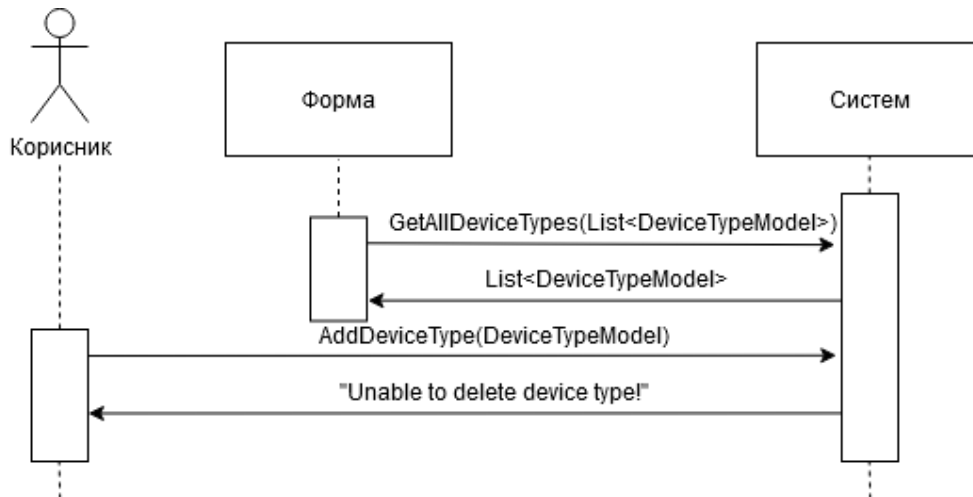
1. Фирма позива систем да прикаже листу типова уређаја (АПСО)
2. Систем враћа форми листу свих типова уређаја (ИА)
3. Корисник позива систем да обрише изабрани тип уређаја (АПСО)
4. Систем приказује кориснику поруку: „Тип уређаја успешно обрисан.“ (ИА)



Слика 31 Брисање типа уређаја

#### Алтернативна сценарија

- 4.1. Уколико систем не може да обрише тип уређаја, приказује кориснику поруку: „Неуспешно брисање типа уређаја!“ (ИА)



Слика 32 Грешка при брисању типа уређаја

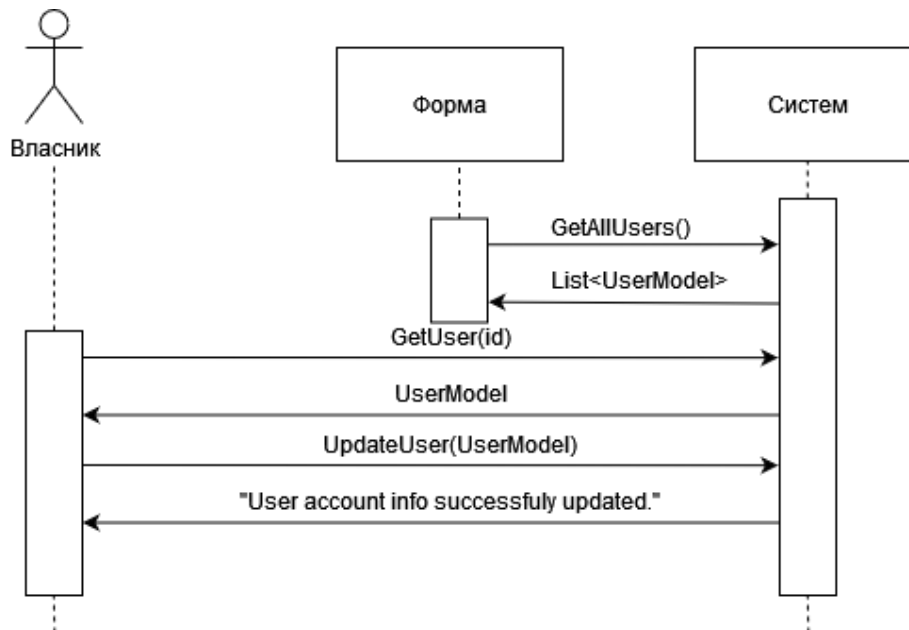
Идентификоване системске операције:

- signal GetAllDeviceTypes(List<DeviceTypeModel>)
- signal RemoveDeviceType(DeviceTypeModel)

### 5.1.11 ДС11: Дијаграм секвенци случаја коришћења – Измена података о налогу корисника

#### Основни сценарио СК

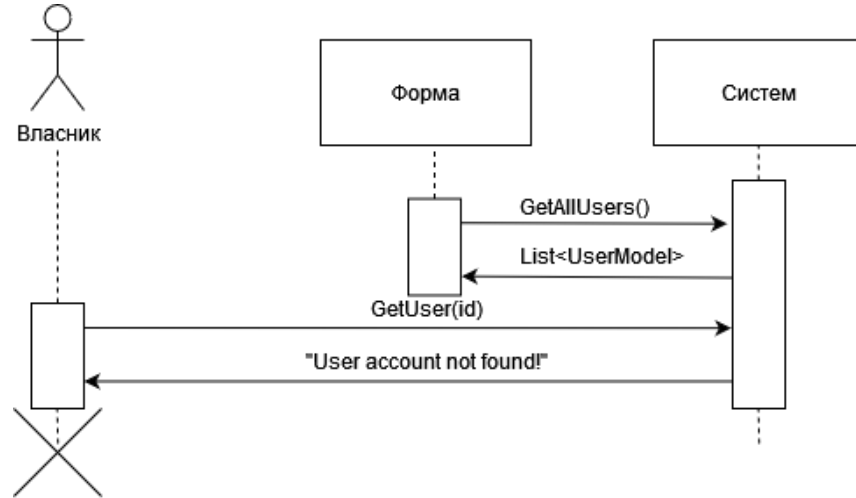
1. Форма позива систем да прикаже листу налога (АПСО)
2. Систем враћа форми листу свих налога (ИА)
3. Власник позива систем да прикаже податке о изабраном налогу (АПСО)
4. Систем приказује власнику податке о изабраном налогу (ИА)
5. Власник позива систем да запамти унете податке (АПСО)
6. Систем приказује власнику поруку: „Подаци о налогу су успешно измењени.“ (ИА)



Слика 33 Измена података налога о кориснику

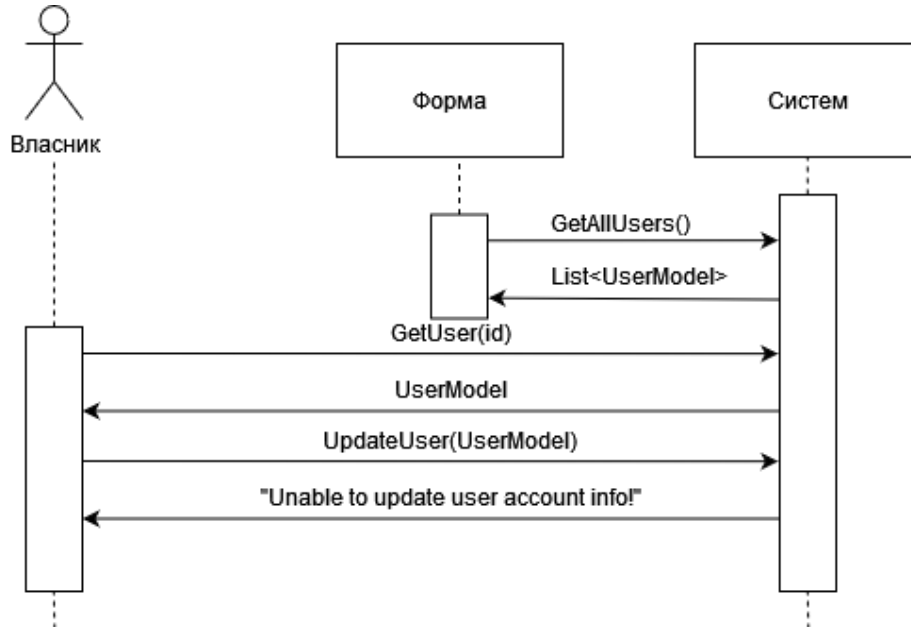
## Алтернативна сценарија

- 4.1. Уколико систем не може да пронађе податке о изабраном налогу, приказује власнику поруку: „Подаци о налогу нису пронађени!“. Прекида се извршење сценарија (ИА)
- 6.1. Уколико систем не може да сачува измењене податке, приказује



Слика 34 Грешка при враћању налога

власнику поруку: „Неуспешна измена података о налогу!“ (ИА)



Слика 35 Грешка при измени података о налогу

Идентификоване системске операције:

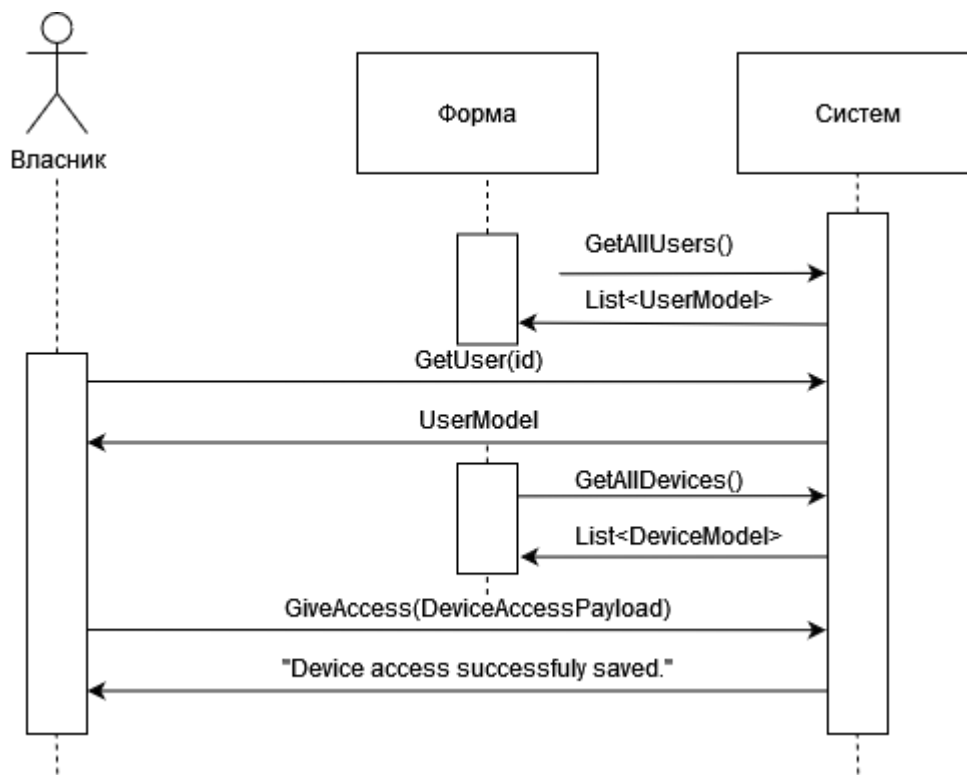
- signal GetAllUsers(List<UserModel>)
- signal GetUser(id, UserModel)
- signal UpdateUser(UserModel)



### 5.1.12 ДС12: Дијаграм секвенци случаја коришћења – Додавање дозволе за коришћење уређаја

#### Основни сценарио СК

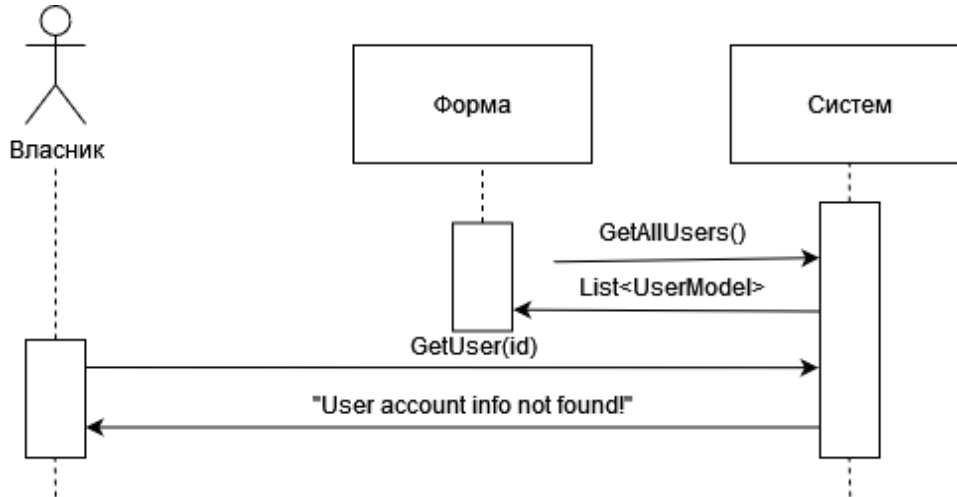
1. Форма позива систем да врати листу налога (АПСО)
2. Систем враћа форми листу свих налога (ИА)
3. Власник позива систем да врати податке о изабраном кориснику (АПСО)
4. Систем приказује податке о изабраном кориснику (ИА)
5. Форма позива систем да врати листу уређаја (АПСО)
6. Систем враћа форми листу свих уређаја (ИА)
7. Власник позива систем да запамти податке о дозволи коришћења изабраног уређаја за претходно изабраног корисника (АПСО)
8. Систем приказује власнику поруку: „Успешно додавање дозволе коришћења уређаја.“ (ИА)



Слика 36 Додавање дозволе за коришћење уређаја

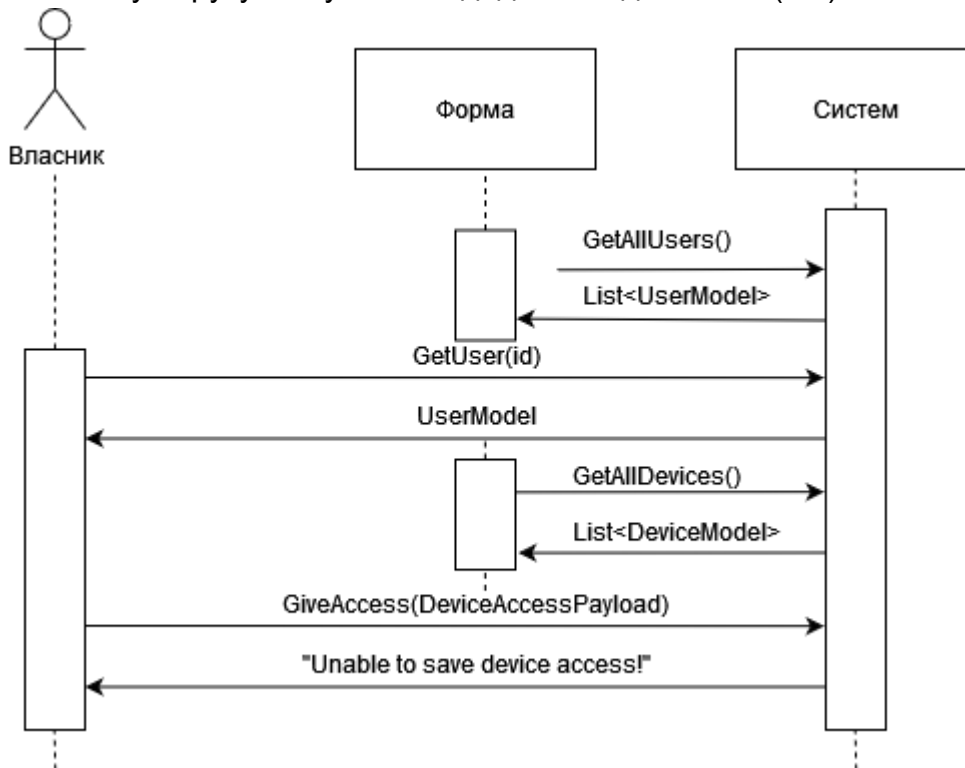
## Алтернативна сценарија

- 4.1. Уколико систем не може да пронађе корисника, приказује власнику поруку: „Подаци о изабраном кориснику нису пронађени!“. Прекида се извршење сценарија (ИА)



Слика 37 Грешка при враћању корисника

- 8.1. Уколико систем не може да сачува дозволу коришћења, приказује власнику поруку: „Неуспешна додавање дозволе!“ (ИА)



Слика 38 Грешка при додавању дозволе

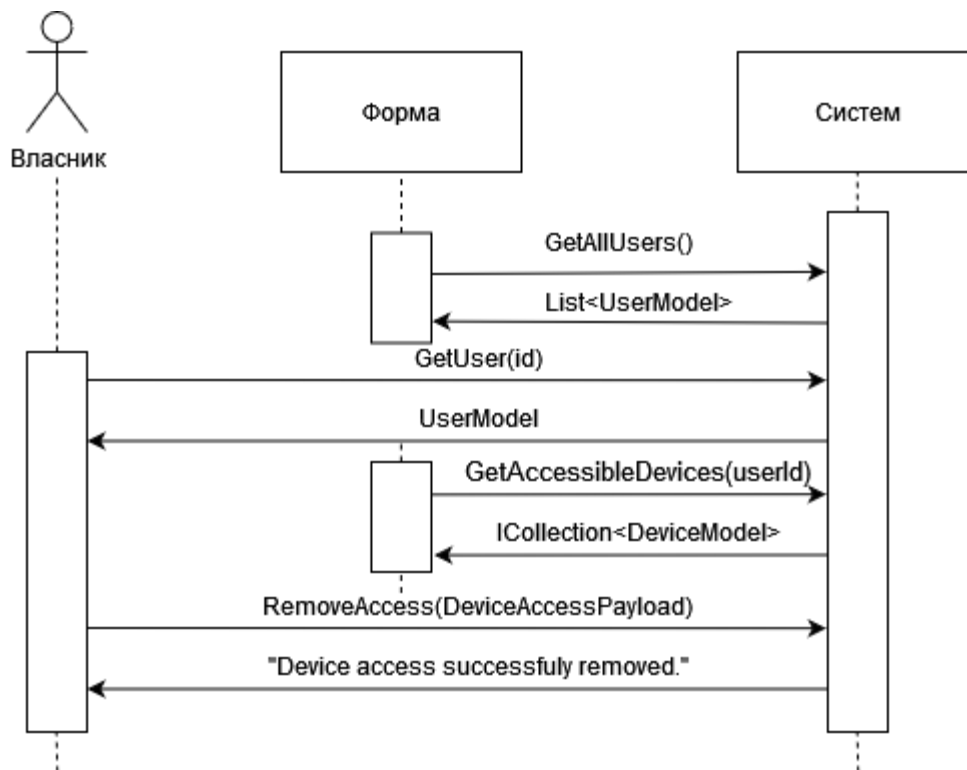
Идентификоване системске операције:

- signal GetAllUsers(List<UserModel>)
- signal GetUser(id, UserModel)
- signal GiveAccess(DeviceAccessPayload)

### 5.1.13 ДС13: Дијаграм секвенци случаја коришћења – Брисање дозволе за коришћење уређаја

#### Основни сценарио

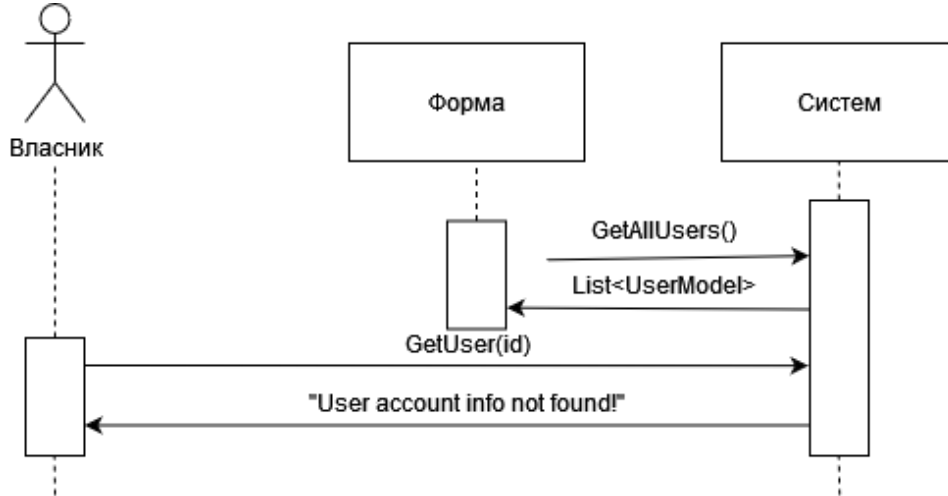
1. Форма позива систем да прикаже листу налога (АПСО)
2. Систем враћа форми листу свих налога (ИА)
3. Власник позива систем да врати податке о изабраном кориснику (АПСО)
4. Систем приказује податке о изабраном кориснику (ИА)
5. Форма позива систем да прикаже листу уређаја којима изабрани корисник има приступ (АПСО)
6. Систем враћа форми листу свих уређаја који одговарају критеријуму (ИА)
7. Власник позива систем да обрише дозволу коришћења изабраног уређаја за претходно изабраног корисника (АПСО)
8. Систем приказује власнику поруку: „Успешно брисање дозволе коришћења уређаја.“ (ИА)



Слика 39 Брисање дозволе за коришћење уређаја

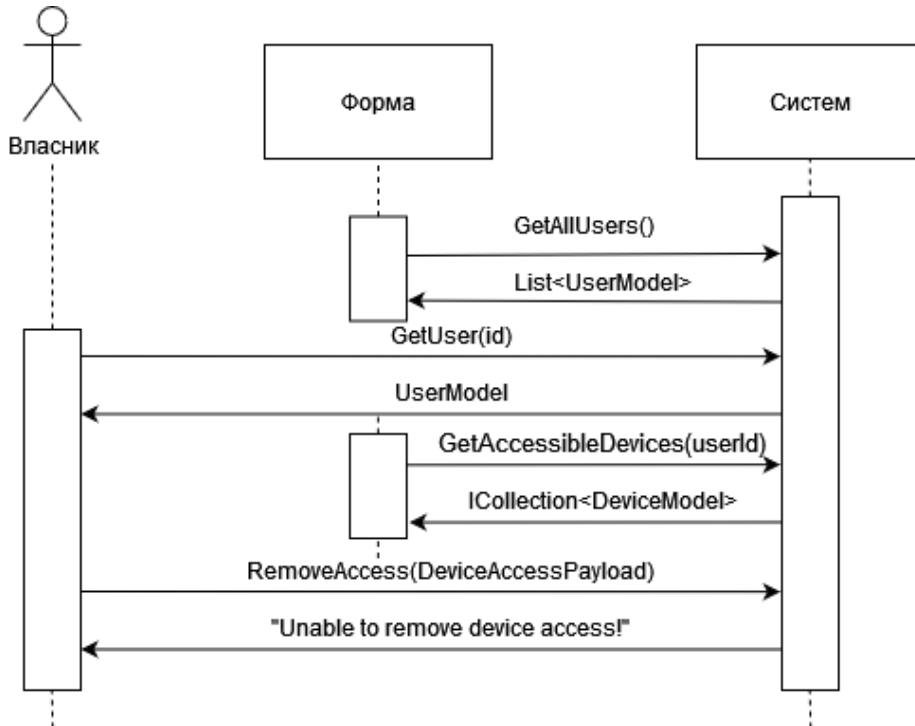
## Алтернативна сценарија

- 4.1. Уколико систем не може да пронађе корисника, приказује власнику поруку: „Подаци о изабраном кориснику нису пронађени!“. Прекида се извршење сценарија (ИА)



Слика 40 Грешка при враћању корисника

- 8.1. Уколико систем не може да обрише дозволу коришћења, приказује власнику поруку: „Неуспешно брисање дозволе!“ (ИА)



Слика 41 Грешка при брисању дозволе

Идентификоване системске операције:

signal GetAllUsers(List<UserModel>)

signal GetUser(id, UserModel)

signal RemoveAccess(DeviceAccessPayload)

Фазом анализе идентификовано је 19 системских операција које је потребно пројектовати:

*signal* Login(LoginRequest, LoginResult)

*signal* ChangeDeviceState(DeviceModel)

*signal* GetUserData(UserModel)

*signal* UpdateUser(UserModel)

*signal* AddDevice(DevicePayload)

*signal* GetAllDevices(List<DeviceModel>)

*signal* GetDevice(id, DeviceModel)

*signal* UpdateDevice(DeviceModel)

*signal* DeleteDevice(DeviceModel)

*signal* GetAllRooms(List<RoomModel>)

*signal* AddRoom(RoomModel)

*signal* DeleteRoom(RoomModel)

*signal* GetAllDeviceTypes(List<DeviceTypeModel>)

*signal* AddDeviceType(DeviceTypeModel)

*signal* RemoveDeviceType(DeviceTypeModel)

*signal* GetAllUsers(List<UserModel>)

*signal* GetUser(id, UserModel)

*signal* GiveAccess(DeviceAccessPayload)

*signal* RemoveAccess(DeviceAccessPayload)

## 5.2 Понашање софтверског система – Дефинисање уговора системским операцијама

### **Уговор УГ1:** ПријавиКорисника

Операција: Login(LoginRequest)

Веза са СК: СК1

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

Постуслови: Корисник је пријављен.

### **Уговор УГ2:** ИзмениСтањеУређаја

Операција: ChangeDeviceState(DeviceModel)

Веза са СК: СК2

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

Постуслови: Стање уређаја је промењено.

### **Уговор УГ3:** ВратиПодаткеОНалогу

Операција: GetUserData(UserModel)

Веза са СК: СК3

Предуслови:

Постуслови:

### **Уговор УГ4:** ИзмениКорисника

Операција: UpdateUser(UserModel)

Веза са СК: СК3, СК11

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

Постуслови: Подаци о кориснику су измењени.

**Уговор УГ5: ДодајУређај**

Операција: AddDevice(DevicePayload)

Веза са СК: СК4

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

Постуслови: Подаци о уређају су сачувани.

**Уговор УГ6: ВратиСвеУређаје**

Операција: GetAllDevices(List<DeviceModel>)

Веза са СК: СК5, СК6

Предуслови: /

Постуслови: /

**Уговор УГ7: ВратиУређај**

Операција: GetDevice(id, DeviceModel)

Веза са СК: СК5, СК6

Предуслови: /

Постуслови: /

**Уговор УГ8: ИзмениУређај**

Операција: UpdateDevice(DeviceModel)

Веза са СК: СК5

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

Постуслови: Подаци о уређају су измењени.



**Уговор УГ9: ОбришиУређај**

Операција: DeleteDevice(DeviceModel)

Веза са СК: СК6

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

Постуслови: Уређај је обрисан.

**Уговор УГ10: ВратиСвеСобе**

Операција: GetAllRooms(List<RoomModel>)

Веза са СК: СК8

Предуслови: /

Постуслови: /

**Уговор УГ11: ДодајСобу**

Операција: AddRoom(RoomModel)

Веза са СК: СК7

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

Постуслови: Подаци о соби су сачувани.

**Уговор УГ12: ОбришиСобу**

Операција: DeleteRoom(RoomModel)

Веза са СК: СК8

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

Постуслови: Соба је обрисана.

**Уговор УГ13:** ВратиСвеТиповеУређаја

Операција: GetAllDeviceTypes(List<DeviceTypeModel>)

Веза са СК: СК10

Предуслови: /

Постуслови: /

**Уговор УГ14:** ДодајТипУређаја

Операција: AddDeviceType(DeviceTypeModel)

Веза са СК: СК9

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

Постуслови: Подаци о типу уређаја су сачувани.

**Уговор УГ15:** ОбришиТипУређаја

Операција: RemoveDeviceType(DeviceTypeModel)

Веза са СК: СК10

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

Постуслови: Тип уређаја је обрисан.

**Уговор УГ16:** ВратиСвеКориснике

Операција: GetAllUsers(List<UserModel>)

Веза са СК: СК11, СК12, СК13

Предуслови: /

Постуслови: /

**Уговор УГ17:** ВратиКорисника

Операција: GetUser(id, UserModel)

Веза са СК: СК11, СК12, СК13

Предуслови: /

Постуслови: /

**Уговор УГ18:** ДодајПристапУређају

Операција: GiveAccess(DeviceAccessPayload)

Веза са СК: СК12

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

Постуслови: Приступ уређају је сачуван.

**Уговор УГ19:** ОбришиПристапУређају

Операција: RemoveAccess(DeviceAccessPayload)

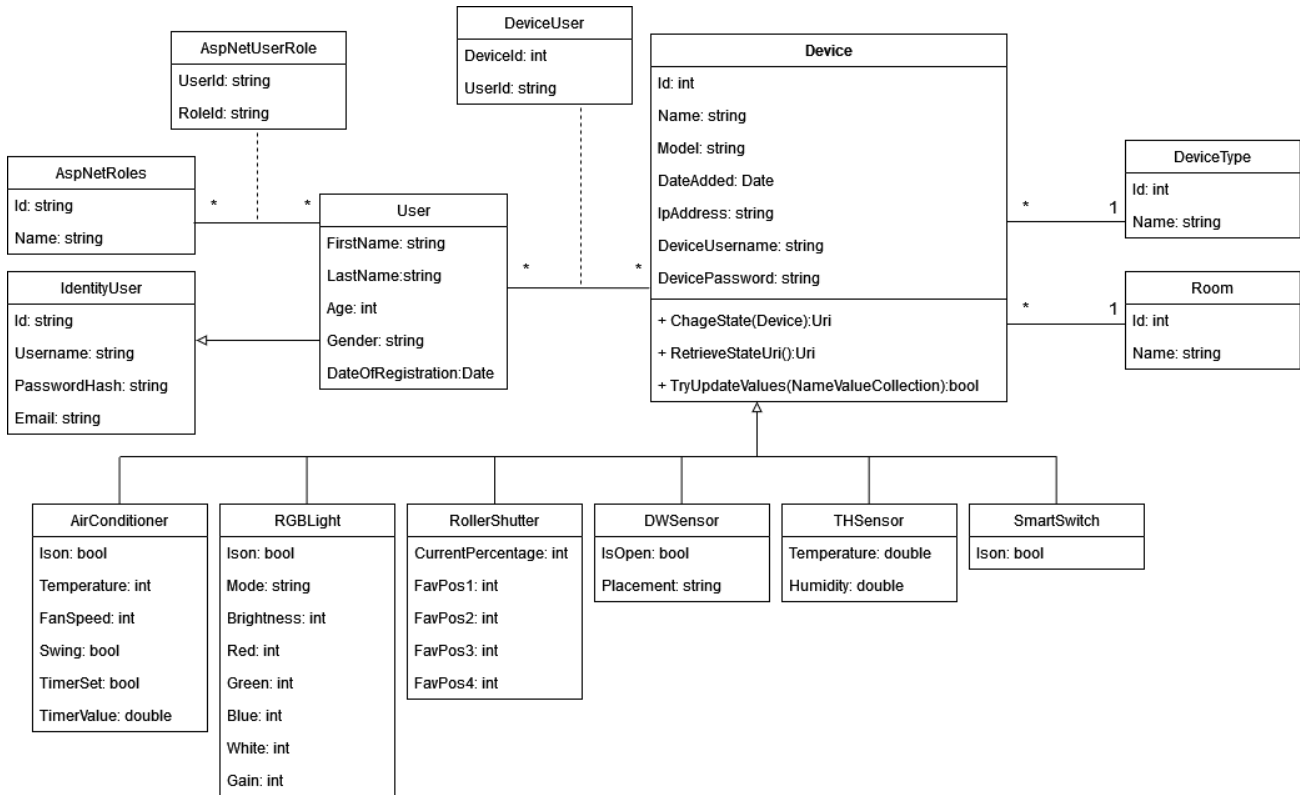
Веза са СК: СК13

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

Постуслови: Обрисан је приступ уређају.

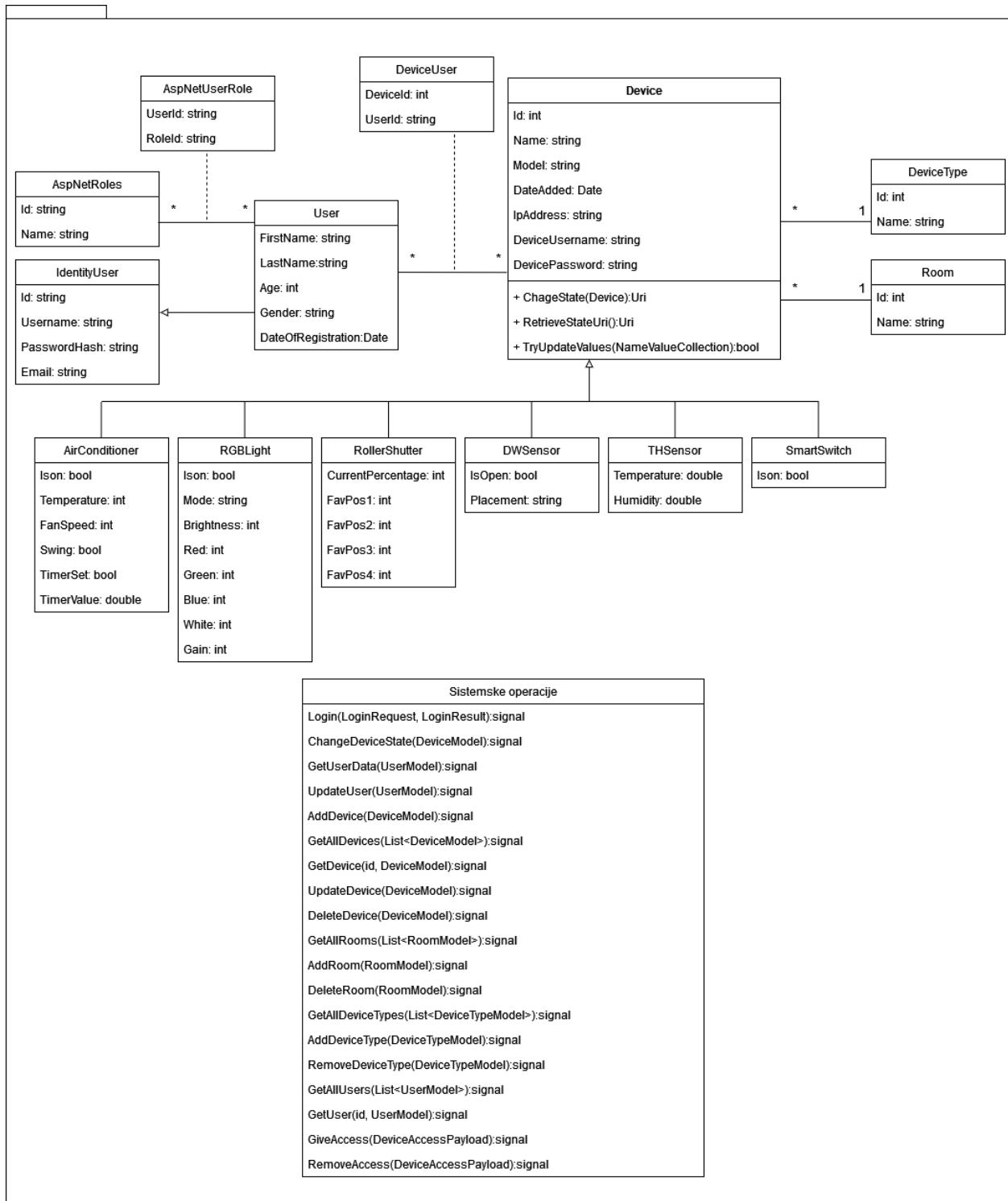
## 5.3 Структура софтверског система – Концептуални модел

Структура софтверског система се описује помоћу концептуалног модела. На концептуалном моделу приказане су концептуалне класе доменског модела као и њихове међусобне везе, односно асоцијације између њих. Називају оменским моделима или моделима објектне анализе. [1]



Слика 42 Концептуални модел

Као резултат фазе анализе добија се логичка структура и понашање софтверског система, што представља пословну логику система:



Слика 43 Концептуални модел и системске операције

## 6. Фаза пројектовања

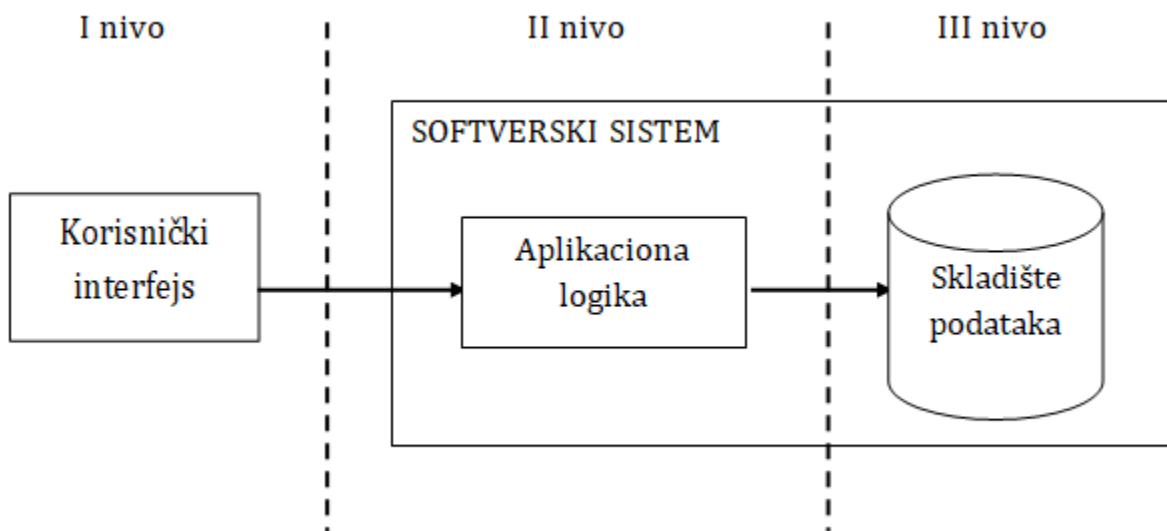
Фаза пројектовања описује физичку структуру и понашање софтверског система (архитектуру софтверског система). Најчешће коришћена архитектура је тронивојска архитектура која се састоји из следећих нивоа [1]:

### 6.1 Архитектура софтверског система

Архитектура софтверског систем је тронивојска и састоји се од следећих нивоа:

- Кориснички интерфејс
- Апликациона логика
- Складиште података

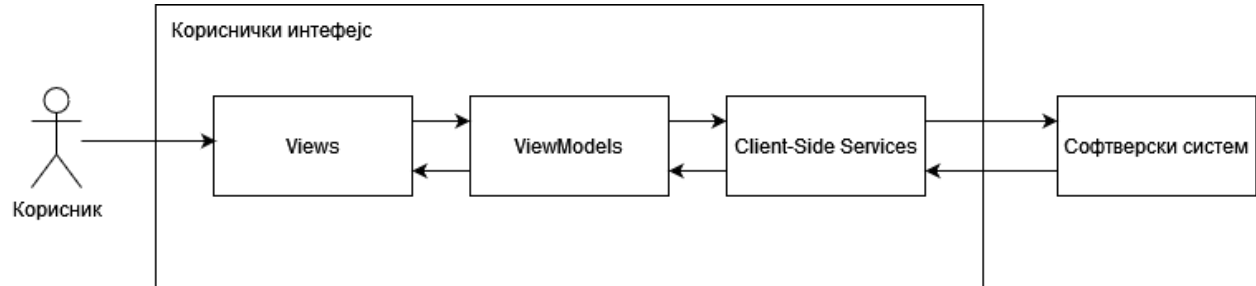
Ниво корисничког интерфејса представља клијентску апликацију, док серверска апликација садржи апликациону логику и складиште података.



Слика 44 Тронивојска архитектура софтверског система

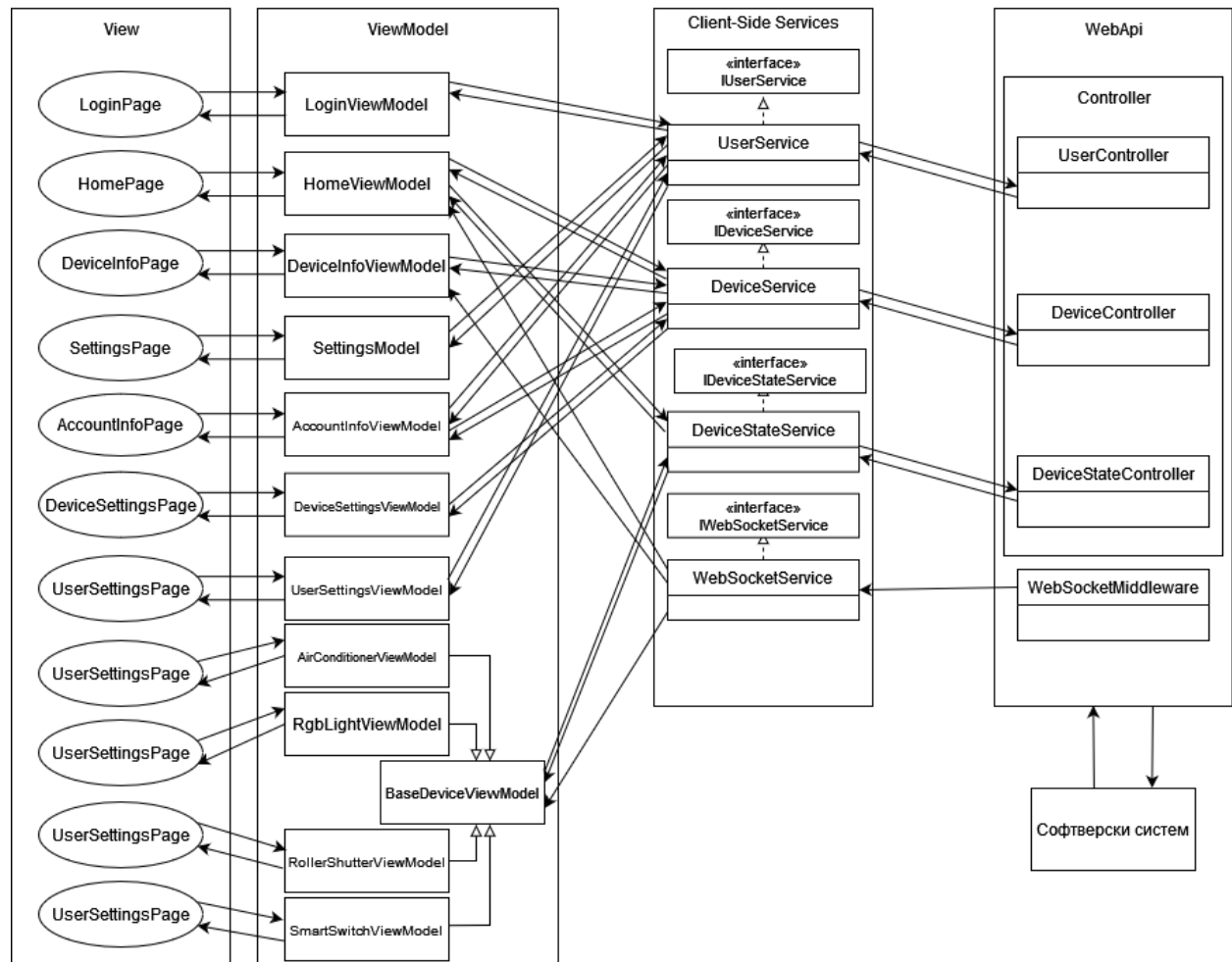
## 6.2 Пројектовање корисничког интерфејса

Кориснички интерфејс представља начин на који корисник комуницира са системом. Базира се на MVVM архитектури, а основна структура се може видети на следећој слици:



Слика 45 Архитектура клијентске апликација

Детаљан начин комуникације корисничког интерфејса са сервером представљен је на следећој слици:



Слика 46 Детаљна комуникација клијента са сервером

## 6.3 Пројектовање апликационе логике

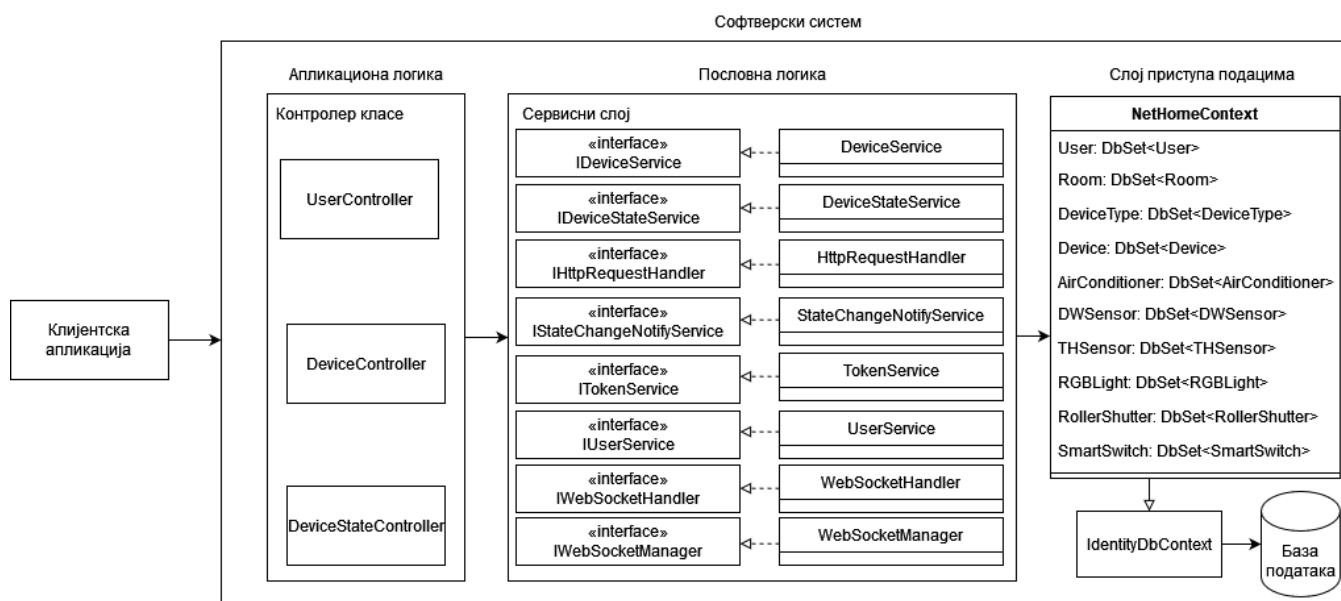
Апликациона логика садржи класе које су неопходне за имплементацију пословне логике:

- Контролери – прихватају долазне захтеве од клијентске апликације и прослеђују их сервисима.
- Сервиси – обрађују пристигле захтеве.
- Контекст базе података – служи за координацију приступа бази података
- Ентитети – класе које садрже атрибуте, али такође, у случајевима када се односе на уређаје и логику за управљање уређајем на који се односе. На основу ентитета, а коришћењем Entity Framework-а креирају се табеле у бази података.

### 6.3.1 Пројектовање контролера апликационе логике

Контролер апликационе логике служи да прими захтеве за извршавање системске операције од клијентске апликације и проследи их сервисном слоју који је одговоран за извршавање захтева. У контролерима се такође дефинише и неопходан ниво приступа свакој крајњој тачки контролера (endpoint). Након што сервисни слој изврши системску операцију, резултат враћа контролеру који је тада задужен да тај резултат проследи клијенту који је послао захтев.

На следећој слици приказана је архитектура софтверског система након пројектовања контролера и класа које чине апликациону логику:



Слика 47 Приказ апликационе логике серверске апликације



## СК1: Случај коришћења – Пријава на систем

### Назив СК

Пријава на систем

### Актер СК

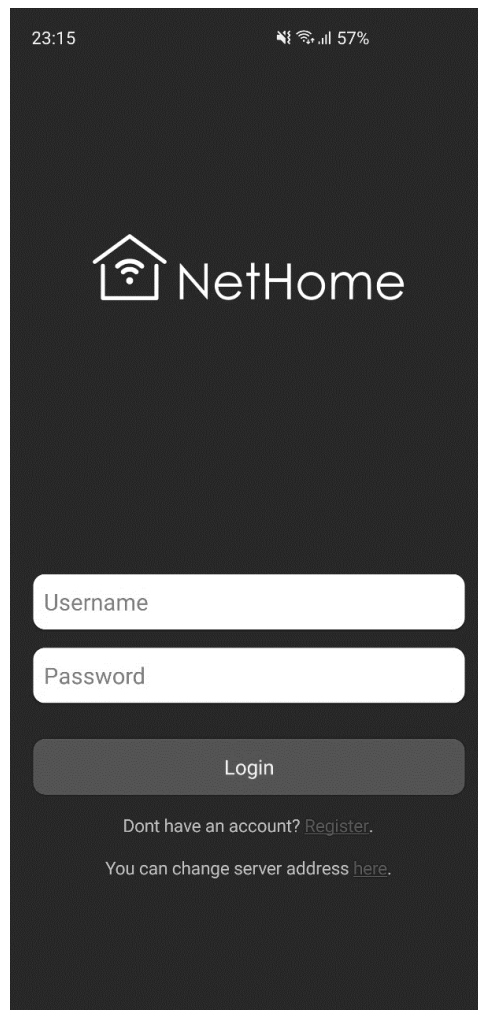
Корисник

### Учесници СК

Корисник и систем (програм)

### Предуслов

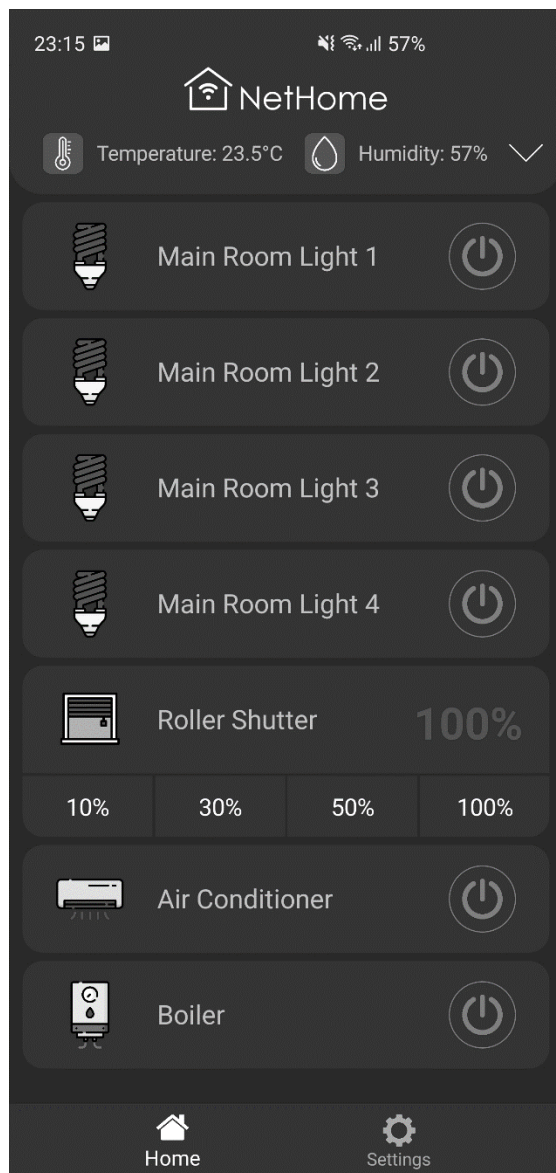
Систем је укључен и приказује страницу за пријаву на систем.



Слика 48 Страница за пријаву на систем

## Основни сценарио

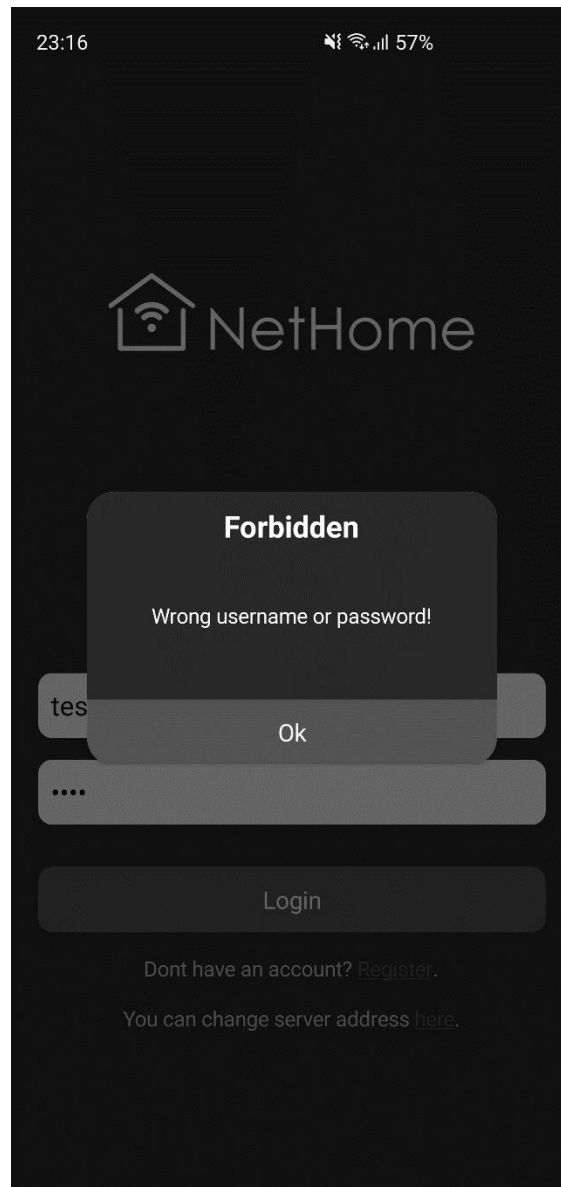
1. Корисник уноси корисничко име и лозинку (АПУСО)
2. Корисник контролише да ли је коректно унео корисничко име и лозинку (АНСО)
3. Корисник позива систем да се пријави (АПСО)
4. Систем проверава податке о кориснику (СО)
5. Систем кориснику враћа податке о налогу и токен за ауторизацију и приказује му почетну страну (ИА)



Слика 49 Почетна страница

## Алтернативна сценарија

- 5.1. Уколико систем не може да нађе корисника са унетим корисничким именом и лозинком, приказује кориснику поруку: „Погрешно корисничко име или лозинка!“ (ИА)



Слика 50 Грешка при пријави

## СК2: Случај коришћења – Промена стања уређаја

### Назив СК

Промена стања уређаја

### Актер СК

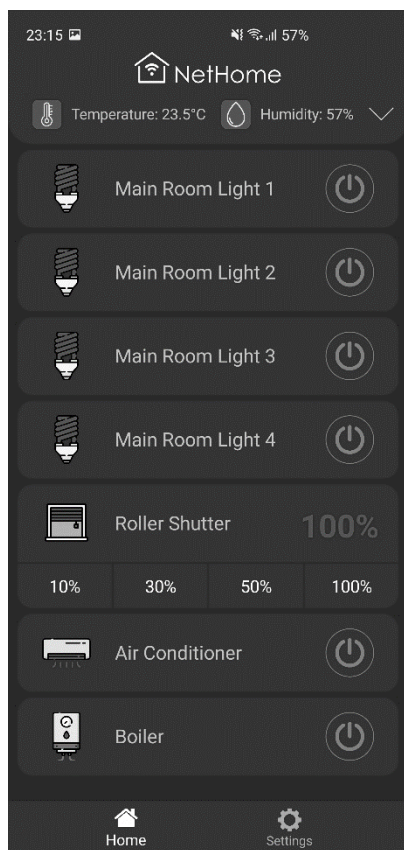
Корисник

### Учесници СК

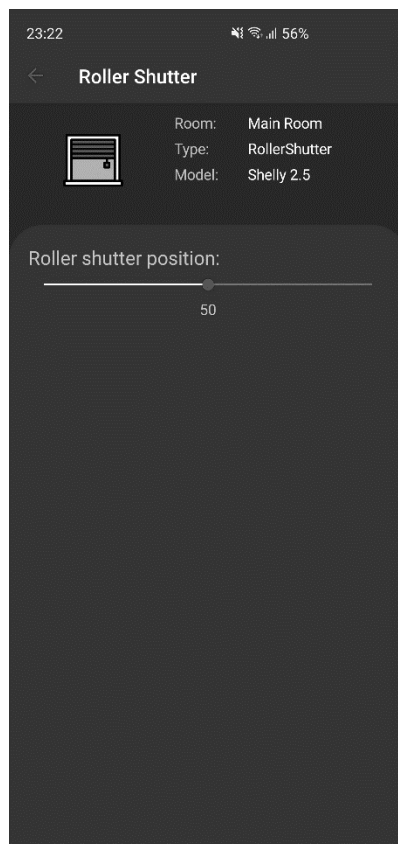
Корисник и систем (програм)

### Предуслов

Систем је укључен и корисник је пријављен на своја налог. Систем приказује почетну страницу или страницу специфичног уређаја.



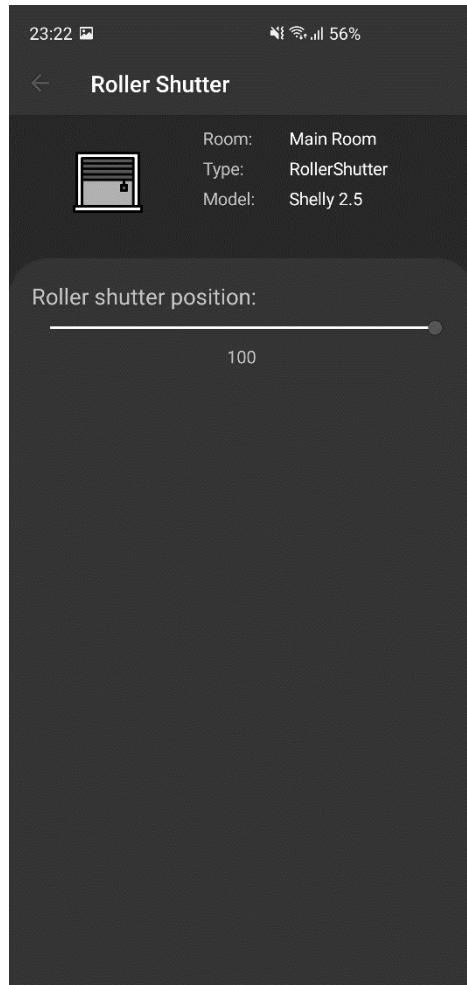
Слика 51 Почетна страница



Слика 52 Страница за контролу ролетни

## Основни сценарио

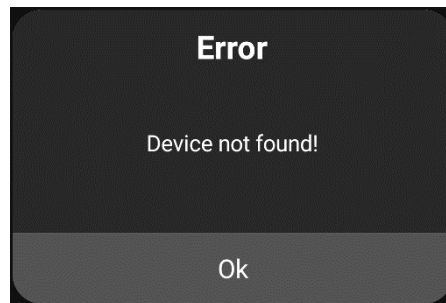
1. Корисник уноси ново стање уређаја (АПУСО)
2. Корисник позива систем да изврши промену стања уређаја (АПСО)
3. Систем проналази уређај, проверава да ли корисник има приступ уређају, а затим врши измену стања уређаја (СО)
4. Систем враћа кориснику информације о уређају са измењеним стањем и приказује ново стање уређаја (ИА)



Слика 53 Измењено стање уређаја

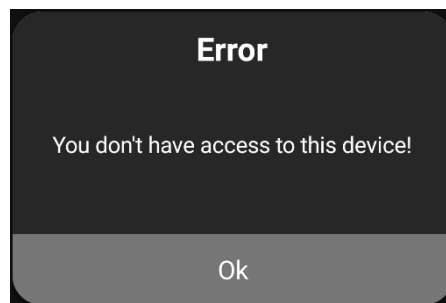
## Алтернативна сценарија

- 4.1. Уколико систем не може да нађе уређај, приказује кориснику поруку: „Уређај није пронађен!“. Прекида се извршење сценарија (ИА)



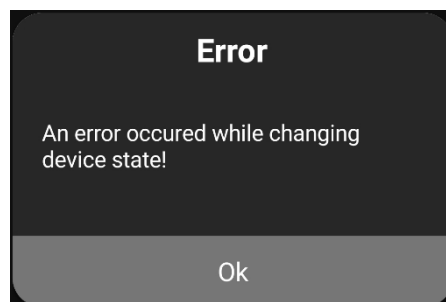
Слика 54 Грешка при враћању уређаја

- 4.2. Уколико корисник нема приступ уређају, приказује поруку: „Немате приступ овом уређају!“. Прекида се извршење сценарија (ИА)



Слика 55 Грешка са дозволом за приступ уређају

- 4.3. Уколико систем није у могућности да изврши тражену измену, приказује проук: „Дошло је до грешке приликом измене стања уређаја!“ (ИА)



Слика 56 Грешка при промени стања

## СКЗ: Случај коришћења – Измена података о налогу

### Назив СК

Измена података о налогу

### Актер СК

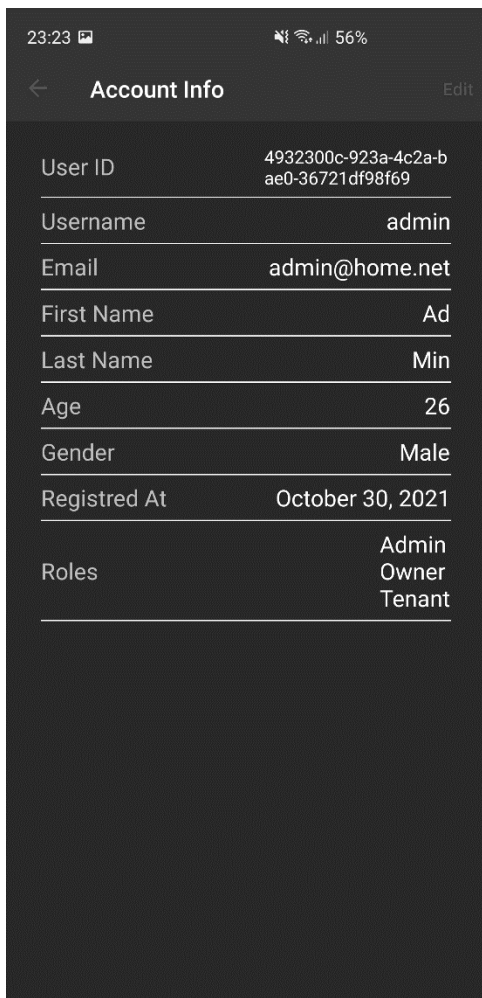
Корисник

### Учесници СК

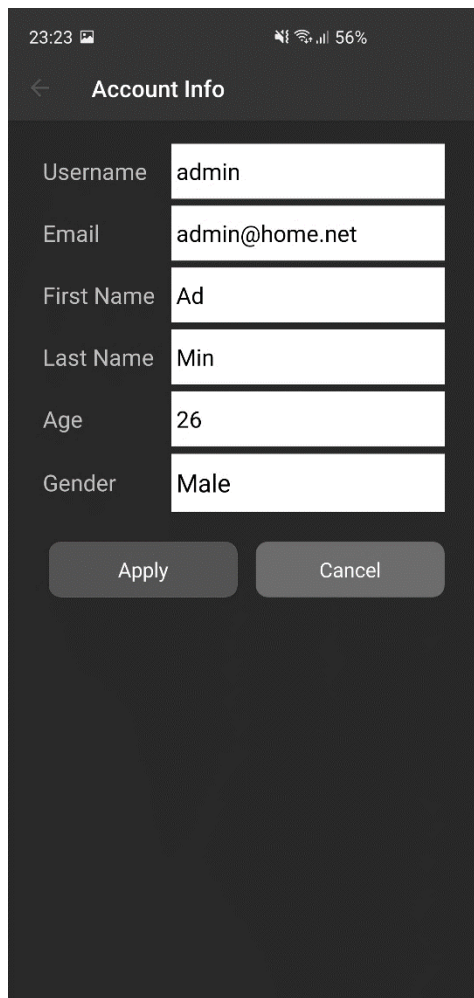
Корисник и систем (програм)

### Предуслов

Систем је укључен и корисник је пријављен на своја налог. Систем приказује страницу за рад са подацима о налогу.



Слика 58 Страница са информацијама о налогу

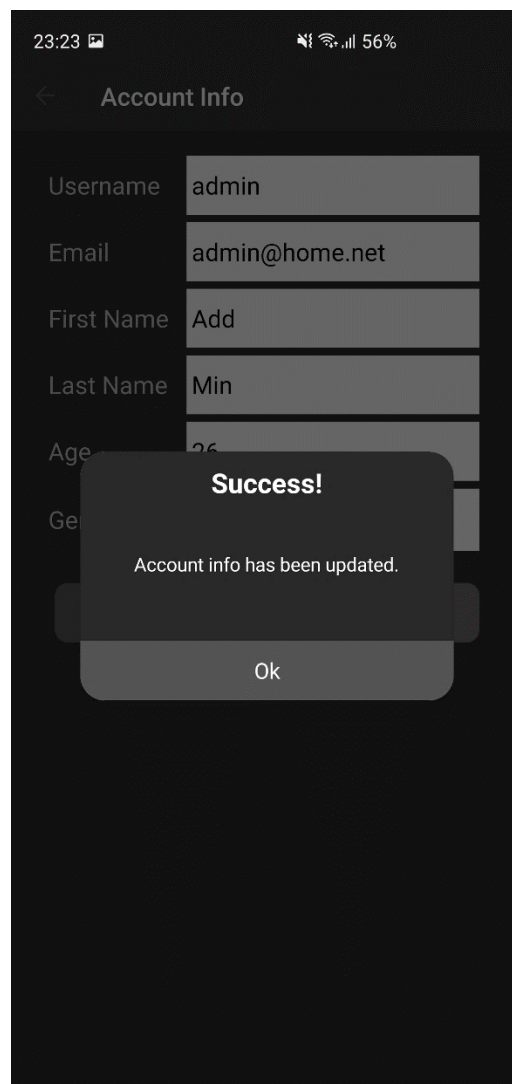


Слика 57 Страница за измену података о налогу



## Основни сценарио

1. Корисник позива систем да прикаже податке о његовом налогу (АПСО)
2. Систем тражи податке о налогу корисника (СО)
3. Систем приказује податке о налогу корисника (ИА)
4. Корисник уноси измене података о налогу (АПУСО)
5. Корисник проверава да ли је исправно унео нове податке (АНСО)
6. Корисник позива систем да запамти податке о налогу (АПСО)
7. Систем памти измењене податке о налогу (СО)
8. Систем приказује кориснику поруку: „Подаци о налогу су успешно сачувани.“ (ИА)

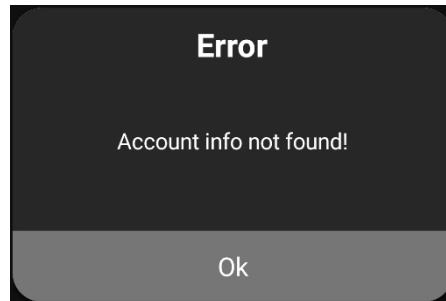


Слика 59 Успешно измењени подаци о налогу



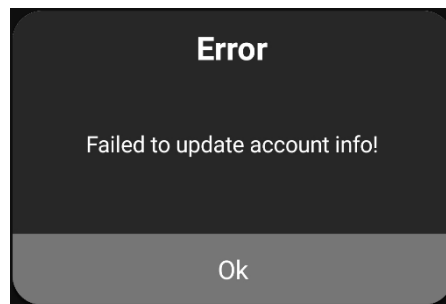
## Алтернативна сценарија

- 4.2. Уколико систем не може да пронађе податке о налогу корисника, приказује кориснику поруку: „Подаци о налогу нису пронађени!“. Прекида се извршење сценарија (ИА)



*Слика 60 Грешка приликом враћања налога*

- 8.2. Уколико систем не може да сачува податке о налогу, приказује кориснику поруку: „Није могуће сачувати податке о налогу!“ (ИА)



*Слика 61 Грешка приликом измене података о налогу*

## СК4: Случај коришћења – Унос уређаја

### Назив СК

Унос уређаја

### Актер СК

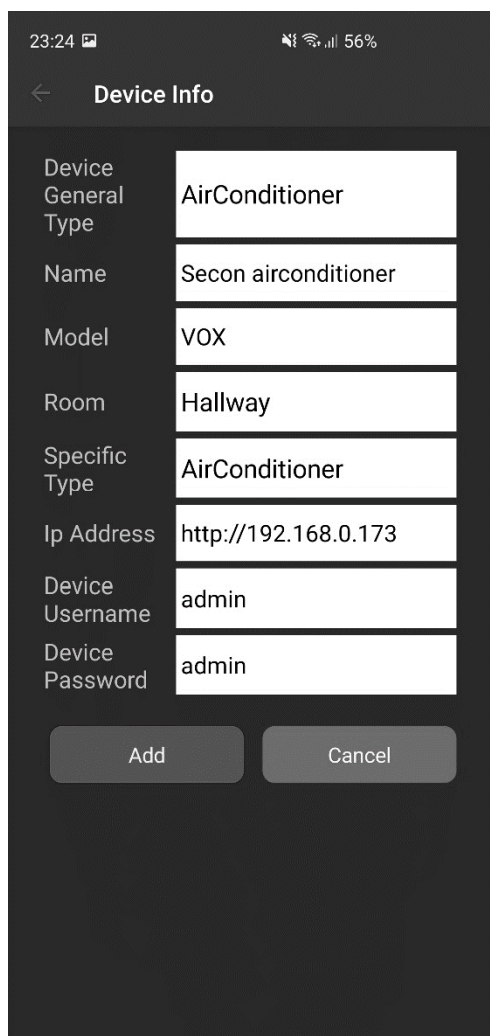
Корисник (Администратор или Власник)

### Учесници СК

Корисник и систем (програм)

### Предуслов

Систем је укључен и корисник је пријављен на своја налог. Систем приказује страницу за унос новог уређаја.



Device General Type	AirConditioner
Name	Secon airconditioner
Model	VOX
Room	Hallway
Specific Type	AirConditioner
Ip Address	http://192.168.0.173
Device Username	admin
Device Password	admin

23:24 56%

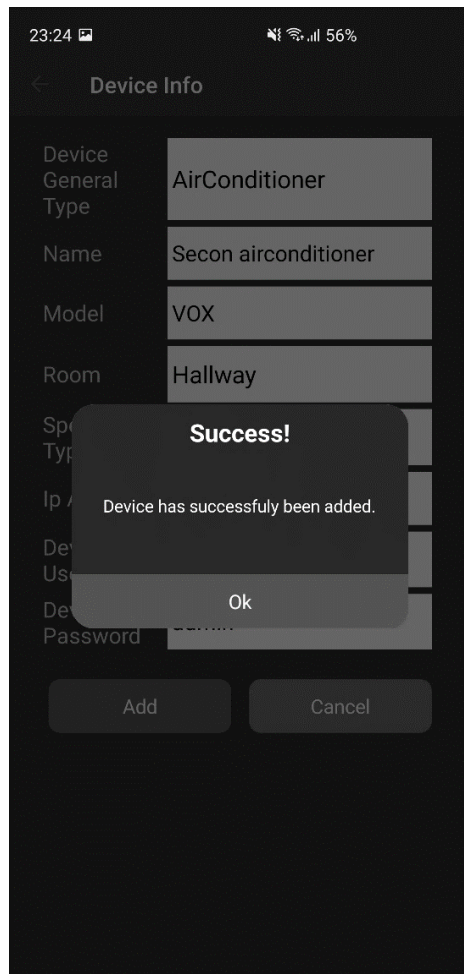
← Device Info

Add Cancel

Слика 62 Страница за унос уређаја

## Основни сценарио

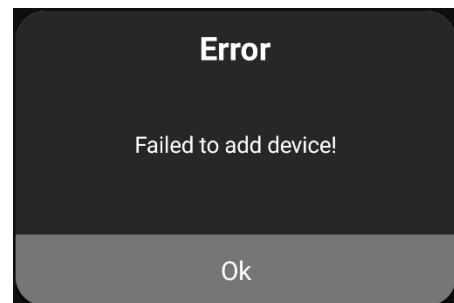
1. Корисник уноси податке о новом уређају (АПУСО)
2. Корисник проверава да ли је исправно унео податке (АНСО)
3. Корисник позива систем да запамти податке о новом уређају (АПСО)
4. Систем памти податке о уређају (СО)
5. Систем приказује кориснику поруку: „Уређај успешно сачуван.“ (ИА)



Слика 64 Успешан унос уређаја

## Алтернативна сценарија

- 5.1. Уколико систем не може да сачува податке о уређају, приказује кориснику поруку: „Неуспешно додавање уређаја!“ (ИА)



Слика 63 Грешка приликом уноса уређаја

## СК5: Случај коришћења – Измена података о уређају

### Назив СК

Измена података о уређају

### Актер СК

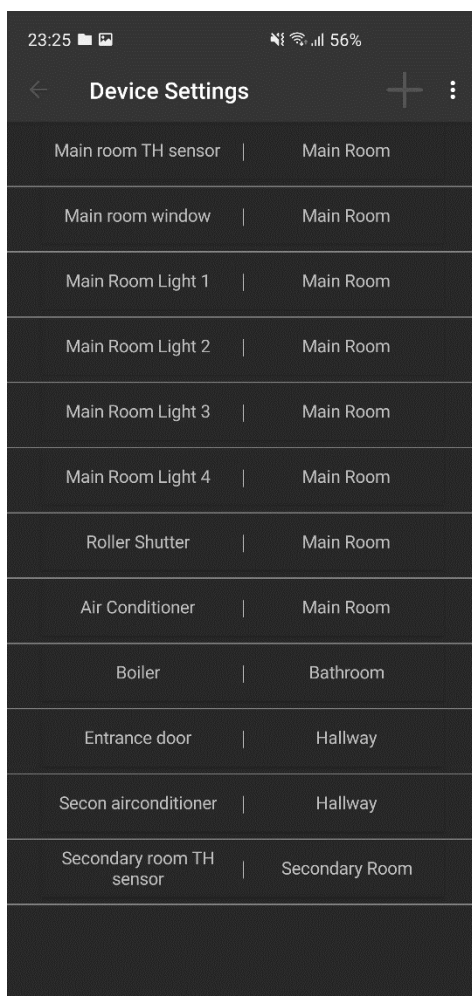
Корисник (Администратор или Власник)

### Учесници СК

Корисник и систем (програм)

### Предуслов

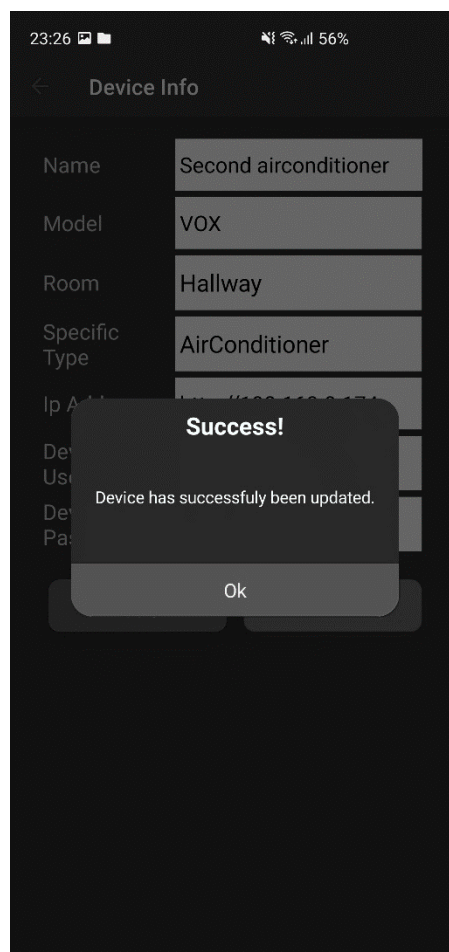
Систем је укључен и корисник је пријављен на своја налог. Систем приказује страницу за управљање уређајима.



Слика 65 Страница за управљање уређајима

## Основни сценарио

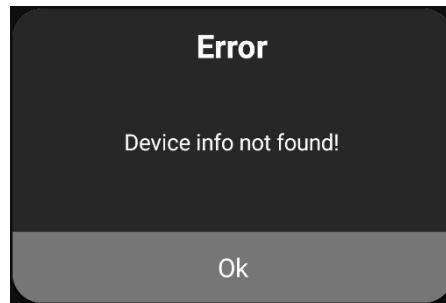
1. Корисник позива систем да прикаже листу уређаја (АПСО)
2. Систем проналази све постојеће уређаје (СО)
3. Систем приказује листу свих уређаја (ИА)
4. Корисник одабира уређај чије податке жели да измени (АПУСО)
5. Корисник позива систем да прикаже податке о изабраном уређају (АПСО)
6. Систем учитава податке о изабраном уређају (СО)
7. Систем приказује кориснику податке о изабраном уређају (ИА)
8. Корисник мења податке о уређају (АПУСО)
9. Корисник проверава да ли је добро унео податке (АНСО)
10. Корисник позива систем да запамти унете податке (АПСО)
11. Систем памти измењене податке о уређају (СО)
12. Систем приказује кориснику поруку: „Подаци о уређају су успешно измењени.“ (ИА)



Слика 66 Подаци о уређају су измењени

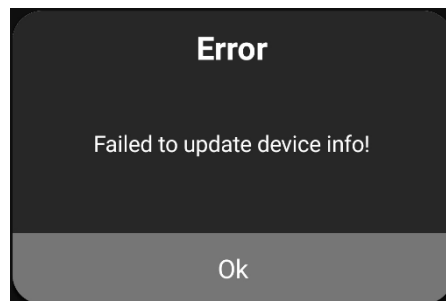
## Алтернативна сценарија

- 7.2. Уколико систем не може да пронађе податке о изабраном уређају, приказује кориснику поруку: „Подаци о уређају нису пронађени!“. Прекида се извршење сценарија (ИА)



*Слика 67 Грешка при враћању уређаја*

- 12.1. Уколико систем не може да сачува измењене податке, приказује кориснику поруку: „Неуспешна измена података о уређају!“ (ИА)



*Слика 68 Грешка при измени података о уређају*

## СК6: Случај коришћења – Брисање уређаја

### Назив СК

Брисање уређаја

### Актер СК

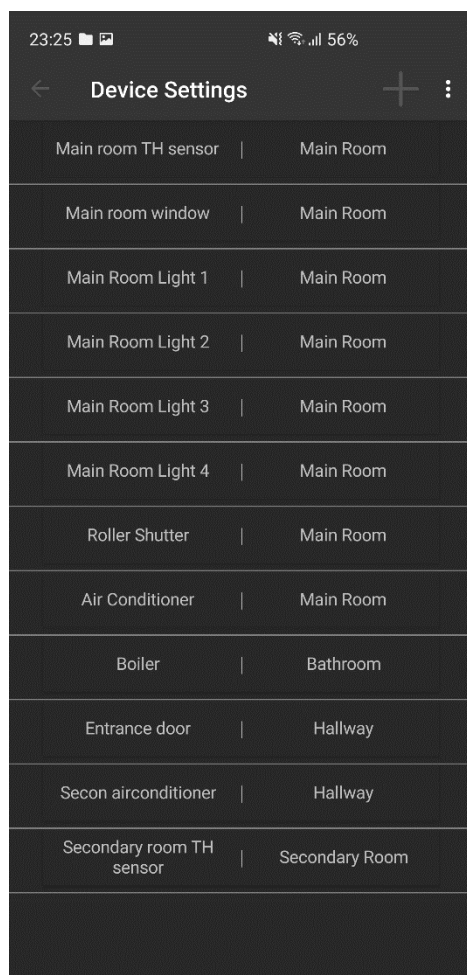
Корисник (Администратор или Власник)

### Учесници СК

Корисник и систем (програм)

### Предуслов

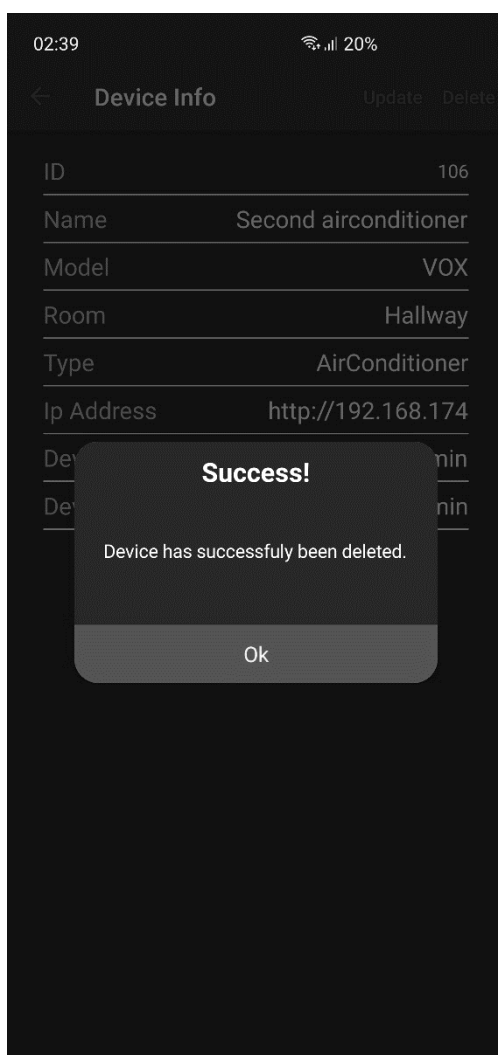
Систем је укључен и корисник је пријављен на своја налог. Систем приказује страницу за управљање уређајима.



Слика 69 Страница за управљање уређајима

## Основни сценарио

1. Корисник позива систем да прикаже листу уређаја (АПСО)
2. Систем проналази све постојеће уређаје (СО)
3. Систем приказује листу свих уређаја (ИА)
4. Корисник одабира уређај који жели да обрише (АПУСО)
5. Корисник позива систем да прикаже податке о изабраном уређају (АПСО)
6. Систем учитава податке о изабраном уређају (СО)
7. Систем приказује кориснику податке о изабраном уређају (ИА)
8. Корисник позива систем да обрише изабрани уређај (АПСО)
9. Систем врши брисање уређаја (СО)
10. Систем приказује кориснику поруку: „Уређај успешно обрисан.“ (ИА)

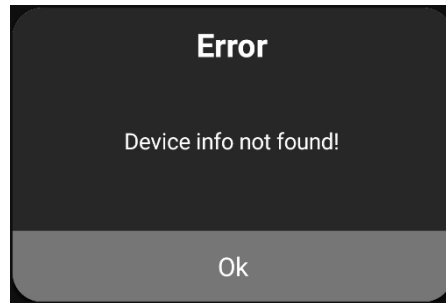


Слика 70 Успешно брисање уређаја



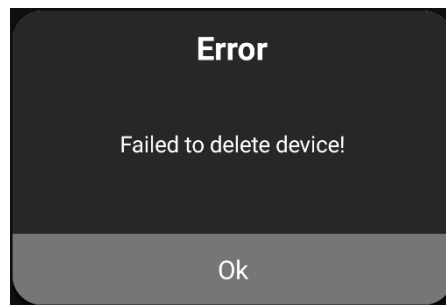
## Алтернативна сценарија

- 7.1. Уколико систем не може да пронађе податке о изабраном уређају, приказује кориснику поруку: „Подаци о уређају нису пронађени!“. Прекида се извршење сценарија (ИА)



Слика 71 Грешка при враћању уређаја

- 10.1. Уколико систем не може да обрише уређај, приказује кориснику поруку: „Неуспешно брисање уређаја!“ (ИА)



Слика 72 Грешка при брисању уређаја

## СК7: Случај коришћења – Унос собе

### Назив СК

Унос собе

### Актер СК

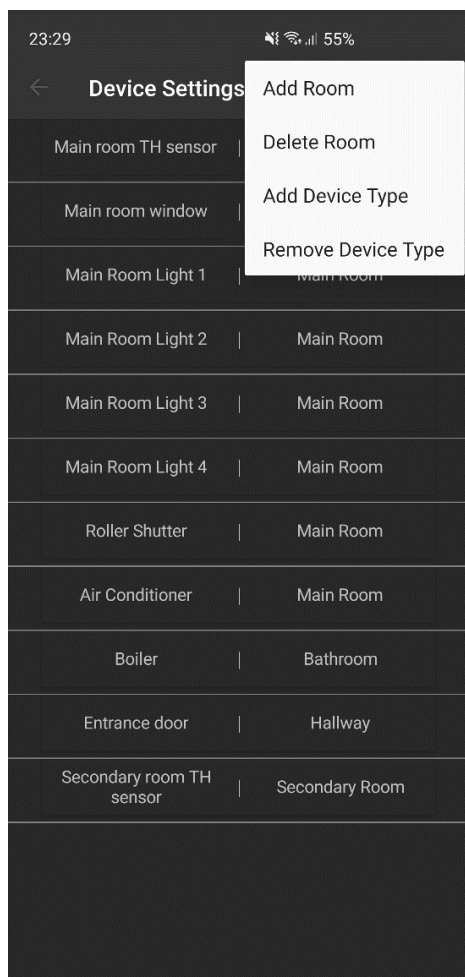
Корисник (Администратор или Власник)

### Учесници СК

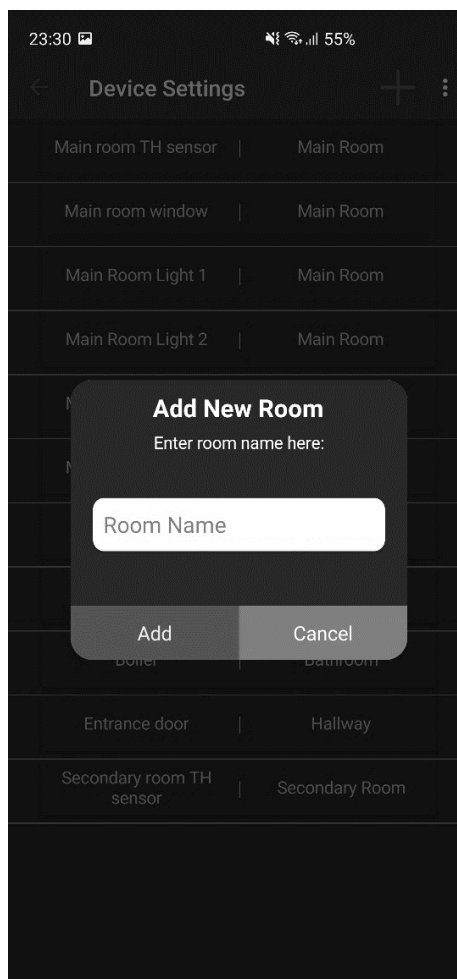
Корисник и систем (програм)

### Предуслов

Систем је укључен и корисник је пријављен на своја налог. Систем приказује прозор за унос собе.



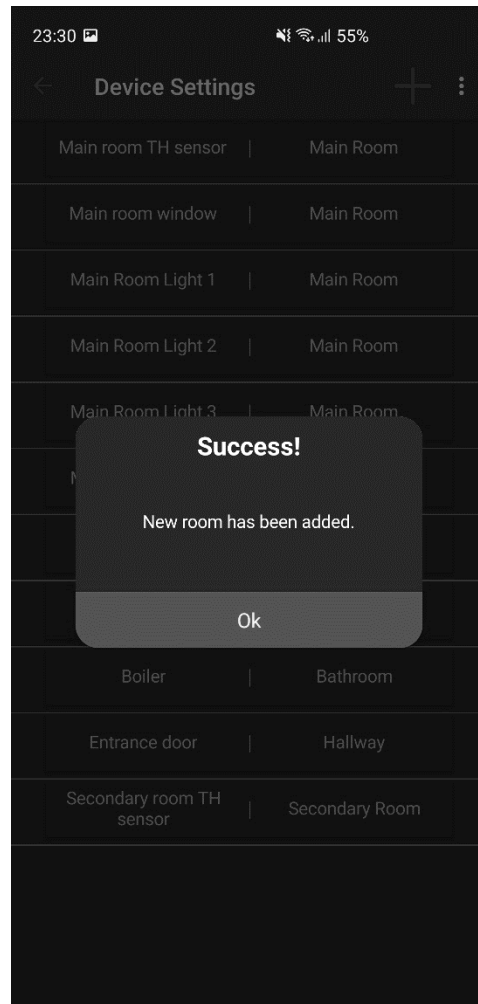
Слика 73 Страница за управљање уређајима



Слика 74 Прозор за унос имена нове собе

## Основни сценарио

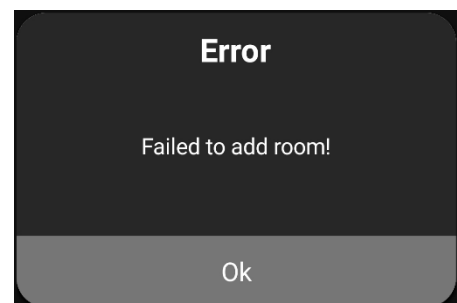
1. Корисник уноси назив нове собе (АПУСО)
2. Корисник проверава да ли је добро унео назив (АНСО)
3. Корисник позива систем да запамти податке о новој соби (АПСО)
4. Систем памти податке о соби (СО)
5. Систем приказује кориснику поруку: „Соба успешно унета.“ (ИА)



Слика 75 Успешан унос собе

## Алтернативна сценарија

- 5.2. Уколико систем не може да сачува податке о новој соби, приказује кориснику поруку: „Неуспешан унос собе!“ (ИА)



Слика 76 Грешка при уносу собе

## СК8: Случај коришћења – Брисање собе

### Назив СК

Брисање собе

### Актер СК

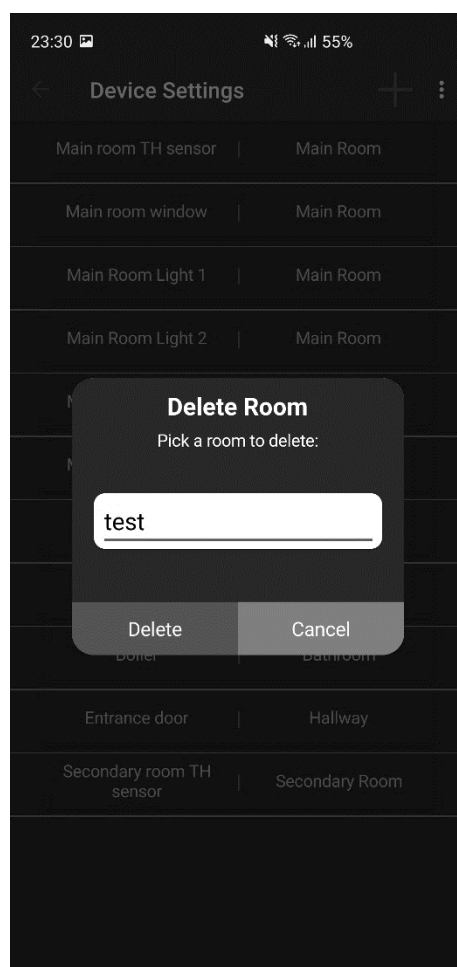
Корисник (Администратор или Власник)

### Учесници СК

Корисник и систем (програм)

### Предуслов

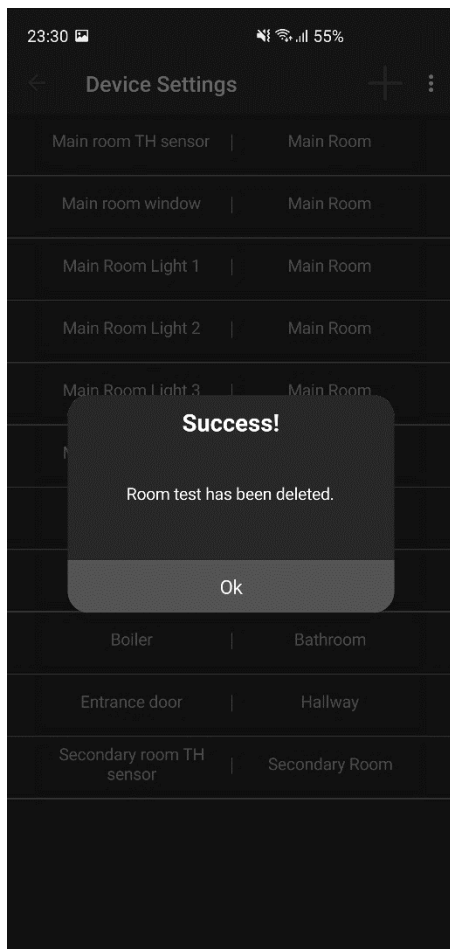
Систем је укључен и корисник је пријављен на своја налог. Систем приказује прозор за брисање собе.



Слика 77 Прозор за брисање собе

## Основни сценарио

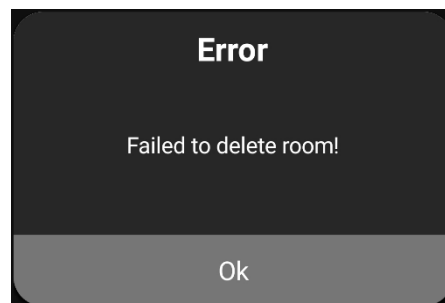
1. Корисник позива систем да прикаже листу соба (АПСО)
2. Систем проналази све постојеће собе (СО)
3. Систем приказује листу свих соба (ИА)
4. Корисник одабира собу коју жели да обрише (АПУСО)
5. Корисник позива систем да обрише изабрану собу (АПСО)
6. Систем врши брисање собе (СО)
7. Систем приказује кориснику поруку: „Соба успешно обрисана.“ (ИА)



Слика 78 Успешно брисање собе

## Алтернативна сценарија

- 7.1. Уколико систем не може да обрише собу, приказује кориснику поруку: „Неуспешно брисање собе!“ (ИА)



Слика 79 Грешка при брисању собе

## СК9: Случај коришћења – Унос типа уређаја

### Назив СК

Унос типа уређаја

### Актер СК

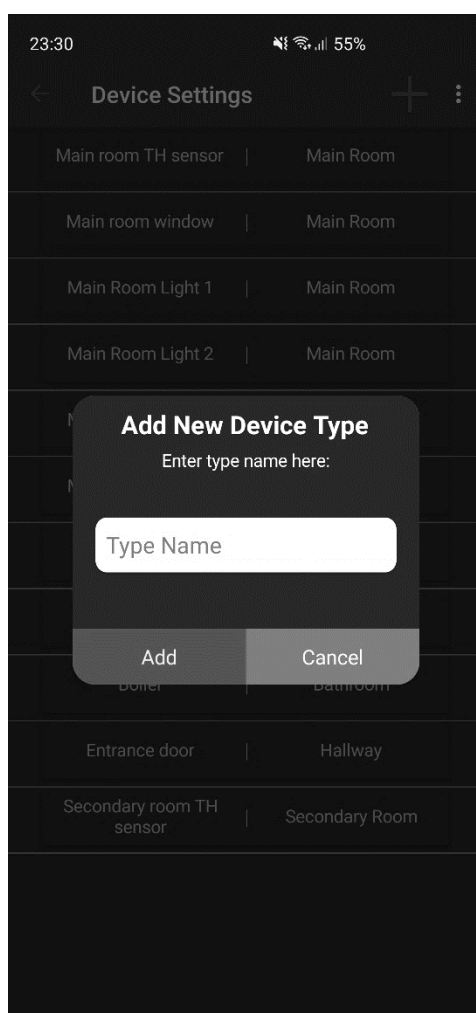
Корисник (Администратор или Власник)

### Учесници СК

Корисник и систем (програм)

### Предуслов

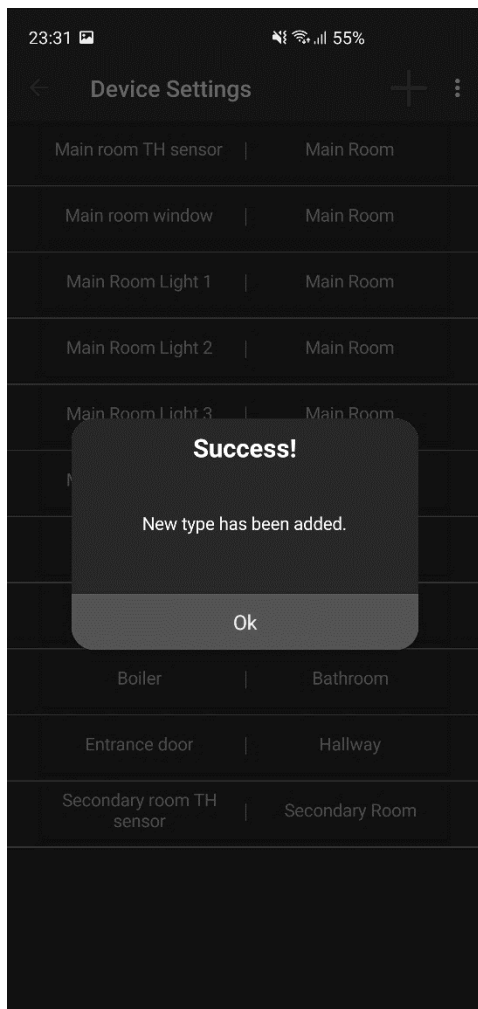
Систем је укључен и корисник је пријављен на своја налог. Систем приказује прозор за унос типа уређаја.



Слика 80 Прозор за унос типа уређаја

## Основни сценарио

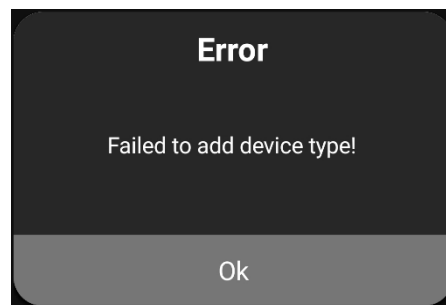
1. Корисник уноси назив новог типа уређаја (АПУСО)
2. Корисник проверава да ли је добро унео назив (АНСО)
3. Корисник позива систем да запами податке о новом типу уређаја (АПСО)
4. Систем памти податке о типу уређаја (СО)
5. Систем приказује кориснику поруку: „Тип уређаја успешно унет.“



Слика 81 Успешан унос типа уређаја

## Алтернативна сценарија

- 5.1. Уколико систем не може да сачува податке о новом типу уређаја, приказује кориснику поруку: „Неуспешан унос типа уређаја!“ (ИА)



Слика 82 Грешка при уносу типа

## СК10: Случај коришћења – Брисање типа уређаја

### Назив СК

Брисање типа уређаја

### Актер СК

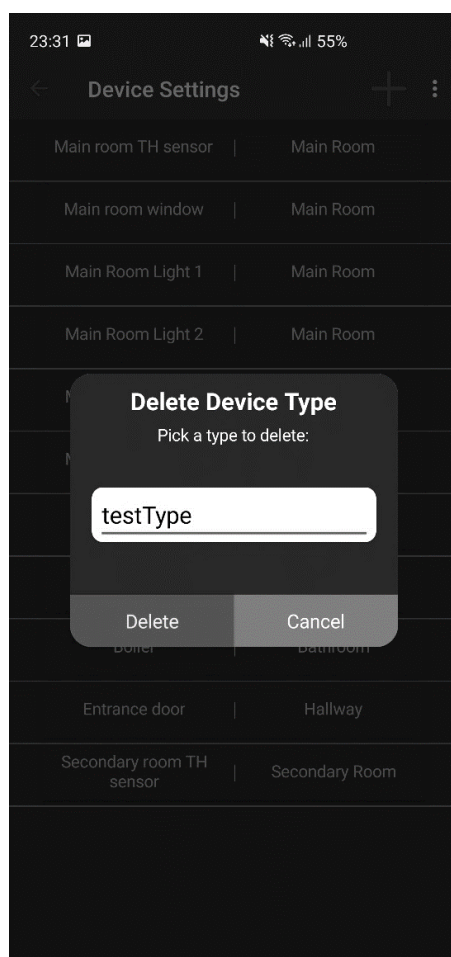
Корисник (Администратор или Власник)

### Учесници СК

Корисник и систем (програм)

### Предуслов

Систем је укључен и корисник је пријављен на своја налог. Систем приказује прозор за брисање типа уређаја.

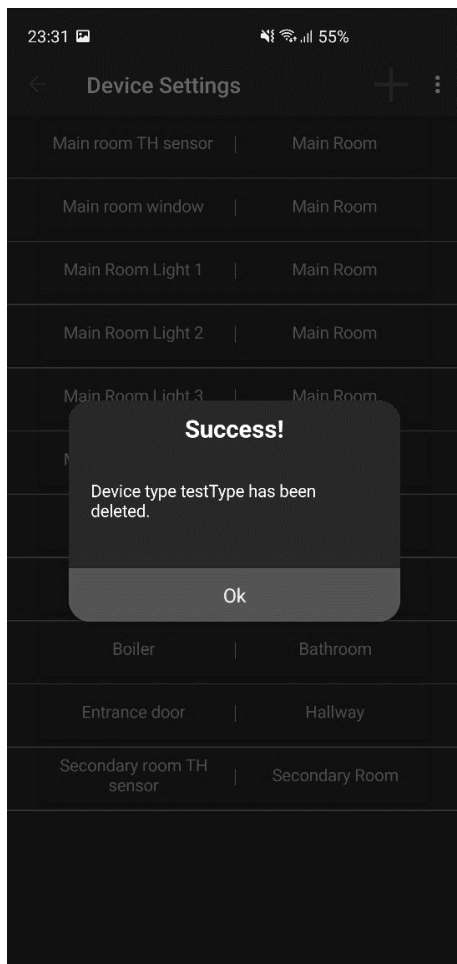


Слика 83 Прозор за брисање типа уређаја



## Основни сценарио

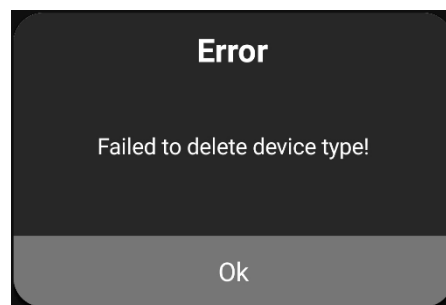
1. Корисник позива систем да прикаже листу типова уређаја (АПСО)
2. Систем проналази све постојеће типове уређаја (СО)
3. Систем приказује листу свих типова уређаја (ИА)
4. Корисник одабира тип уређаја који жели да обрише (АПУСО)
5. Корисник позива систем да обрише изабрани тип уређаја (АПСО)
6. Систем врши брисање типа уређаја (СО)
7. Систем приказује кориснику поруку: „Тип уређаја успешно обрисан.“ (ИА)



Слика 84 Успешно брисање типа уређаја

## Алтернативна сценарија

- 7.1. Уколико систем не може да обрише тип уређаја, приказује кориснику поруку: „Неуспешно брисање типа уређаја!“ (ИА)



Слика 85 Грешка при брисању типа

## СК11: Случај коришћења – Измена података о налогу корисника

### Назив СК

Измена података о уређају

### Актер СК

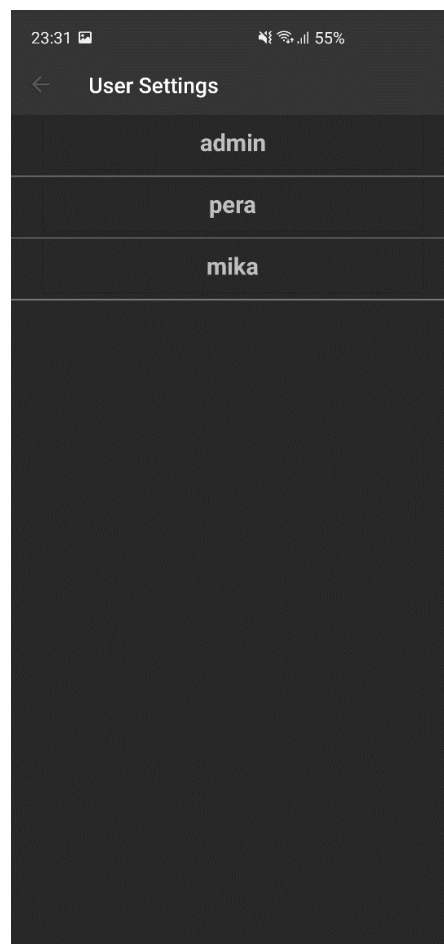
Власник

### Учесници СК

Власник и систем (програм)

### Предуслов

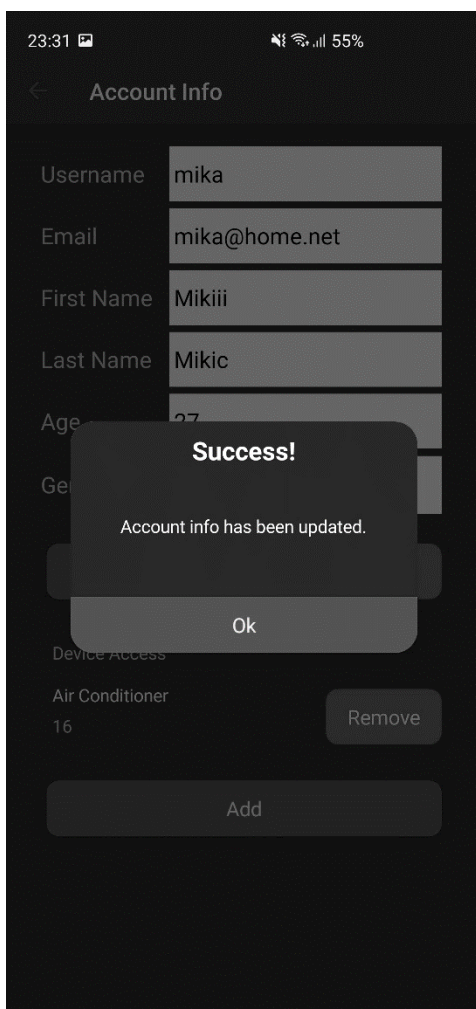
Систем је укључен и власник је пријављен на своја налог. Систем приказује страницу за управљање налозима.



Слика 86 Страница за управљање налозима

## Основни сценарио

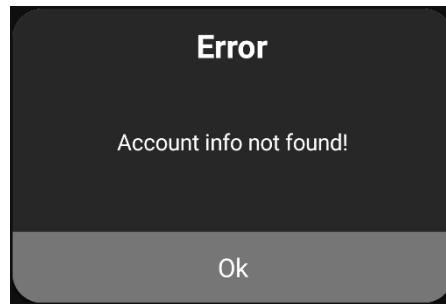
1. Власник позива систем да прикаже листу налога (АПСО)
2. Систем проналази све постојеће налоге (СО)
3. Систем приказује листу свих налога (ИА)
4. Власник одабира налог чије податке жели да измени (АПУСО)
5. Власник позива систем да прикаже податке о изабраном налогу (АПСО)
6. Систем учитава податке о изабраном налогу (СО)
7. Систем приказује власнику податке о изабраном налогу (ИА)
8. Власник мења податке о налогу (АПУСО)
9. Власник проверава да ли је добро унео податке (АНСО)
10. Власник позива систем да запамти унете податке (АПСО)
11. Систем памти измењене податке о налогу (СО)
12. Систем приказује власнику поруку: „Подаци о налогу су успешно измењени.“ (ИА)



Слика 87 Успешна измена података о налогу

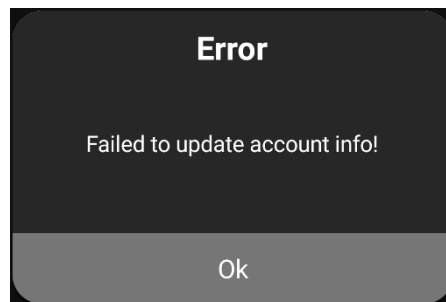
## Алтернативна сценарија

- 7.2. Уколико систем не може да пронађе податке о изабраном налогу, приказује власнику поруку: „Подаци о налогу нису пронађени!“. Прекида се извршење сценарија (ИА)



*Слика 88 Грешка при враћању налога*

- 12.1. Уколико систем не може да сачува измењене податке, приказује власнику поруку: „Неуспешна измена података о налогу!“ (ИА)



*Слика 89 Грешка при измени података о налогу*

## СК12: Случај коришћења – Додавање дозволе за коришћење уређаја

### Назив СК

Додавање дозволе за коришћење уређаја

### Актер СК

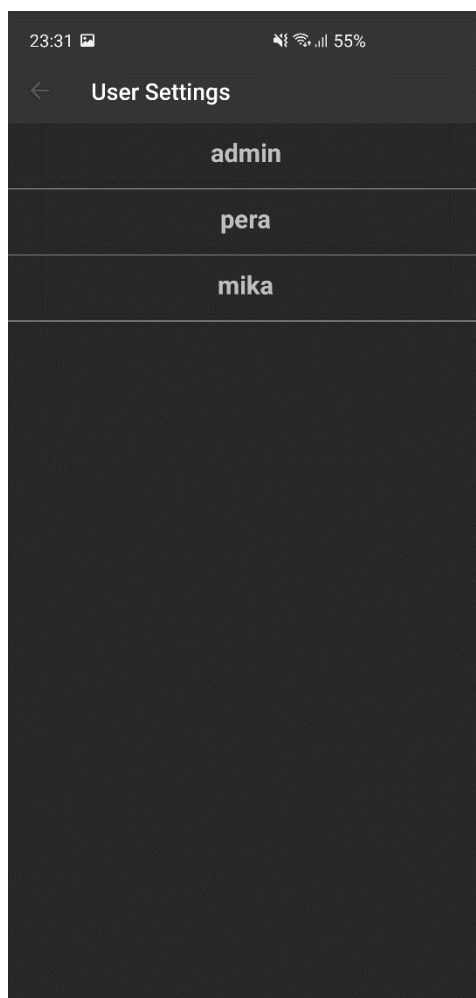
Власник

### Учесници СК

Власник и систем (програм)

### Предуслов

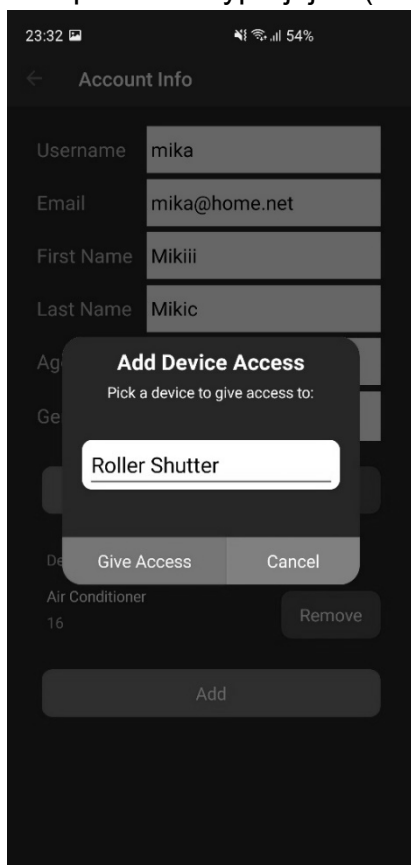
Систем је укључен и власник је пријављен на своја налог. Систем приказује страницу за управљање налозима.



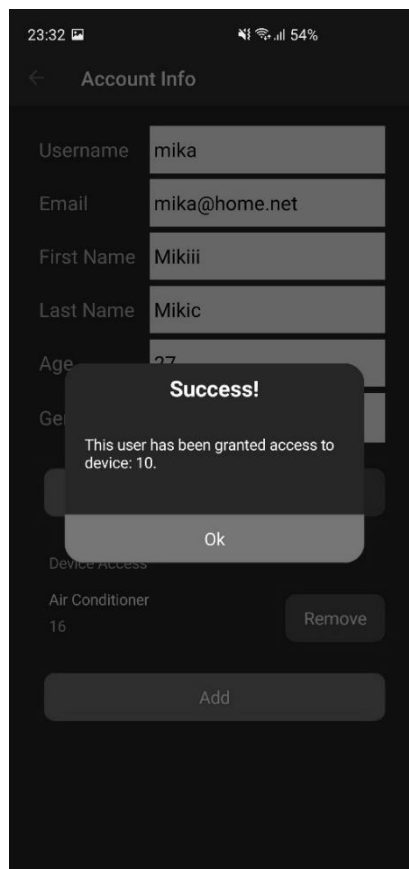
Слика 90 Страница за управљање корисницима

## Основни сценарио

1. Власник позива систем да прикаже листу налога (АПСО)
2. Систем проналази све постојеће налоге (СО)
3. Систем приказује листу свих налога (ИА)
4. Власник одабира налог чије коме жели да дода дозволу за коришћење уређаја (АПУСО)
5. Власник позива систем да врати податке о изабраном кориснику (АПСО)
6. Систем проналази изабраног корисника (СО)
7. Систем приказује податке о изабраном кориснику (ИА)
8. Власник позива систем да прикаже листу уређаја (АПСО)
9. Систем проналази све постојеће уређаје (СО)
10. Систем приказује листу свих уређаја (ИА)
11. Власник одабира уређај за који даје дозволу коришћења (АПУСО)
12. Власник позива систем да запамти податке о дозволи коришћења изабраног уређаја за претходно изабраног корисника (АПСО)
13. Систем памти дозволу коришћења уређаја (СО)
14. Систем приказује власнику поруку: „Успешно додавање дозволе коришћења уређаја.“ (ИА)



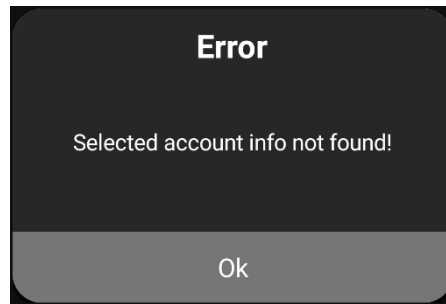
Слика 91 Прозор за одабир уређаја



Слика 92 Успешан унос дозволе

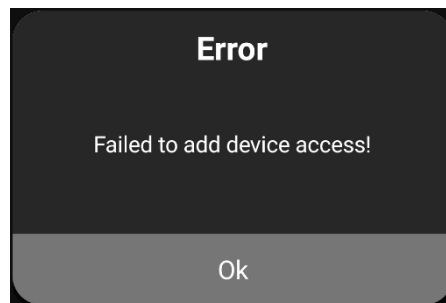
## Алтернативна сценарија

- 7.2. Уколико систем не може да пронађе корисника, приказује власнику поруку: „Подаци о изабраном кориснику нису пронађени!“. Прекида се извршење сценарија (ИА)



*Слика 93 Грешка при враћању налога*

- 14.1. Уколико систем не може да сачува дозволу коришћења, приказује власнику поруку: „Неуспешна додавање дозволе!“ (ИА)



*Слика 94 Грешка при додавању дозволе*

## СК13: Случај коришћења – Брисање дозволе за коришћење уређаја

### Назив СК

Брисање дозволе за коришћење уређаја

### Актер СК

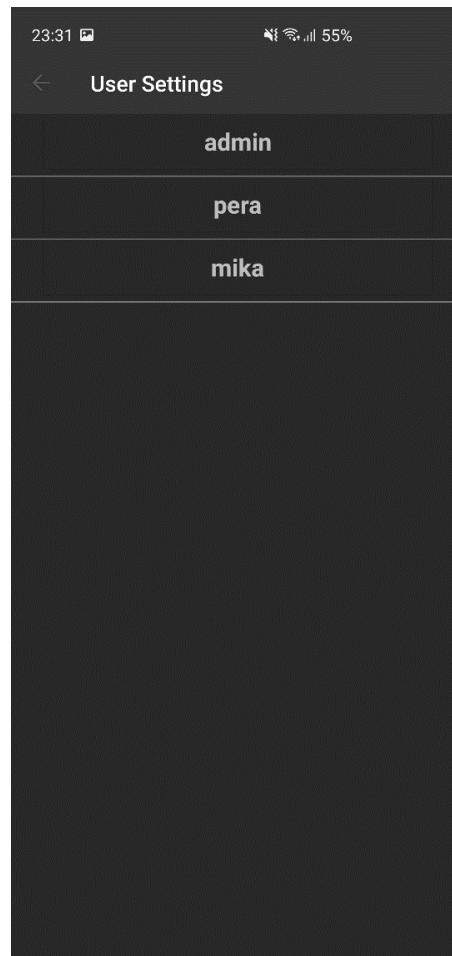
Власник

### Учесници СК

Власник и систем (програм)

### Предуслов

Систем је укључен и власник је пријављен на своја налог. Систем приказује страницу за управљање налозима.

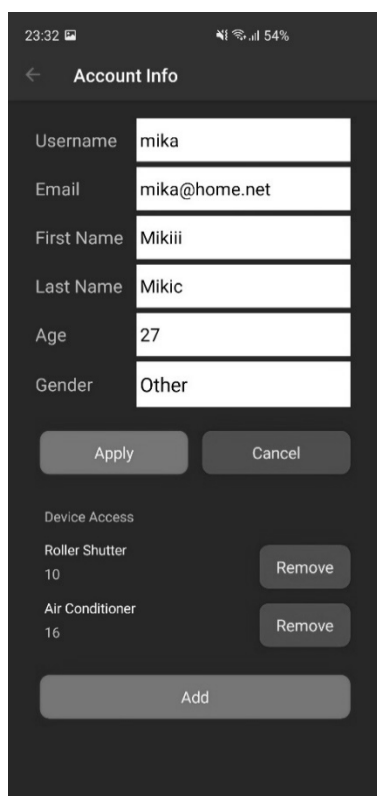


Слика 95 Страница за управљање налозима

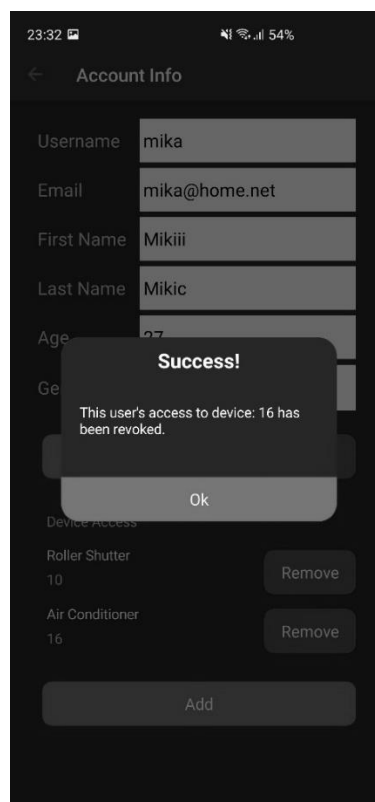


## Основни сценарио

1. Власник позива систем да прикаже листу налога (АПСО)
2. Систем проналази све постојеће налоге (СО)
3. Систем приказује листу свих налога (ИА)
4. Власник одабира налог чије коме жели да обрише дозволу за коришћење уређаја (АПУСО)
5. Власник позива систем да врати податке о изабраном кориснику (АПСО)
6. Систем проналази изабраног корисника (СО)
7. Систем приказује податке о изабраном кориснику (ИА)
8. Власник позива систем да прикаже листу уређаја којима изабрани корисник има приступ (АПСО)
9. Систем проналази све постојеће уређаје који одговарају критеријуму (СО)
10. Систем приказује листу свих уређаја који одговарају критеријуму (ИА)
11. Власник одабира уређај за који брише дозволу коришћења (АПУСО)
12. Власник позива систем да обрише дозволу коришћења изабраног уређаја за претходно изабраног корисника (АПСО)
13. Систем брише дозволу коришћења уређаја (СО)
14. Систем приказује власнику поруку: „Успешно брисање дозволе коришћења уређаја.“ (ИА)



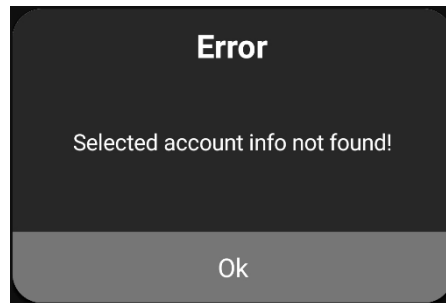
Слика 96 Страница за управљање подацима о налогу



Слика 97 Успешно брисање дозволе

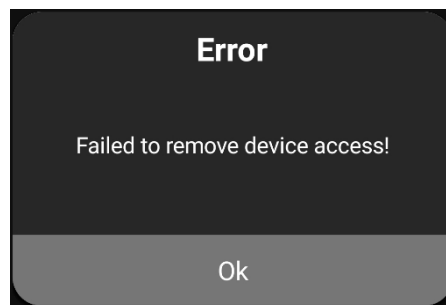
## Алтернативна сценарија

- 7.2. Уколико систем не може да пронађе корисника, приказује власнику поруку: „Подаци о изабраном кориснику нису пронађени!“. Прекида се извршење сценарија (ИА)



Слика 98 Грешка при враћању налога

- 14.1. Уколико систем не може да обрише дозволу коришћења, приказује власнику поруку: „Неуспешно брисање дозволе!“ (ИА)



Слика 99 Грешка при брисању дозволе

### 6.3.2 Пословна логика

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 [1].

У наставку дати су дијаграми секвенци за сваки од идентификованих уговора који описују комуникацију контролера, апликационе логике и слоја приступа подацима.

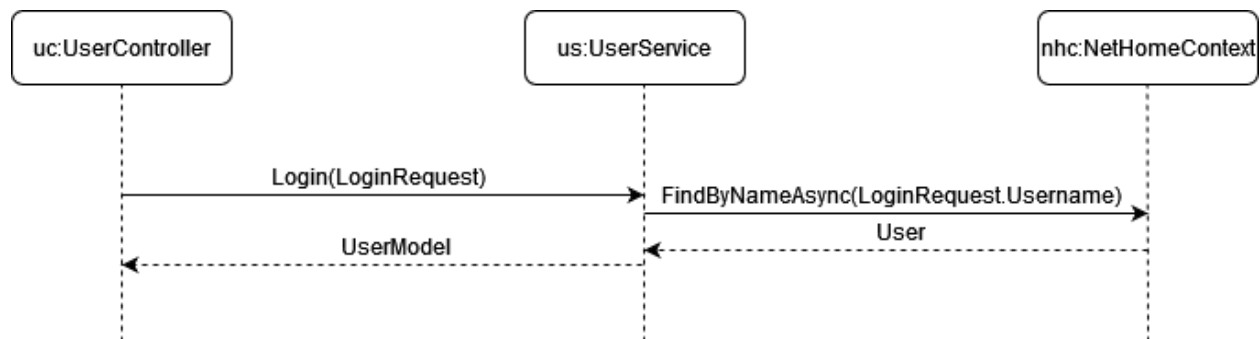
#### Уговор УГ1: ПријавиКорисника

Операција: Login(LoginRequest)

Веза са СК: СК1

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

Постуслови: Корисник је пријављен.



Слика 100 ДС Пријава корисника

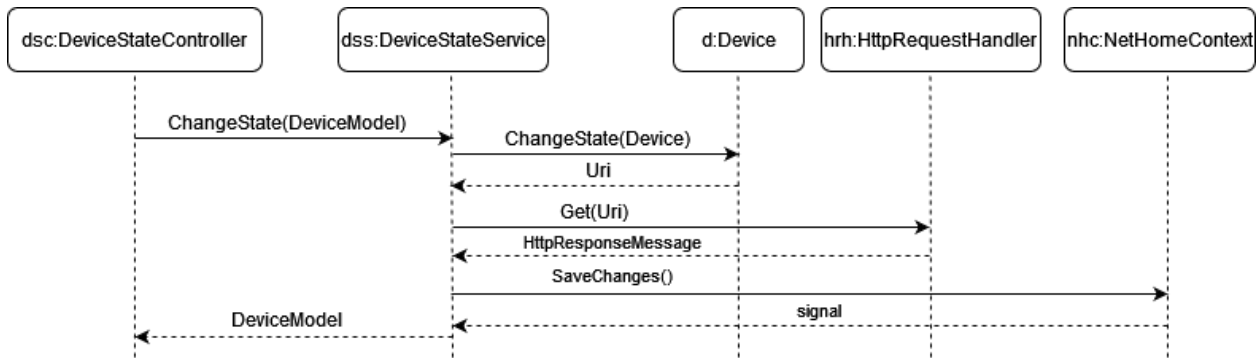
#### Уговор УГ2: ИзмениСтањеУређаја

Операција: ChangeDeviceState(DeviceModel)

Веза са СК: СК2

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

Постуслови: Стање уређаја је промењено.



Слика 101 ДС Измена стања уређаја

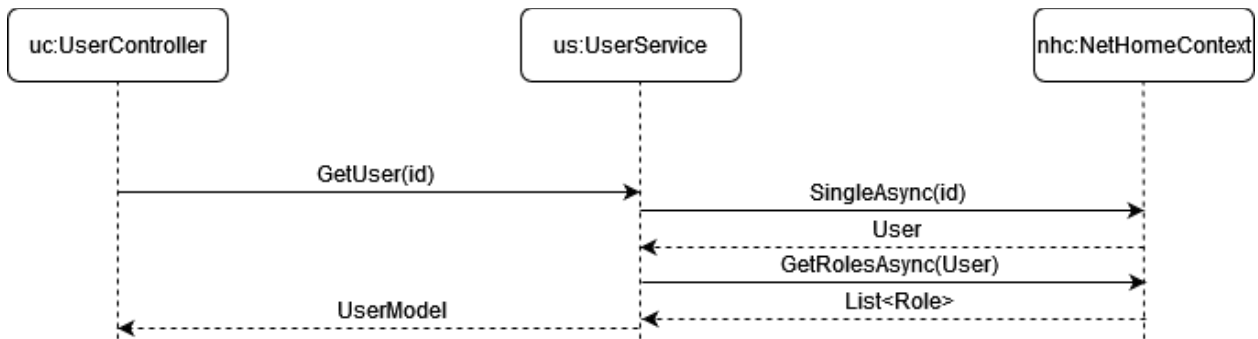
### Уговор УГ3: ВратиПодаткеОНалогу

Операција: GetUserData(UserModel)

Веза са СК: СК3

Предуслови:

Постуслови:



Слика 102 ДС Врати податке о налогу

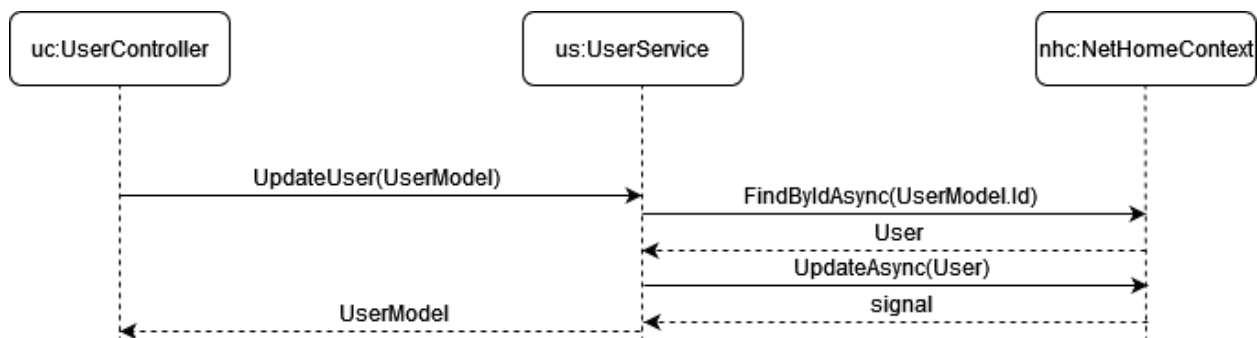
### Уговор УГ4: ИзмениКорисника

Операција: UpdateUser(UserModel)

Веза са СК: СК3, СК11

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

Постуслови: Подаци о кориснику су измењени.



Слика 103 ДС Измена корисника

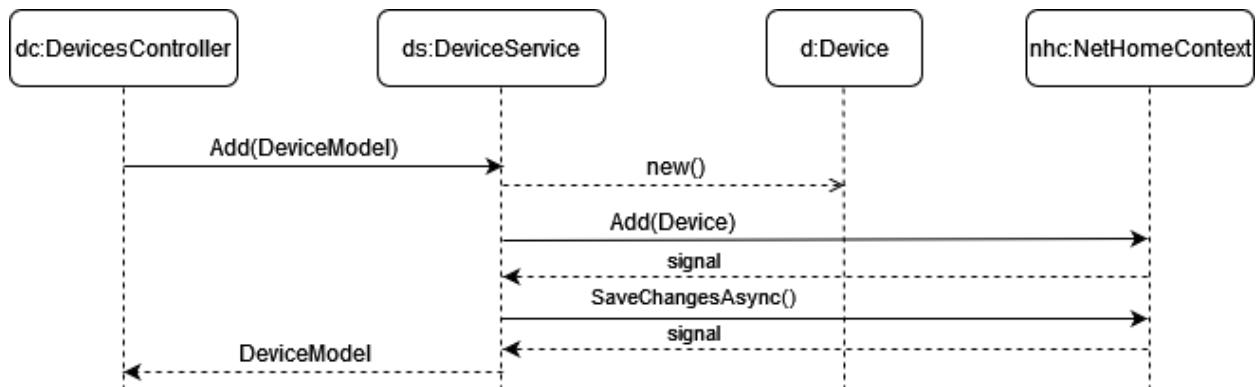
### Уговор УГ5: ДодајУређај

Операција: AddDevice(DeviceModel)

Веза са СК: СК4

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

Постуслови: Подаци о уређају су сачувани.



Слика 104 ДС Додавање уређаја

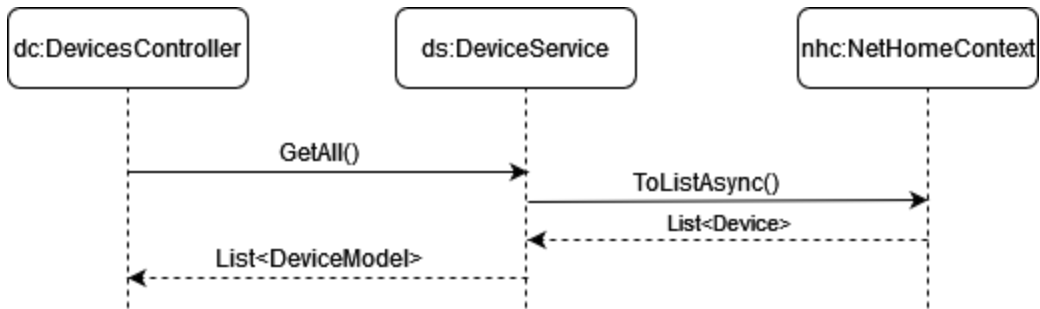
### Уговор УГ6: ВратиСвеУређаје

Операција: GetAllDevices(List<DeviceModel>)

Веза са СК: СК5, СК6

Предуслови: /

Постуслови: /



Слика 105 ДС Врати све уређаје

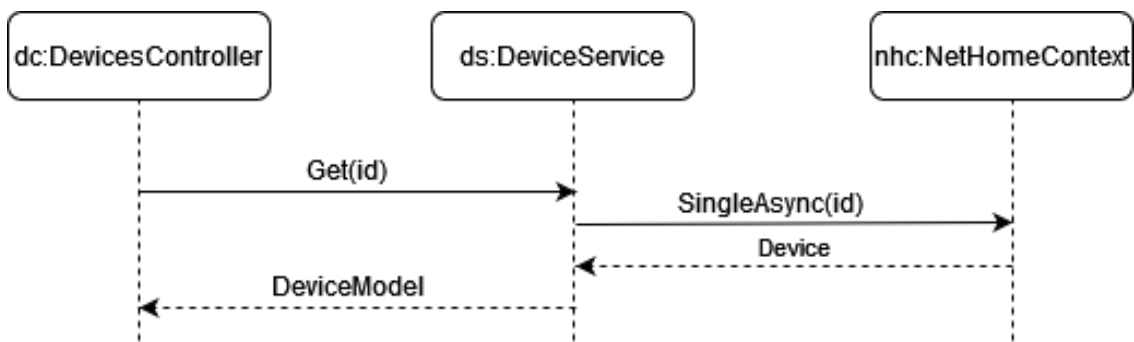
### Уговор УГ7: ВратиУређај

Операција: GetDevice(id, DeviceModel)

Веза са СК: СК5, СК6

Предуслови: /

Постуслови: /



Слика 106 ДС Врати уређај

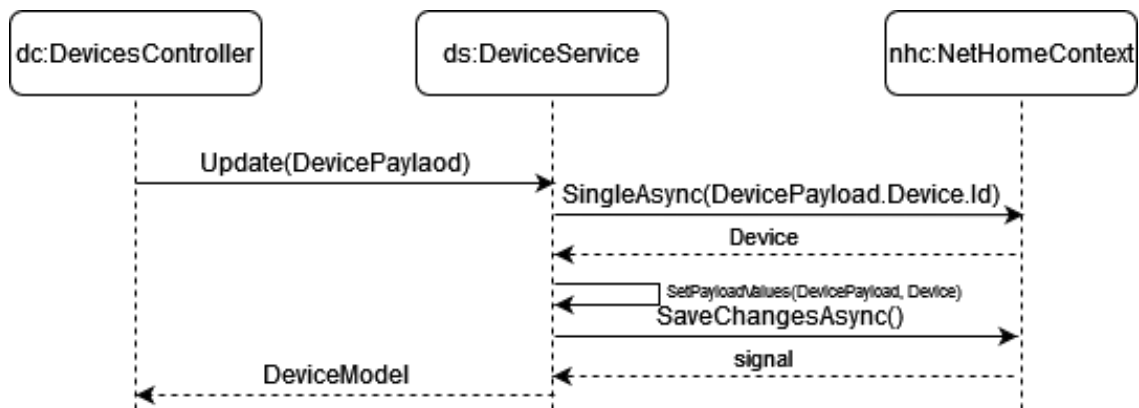
### Уговор УГ8: ИзмениУређај

Операција: UpdateDevice(DeviceModel)

Веза са СК: СК5

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

Постуслови: Подаци о уређају су измењени.



Слика 107 ДС Измена уређаја

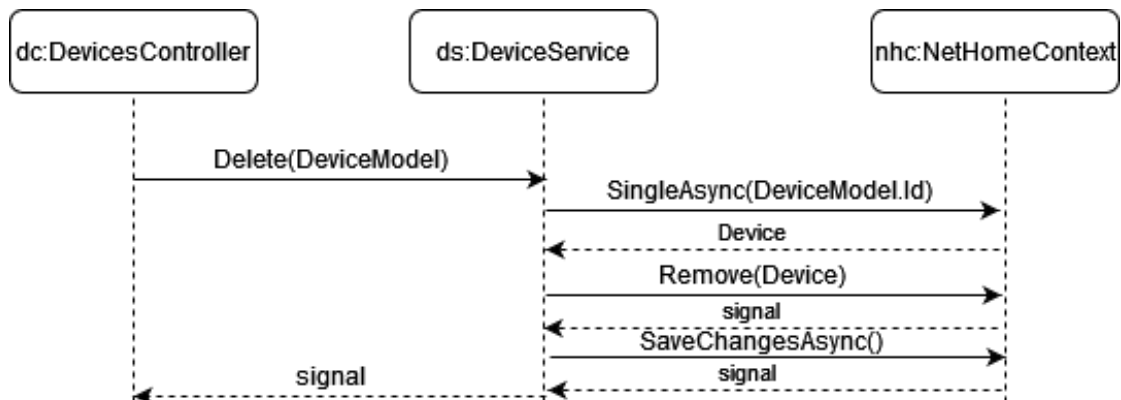
### Уговор УГ9: ОбришиУређај

Операција: DeleteDevice(DeviceModel)

Веза са СК: СК6

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

Постуслови: Уређај је обрисан.



Слика 108 ДС Брисање уређаја

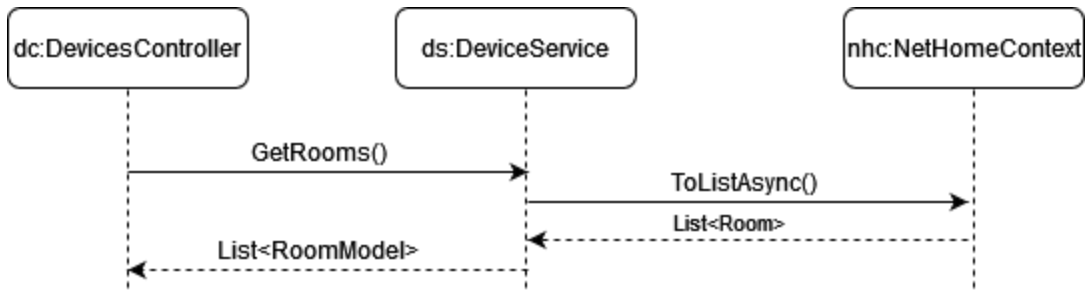
### Уговор УГ10: ВратиСвеСобе

Операција: GetAllRooms(List<RoomModel>)

Веза са СК: СК8

Предуслови: /

Постуслови: /



Слика 109 ДС Врати све собе

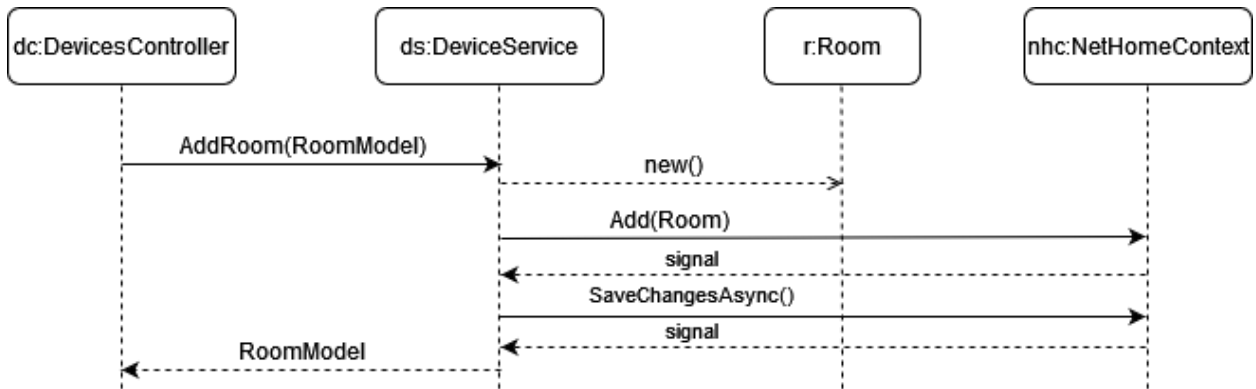
### Уговор УГ11: ДодајСобу

Операција: AddRoom(RoomModel)

Веза са СК: СК7

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

Постуслови: Подаци о соби су сачувани.



Слика 110 ДС Додавање собе

### Уговор УГ12: ОбришиСобу

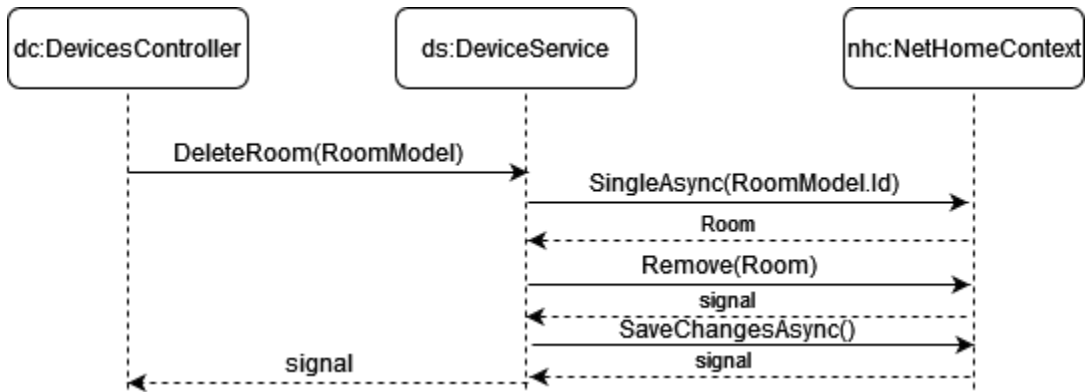
Операција: DeleteRoom(RoomModel)

Веза са СК: СК8

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

Постуслови: Соба је обрисана.





Слика 111 ДС Брисање собе

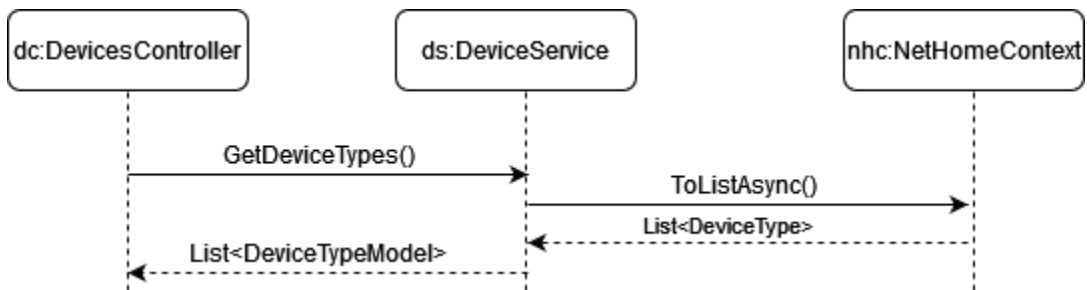
### Уговор УГ13: ВратиСвеТиповеУређаја

Операција: GetAllDeviceTypes(List<DeviceTypeModel>)

Веза са СК: СК10

Предуслови: /

Постуслови: /



Слика 112 ДС Врати све типове уређаја

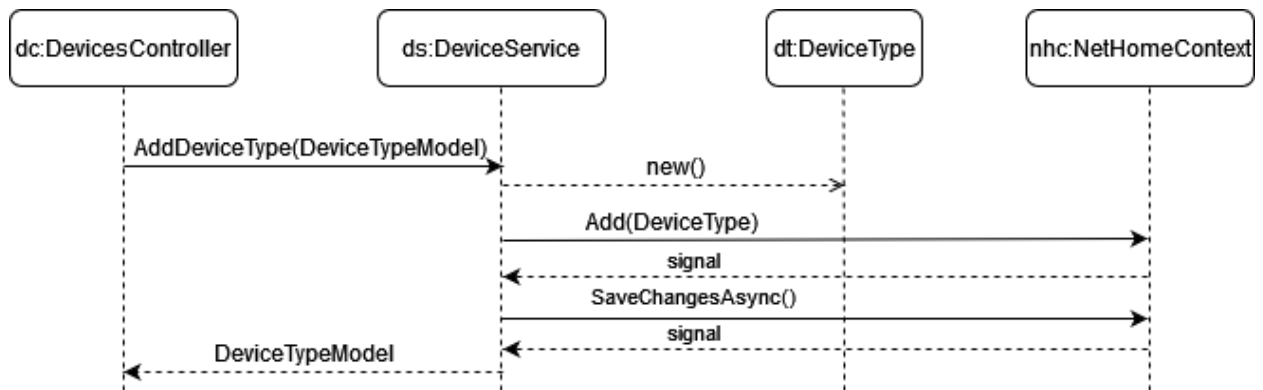
### Уговор УГ14: ДодајТипУређаја

Операција: AddDeviceType(DeviceTypeModel)

Веза са СК: СК9

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

Постуслови: Подаци о типу уређаја су сачувани.



Слика 113 ДС Додавање типа уређаја

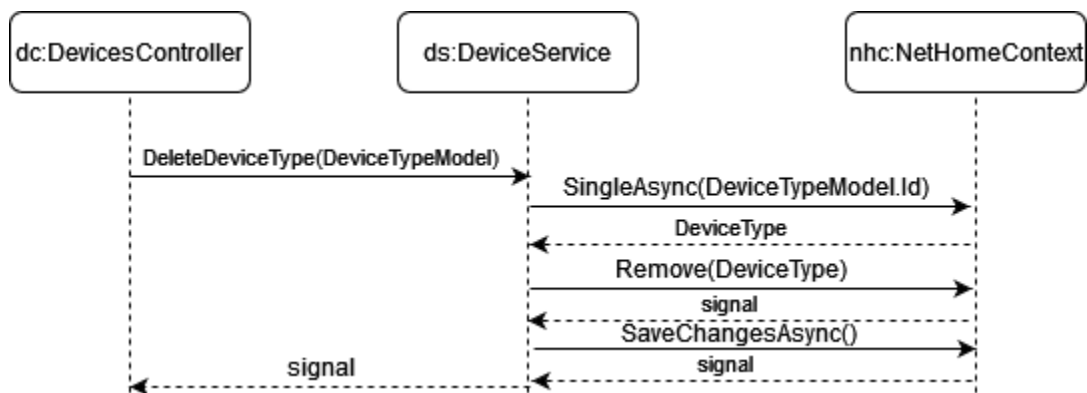
### Уговор УГ15: ОбришиТипУређаја

Операција: RemoveDeviceType(DeviceTypeModel)

Веза са СК: СК10

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

Постуслови: Тип уређаја је обрисан.



Слика 114 ДС Брисање типа уређаја

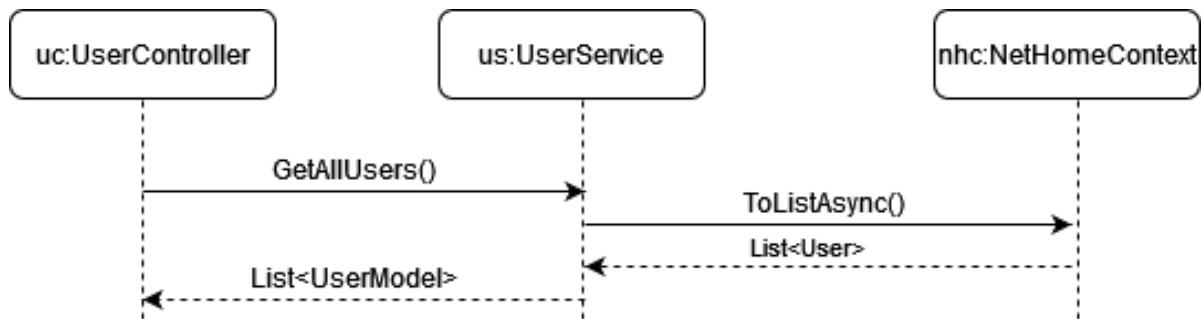
### Уговор УГ16: ВратиСвеКориснике

Операција: GetAllUsers(List<UserModel>)

Веза са СК: СК11, СК12, СК13

Предуслови: /

Постуслови: /



Слика 115 ДС Врати све кориснике

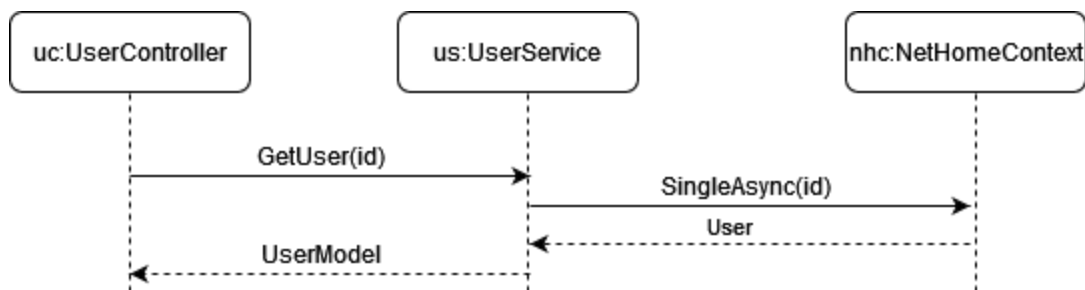
### Уговор УГ17: ВратиКорисника

Операција: GetUser(id, UserModel)

Веза са СК: СК11, СК12, СК13

Предуслови: /

Постуслови: /



Слика 116 ДС Врати корисника

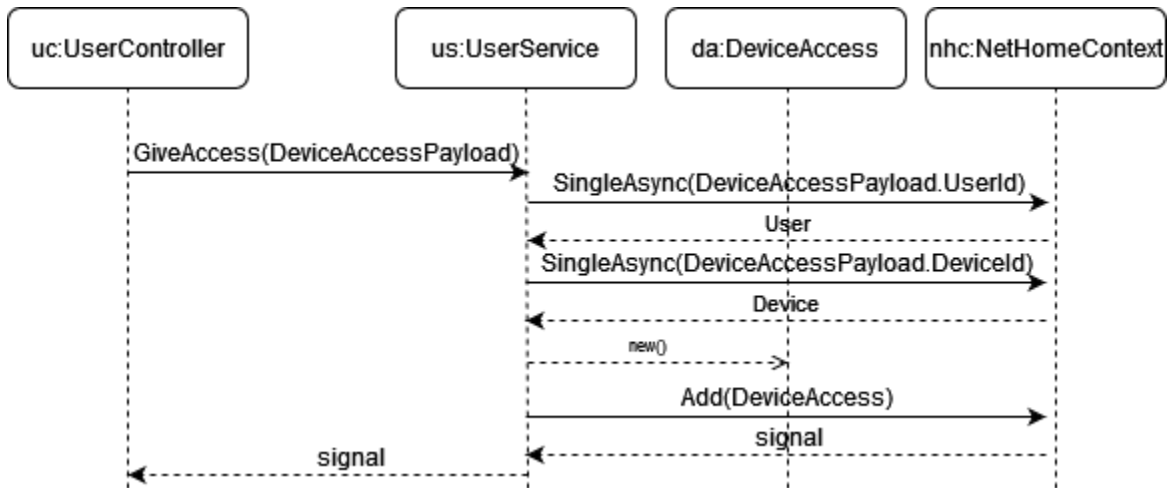
### Уговор УГ18: ДодајПристапУређају

Операција: GiveAccess(DeviceAccessPayload)

Веза са СК: СК12

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

Постуслови: Приступ уређају је сачуван.



Слика 117 ДС Додавање приступа уређају

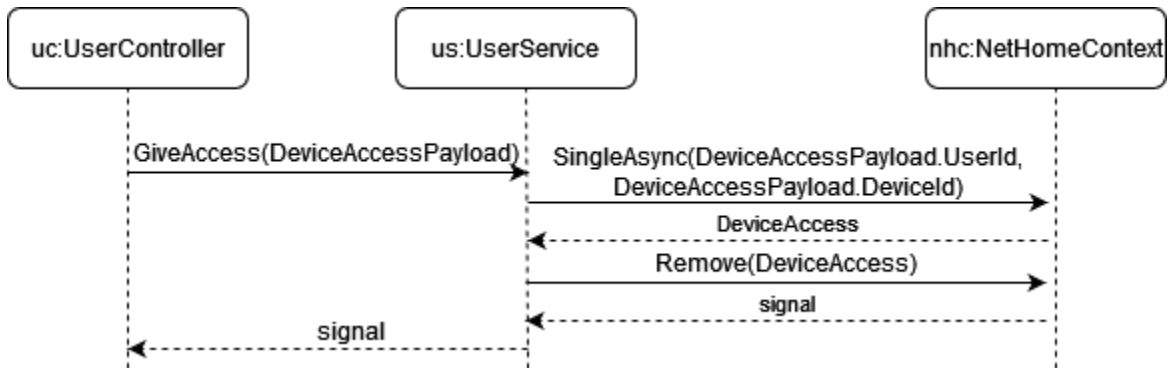
### Уговор УГ19: ОбришиПриступУређају

Операција: RemoveAccess(DeviceAccessPayload)

Веза са СК: СК13

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

Постуслови: Обрисан је приступ уређају.



Слика 118 Брисање приступа уређају

### 6.3.3 Пројектовање базе података

Релациони модел добијен је на основу концептуалног модела.

Класа User наслеђује класу IdentityUser и на основу тога постојаће табелаAspNetUsers која ће садржати како атрибуте класе User, тако и атрибуте базне класе IdentityUser.

На сличан начин, сви атрибути класа које наслеђују класу Device биће обејдињени у једну табелу Device која ће такође садржати и заједничке атрибуте из класе Device.

- Device(Id, Name, Model, DateAdded, IpAddress, DeviceUsername, DevicePassword, *DeviceType.Id*, *Room.Id*)
- DeviceType(Id, Name)
- Room(Id, Name)
- AirConditioner(Device.Id, Ison, Temperature, FanSpeed, Swing, TimerSet, TimerValue)
- RGBLight(Device.Id, Ison, Mode, Brightness, Red, Green, Blue, White, Gain)
- RollerShutter(Device.Id, CurrentPercentage, FavPos1, FavPos2, FavPos3, FavPos4)
- DWSensor(Device.Id, IsOpen, Placement)
- THSensor(Device.Id, Temperature, Humidity)
- SmartSwitch(Device.Id, Ison)
- User(IdentityUser.Id, FirstName, LastName, Age, Gender, DateOfRegistration)
- DeviceUser(DeviceId, UserId)
- IdentityUser(Id, Username, PasswordHash, Email)
- AspNetRoles(Id, Name)
- AspNetUserRole(UserId, RoleId)

На основу креираних ентитета, а уз помоћ Entity Framework-а, генерисане су табеле у бази података:

Табела <b>Device</b>		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT RESTRICTED DeviceType, Room
	Id	Integer	not null			UPDATE RESTRICTED DeviceType, Room
	Name	String	not null			CASCADES DeviceUser
	Model	String	not null			
	DateAdded	Date	not null			
	IpAddress	String	not null			DELETE RESTRICTED DeviceUser
	DeviceUsername	String	not null			
	DevicePassword	String	not null			
	RoomId	Integer				
	TypeId	Integer				
	Discriminator	String	not null			
	Ison	Boolean				
	Temperature	Double				
	FanSpeed	Integer				
	Swing	Boolean				
	TimerSet	Boolean				
	TimerValue	Double				
	IsOpen	Boolean				
	Placement	String				
	RGBLight_Ison	Boolean				
Mode	String					
Brightness	Integer					

Red	Integer			
Green	Integer			
Blue	Integer			
White	Integer			
Gain	Integer			
CurrentPercentage	Integer			
FavPos1	Integer			
FavPos2	Integer			
FavPos3	Integer			
FavPos4	Integer			
SmartSwitch_Ison	Boolean			
THSensor_ Tempaerature	Double			
Humidity	Double			

Табела 1 - Device

Табела <b>DeviceType</b>		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 Device DELETE CASCADES Device
	Id	Integer	not null			
	Name	String	not null			

Табела 2 - DeviceType

Табела <b>DeviceUser</b>		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT RESTRICTED Device,AspNetUsers UPDATE RESTRICTED Device,AspNetUsers DELETE /
	DeviceId	Integer	not null			
	UserId	String	not null			

Табела 3 DeviceUser

Табела <b>AspNetUsers</b>		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 AspNetUserRole, DeviceUser  DELETE RESTRICTED AspNetUserRole, DeviceUser
	Id	String	not null			
	Username	String	not null			
	PasswordHash	String	not null			
	Email	String	not null			
	FirstName	String	not null			
	LastName	String	not null			
	Age	Integer	not null			
	Gender	String	not null			
	DateOfRegistration	Date	not null			

Табела 4 AspNetUser



Табела <b>AspNetUserRole</b>		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, AspNetRoles UPDATE RESTRICTED AspNetUsers, AspNetRoles DELETE /
	UserId	String	not null			
	RoleId	String	not null			

Табела 5 AspNetUserRole

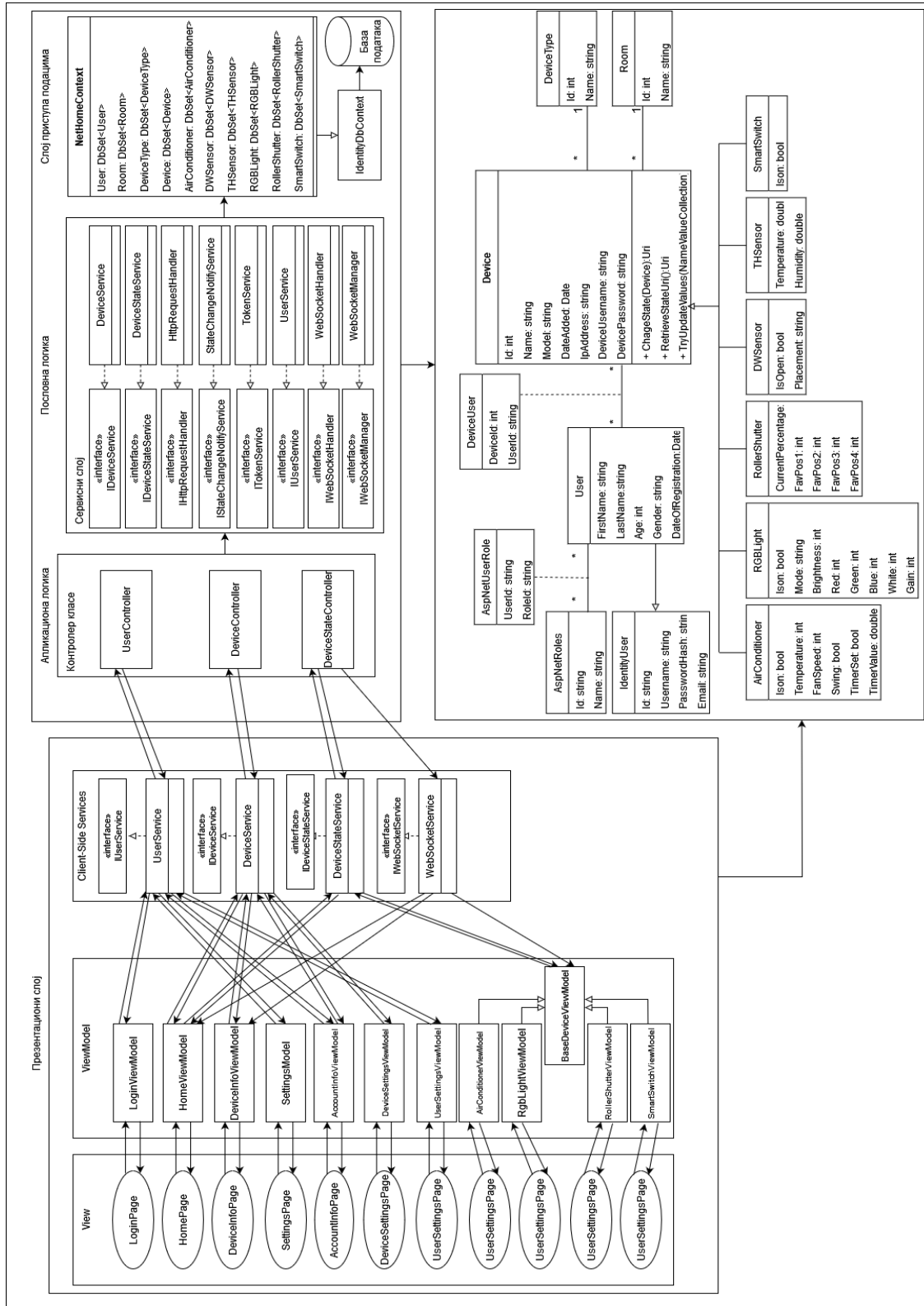
Табела <b>Room</b>		Просто вредносно ограничење		Сложено вредносно ограничење		Структурно ограничење
Атрибути	Име	Тип атрибута	Вредност атрибута	Међузависност атрибута једне табеле	Међузависност атрибута више табела	INSERT / UPDATE CASCADES Device DELETE CASCADES Device
	Id	Integer	not null			
	Name	String	not null			

Табела 6 Room

Табела <b>AspNetRoles</b>		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 AspNetUserRole DELETE RESTRICTED AspNetUserRole
	RoleId	String	not null			
	Name	String	not null			

Табела 7 AspNetRoles

## 6.4 Коначан изглед архитектуре софтверског система



Слика 119 Коначна архитектура

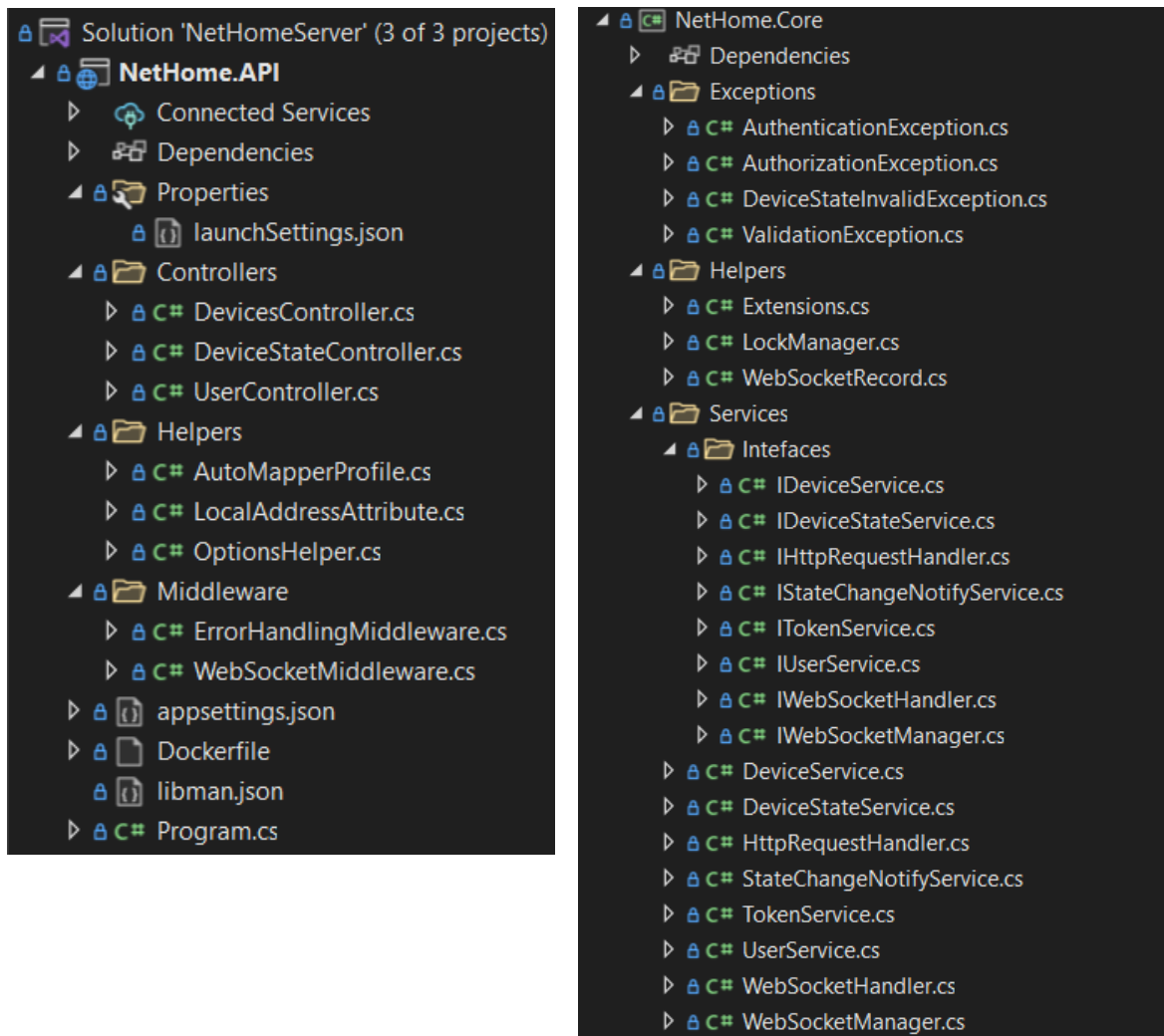
## 7. Фаза имплементације

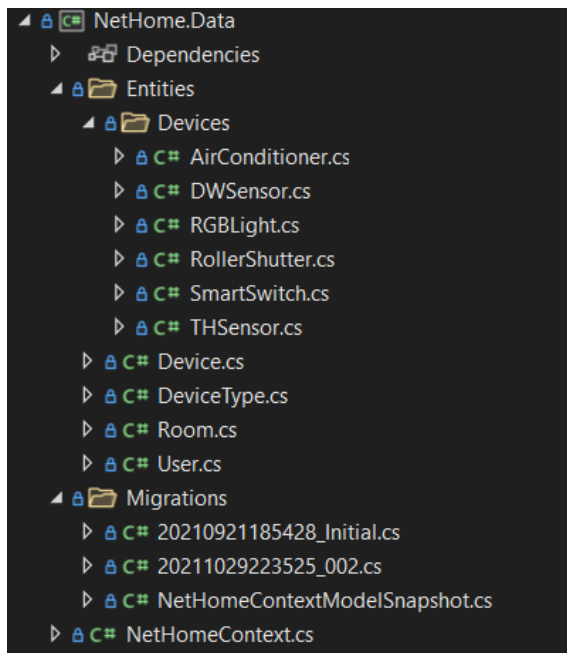
Софтверски систем развијен је употребом .NET технологија, по клијент-сервер архитектури. Клијентска апликација представља Xamarin апликацију која је намењена да се извршава на мобилном телефону, док серверска апликација представља WebApi. Систем за управљање базом података је SQLite, док је за мапирање и комуникацију са базом коришћен EntityFrameworkCore. Као развојно окружење коришћен је Visual Studio 2022.

### 7.1 Структура софтверског система

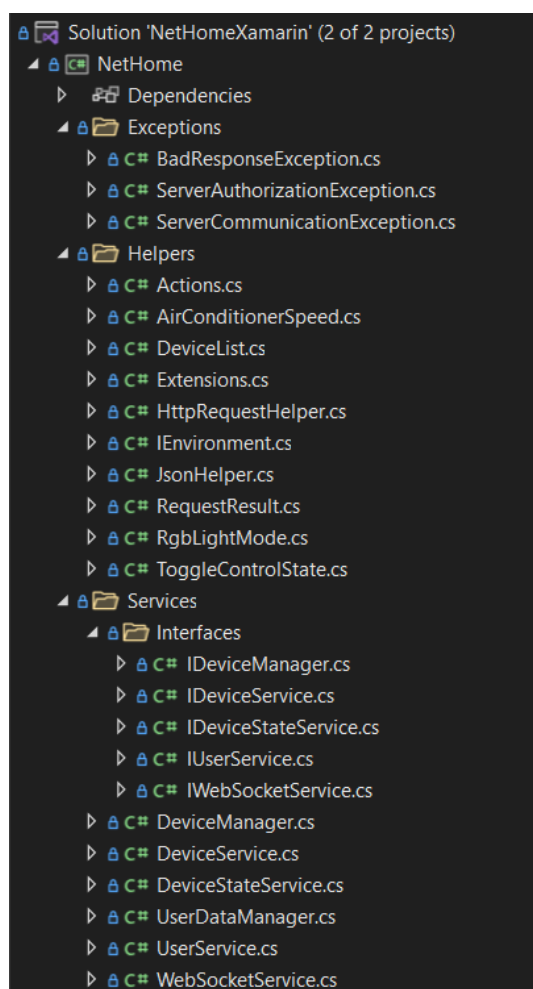
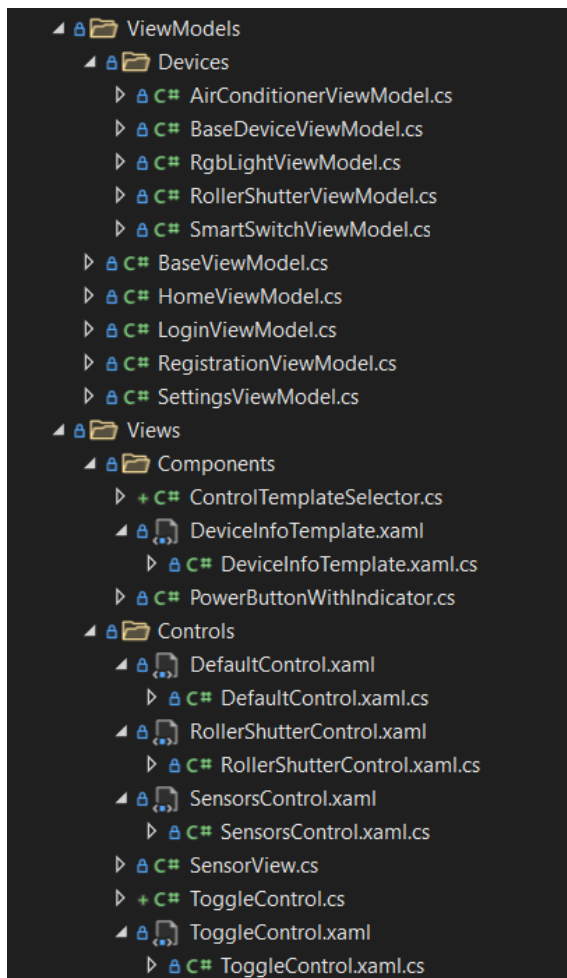
На основу архитектуре софтверског система направљени су следећи пројекти са својим неопходним класама и они су приказани на следећим сликама:

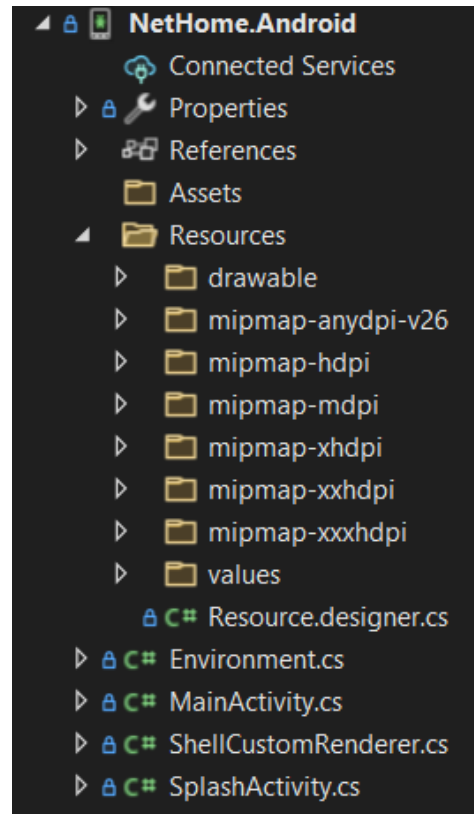
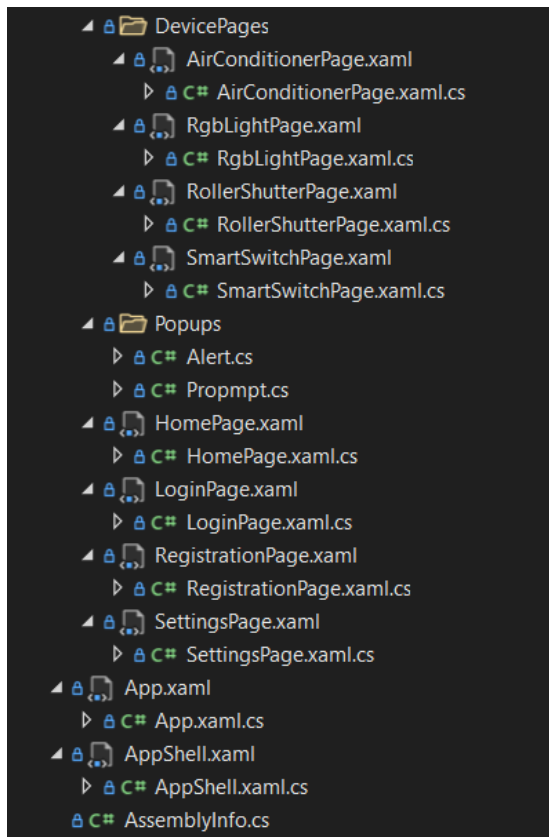
Серверска апликација:





### Клијентска апликација:





## 7.2 Имплементација апликационе логике

У овом наставку биће приказана имплементација претходно пројектоване апликационе логике.

### 7.2.1 Комуникација са клијентом

Почетак извршавања серверске апликације преставља фајл Program.cs, односно класа Program, која након промена уведених у .NET 6 више не садржи Main методу, већ се сав код у овом фајлу извршава приликом покретања апликације.

```
var builder = WebApplication.CreateBuilder(args);
var services = builder.Services;

services.AddAuthentication(OptionsHelper.AuthenticationOptions())
    .AddJwtBearer(OptionsHelper.JwtBearerOptions(builder.Configuration["Jwt:Key"]));
services.AddAuthorization();
services.AddDbContext<NetHomeContext>(options =>
    options.UseSqlite(builder.Configuration["ConnectionStrings:NetHomeContextConnection"]));
services.AddIdentityCore<User>(OptionsHelper.IdentityOptions())
    .AddRoles<IdentityRole>().AddEntityFrameworkStores<NetHomeContext>();
services.AddScoped<IHttpRequestHandler, HttpRequestHandler>();
services.AddSingleton<IWebSocketManager, WebSocketManager>();
services.AddScoped<IWebSocketHandler, WebSocketHandler>();
services.AddAutoMapper(AppDomain.CurrentDomain.GetAssemblies());
services.AddScoped<IUserService, UserService>();
services.AddScoped<ITokenService, TokenService>();
services.AddScoped<IDeviceService, DeviceService>();
services.AddScoped<IStateChangeNotifyService, StateChangeNotifyService>();
services.AddScoped<IDeviceStateService, DeviceStateService>();
services.AddControllers().AddJsonOptions(OptionsHelper.JsonOptions());

var app = builder.Build();

app.UseMiddleware<ErrorHandlingMiddleware>();

app.UseRouting();

app.UseAuthentication();
app.UseAuthorization();

app.UseWebSockets(new WebSocketOptions()
{
    KeepAliveInterval = TimeSpan.FromSeconds(10)
});
app.UseMiddleware<WebSocketMiddleware>();

app.UseEndpoints(endpoints =>
{
    endpoints.MapControllers();
});

app.Run();
```

На претходној слици види се да се у првом делу класе Program дефинишу неопходне функционалности и сервиси, потребни за потпуну оперативност. У другом делу дефинишу се компоненте проточне обраде (*pipeline*) апликације. У

том делу веома је битан редослед позивања тих компоненти, обзиром да приликом пристизања сваког HTTP захтева, тај захтев пролази кроз цео *pipeline* и тек на крају долази до контролера који ће тај захтев усмерити тако да он буде извршен. Ове компоненте називају се *Middleware*.

Комуникација са клијентском апликацијом одвија се примарно преко стандардних HTTP захтева, али се такође, паралелно, користе и *WebSocket*-и, који служе да у реалном времену обавештавају кориснике о променама које други корисници праве или које се дешавају у стану.

Ради обраде креиран је *WebSocketMiddleware*, који „пресреће“ *WebSocket* захтеве и обрађује их, док остале захтеве прослеђује на даље извршавање.

2 references

```
public class WebSocketMiddleware
{
    private readonly RequestDelegate _next;

    0 references
    public WebSocketMiddleware(RequestDelegate next)
    {
        _next = next;
    }

    0 references
    public async Task Invoke(HttpContext context, IWebSocketHandler _webSocketHandler)
    {
        if (context.WebSockets.IsWebSocketRequest && context.User.Identity.IsAuthenticated)
        {
            using var socket = await context.WebSockets.AcceptWebSocketAsync();
            var userId = context.User.FindFirst(ClaimTypes.NameIdentifier).Value;
            if (userId is null) throw new AuthenticationException("WebSocket authentication error!");
            var wsr = _webSocketHandler.OnConnected(userId, socket);
            await _webSocketHandler.ReceiveAsync(wsr);
        }
        else
        {
            await _next(context);
        }
    }
}
```

*WebSocketMiddleware* путем *DependencyInjection*-а захтева инстанцу сервиса за рад са *WebSocet*-има и за ауторизоване кориснике отвара *WebSocket* конекцију. Приликом промена стања уређаја, из сервиса за управљање стањима уређаја, поново се позива овај сервис и сви тренутно повезани корисници обавештавају се о насталој промени.

Ради стабилности серверске апликације такође се имплементира и посебан *Middleware* за обраду грешака. Он позив за даљу обраду сваког захтева обавија у *Try/Catch* блок, што значи да свака грешка која настане у даљој обради захтева, а није обрађена, бива обрађена у овом сегменту.

2 references

```
public class ErrorHandlingMiddleware
{
    private readonly RequestDelegate _next;

    0 references
    public ErrorHandlingMiddleware(RequestDelegate next)
    {
        _next = next;
    }

    0 references
    public async Task Invoke(HttpContext context)
    {
        try
        {
            await _next(context);
        }
        catch (Exception ex)
        {
            await HandleExceptionAsync(context, ex);
        }
    }

    1 reference
    public static Task HandleExceptionAsync(HttpContext context, Exception exception)
    {
        var code = HttpStatusCode.InternalServerError;
        if (exception is ValidationException) code = HttpStatusCode.BadRequest;
        else if (exception is AuthenticationException) code = HttpStatusCode.Unauthorized;
        else if (exception is AuthorizationException) code = HttpStatusCode.Forbidden;

        var result = JsonSerializer.Serialize(new { exception.Message });
        context.Response.ContentType = "application/json";
        context.Response.StatusCode = (int)code;
        return context.Response.WriteAsync(result);
    }
}
```

За аутентификацију и ауторизацију, апликација користи *ASP.NET Core Identity*, који у апликацији генерише свој *Middleware* и приликом сваког захтева који очекује да корисник буде ауторизован, ауторизацију врши на следећи начин. Приликом пријаве на систем кориснику се уз податке о његовом налогу прослеђује и токен за ауторизацију (*JWT*). Тај токен мора да буде постављен у заглављу сваког HTTP захтева, чија циљана крајња тачка (*Endpoint*) захтева ауторизованог корисника. *Middleware* за ауторизацију затим, по пристизању захтева, врши дешифровање токена и проверава његов садржај, а у инстанци *HttpContext* класе, која је везана



за тај специфичан HTTP захтев, поставља податке о ауторизованом кориснику уколико је ауторизација била успешна.

```
1 reference
public static Action<JwtBearerOptions> JwtBearerOptions(string key) => options =>
{
    options.RequireHttpsMetadata = false;
    options.SaveToken = true;
    options.TokenValidationParameters = new TokenValidationParameters()
    {
        ValidateIssuerSigningKey = true,
        IssuerSigningKey = new SymmetricSecurityKey(Encoding.ASCII.GetBytes(key)),
        ValidateIssuer = false,
        ValidateAudience = false,
        ValidateLifetime = true,
        RequireExpirationTime = true
    };
    options.Events = new JwtBearerEvents
    {
        OnTokenValidated = async context =>
        {
            var id = context.Principal.FindFirst(ClaimTypes.NameIdentifier).Value;
            var userManager = context.HttpContext.RequestServices.GetRequiredService<UserManager<User>>();
            var user = await userManager.FindByIdAsync(id) ?? throw new AuthorizationException("Unable to verify token!");
            if (user.LockoutEnd > DateTime.Now)
            {
                var timeSpan = user.LockoutEnd - DateTime.Now;
                var years = timeSpan.Value.Days / 365;
                throw new AuthorizationException($"Your account has been locked for {years} years. Please contact an owner.");
            }
            var claims = new ClaimsIdentity();
            foreach (string role in await userManager.GetRolesAsync(user))
                claims.AddClaim(new Claim(ClaimTypes.Role, role));
            context.Principal.AddIdentity(claims);
        }
    };
};
```

Да би се ауторизација обавила неопходно је да код сваке крајње тачке у контролеру буде постављен атрибут за ауторизацију као на следећем примеру:

```
[Authorize]
[HttpGet("get")]
0 references
public async Task<DeviceModel> Get([FromQuery] int id)
{
    return await _deviceService.GetDevice(id, User.FindFirstValue(ClaimTypes.NameIdentifier));
}
```

Могуће је такође, у оквиру атрибута, дефинисати и које роле које корисник мора имати за приступ крајњој тачки:

```
[Authorize(Roles = "Admin, Owner")]
[HttpPost("add")]
0 references
public async Task<DeviceModel> Add([FromBody] DevicePayload device)
{
    return await _deviceService.Add(device);
}
```

## 7.2.2 Пословна логика

Пословна логика је описана структуром коју чине доменске класе и понашањем које се састоји из системских операција [1].

### Имплементација доменских класа

Доменске класе се имплементирају на основу концептуалног модела, садрже јавне *Property*-је и у случају уређаја, јавне методе, које садрже део логике за специфичну врсту уређаја.

На следећој слици приказана је једна од једноставнијих доменских класа, *SmartSwitch*, која се односи на паметни прекидач, на сличан начин имплементирани су и остале доменске класе.

```
6 references
public class SmartSwitch : Device
{
    4 references
    public bool Ison { get; set; }
    2 references
    public override Uri ChangeState(Device newValue)
    {
        SmartSwitch ss = (SmartSwitch)newValue;
        Ison = ss.Ison;
        switch (Model)
        {
            case "Shelly 1":
            case "Shelly 1PM":
                var baseUri = new Uri($"{IpAddress}/relay/0");
                var uri = new UriBuilder(baseUri);
                string isonQuery = $"turn={{Ison ? "on" : "off"}}";
                uri.Query = isonQuery;
                return uri.Uri;
            default:
                throw new NotImplementedException();
        }
    }
}

2 references
public override Uri RetrieveStateUri() => new($"{IpAddress}/relay/0");

3 references
public override bool TryUpdateValues(NameValueCollection values)
{
    bool result = bool.TryParse(values["ison"], out bool newValue);
    if (result) Ison = newValue;
    return result;
}
}
```

## Имплементација системских операција

**Уговор УГ1:** ПријавиКорисника

Операција: Login(LoginRequest)

Веза са СК: СК1

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

Постуслови: Корисник је пријављен.

```
[HttpPost("login")]
0 references
public async Task<LoginResponse> Login([FromBody] LoginRequest login)
{
    return await _userService.Login(login);
}
```

**Уговор УГ2:** ИзмениСтањеУређаја

Операција: ChangeDeviceState(DeviceModel)

Веза са СК: СК2

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

Постуслови: Стање уређаја је промењено.

```
[Authorize]
[HttpPost("change")]
0 references
public async Task<DeviceModel> ChangeDeviceState([FromBody] DeviceModel device)
{
    return await _deviceStateService.ChangeState(device, User.FindFirstValue(ClaimTypes.NameIdentifier));
}
```

**Уговор УГ3:** ВратиПодаткеОНалогу

Операција: GetUserData(UserModel)

Веза са СК: СК3

Предуслови:

Постуслови:

```

[Authorize]
[HttpGet("getown")]
0 references
public async Task<UserModel> GetOwnUserData()
{
    return await _userService.GetUser(User.FindFirstValue(ClaimTypes.NameIdentifier));
}

```

#### Уговор УГ4: ИзмениКорисника

Операција: UpdateUser(UserModel)

Веза са СК: СК3, СК11

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

Постуслови: Подаци о кориснику су измењени.

```

[Authorize]
[HttpPost("update")]
0 references
public async Task<UserModel> Update([FromBody] UserModel userModel)
{
    return await _userService.UpdateUser(userModel, User.FindFirstValue(ClaimTypes.NameIdentifier));
}

```

#### Уговор УГ5: ДодајУређај

Операција: AddDevice(DevicePayload)

Веза са СК: СК4

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

Постуслови: Подаци о уређају су сачувани.

```

[Authorize(Roles = "Admin, Owner")]
[HttpPost("add")]
0 references
public async Task<DeviceModel> Add([FromBody] DevicePayload device)
{
    return await _deviceService.Add(device);
}

```

#### Уговор УГ6: ВратиСвеУређаје

Операција: GetAllDevices(List<DeviceModel>)

Веза са СК: СК5, СК6

Предуслови: /

Постуслови: /

```
[Authorize]
[HttpGet("getall")]
0 references
public async Task<ICollection<DeviceModel>> GetAll()
{
    return await _deviceService.GetAllDevices(User.FindFirstValue(ClaimTypes.NameIdentifier));
}
```

**Уговор УГ7:** ВратиУређај

Операција: GetDevice(id, DeviceModel)

Веза са СК: СК5, СК6

Предуслови: /

Постуслови: /

```
[Authorize]
[HttpGet("get")]
0 references
public async Task<DeviceModel> Get([FromQuery] int id)
{
    return await _deviceService.GetDevice(id, User.FindFirstValue(ClaimTypes.NameIdentifier));
}
```

**Уговор УГ8:** ИзмениУређај

Операција: UpdateDevice(DeviceModel)

Веза са СК: СК5

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

Постуслови: Подаци о уређају су измењени.

```
[Authorize(Roles = "Admin, Owner")]
[HttpPost("update")]
0 references
public async Task<DeviceModel> Update([FromBody] DevicePayload device)
{
    return await _deviceService.Update(device);
}
```

**Уговор УГ9:** ОбришиУређај

Операција: DeleteDevice(DeviceModel)

Веза са СК: СК6

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

Постуслови: Уређај је обрисан.

```
[Authorize(Roles = "Admin, Owner")]
[HttpPost("delete")]
0 references
public async Task Delete([FromBody] DeviceModel device)
{
    await _deviceService.Delete(device);
}
```

**Уговор УГ10:** ВратиСвеСобе

Операција: GetAllRooms(List<RoomModel>)

Веза са СК: СК8

Предуслови: /

Постуслови: /

```
[Authorize(Roles = "Admin, Owner")]
[HttpGet("getrooms")]
0 references
public async Task<ICollection<RoomModel>> GetRooms()
{
    return await _deviceService.GetRooms();
}
```

**Уговор УГ11:** ДодајСобу

Операција: AddRoom(RoomModel)

Веза са СК: СК7

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

Постуслови: Подаци о соби су сачувани.

```
[Authorize(Roles = "Admin, Owner")]
[HttpPost("addroom")]
0 references
public async Task AddRoom([FromBody] RoomModel room)
{
    await _deviceService.AddRoom(room);
}
```

### Уговор УГ12: ОбришиСобу

Операција: DeleteRoom(RoomModel)

Веза са СК: СК8

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

Постуслови: Соба је обрисана.

```
[Authorize(Roles = "Admin, Owner")]
[HttpPost("deleteroom")]
0 references
public async Task DeleteRoom([FromBody] RoomModel room)
{
    await _deviceService.DeleteRoom(room);
}
```

### Уговор УГ13: ВратиСвеТиповеУређаја

Операција: GetAllDeviceTypes(List<DeviceTypeModel>)

Веза са СК: СК10

Предуслови: /

Постуслови: /

```
[Authorize(Roles = "Admin, Owner")]
[HttpGet("gettypes")]
0 references
public async Task<ICollection<DeviceTypeModel>> GetDeviceTypes()
{
    return await _deviceService.GetDeviceTypes();
}
```

### Уговор УГ14: ДодајТипУређаја

Операција: AddDeviceType(DeviceTypeModel)

Веза са СК: СК9

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

Постуслови: Подаци о типу уређаја су сачувани.

```

[Authorize(Roles = "Admin, Owner")]
[HttpPost("addtype")]
0 references
public async Task AddDeviceType([FromBody] DeviceTypeModel type)
{
    await _deviceService.AddType(type);
}

```

**Уговор УГ15:** Обриши Тип Уређаја

Операција: RemoveDeviceType(DeviceTypeModel)

Веза са СК: СК10

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

Постуслови: Тип уређаја је обрисан.

```

[Authorize(Roles = "Admin, Owner")]
[HttpPost("deletetype")]
0 references
public async Task DeleteDeviceType([FromBody] DeviceTypeModel type)
{
    await _deviceService.DeleteType(type);
}

```

**Уговор УГ16:** Врати Све Кориснике

Операција: GetAllUsers(List<UserModel>)

Веза са СК: СК11, СК12, СК13

Предуслови: /

Постуслови: /

```

[Authorize(Roles = "Owner")]
[HttpGet("getall")]
0 references
public async Task<ICollection<UserModel>> GetAllUsers()
{
    return await _userService.GetAllUsers();
}

```

**Уговор УГ17:** Врати Корисника

Операција: GetUser(id, UserModel)

Веза са СК: СК11, СК12, СК13



Предуслови: /

Постуслови: /

```
[Authorize(Roles = "Owner")]
[HttpGet("get")]
0 references
public async Task<UserModel> GetUser([FromQuery] string id)
{
    return await _userService.GetUser(id);
}
```

**Уговор УГ18:** ДодајПриступУређају

Операција: GiveAccess(DeviceAccessPayload)

Веза са СК: СК12

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

Постуслови: Приступ уређају је сачуван.

```
[Authorize(Roles = "Owner")]
[HttpPost("giveaccess")]
0 references
public async Task GiveAccess([FromBody] DeviceAccessPayload deviceAccess)
{
    await _userService.GiveUserDeviceAccess(deviceAccess);
}
```

**Уговор УГ19:** ОбришиПриступУређају

Операција: RemoveAccess(DeviceAccessPayload)

Веза са СК: СК13

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

Постуслови: Обрисан је приступ уређају.

```
[Authorize(Roles = "Owner")]
[HttpPost("removeaccess")]
0 references
public async Task RemoveAccess([FromBody] DeviceAccessPayload deviceAccess)
{
    await _userService.RemoveUserDeviceAccess(deviceAccess);
}
```

### 7.2.3 Складиште података






















На основу класа ентитета генерисане су табеле у бази података. Коришћена база података је *SQLite*.

▼	 AspNetRoles	CREATE TABLE "AspNetRoles"
	 Id	TEXT "Id" TEXT NOT NULL
	 Name	TEXT "Name" TEXT
	 NormalizedName	TEXT "NormalizedName" TEXT
	 ConcurrencyStamp	TEXT "ConcurrencyStamp" TEXT




Слика 120 Табела AspNetRoles

▼	 AspNetUserRoles	CREATE TABLE "AspNetU
	 UserId	TEXT "UserId" TEXT NOT NULL
	 RoleId	TEXT "RoleId" TEXT NOT NULL

Слика 121 Табела AspNetUserRoles

▼	 AspNetUsers	CREATE TABLE "AspNetUsers" ( "Id" TEXT NOT
	 Id	TEXT "Id" TEXT NOT NULL
	 FirstName	TEXT "FirstName" TEXT
	 LastName	TEXT "LastName" TEXT
	 Age	INTEGER "Age" INTEGER NOT NULL
	 Gender	TEXT "Gender" TEXT
	 DateOfRegistration	TEXT "DateOfRegistration" TEXT NOT NULL
	 UserName	TEXT "UserName" TEXT
	 NormalizedUserName	TEXT "NormalizedUserName" TEXT
	 Email	TEXT "Email" TEXT
	 NormalizedEmail	TEXT "NormalizedEmail" TEXT
	 EmailConfirmed	INTEGER "EmailConfirmed" INTEGER NOT NULL
	 PasswordHash	TEXT "PasswordHash" TEXT
	 SecurityStamp	TEXT "SecurityStamp" TEXT
	 ConcurrencyStamp	TEXT "ConcurrencyStamp" TEXT
	 PhoneNumber	TEXT "PhoneNumber" TEXT
	 PhoneNumberConfirmed	INTEGER "PhoneNumberConfirmed" INTEGER NOT NULL
	 TwoFactorEnabled	INTEGER "TwoFactorEnabled" INTEGER NOT NULL
	 LockoutEnd	TEXT "LockoutEnd" TEXT
	 LockoutEnabled	INTEGER "LockoutEnabled" INTEGER NOT NULL
	 AccessFailedCount	INTEGER "AccessFailedCount" INTEGER NOT NULL

Слика 122 Табела AspNetUsers

▼	 DeviceUser	CREATE TABLE "DeviceUser" ( "[
	 DevicesId	INTEGER "DevicesId" INTEGER NOT NULL
	 UsersId	TEXT "UsersId" TEXT NOT NULL

Слика 123 Табела DeviceUser

Room			CREATE TABLE "Room" (
Id	INTEGER	"Id" INTEGER NOT NULL	
Name	TEXT	"Name" TEXT	

Слика 124 Табела Room

DeviceType			CREATE TABLE "DeviceT
Id	INTEGER	"Id" INTEGER NOT NULL	
Name	TEXT	"Name" TEXT	

Слика 125 Табела DeviceType

Device			CREATE TABLE "Device" ( "Id" INTE
Id	INTEGER	"Id" INTEGER NOT NULL	
Name	TEXT	"Name" TEXT	
Model	TEXT	"Model" TEXT	
DateAdded	TEXT	"DateAdded" TEXT NOT NULL	
IpAddress	TEXT	"IpAddress" TEXT	
DeviceUsername	TEXT	"DeviceUsername" TEXT	
DevicePassword	TEXT	"DevicePassword" TEXT	
RoomId	INTEGER	"RoomId" INTEGER	
TypeId	INTEGER	"TypeId" INTEGER	
Discriminator	TEXT	"Discriminator" TEXT NOT NULL	
Ison	INTEGER	"Ison" INTEGER	
Temperature	INTEGER	"Temperature" INTEGER	
FanSpeed	INTEGER	"FanSpeed" INTEGER	
Swing	INTEGER	"Swing" INTEGER	
TimerSet	INTEGER	"TimerSet" INTEGER	
TimerValue	REAL	"TimerValue" REAL	
IsOpen	INTEGER	"IsOpen" INTEGER	
Placement	TEXT	"Placement" TEXT	
RGBLight_Ison	INTEGER	"RGBLight_Ison" INTEGER	
Mode	TEXT	"Mode" TEXT	
Brightness	INTEGER	"Brightness" INTEGER	
Red	INTEGER	"Red" INTEGER	
Green	INTEGER	"Green" INTEGER	
Blue	INTEGER	"Blue" INTEGER	
White	INTEGER	"White" INTEGER	
Gain	INTEGER	"Gain" INTEGER	
CurrentPercentage	INTEGER	"CurrentPercentage" INTEGER	
FavPos1	INTEGER	"FavPos1" INTEGER	
FavPos2	INTEGER	"FavPos2" INTEGER	
FavPos3	INTEGER	"FavPos3" INTEGER	
FavPos4	INTEGER	"FavPos4" INTEGER	
SmartSwitch_Ison	INTEGER	"SmartSwitch_Ison" INTEGER	
THSensor_Temperature	REAL	"THSensor_Temperature" REAL	
Humidity	REAL	"Humidity" REAL	

Слика 126 Табела Device

## 7.3 Имплементација презентационог слоја

Презентациони слој обухвата клијентску апликацију израђену помоћу .NET платформе за развој мобилних апликација *Xamarin*. Апликација је базирана на *MVVM* архитектури, али садржи и своје сервисе, помоћу којих *ViewModel*-и комуницирају са серверском апликацијом.

СК1: Случај коришћења – Пријава на систем

### Назив СК

Пријава на систем

### Актер СК

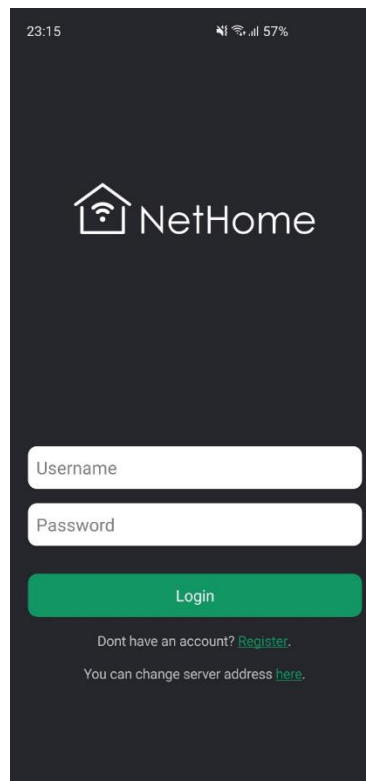
Корисник

### Учесници СК

Корисник и систем (програм)

### Предуслов

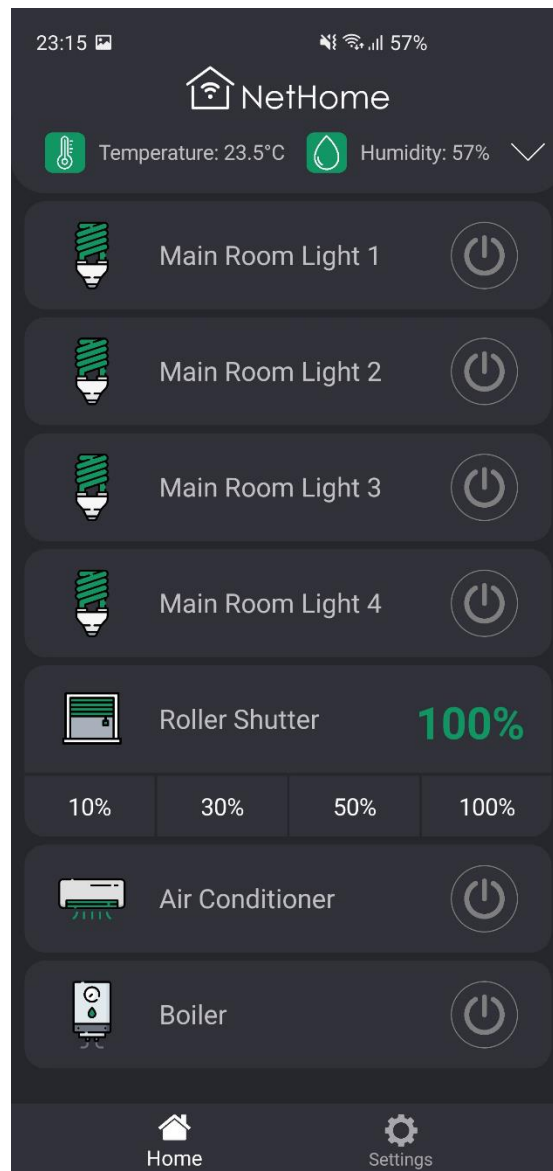
Систем је укључен и приказује страницу за пријаву на систем.



Слика 127 Страница за пријаву на систем

## Основни сценарио

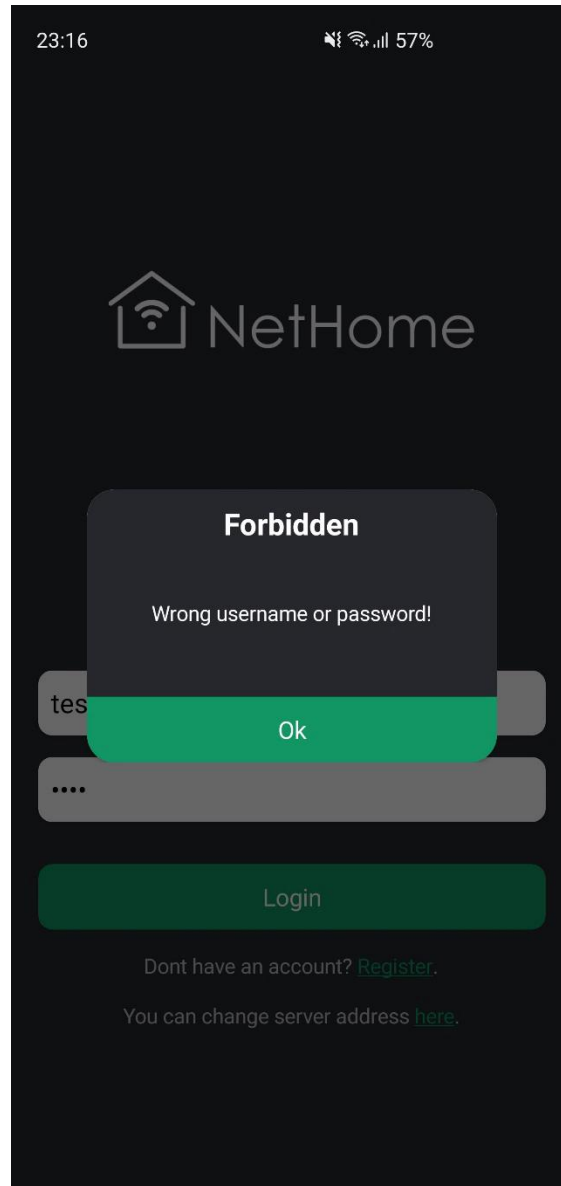
6. Корисник уноси корисничко име и лозинку (АПУСО)
7. Корисник контролише да ли је коректно унео корисничко име и лозинку (АНСО)
8. Корисник позива систем да се пријави (АПСО)
9. Систем проверава податке о кориснику (СО)
10. Систем кориснику враћа податке о налогу и токен за ауторизацију и приказује му почетну страну (ИА)



Слика 128 Почетна страница

## Алтернативна сценарија

- 10.1. Уколико систем не може да нађе корисника са унетим корисничким именом и лозинком, приказује кориснику поруку: „Погрешно корисничко име или лозинка!“ (ИА)



Слика 129 Грешка при пријави

## СК2: Случај коришћења – Промена стања уређаја

### Назив СК

Промена стања уређаја

### Актер СК

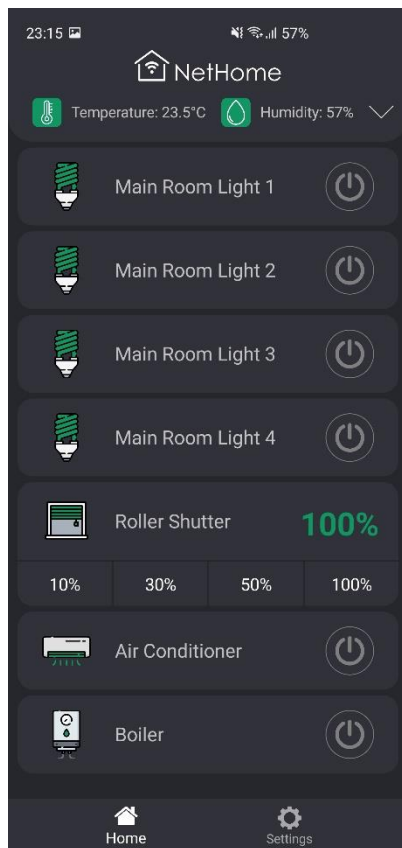
Корисник

### Учесници СК

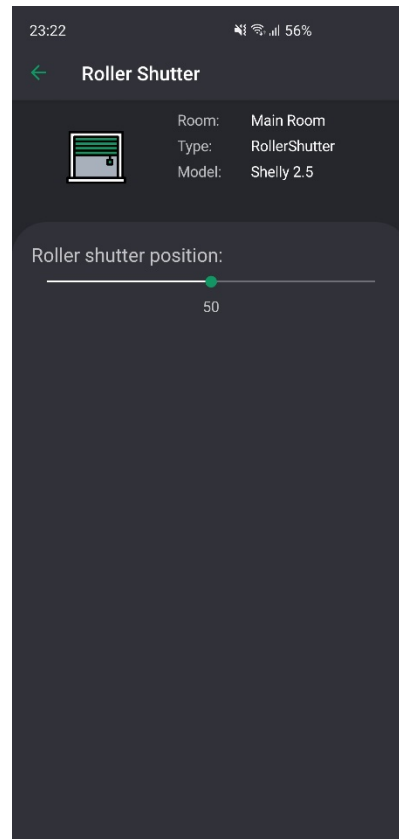
Корисник и систем (програм)

### Предуслов

Систем је укључен и корисник је пријављен на своја налог. Систем приказује почетну страницу или страницу специфичног уређаја.



Слика 130 Почетна страница



Слика 131 Страница за контролу ролетни

## Основни сценарио

5. Корисник уноси ново стање уређаја (АПУСО)
6. Корисник позива систем да изврши промену стања уређаја (АПСО)
7. Систем проналази уређај, проверава да ли корисник има приступ уређају, а затим врши измену стања уређаја (СО)
8. Систем враћа кориснику информације о уређају са измењеним стањем и приказује ново стање уређаја (ИА)

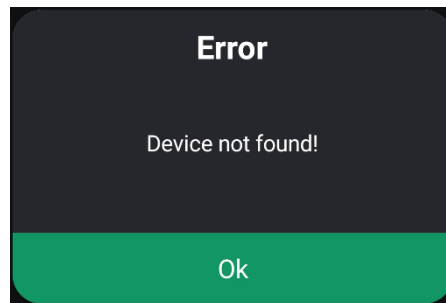


Слика 132 Измењено стање уређаја



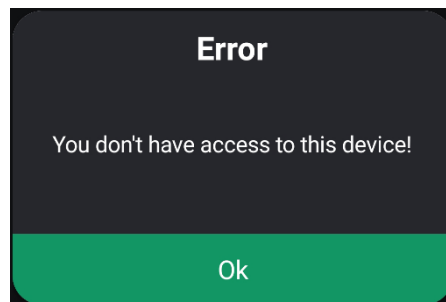
## Алтернативна сценарија

- 8.1. Уколико систем не може да нађе уређај, приказује кориснику поруку: „Уређај није пронађен!“. Прекида се извршење сценарија (ИА)



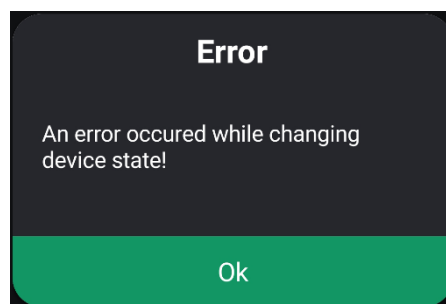
*Слика 133 Грешка при враћању уређаја*

- 8.2. Уколико корисник нема приступ уређају, приказује поруку: „Немате приступ овом уређају!“. Прекида се извршење сценарија (ИА)



*Слика 134 Грешка са дозволом за приступ уређају*

- 8.3. Уколико систем није у могућности да изврши тражену измену, приказује прокук: „Дошло је до грешке приликом измене стања уређаја!“ (ИА)



*Слика 135 Грешка при промени стања*

### СКЗ: Случај коришћења – Измена података о налогу

#### Назив СК

Измена података о налогу

#### Актер СК

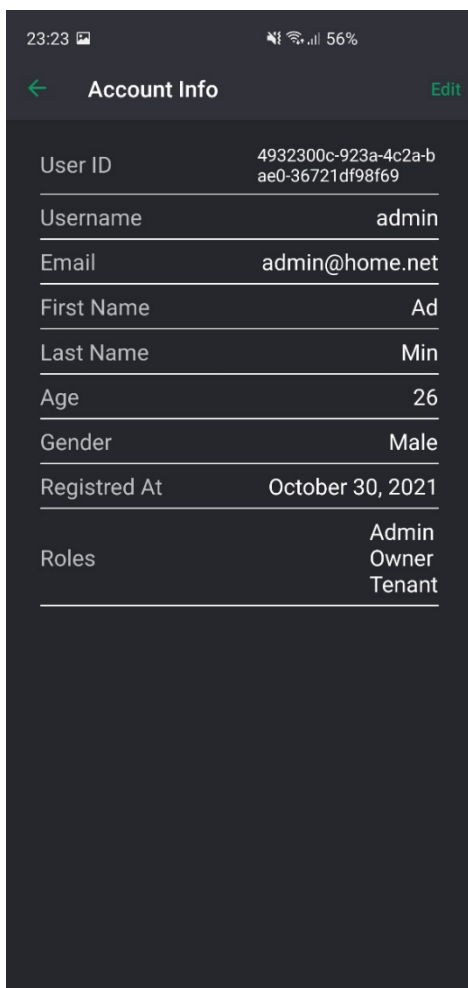
Корисник

#### Учесници СК

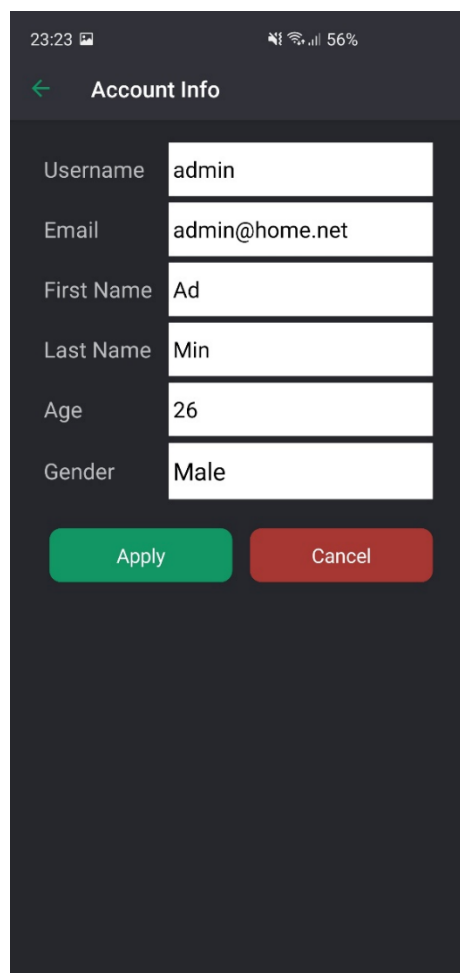
Корисник и систем (програм)

#### Предуслов

Систем је укључен и корисник је пријављен на своја налог. Систем приказује страницу за рад са подацима о налогу.



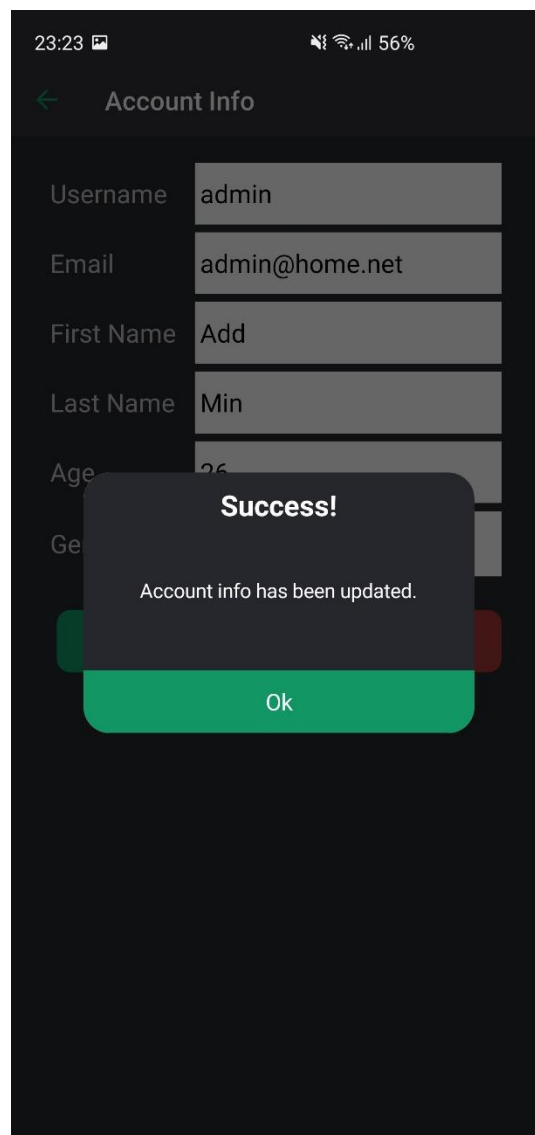
Слика 136 Страница са информацијама о налогу



Слика 137 Страница за измену података о налогу

## Основни сценарио

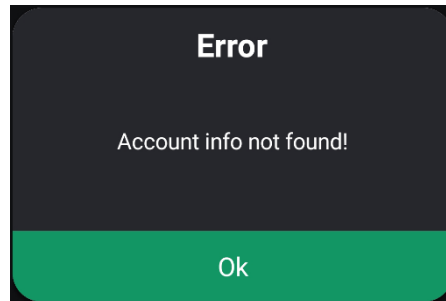
9. Корисник позива систем да прикаже податке о његовом налогу (АПСО)
10. Систем тражи податке о налогу корисника (СО)
11. Систем приказује податке о налогу корисника (ИА)
12. Корисник уноси измене података о налогу (АПУСО)
13. Корисник проверава да ли је исправно унео нове податке (АНСО)
14. Корисник позива систем да запамти податке о налогу (АПСО)
15. Систем памти измењене податке о налогу (СО)
16. Систем приказује кориснику поруку: „Подаци о налогу су успешно сачувани.“ (ИА)



Слика 138 Успешно измењени подаци о налогу

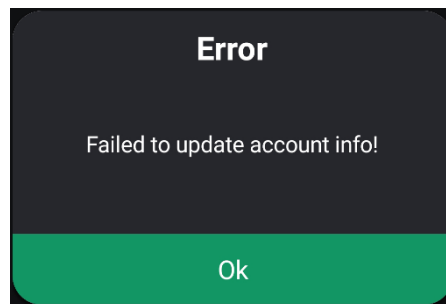
## Алтернативна сценарија

- 4.3. Уколико систем не може да пронађе податке о налогу корисника, приказује кориснику поруку: „Подаци о налогу нису пронађени!“. Прекида се извршење сценарија (ИА)



Слика 139 Грешка приликом враћања налога

- 8.3. Уколико систем не може да сачува податке о налогу, приказује кориснику поруку: „Није могуће сачувати податке о налогу!“ (ИА)



Слика 140 Грешка приликом измене података о налогу

## СК4: Случај коришћења – Унос уређаја

### Назив СК

Унос уређаја

### Актер СК

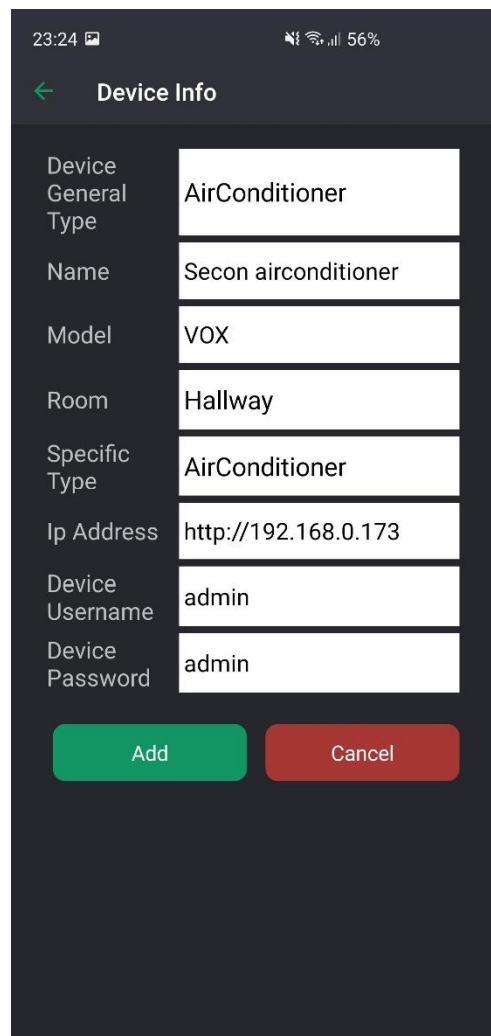
Корисник (Администратор или Власник)

### Учесници СК

Корисник и систем (програм)

### Предуслов

Систем је укључен и корисник је пријављен на своја налог. Систем приказује страницу за унос новог уређаја.



23:24 56%

← Device Info

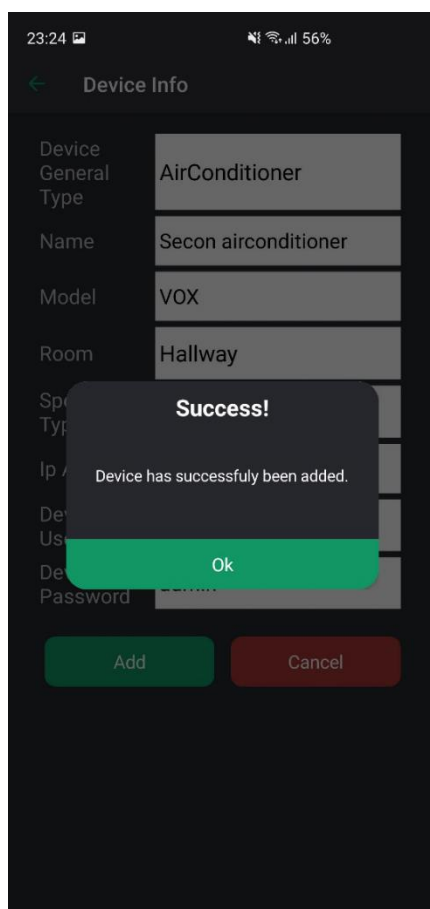
Device General Type	AirConditioner
Name	Secon airconditioner
Model	VOX
Room	Hallway
Specific Type	AirConditioner
Ip Address	http://192.168.0.173
Device Username	admin
Device Password	admin

Add Cancel

Слика 141 Страница за унос уређаја

## Основни сценарио

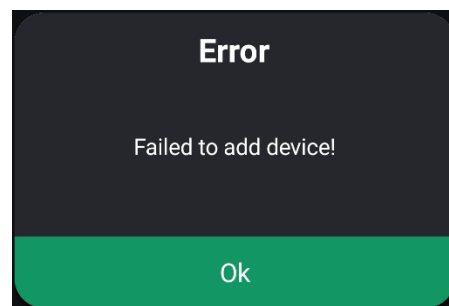
6. Корисник уноси податке о новом уређају (АПУСО)
7. Корисник проверава да ли је исправно унео податке (АНСО)
8. Корисник позива систем да запами податке о новом уређају (АПСО)
9. Систем памти податке о уређају (СО)
10. Систем приказује кориснику поруку: „Уређај успешно сачуван.“ (ИА)



Слика 142 Успешан унос уређаја

## Алтернативна сценарија

- 10.1. Уколико систем не може да сачува податке о уређају, приказује кориснику поруку: „Неуспешно додавање уређаја!“ (ИА)



Слика 143 Грешка приликом уноса уређаја

## СК5: Случај коришћења – Измена података о уређају

### Назив СК

Измена података о уређају

### Актер СК

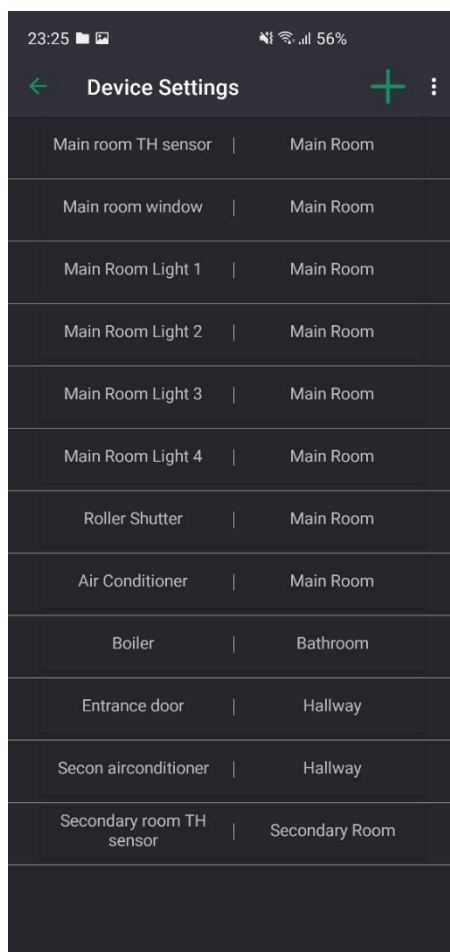
Корисник (Администратор или Власник)

### Учесници СК

Корисник и систем (програм)

### Предуслов

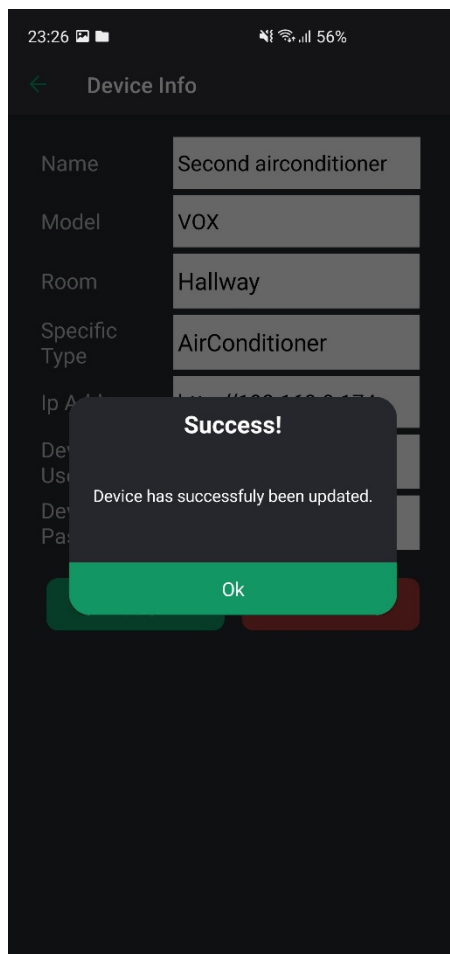
Систем је укључен и корисник је пријављен на своја налог. Систем приказује страницу за управљање уређајима.



Слика 144 Страница за управљање уређајима

## Основни сценарио

13. Корисник позива систем да прикаже листу уређаја (АПСО)
14. Систем проналази све постојеће уређаје (СО)
15. Систем приказује листу свих уређаја (ИА)
16. Корисник одабира уређај чије податке жели да измени (АПУСО)
17. Корисник позива систем да прикаже податке о изабраном уређају (АПСО)
18. Систем учитава податке о изабраном уређају (СО)
19. Систем приказује кориснику податке о изабраном уређају (ИА)
20. Корисник мења податке о уређају (АПУСО)
21. Корисник проверава да ли је добро унео податке (АНСО)
22. Корисник позива систем да запамти унете податке (АПСО)
23. Систем памти измењене податке о уређају (СО)
24. Систем приказује кориснику поруку: „Подаци о уређају су успешно измењени.“ (ИА)

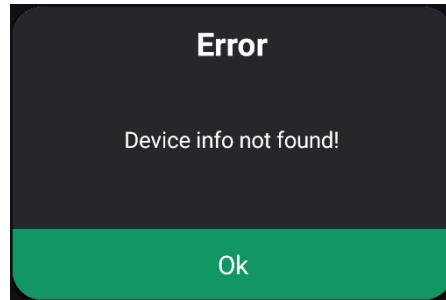


Слика 145 Подаци о уређају су измењени



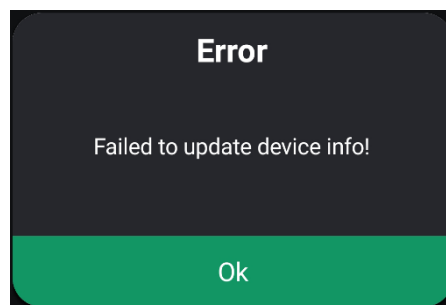
## Алтернативна сценарија

- 7.3. Уколико систем не може да пронађе податке о изабраном уређају, приказује кориснику поруку: „Подаци о уређају нису пронађени!“. Прекида се извршење сценарија (ИА)



*Слика 146 Грешка при враћању уређаја*

- 24.1. Уколико систем не може да сачува измењене податке, приказује кориснику поруку: „Неуспешна измена података о уређају!“ (ИА)



*Слика 147 Грешка при измени података о уређају*

## СК6: Случај коришћења – Брисање уређаја

### Назив СК

Брисање уређаја

### Актер СК

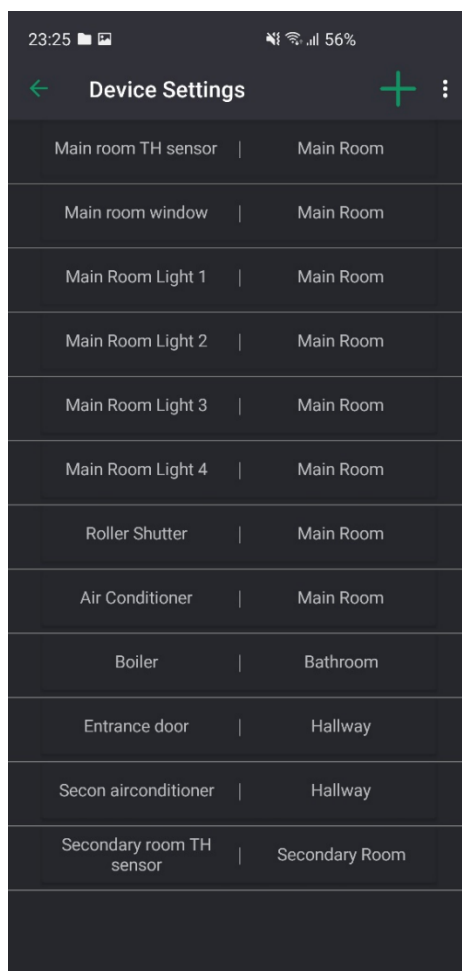
Корисник (Администратор или Власник)

### Учесници СК

Корисник и систем (програм)

### Предуслов

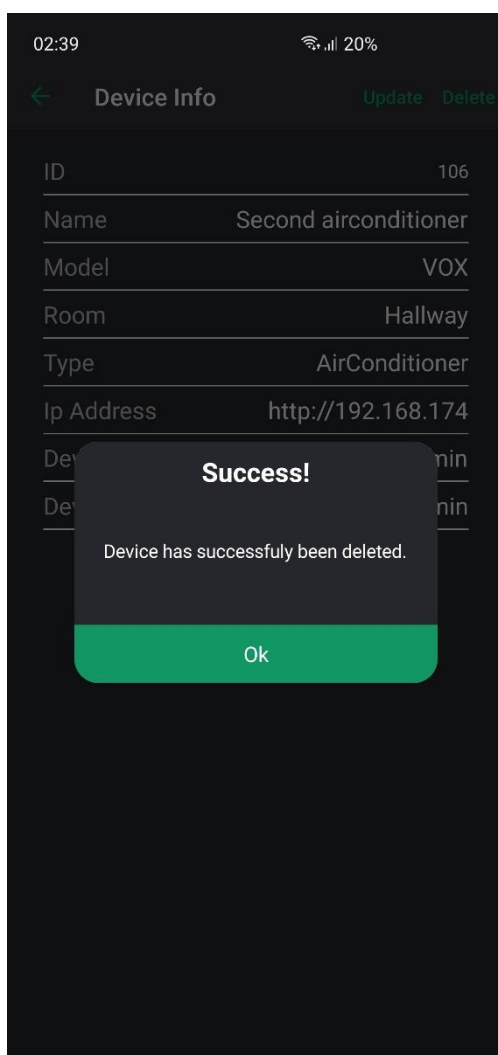
Систем је укључен и корисник је пријављен на своја налог. Систем приказује страницу за управљање уређајима.



Слика 148 Страница за управљање уређајима

## Основни сценарио

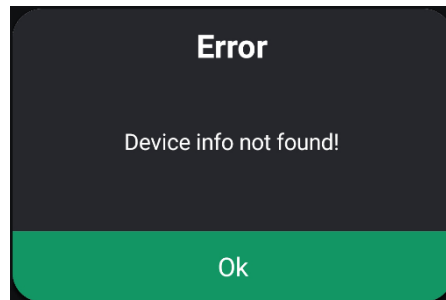
11. Корисник позива систем да прикаже листу уређаја (АПСО)
12. Систем проналази све постојеће уређаје (СО)
13. Систем приказује листу свих уређаја (ИА)
14. Корисник одабира уређај који жели да обрише (АПУСО)
15. Корисник позива систем да прикаже податке о изабраном уређају (АПСО)
16. Систем учитава податке о изабраном уређају (СО)
17. Систем приказује кориснику податке о изабраном уређају (ИА)
18. Корисник позива систем да обрише изабрани уређај (АПСО)
19. Систем врши брисање уређаја (СО)
20. Систем приказује кориснику поруку: „Уређај успешно обрисан.“ (ИА)



Слика 149 Успешно брисање уређаја

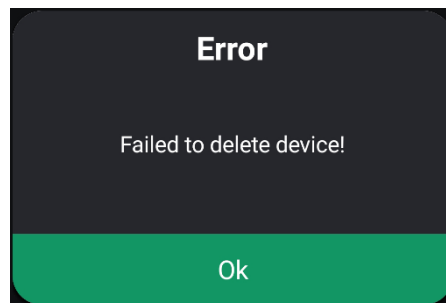
## Алтернативна сценарија

- 7.2. Уколико систем не може да пронађе податке о изабраном уређају, приказује кориснику поруку: „Подаци о уређају нису пронађени!“. Прекида се извршење сценарија (ИА)



*Слика 150 Грешка при враћању уређаја*

- 20.1. Уколико систем не може да обрише уређај, приказује кориснику поруку: „Неуспешно брисање уређаја!“ (ИА)



*Слика 151 Грешка при брисању уређаја*

## СК7: Случај коришћења – Унос собе

### Назив СК

Унос собе

### Актер СК

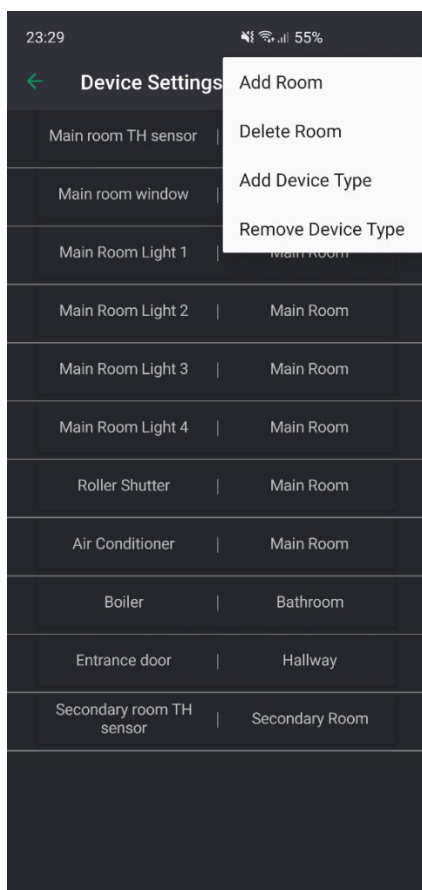
Корисник (Администратор или Власник)

### Учесници СК

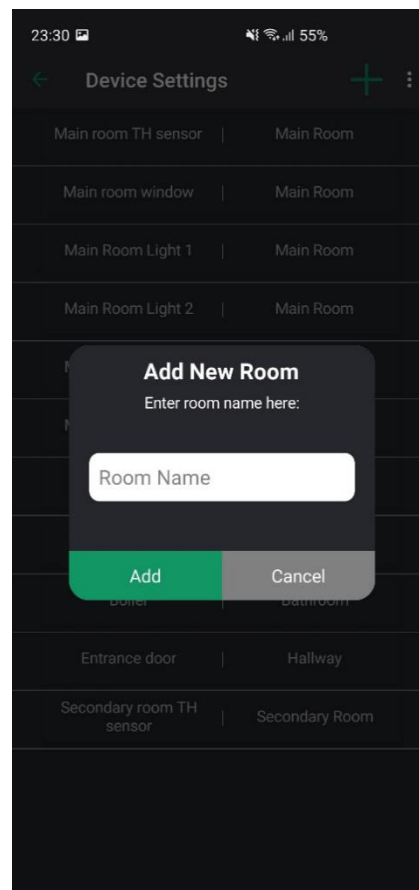
Корисник и систем (програм)

### Предуслов

Систем је укључен и корисник је пријављен на своја налог. Систем приказује прозор за унос собе.



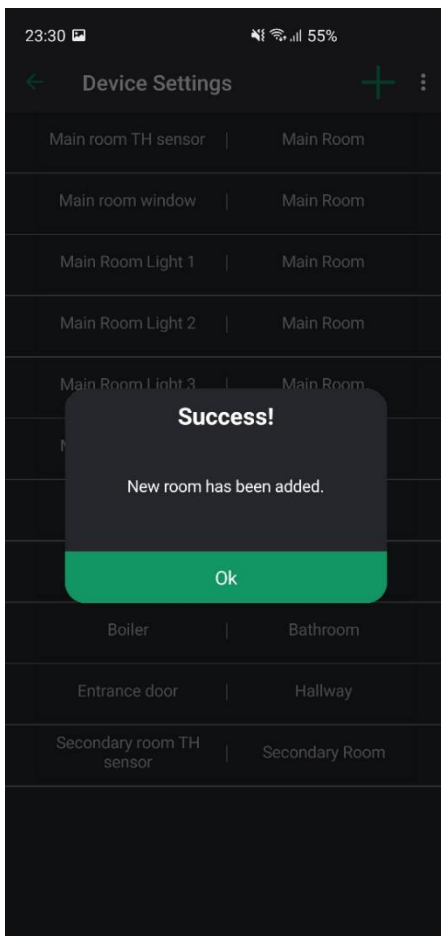
Слика 153 Страница за управљање уређајима



Слика 152 Прозор за унос имена нове собе

## Основни сценарио

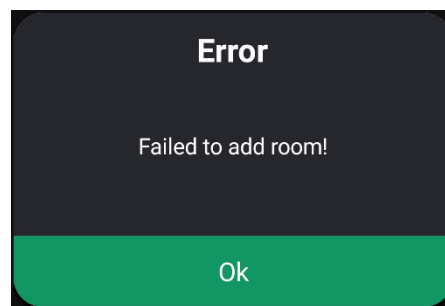
6. Корисник уноси назив нове собе (АПУСО)
7. Корисник проверава да ли је добро унео назив (АНСО)
8. Корисник позива систем да запами податке о новој соби (АПСО)
9. Систем памти податке о соби (СО)
10. Систем приказује кориснику поруку: „Соба успешно унета.“ (ИА)



Слика 154 Успешан унос собе

## Алтернативна сценарија

- 5.3. Уколико систем не може да сачува податке о новој соби, приказује кориснику поруку: „Неуспешан унос собе!“ (ИА)



Слика 155 Грешка при уносу собе

## СК8: Случај коришћења – Брисање собе

### Назив СК

Брисање собе

### Актер СК

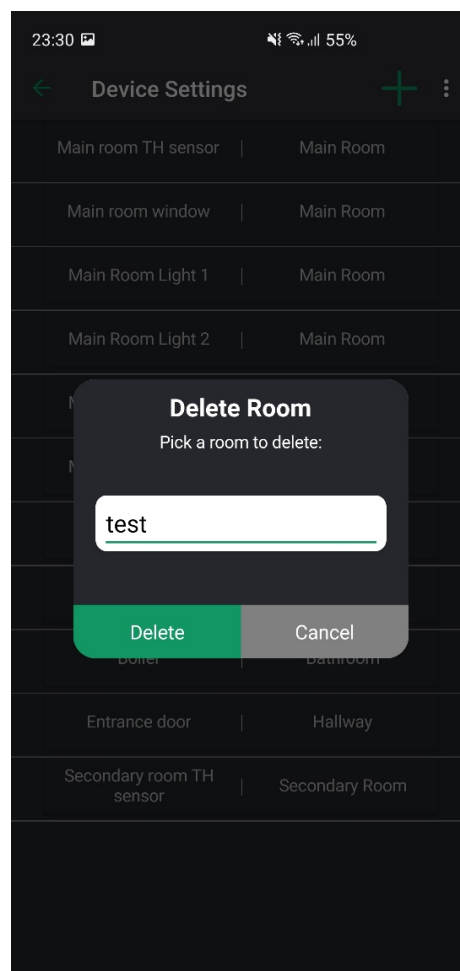
Корисник (Администратор или Власник)

### Учесници СК

Корисник и систем (програм)

### Предуслов

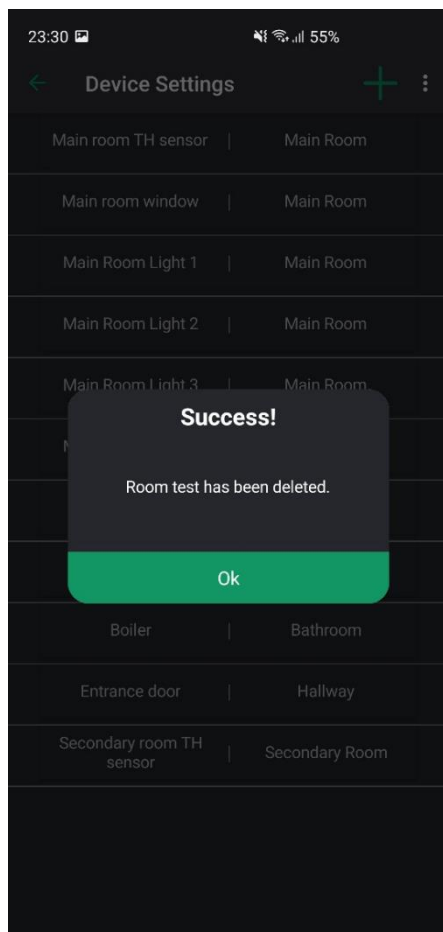
Систем је укључен и корисник је пријављен на своја налог. Систем приказује прозор за брисање собе.



Слика 156 Прозор за брисање собе

## Основни сценарио

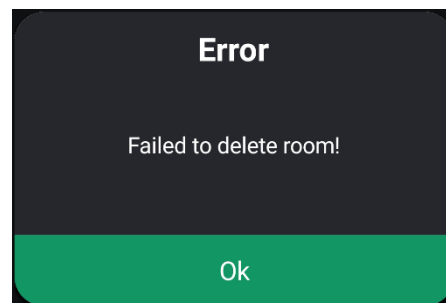
8. Корисник позива систем да прикаже листу соба (АПСО)
9. Систем проналази све постојеће собе (СО)
10. Систем приказује листу свих соба (ИА)
11. Корисник одабира собу коју жели да обрише (АПУСО)
12. Корисник позива систем да обрише изабрану собу (АПСО)
13. Систем врши брисање собе (СО)
14. Систем приказује кориснику поруку: „Соба успешно обрисана.“ (ИА)



Слика 157 Успешно брисање собе

## Алтернативна сценарија

- 14.1. Уколико систем не може да обрише собу, приказује кориснику поруку: „Неуспешно брисање собе!“ (ИА)



Слика 158 Грешка при брисању собе



## СК9: Случај коришћења – Унос типа уређаја

### Назив СК

Унос типа уређаја

### Актер СК

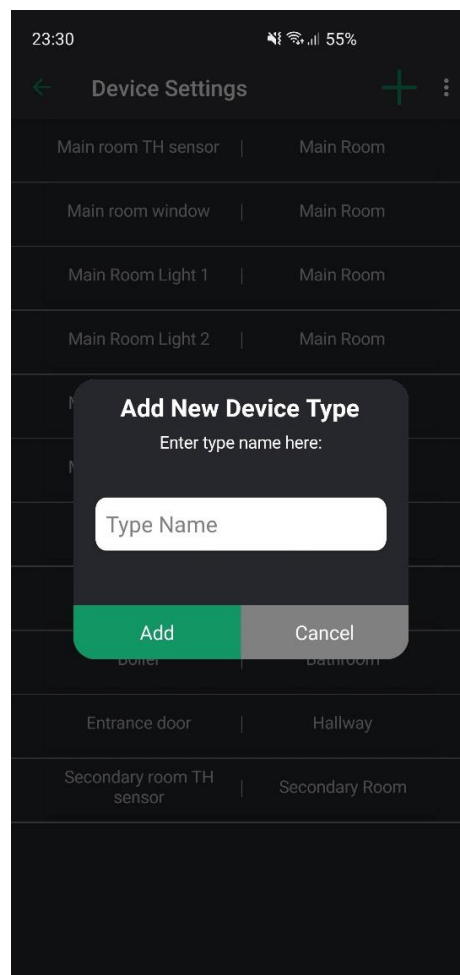
Корисник (Администратор или Власник)

### Учесници СК

Корисник и систем (програм)

### Предуслов

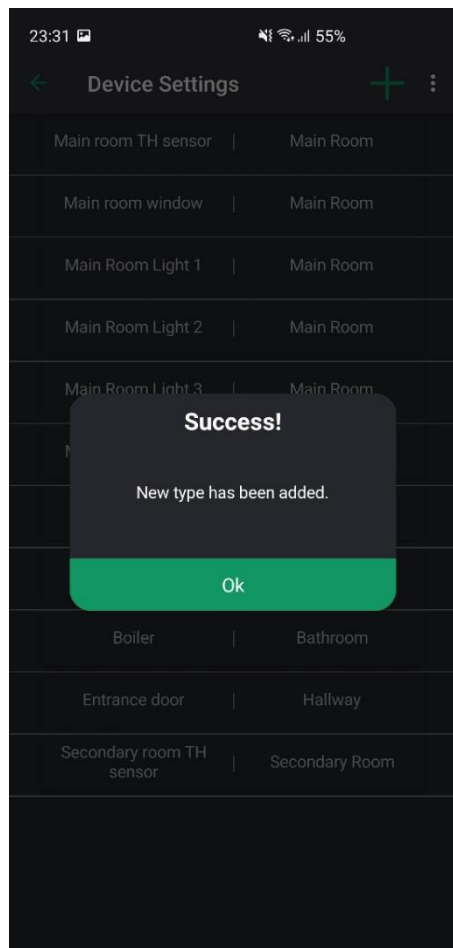
Систем је укључен и корисник је пријављен на своја налог. Систем приказује прозор за унос типа уређаја.



Слика 159 Прозор за унос типа уређаја

## Основни сценарио

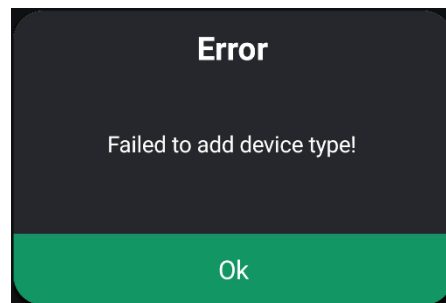
6. Корисник уноси назив новог типа уређаја (АПУСО)
7. Корисник проверава да ли је добро унео назив (АНСО)
8. Корисник позива систем да запами податке о новом типу уређаја (АПСО)
9. Систем памти податке о типу уређаја (СО)
10. Систем приказује кориснику поруку: „Тип уређаја успешно унет.“



Слика 160 Успешан унос типа уређаја

## Алтернативна сценарија

- 10.1. Уколико систем не може да сачува податке о новом типу уређаја, приказује кориснику поруку: „Неуспешан унос типа уређаја!“ (ИА)



Слика 161 Грешка при уносу типа

## СК10: Случај коришћења – Брисање типа уређаја

### Назив СК

Брисање типа уређаја

### Актер СК

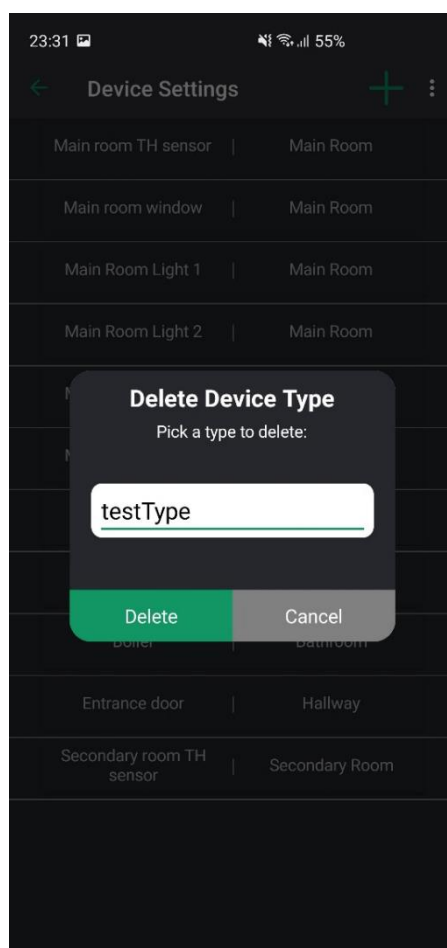
Корисник (Администратор или Власник)

### Учесници СК

Корисник и систем (програм)

### Предуслов

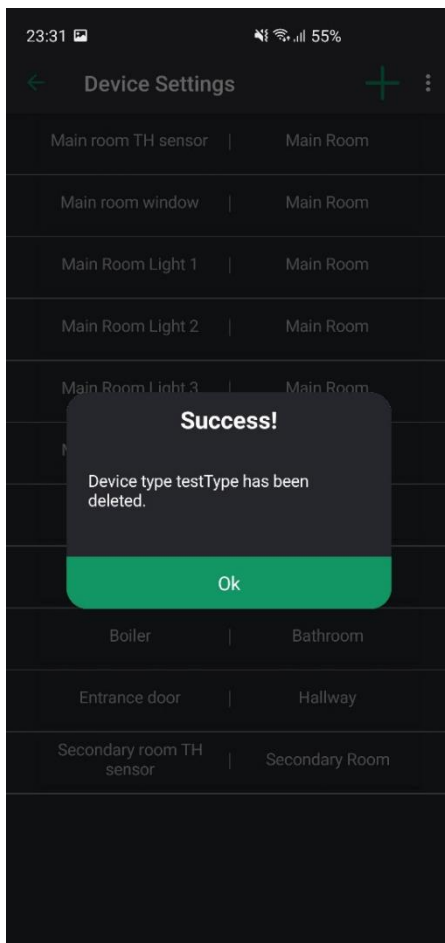
Систем је укључен и корисник је пријављен на своја налог. Систем приказује прозор за брисање типа уређаја.



Слика 162 Прозор за брисање типа уређаја

## Основни сценарио

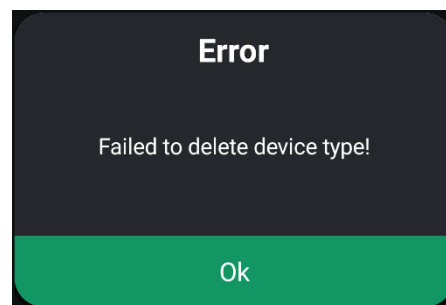
8. Корисник позива систем да прикаже листу типова уређаја (АПСО)
9. Систем проналази све постојеће типове уређаја (СО)
10. Систем приказује листу свих типова уређаја (ИА)
11. Корисник одабира тип уређаја који жели да обрише (АПУСО)
12. Корисник позива систем да обрише избрани тип уређаја (АПСО)
13. Систем врши брисање типа уређаја (СО)
14. Систем приказује кориснику поруку: „Тип уређаја успешно обрисан.“ (ИА)



Слика 163 Успешно брисање типа уређаја

## Алтернативна сценарија

- 14.1. Уколико систем не може да обрише тип уређаја, приказује кориснику поруку: „Неуспешно брисање типа уређаја!“ (ИА)



Слика 164 Грешка при брисању типа

## СК11: Случај коришћења – Измена података о налогу корисника

### Назив СК

Измена података о уређају

### Актер СК

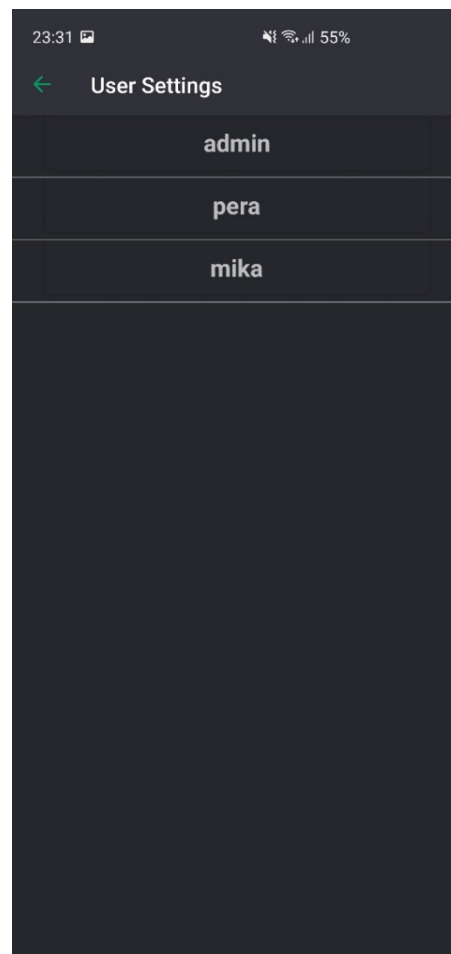
Власник

### Учесници СК

Власник и систем (програм)

### Предуслов

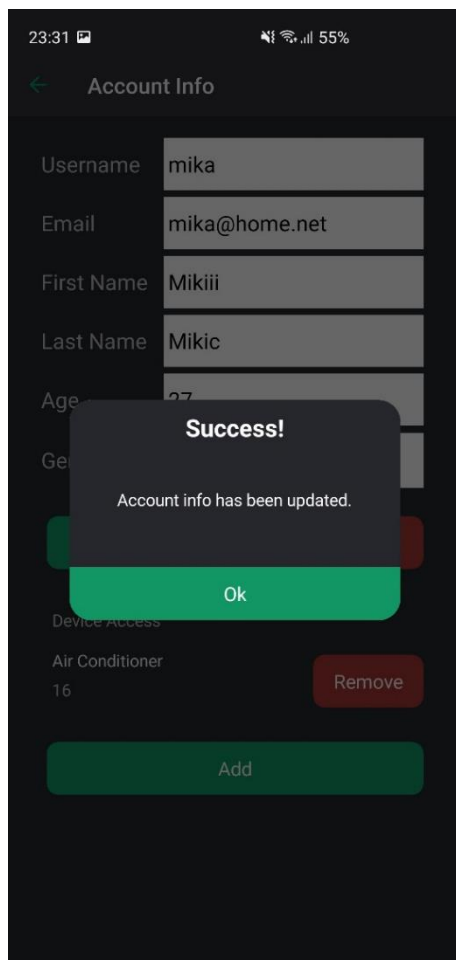
Систем је укључен и власник је пријављен на своја налог. Систем приказује страницу за управљање налозима.



Слика 165 Страница за управљање налозима

## Основни сценарио

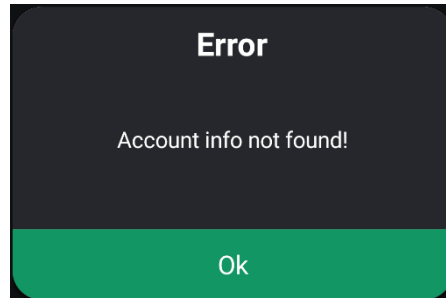
13. Власник позива систем да прикаже листу налога (АПСО)
14. Систем проналази све постојеће налоге (СО)
15. Систем приказује листу свих налога (ИА)
16. Власник одабира налог чије податке жели да измени (АПУСО)
17. Власник позива систем да прикаже податке о изабраном налогу (АПСО)
18. Систем учитава податке о изабраном налогу (СО)
19. Систем приказује власнику податке о изабраном налогу (ИА)
20. Власник мења податке о налогу (АПУСО)
21. Власник проверава да ли је добро унео податке (АНСО)
22. Власник позива систем да запамти унете податке (АПСО)
23. Систем памти измењене податке о налогу (СО)
24. Систем приказује власнику поруку: „Подаци о налогу су успешно измењени.“ (ИА)



Слика 166 Успешна измена података о налогу

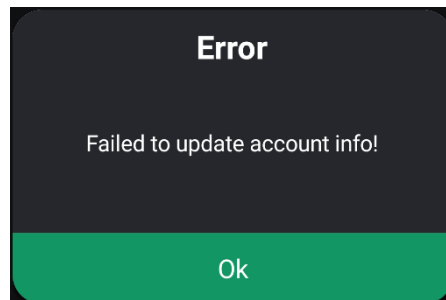
## Алтернативна сценарија

- 7.3. Уколико систем не може да пронађе податке о изабраном налогу, приказује власнику поруку: „Подаци о налогу нису пронађени!“. Прекида се извршење сценарија (ИА)



Слика 167 Грешка при враћању налога

- 24.1. Уколико систем не може да сачува измењене податке, приказује власнику поруку: „Неуспешна измена података о налогу!“ (ИА)



Слика 168 Грешка при измени података о налогу

## СК12: Случај коришћења – Додавање дозволе за коришћење уређаја

### Назив СК

Додавање дозволе за коришћење уређаја

### Актер СК

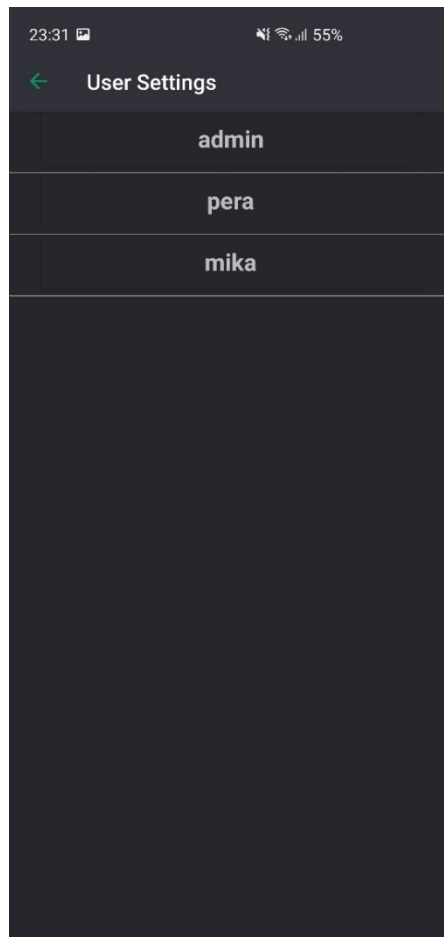
Власник

### Учесници СК

Власник и систем (програм)

### Предуслов

Систем је укључен и власник је пријављен на своја налог. Систем приказује страницу за управљање налозима.

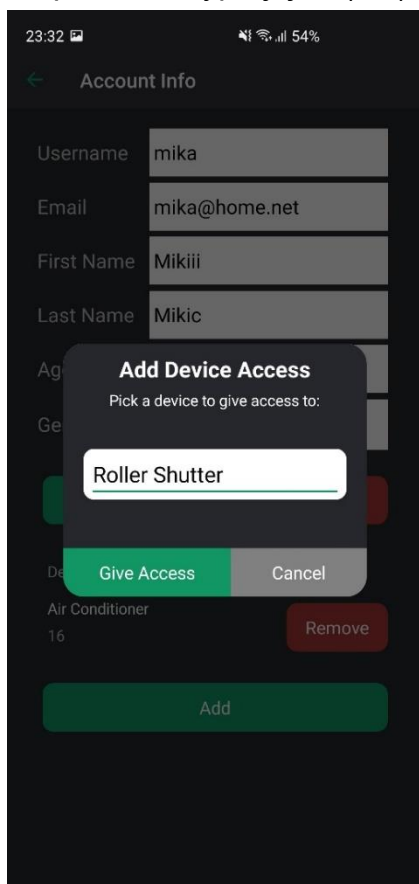


Слика 169 Страница за управљање корисницима

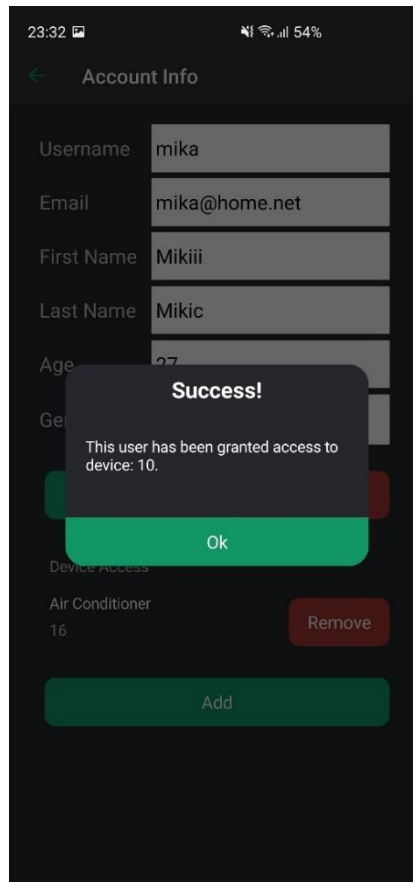


## Основни сценарио

15. Власник позива систем да прикаже листу налога (АПСО)
16. Систем проналази све постојеће налоге (СО)
17. Систем приказује листу свих налога (ИА)
18. Власник одабира налог чије коме жели да дода дозволу за коришћење уређаја (АПУСО)
19. Власник позива систем да врати податке о изабраном кориснику (АПСО)
20. Систем проналази изабраног корисника (СО)
21. Систем приказује податке о изабраном кориснику (ИА)
22. Власник позива систем да прикаже листу уређаја (АПСО)
23. Систем проналази све постојеће уређаје (СО)
24. Систем приказује листу свих уређаја (ИА)
25. Власник одабира уређај за који даје дозволу коришћења (АПУСО)
26. Власник позива систем да запамти податке о дозволи коришћења изабраног уређаја за претходно изабраног корисника (АПСО)
27. Систем памти дозволу коришћења уређаја (СО)
28. Систем приказује власнику поруку: „Успешно додавање дозволе коришћења уређаја.“ (ИА)



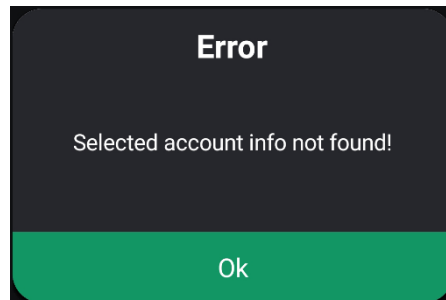
Слика 171 Прозор за одабир уређаја



Слика 170 Успешан унос дозволе

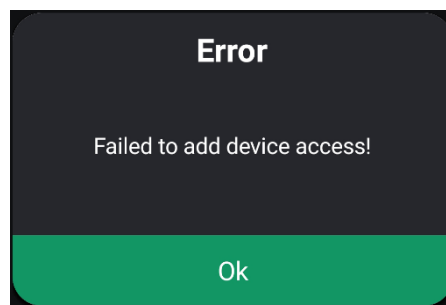
## Алтернативна сценарија

- 7.3. Уколико систем не може да пронађе корисника, приказује власнику поруку: „Подаци о изабраном кориснику нису пронађени!“. Прекида се извршење сценарија (ИА)



Слика 172 Грешка при враћању налога

- 28.1. Уколико систем не може да сачува дозволу коришћења, приказује власнику поруку: „Неуспешна додавање дозволе!“ (ИА)



Слика 173 Грешка при додавању дозволе

## СК13: Случај коришћења – Брисање дозволе за коришћење уређаја

### Назив СК

Брисање дозволе за коришћење уређаја

### Актер СК

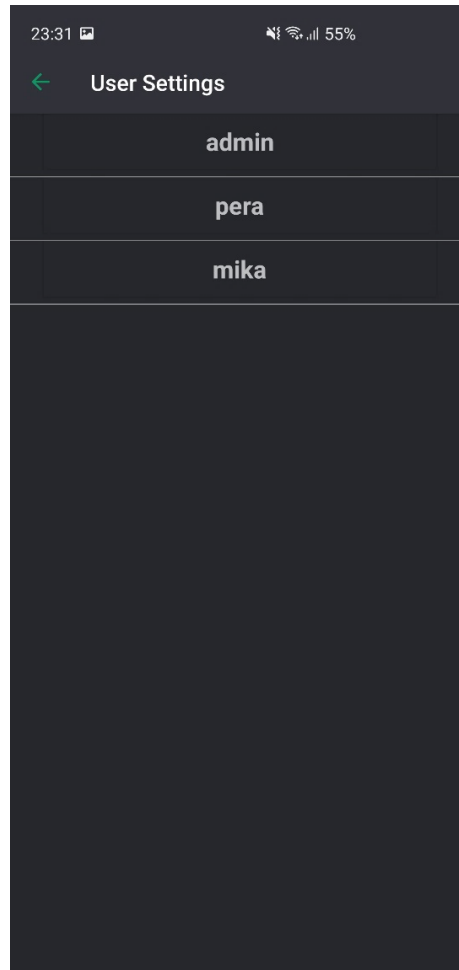
Власник

### Учесници СК

Власник и систем (програм)

### Предуслов

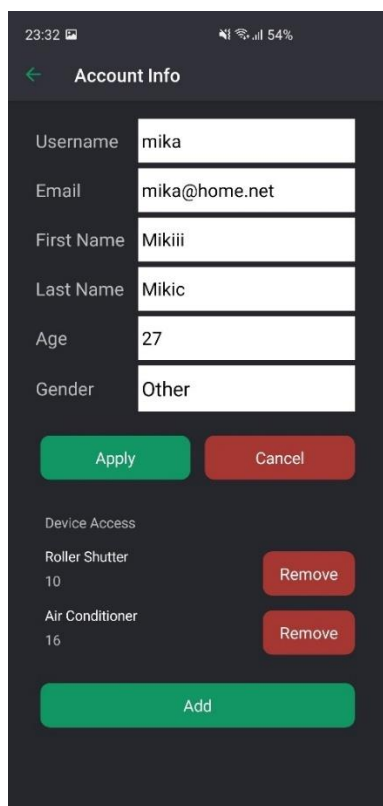
Систем је укључен и власник је пријављен на своја налог. Систем приказује страницу за управљање налозима.



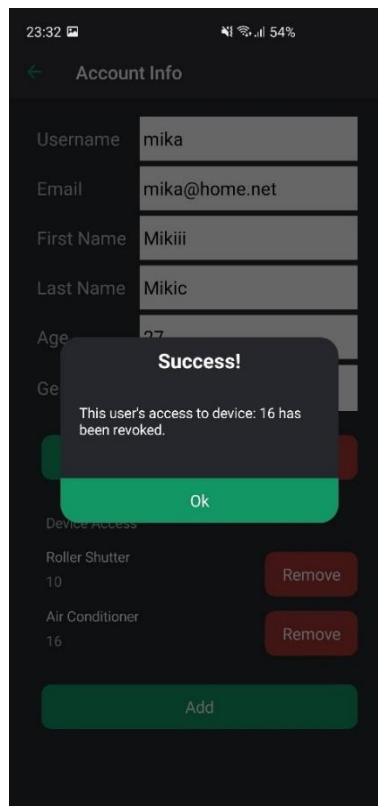
Слика 174 Страница за управљање налозима

## Основни сценарио

15. Власник позива систем да прикаже листу налога (АПСО)
16. Систем проналази све постојеће налоге (СО)
17. Систем приказује листу свих налога (ИА)
18. Власник одабира налог чије коме жели да обрише дозволу за коришћење уређаја (АПУСО)
19. Власник позива систем да врати податке о изабраном кориснику (АПСО)
20. Систем проналази изабраног корисника (СО)
21. Систем приказује податке о изабраном кориснику (ИА)
22. Власник позива систем да прикаже листу уређаја којима изабрани корисник има приступ (АПСО)
23. Систем проналази све постојеће уређаје који одговарају критеријуму (СО)
24. Систем приказује листу свих уређаја који одговарају критеријуму (ИА)
25. Власник одабира уређај за који брише дозволу коришћења (АПУСО)
26. Власник позива систем да обрише дозволу коришћења изабраног уређаја за претходно изабраног корисника (АПСО)
27. Систем брише дозволу коришћења уређаја (СО)
28. Систем приказује власнику поруку: „Успешно брисање дозволе коришћења уређаја.“ (ИА)



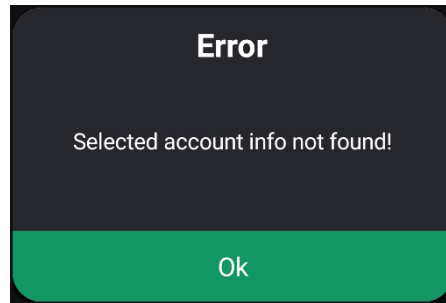
Слика 176 Страница за управљање подацима о налогу



Слика 175 Успешно брисање дозволе

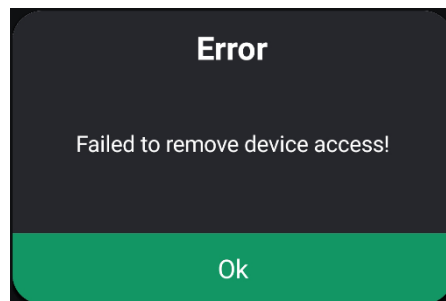
## Алтернативна сценарија

- 7.3. Уколико систем не може да пронађе корисника, приказује власнику поруку: „Подаци о изабраном кориснику нису пронађени!“. Прекида се извршење сценарија (ИА)



Слика 177 Грешка при враћању налога

- 28.1. Уколико систем не може да обрише дозволу коришћења, приказује власнику поруку: „Неуспешно брисање дозволе!“ (ИА)



Слика 178 Грешка при брисању дозволе

## 8. Фаза тестирања

Тестирање апликације вршено је ручно, уносом како исправних, тако и неисправних података, тестирани су механизми валидације, али и сама функционалност апликације. На овај начин све уочене грешке бивале су уклањане и исправљане. Након довољно поновљених тестова који нису давали грешке закључујемо да је апликација потпуно функционална и испуњава постављене захтеве.

## 9. Закључак

У раду „Развој софтверског система за управљање уређајима у паметној кући употребом .NET технологија“ описано је развијање софтверског система који служи за управљање паметним уређајима у домаћинству.

У теоријском делу приказани су основни концепти технологија коришћених у изради самог пројекта, као и процеса који је до његовог настанка довео. Практични део даје нам увид у развој софтверског система реализованог имплементацијом датих технологија.

Креирање овог софтверског система инспирисано је пре свега све већом дигитализацијом света у коме живимо, док се у својим домовима некако и даље служимо застарелим, мануелним технологијама и начинима рада. Приметио сам да би данашњем човеку, у складу са ритмом живота који живи, више одговарало да контролу над својим домом има „на дохват руке“ и да му функције његовог дома буду увек доступне. Вођен бенефитима које овај систем већ доноси у мом стану, креирао сам апликацију која би многим људима олакшала убрзан живот, као и смањила напор старијим суграђанима и помогла им у испуњавању свакодневних задатака у њиховом дому.

Представивши идеју, а касније и саму апликацију многим колегама, пријатељима и члановима породице сусрео сам се првенствено са великим бројем позитивних критика, али још важније са великим интересовањем за поседовање и инсталирање поменутог система и у њиховим домовима. Иако је апликација зарадила много похвала, она, наравно, има доста простора за унапређивање.

## 10. Литература

- [1] Vlajić, S. (2015). Projektovanje softvera (skripta). Београд, Србија: Факултет организационих наука
- [2] Microsoft. (2021, Март 18). A tour of the C# language. Преузето са: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/> [последњи приступ 2022, Јул 12]
- [3] Microsoft. (2022, Март 30). Overview of classes, structs, and records in C#. Преузето са: <https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/object-oriented> [последњи приступ 2022, Јул 12]
- [4] Microsoft. (2021, Фебруар 16). Inheritance - derive types to create more specialized behavior. Преузето са: <https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/object-oriented/inheritance> [последњи приступ 2022, Јул 12]
- [5] Microsoft. (2021, Новембар 6). Polymorphism. Преузето са: <https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/object-oriented/polymorphism> [последњи приступ 2022, Јул 12]
- [6] Microsoft. (2022, Март 11). Asynchronous programming. Преузето са: <https://docs.microsoft.com/en-us/dotnet/csharp/async> [последњи приступ 2022, Јул 12]
- [7] Mahesh Chand (2022, Фебруар 21). What Is New In .NET 6.0. Преузето са: <https://www.c-sharpcorner.com/article/what-is-new-in-net-6-0/> [последњи приступ 2022, Јул 12]
- [8] Microsoft. What is Xamarin. Преузето са: <https://dotnet.microsoft.com/en-us/learn/xamarin/what-is-xamarin> [последњи приступ 2022, Јул 12]
- [9] Microsoft. Xamarin.Forms. Преузето са: <https://dotnet.microsoft.com/en-us/apps/xamarin/xamarin-forms> [последњи приступ 2022, Јул 12]
- [10] Microsoft. ASP.NET. Преузето са: <https://dotnet.microsoft.com/en-us/apps/aspnet> [последњи приступ 2022, Јул 12]
- [11] Microsoft. (2021, Мај 25). Entity Framework Core. Преузето са: <https://docs.microsoft.com/en-us/ef/core/> [последњи приступ 2022, Јул 12]
- [12] Jwt.io. Introduction to JSON Web Tokens. Преузето са: <https://jwt.io/introduction> [последњи приступ 2022, Јул 12]
- [13] SuperTokens Team. What is a JWT? Understanding JSON Web Tokens. Преузето са: <https://supertokens.com/blog/what-is-jwt> [последњи приступ 2022, Јул 12]
- [14] Microsoft. (2022, Јул 10). Introduction to ASP.NET Identity. Преузето са: <https://docs.microsoft.com/en-us/aspnet/identity/overview/getting-started/introduction-to-aspnet-identity> [последњи приступ 2022, Јул 12]



- [15] Abhishek Arora. (2021, Април 12). Introduction to ASP.Net Identity 2.0. Преузето са: <https://www.c-sharpcorner.com/UploadFile/16101a/introduction-to-Asp-Net-identity-2-0/> [последњи приступ 2022, Јул 12]
- [16] Wallarm. A simple explanation of what a WebSocket is. Преузето са: <https://www.wallarm.com/what/a-simple-explanation-of-what-a-websocket-is> [последњи приступ 2022, Јул 12]
- [17] Microsoft. (2021, Јул 8). The Model-View-ViewModel Pattern. Преузето са: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm>
- [18] Microsoft. (2021, Март 3). Xamarin.Forms Shell. Преузето са: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/app-fundamentals/shell/> [последњи приступ 2022, Јул 12]
- [19] Microsoft. (2021, Јул 8). Xamarin.Forms Data Binding. Преузето са: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/app-fundamentals/data-binding/> [последњи приступ 2022, Јул 12]
- [20] ESP8266. Преузето са: <https://en.wikipedia.org/wiki/ESP8266> [последњи приступ 2022, Јул 12]