

MENDELOVA UNIVERZITA V BRNĚ

PROVOZNĚ EKONOMICKÁ FAKULTA

SOFTWARE INŽENÝRSTVÍ I

System pro správu 3D tisku

Author:

Bc. Ladislav KOLÁČEK

Bc. Luboš JURÁNEK

Bc. Jirka PAVLÍK

Supervisor:

Ing. Jan KOLOMAZNÍK

12. ledna 2015

Obsah

1	Úvod	3
2	Eriksson-Penker diagram	3
2.1	Správa modelů	4
2.2	Správa front	5
2.3	Správa notifikací	6
2.4	Správa barev	6
2.5	Správa uživatelů	8
3	Správa uživatelů	8
3.1	Use case diagram	8
3.2	Přihlášení uživatele	11
3.2.1	Scénář případů užití	11
3.2.2	Diagram aktivit	12
3.3	Změna hesla	13
3.3.1	Scénář případů užití	13
3.3.2	Diagram aktivit	14
3.4	Vymazání uživatele	15
3.4.1	Scénář případů užití	15
3.4.2	Diagram aktivit	16
4	Správa notifikací	17
4.1	Diagram tříd	17
4.2	Diagram tříd - oprava	17
4.3	Sekvenční diagramy	18
4.3.1	Vytvoření notifikace	18
4.3.2	Modifikace notifikace	19
4.3.3	Vymazání notifikace	20
4.4	Stavový diagram	22
5	Závěr	23

Seznam obrázků

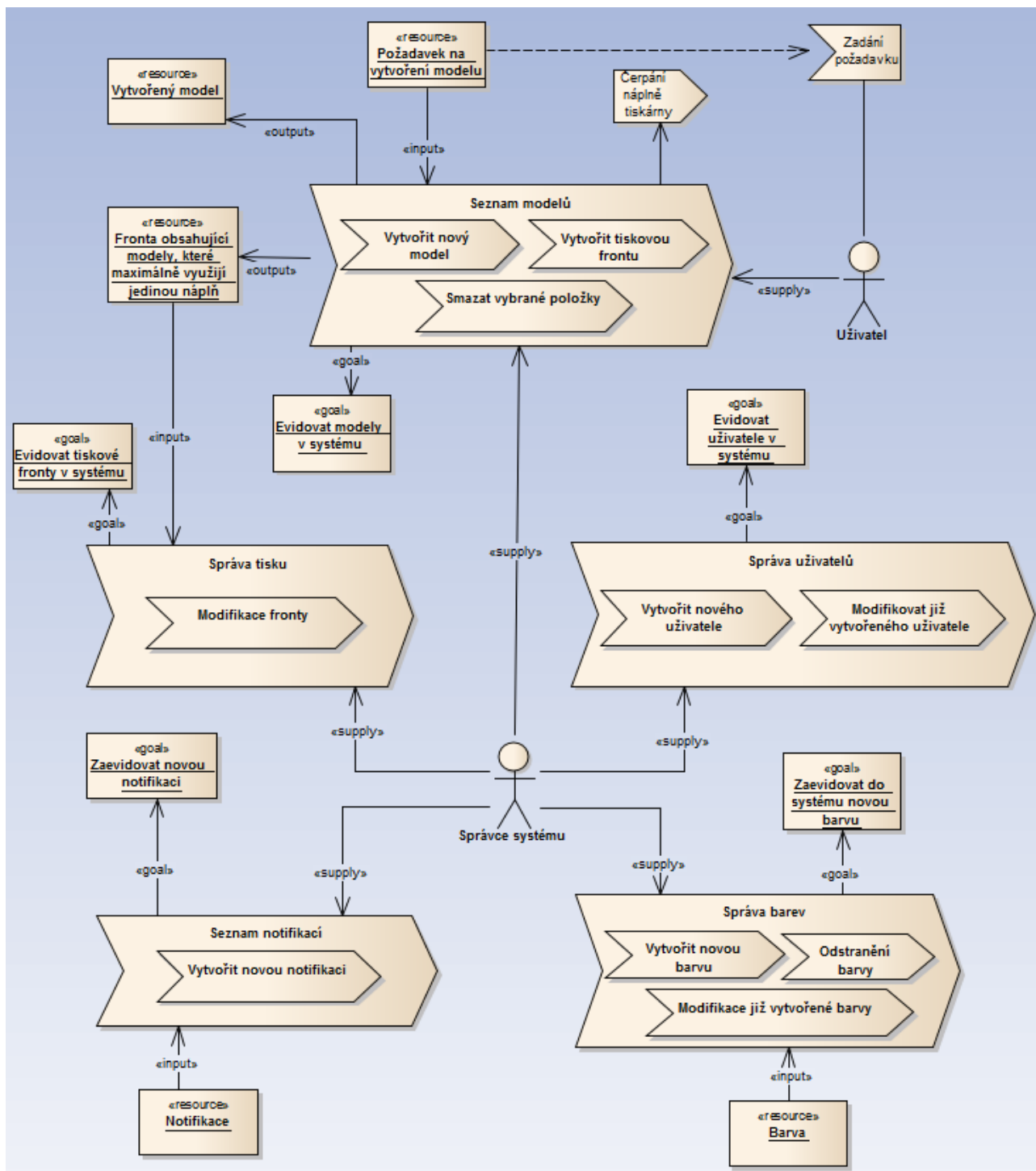
1	Diagram Eriksson-Penker 1. vrstva	4
2	Diagram Eriksson-Penker 2. vrstva - systém pro správu modelů	5
3	Diagram Eriksson-Penker 2. vrstva - systém pro správu front	6
4	Diagram Eriksson-Penker 2. vrstva - systém pro správu notifikací	6
5	Diagram Eriksson-Penker 2. vrstva - systém pro správu barev	7
6	Diagram Eriksson-Penker 2. vrstva - systém pro správu uživatelů	8
7	Use case diagram - systém pro správu uživatelů	10
8	Diagram aktivit - přihlášení uživatele	12
9	Diagram aktivit - změna hesla	14
10	Diagram aktivit - vymazání uživatele	16
11	Diagram tříd	17
12	Diagram tříd - upravený návrh	18
13	Sekvenční diagram - vytvoření notifikace	19
14	Sekvenční diagram - modifikace notifikace	20
15	Sekvenční diagram - vymazání notifikace	21
16	Stavový diagram	22

1 Úvod

V tomto semestrálním projektu je popsána analýza a návrh částí systému pro správu 3D tisku na Mendelově univerzitě. Konkrétně obsahuje Eriksson-Penker diagram pro celý systém, a dále se zabývá pouze některými vybranými částmi systému: Správou uživatelů, která byla popsána diagramem a scénáři případů užití a diagramy aktivit. A poté správou notifikací, kterou popisuje diagram tříd, sekvenční diagramy a stavový diagram.

2 Eriksson-Penker diagram

V prvopočátku byl navržen kompletní systém pro správu 3D tiskárny v podobě diagramu Eriksson-Penker. Všechny následující návrhy budou založeny právě na následujícím diagramu. Při samotném návrhu diagramu bylo nutné určit, kteří aktéři budou s daným systémem participovat. Dále bylo nutné daný systém rozdělit do několika procesů, aby byla jeho realizace co nejjednodušší. U každého z pěti hlavních procesů jsme návrh provedli až do 2. vrstvy. Každý proces má v návrhu své zdroje a cíle. Je třeba zdůraznit, že se jedná pouze o hrubý návrh systému a spousta detailů bude teprve upřesněna v následujících diagramech.



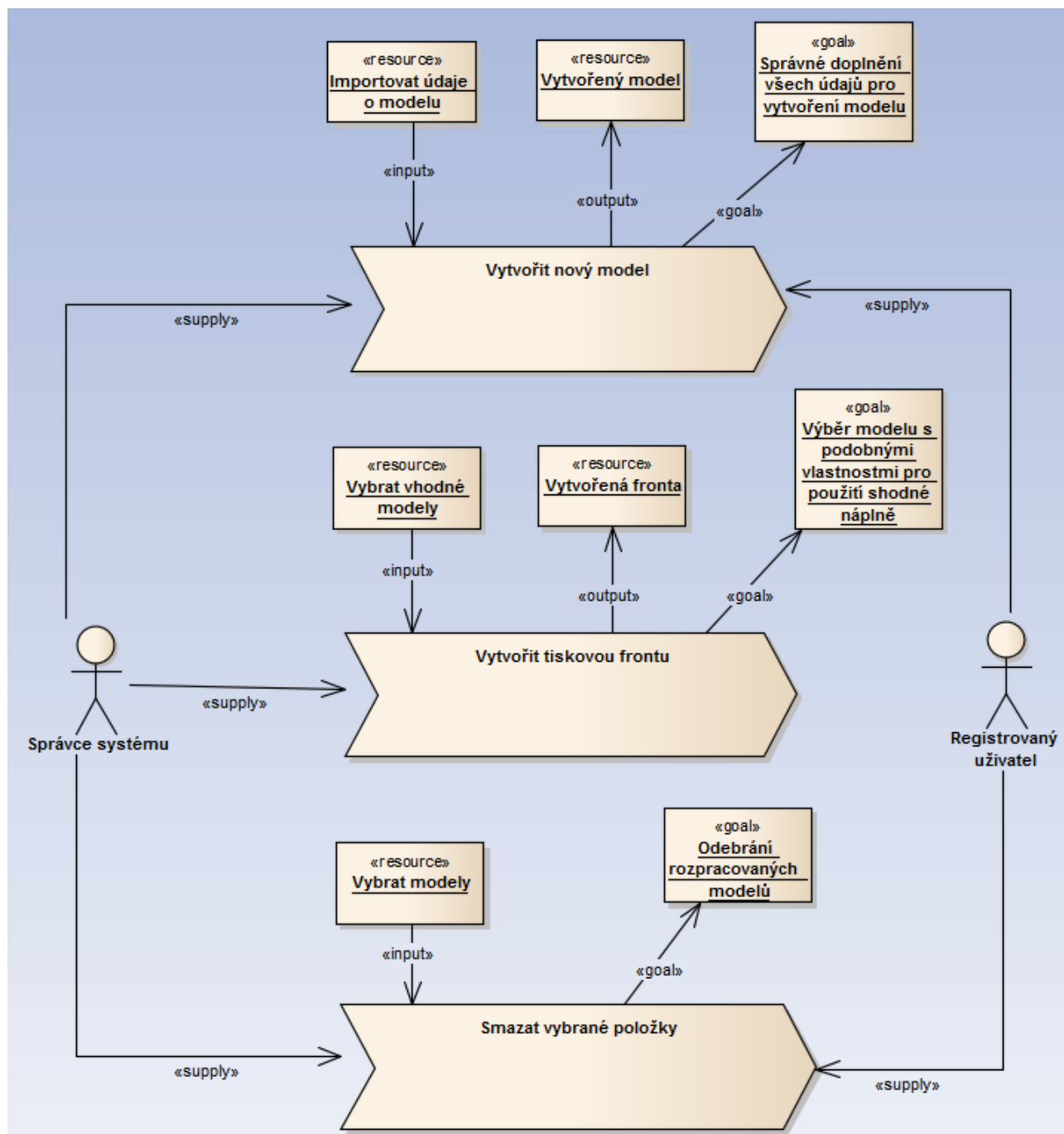
Obrázek 1: Diagram Eriksson-Penker 1. vrstva

Z návrhu na obrázku 1 je patrné, že uživatel by měl komunikovat pouze se správou modelů. Naproti tomu správce systému má samozřejmě pravomoce ke komunikaci se všemi procesy.

2.1 Správa modelů

V procesu správa modelů v systému participují oba aktéři, správce systému i uživatel. Aby uživatel byl schopen vytvořit model, anebo smazat svoji položku v systému, musí v něm být nejprve registrován. Vytvořit tiskovou frontu, dle našeho návrhu, smí pouze správce systému.

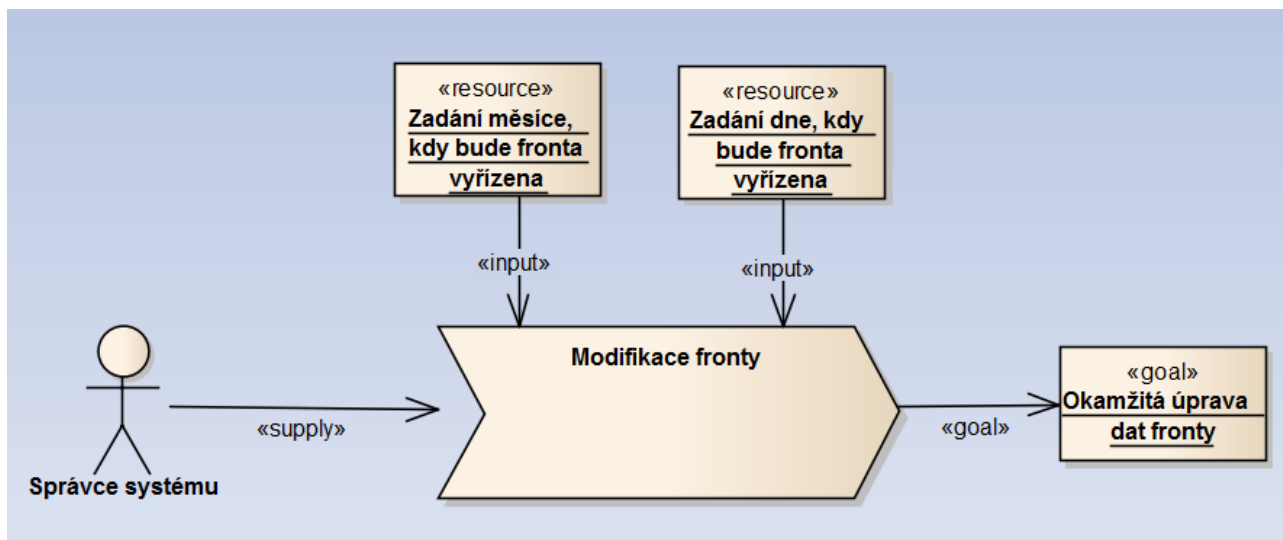
Tisková fronta se vytváří s cílem efektivně využít barvu při tisku. Uživatel není povinen se starat o využívání barvy v tiskárně.



Obrázek 2: Diagram Eriksson-Penker 2. vrstva - systém pro správu modelů

2.2 Správa front

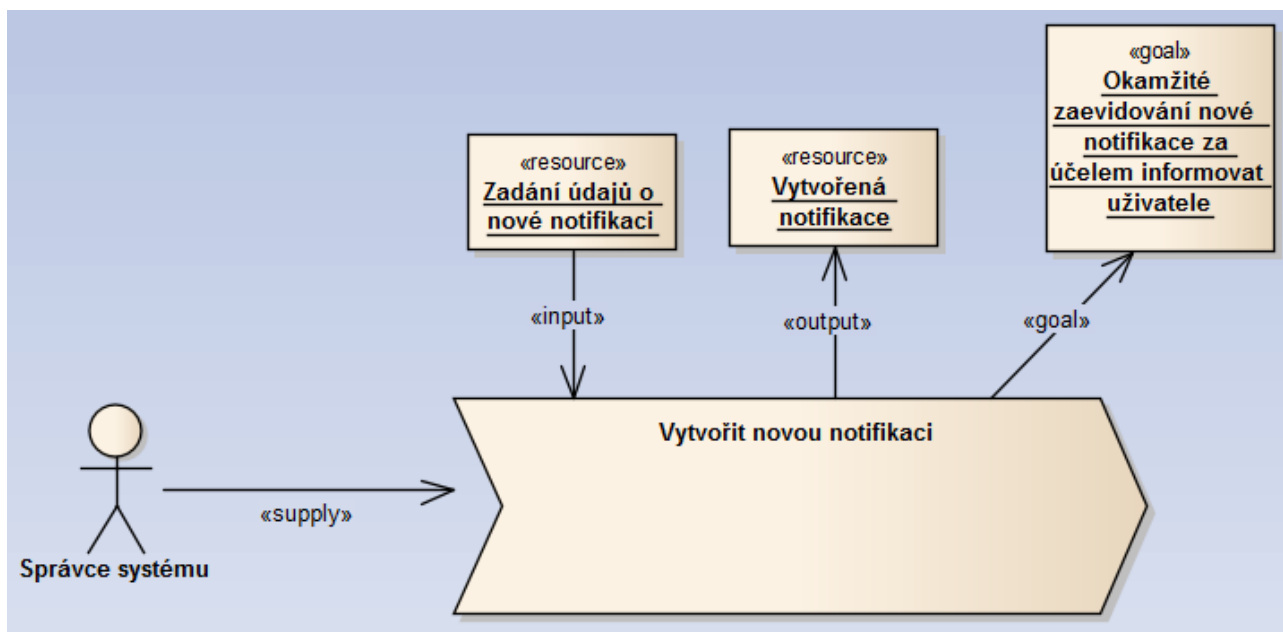
V procesu správa front má správce systému jedinou možnost, a to modifikovat již vytvořenou frontu v rámci správy modelů. Proces modifikace fronty vyžaduje doplnění časových údajů o vyřízení fronty.



Obrázek 3: Diagram Eriksson-Penker 2. vrstva - systém pro správu front

2.3 Správa notifikací

V procesu správa notifikací má správce systému jedinou možnost, a to vytvořit novou notifikaci. Tento proces vyžaduje zadání údajů o nově vzniklé notifikaci, která má za úkol informovat uživatele.

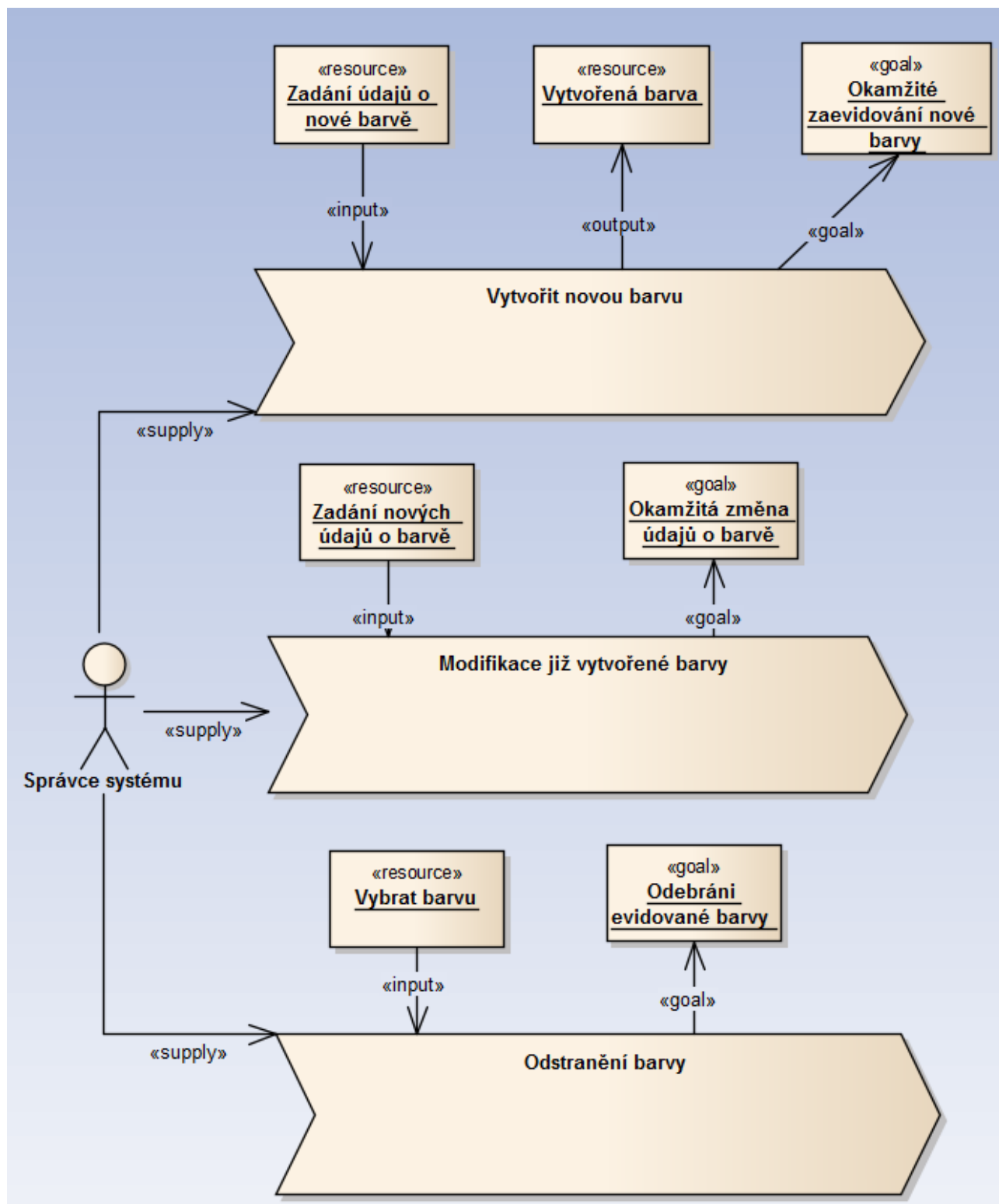


Obrázek 4: Diagram Eriksson-Penker 2. vrstva - systém pro správu notifikací

2.4 Správa barev

V procesu správa barev má správce systému možnost vytvořit novou barvu, modifikovat ji, anebo ji ze systému úplně odebrat v případě, že byla používána jen minimálně a nevyplatí se ji

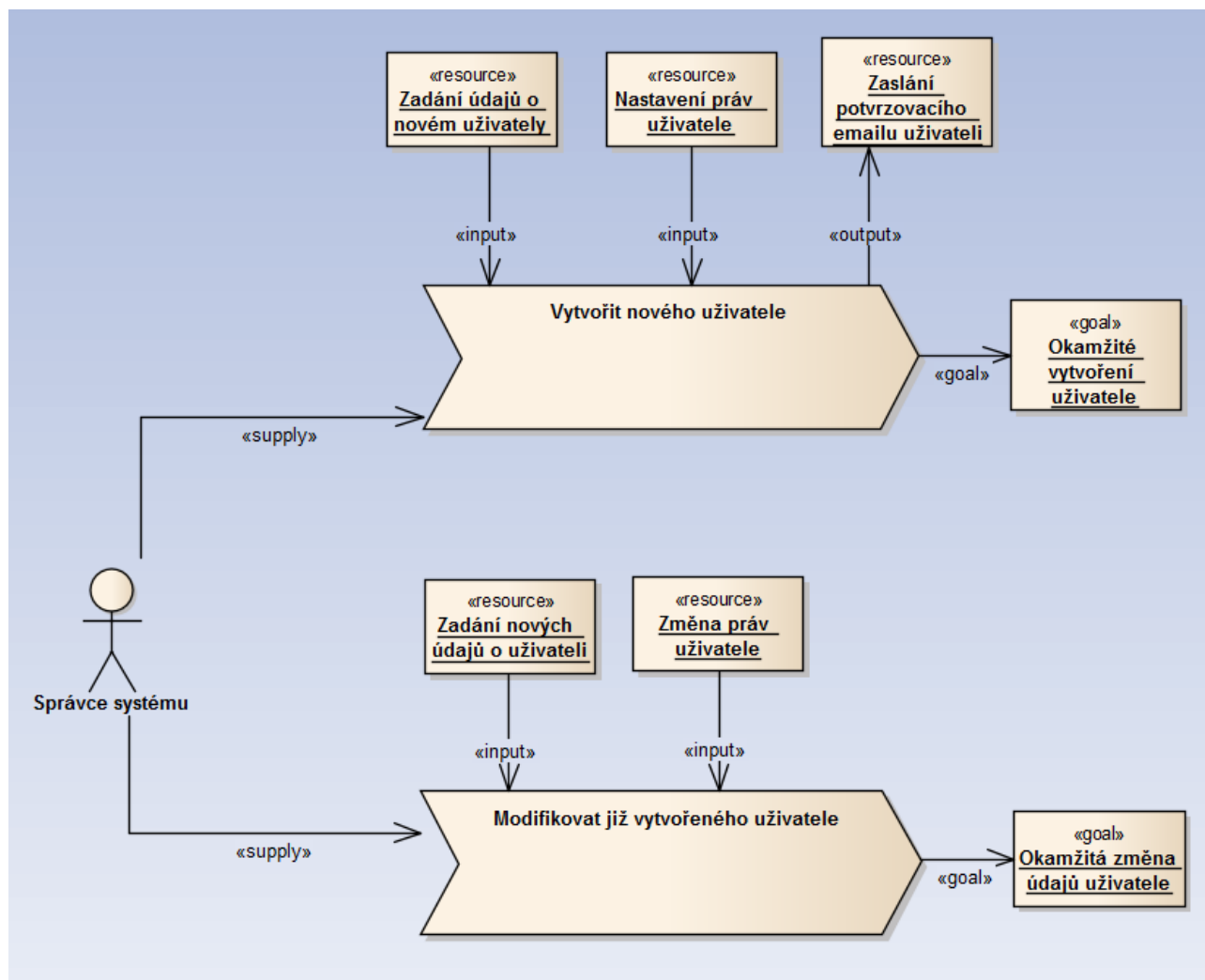
znovu doplnit. Proces vytvoření a modifikace barvy vyžaduje zadání údajů o právě zpracovávané barvě.



Obrázek 5: Diagram Eriksson-Penker 2. vrstva - systém pro správu barev

2.5 Správa uživatelů

V procesu správa uživatelů má správce systému možnost vytvořit úplně nového uživatele, anebo modifikovat údaje o stávajícím uživateli. Obě tyto akce vyžadují zadání údajů o uživateli a nastavení jeho práv v systému.



Obrázek 6: Diagram Eriksson-Penker 2. vrstva - systém pro správu uživatelů

3 Správa uživatelů

Druhou částí projektu bylo rozebrat jednu z částí celého systému pro Správu 3D tisku, a to správu uživatelů. Tuto část jsme následně měli popsat use case diagramem a diagramy aktivit.

3.1 Use case diagram

S jednotlivými případy užití systému komunikuje pět aktérů. Tři z nich jsou aktivní, nepřihlášený uživatel, přihlášený uživatel a administrátor. A dva z nich jsou pasivní, čas a mail-server.

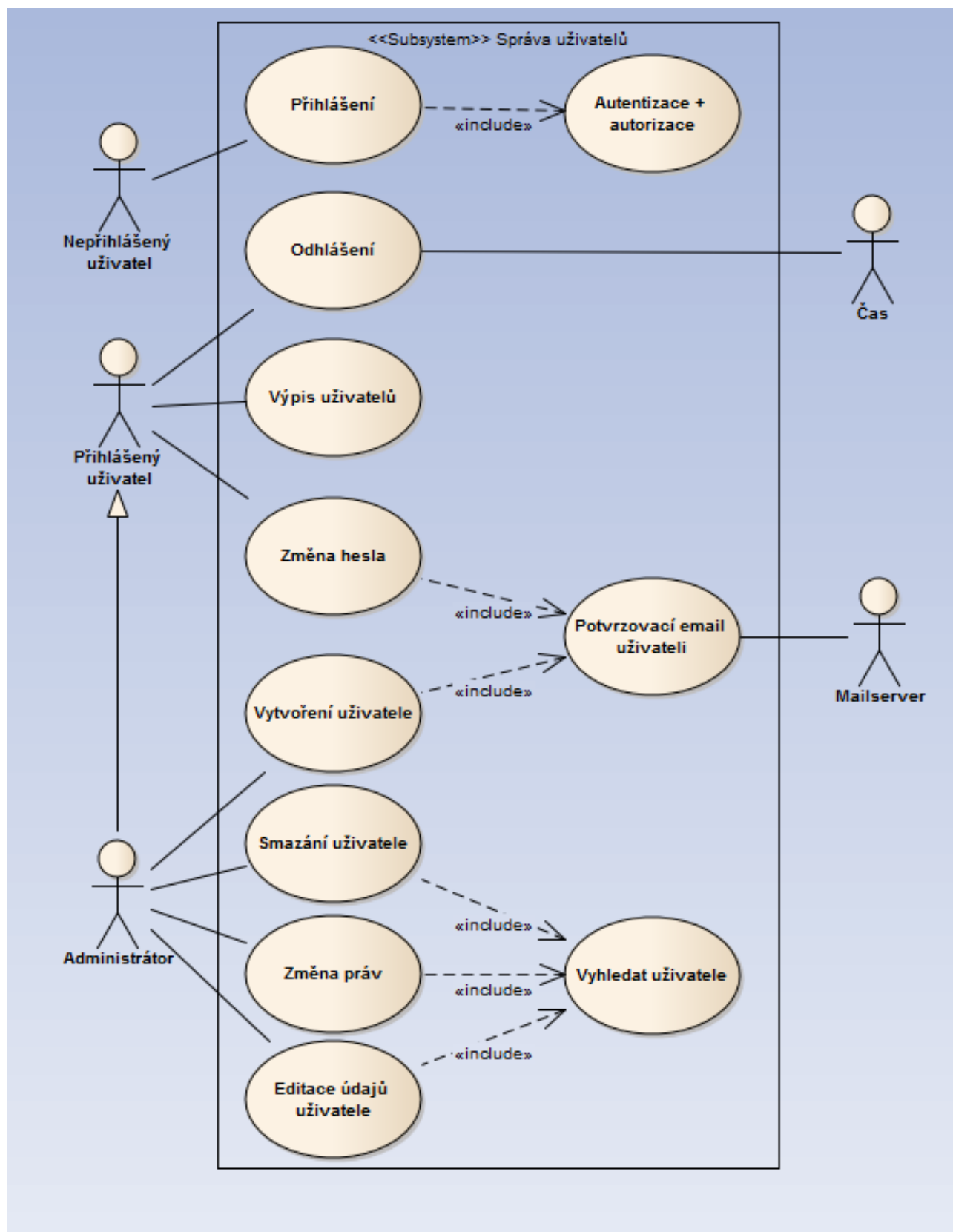
Nepřihlášený uživatel je aktér, který nemá žádná oprávnění v systému cokoliv provádět,

kromě vlastního přihlášení, po kterém, pokud proběhne správně autentizace a autorizace uživatele, se nepřihlášený uživatel stane uživatelem přihlášeným.

Přihlášený uživatel má v rámci správy uživatelů tři případy užití, kterými může komunikovat se systémem. Může se odhlásit, čímž by se znovu stal nepřihlášeným uživatelem. Pokud toto neudělá po nějakou stanovenou dobu, například 24 hodin, systém jej sám odhlásí pomocí pasivního aktéra čas. Dále si může nechat systémem vypsat všechny uživatele nebo si změnit heslo.

Administrátor má stejné možnosti jako přihlášený uživatel, ale navíc může vytvářet účty uživatelům nebo ovlivňovat již vytvořené účty uživatelů. Při vytvoření nového účtu uživatele je tomuto uživateli odeslán potvrzovací email pasivním aktérem mailserver, aby se o této skutečnosti dozvěděl. Pro bezpečnost je email posílán také v případě změny hesla. Ovlivňováním již vytvořených účtů uživatelů se v tomto případě myslí jejich mazání, změna jejich práv či editace údajů. Ve všech těchto případech musí administrátor nejdříve vyhledat konkrétní účet uživatele přes výpis uživatelů.

Dále jsou podrobněji popsány tři případy užití, přihlášení uživatele, změna hesla a vymazání uživatele pomocí scénářů případů užití a diagramů aktivit.



Obrázek 7: Use case diagram - systém pro správu uživatelů

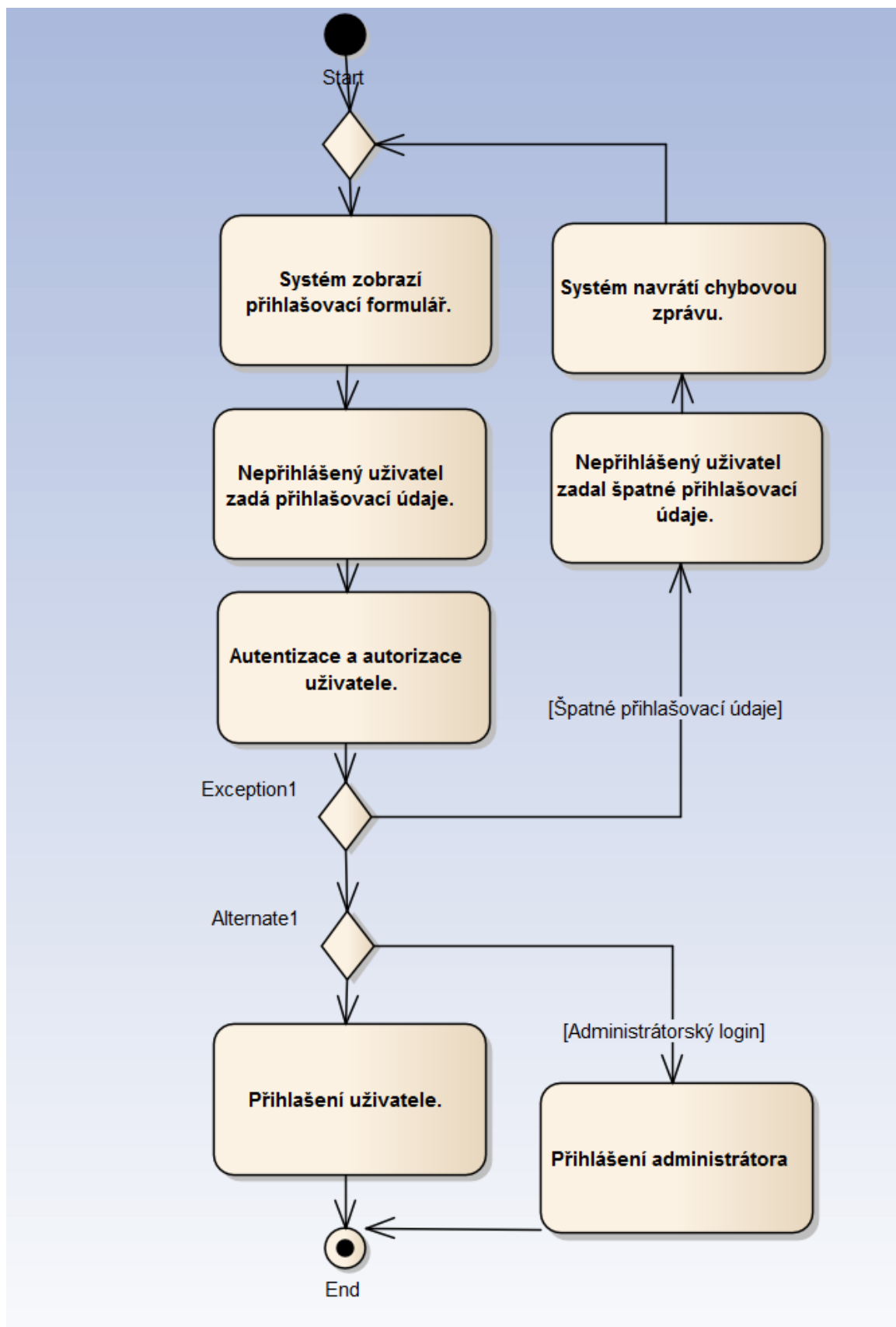
3.2 Přihlášení uživatele

3.2.1 Scénář případů užití

Umožňuje se přihlásit - změnit se v aktéra *Přihlášený uživatel* nebo *Administrátor*.

- Aktéři
 - Nepřihlášený uživatel
- Podmínky pro spuštění
 - Uživatel musí být nepřihlášený.
- Základní tok
 1. Nepřihlášený uživatel se chce přihlásit.
 2. Systém zobrazí přihlašovací formulář.
 3. Nepřihlášený uživatel zadá přihlašovací údaje.
 4. Systém zvaliduje data od uživatele pomocí «include» autentizace + autorizace uživatele.
 5. Systém přihlásí uživatele.
- Vyjímka
 - 4.1 Pokud nepřihlášený uživatel zadal špatné přihlašovací údaje, systém zobrazí chybovou zprávu a pokračuje na 2. bodu základního toku.
- Alternativní tok
 - 5.1 Pokud systém rozezná administrátorské přihlašovací údaje, přihlásí uživatele jako administrátora.
- Podmínky pro dokončení
 - Systém změní roli uživatele v systému na aktéra *Přihlášený uživatel* nebo *Administrátor*.

3.2.2 Diagram aktivit



Obrázek 8: Diagram aktivit - přihlášení uživatele

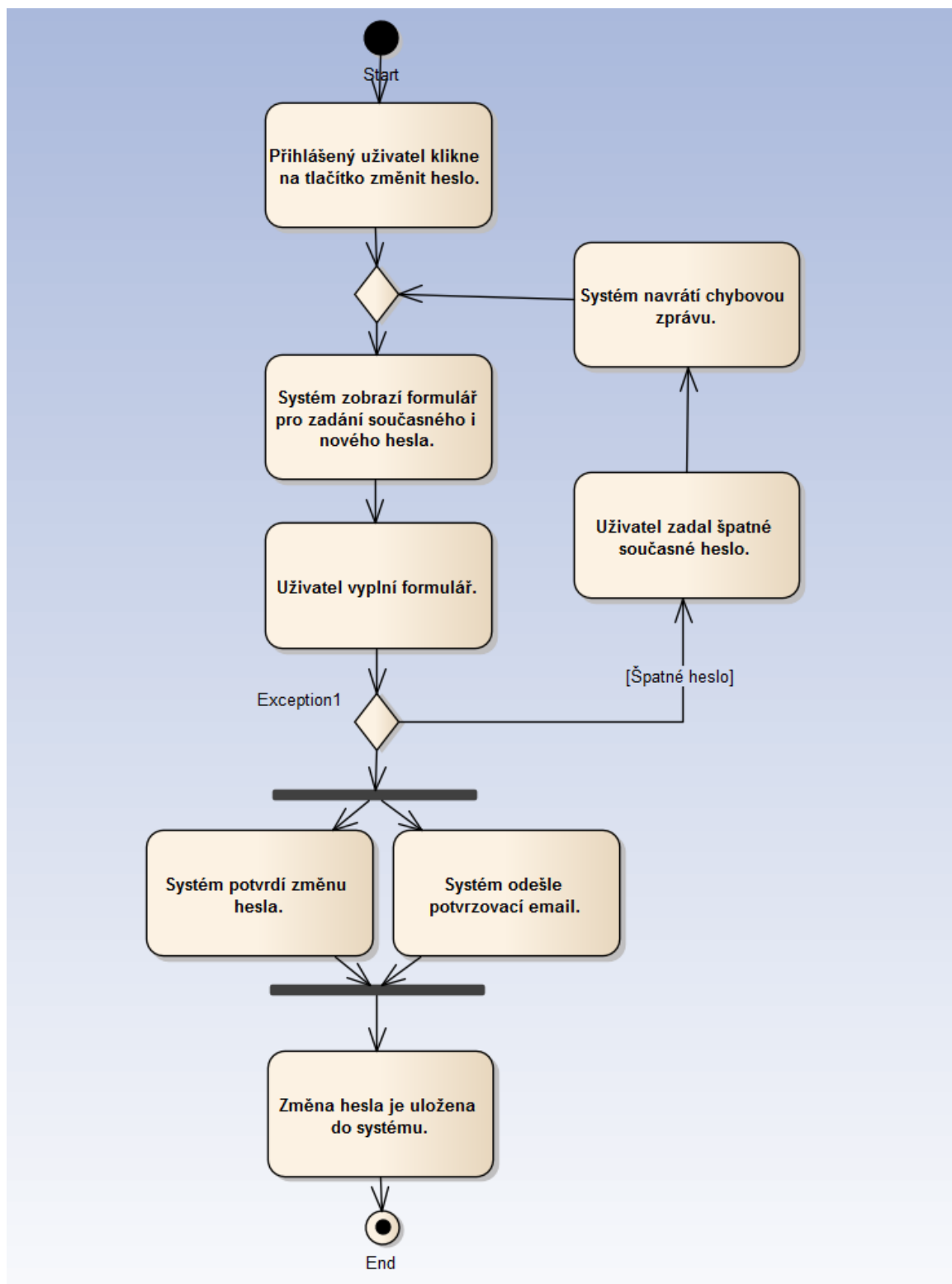
3.3 Změna hesla

3.3.1 Scénář případů užití

Umožňuje změnit přihlašovací heslo.

- Aktéři
 - Přihlášený uživatel
 - Administrátor
 - Mailserver
- Podmínky pro spuštění
 - Uživatel musí být přihlášený.
- Základní tok
 1. Přihlášený uživatel si chce změnit heslo.
 2. Systém zobrazí formulář pro zadání současného a nového hesla.
 3. Uživatel zadá současné a nové heslo.
 4. Systém zvaliduje data od uživatele.
 5. Systém změní heslo uživatele, potvrdí změnu a «include» odešle potvrzovací email z mailserveru.
- Vyjímka
 - 4.1 Pokud uživatel zadal špatné současné heslo, systém zobrazí chybovou zprávu a pokračuje na 2. bodu základního toku.
- Podmínky pro dokončení
 - Změněné heslo bude korektně uloženo v databázi.

3.3.2 Diagram aktivit



Obrázek 9: Diagram aktivit - změna hesla

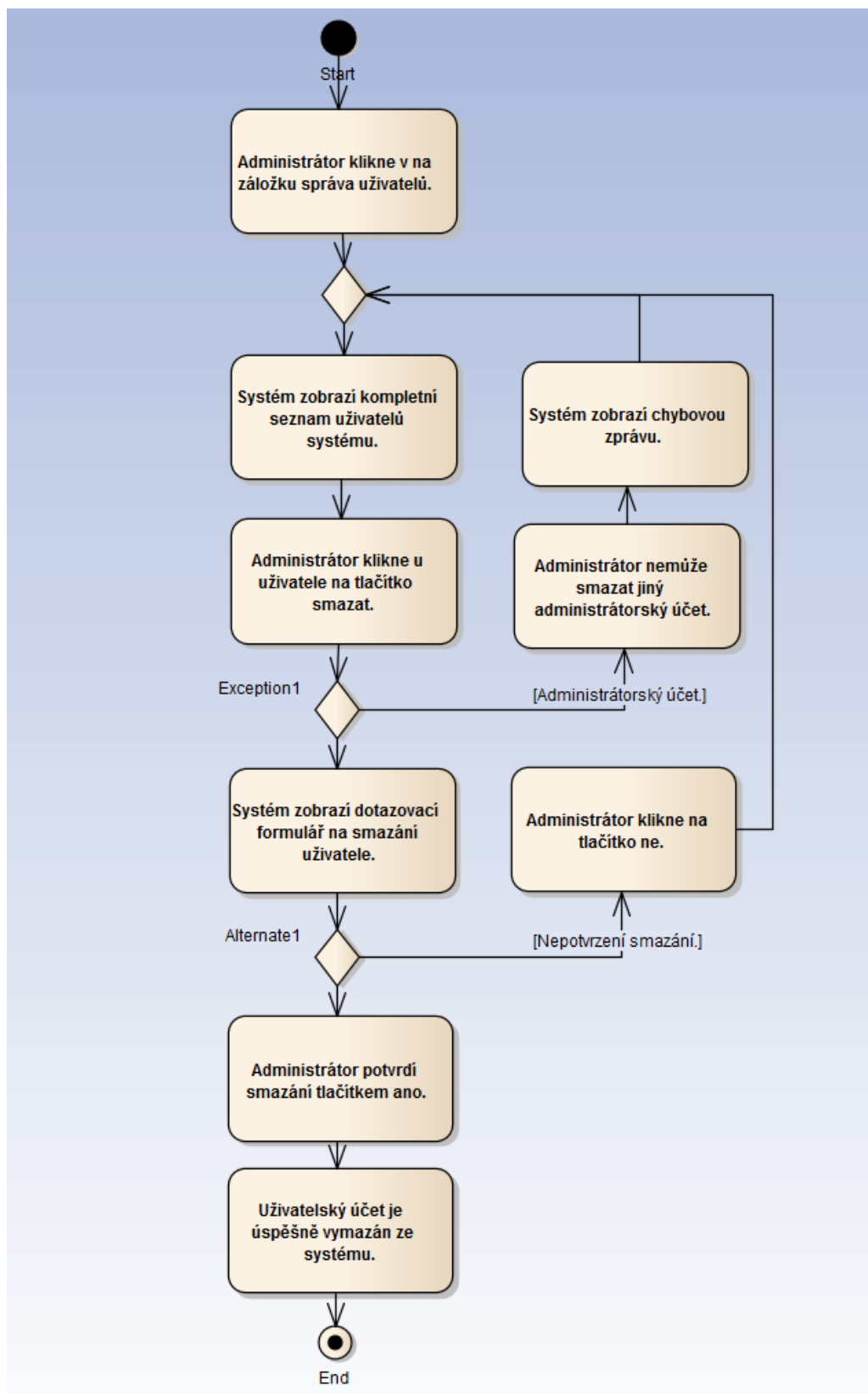
3.4 Vymazání uživatele

3.4.1 Scénář případů užití

Umožňuje smazat účet uživatele.

- Aktéři
 - Administrátor
- Podmínky pro spuštění
 - Administrátor se musí nacházet na výpisu registrovaných uživatelů. Účet určen ke smazání nesmí být administrátorský.
- Základní tok
 1. Administrátor požádá systém o smazání účtu konkrétního uživatele ve výpisu registrovaných uživatelů, který «include» vyhledá účet daného uživatele.
 2. Systém zobrazí dotaz k potvrzení smazání účtu.
 3. Administrátor kladně potvrdí dotaz.
 4. Systém vymaže uživatelský účet.
- Alternativní tok
 - 3.1 Pokud administrátor záporně potvrdí dotaz, tento případ užití je ukončen a administrátor se vrací na výpis registrovaných uživatelů.
- Podmínky pro dokončení
 - Smazání účtu bude korektně provedeno v databázi.

3.4.2 Diagram aktivit



Obrázek 10: Diagram aktivit - vymazání uživatele

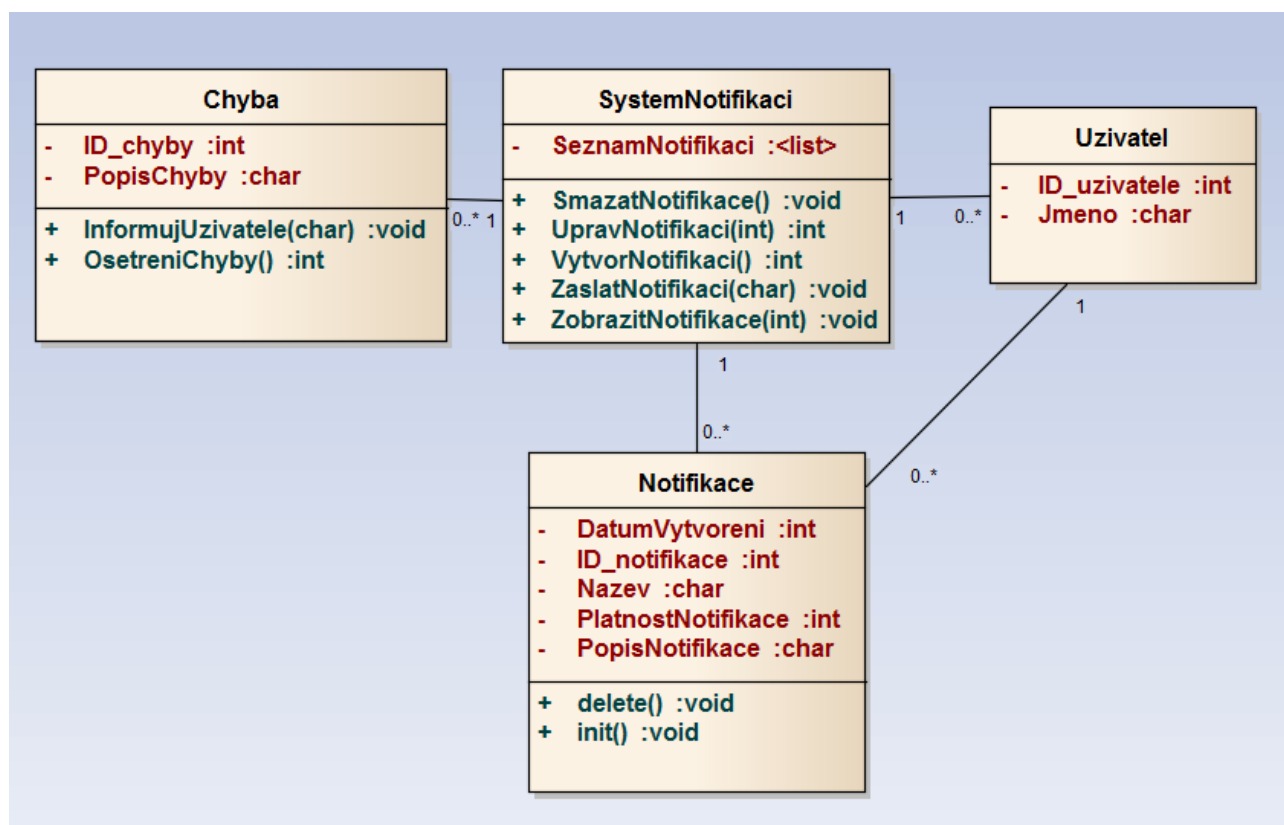
4 Správa notifikací

Poslední částí projektu byl analytický rozbor přiřazeného use case diagramu navrženého týmem pana Pirohta. Ze zadaného use case diagramu pro správu notifikací jsme vytvořili diagram tříd, který jsme ověřili pomocí sekvenčních diagramů a stavového diagramu.

4.1 Diagram tříd

Na základě přiřazeného use case diagramu modulu notifikací, navrženého skupinou pana Pirohty, byl vytvořen diagram analytických tříd. Rozbor přiřazeného use case diagramu byl proveden metodou vyčlenění podstatných slov a sloves. Z těchto vybraných pojmů byly určeny třídy, atributy a metody použité pro tento modul. Celkem byly navrženy čtyři třídy, které jsou v diagramu doplněny vazbami mezi těmito třídami a dále byly k jednotlivým třídám přiřazeny jejich atributy a metody.

Hlavní třídou je třída *SystemNotifikaci* starající se o běh celého modulu notifikací v systému, která byla navržena jako singleton. Na následujícím diagramu jsou zobrazeny třídy a vztahy mezi nimi. Metody, které jsou implementovány v těchto třídách by měly zajistit, aby uživatelé byli schopni vykonávat všechny akce, které byly navrženy v use case diagramu.

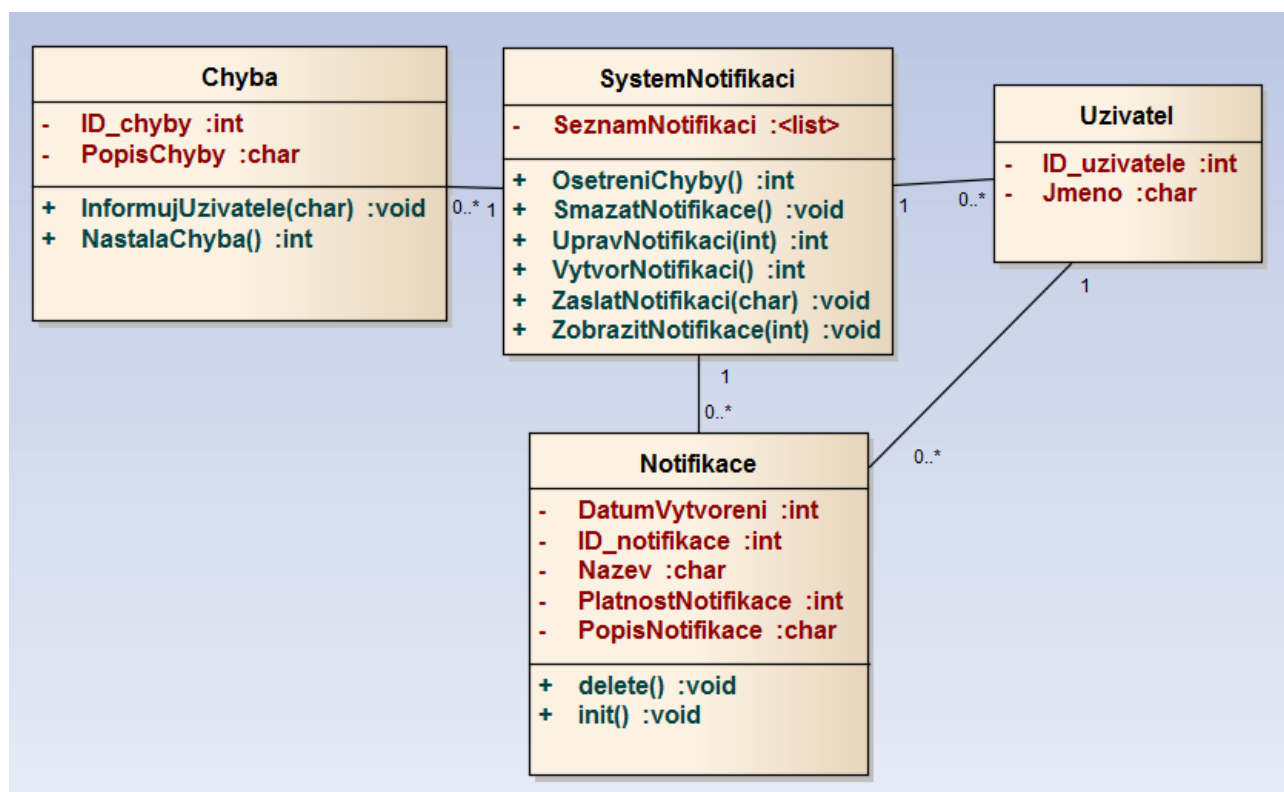


Obrázek 11: Diagram tříd

4.2 Diagram tříd - oprava

Při návrhu sekvenčního diagramu bylo zjištěno, že diagram tříd není navržen zcela správně. Sekvenční diagram je zobrazen na obrázku 13. Na základě tohoto diagramu byla metoda *OsetreniChyby()* přesunuta do třídy *SystemNotifikaci*. Dále do třídy *Chyba* byla přidána metoda

NastalaChyba(), aby bylo možné o této nově vzniklé skutečnosti informovat Systém notifikací, který následně nalezenou chybu se pokusí ošetřit.



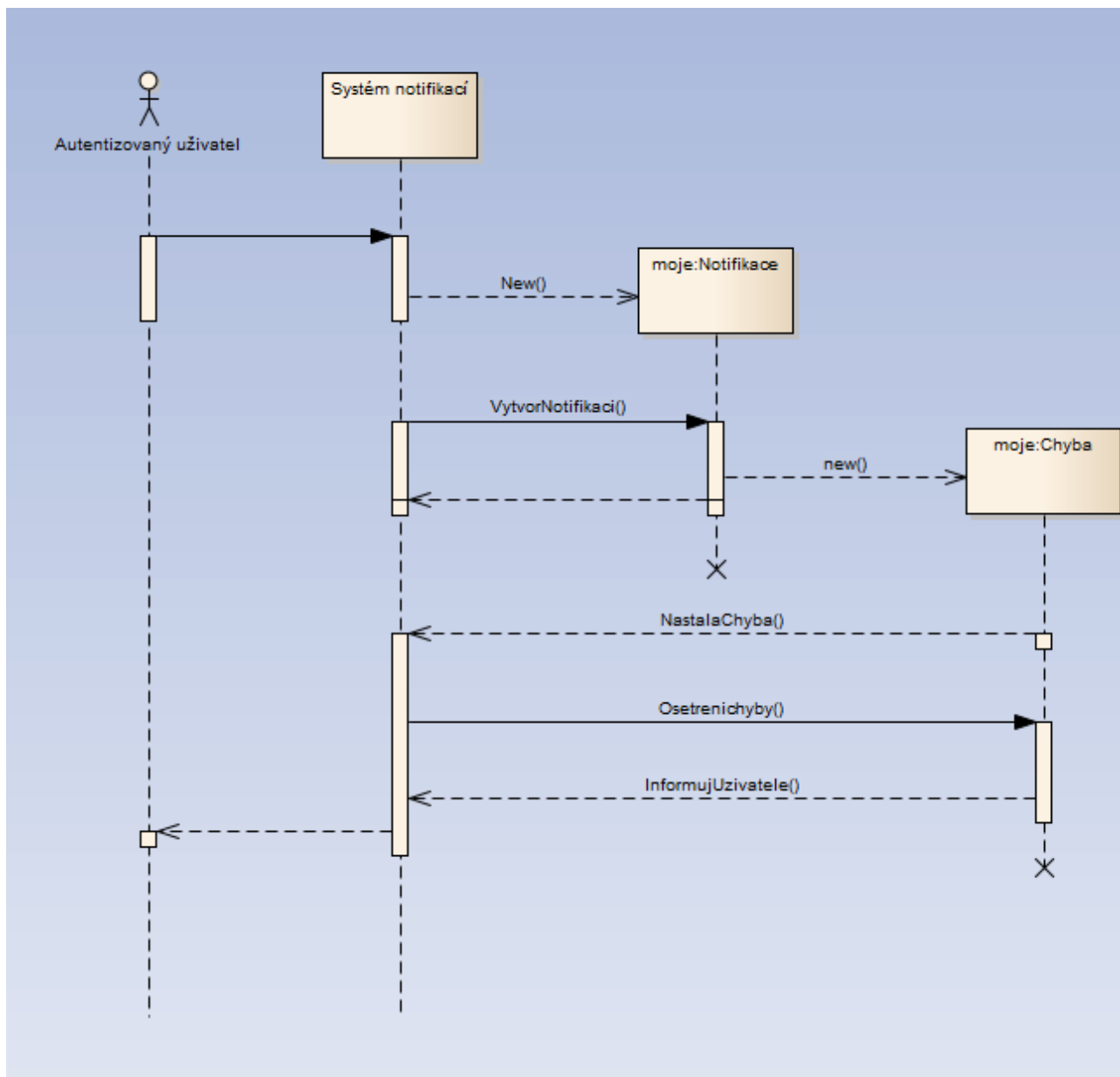
Obrázek 12: Diagram tříd - upravený návrh

4.3 Sekvenční diagramy

Návrh sekvenčního diagramu byl proveden za účelem ověření validity. Na základě tohoto diagramu byly zjištěny nedostatky v návrhu diagramu tříd. Tyto nedostatky jsou podrobněji popsány v kapitole 4.2. Na následujícím diagramu je zobrazeno vytváření nové notifikace. Je zde navržen *autentizovaný uživatel*, který komunikuje se systémem notifikací. Dále je zde navržena třída *Systém notifikací* a dva objekty *moje:Notifikace* a *moje:Chyba*. Všechny objekty zde navzájem komunikují zasíláním zpráv.

4.3.1 Vytvoření notifikace

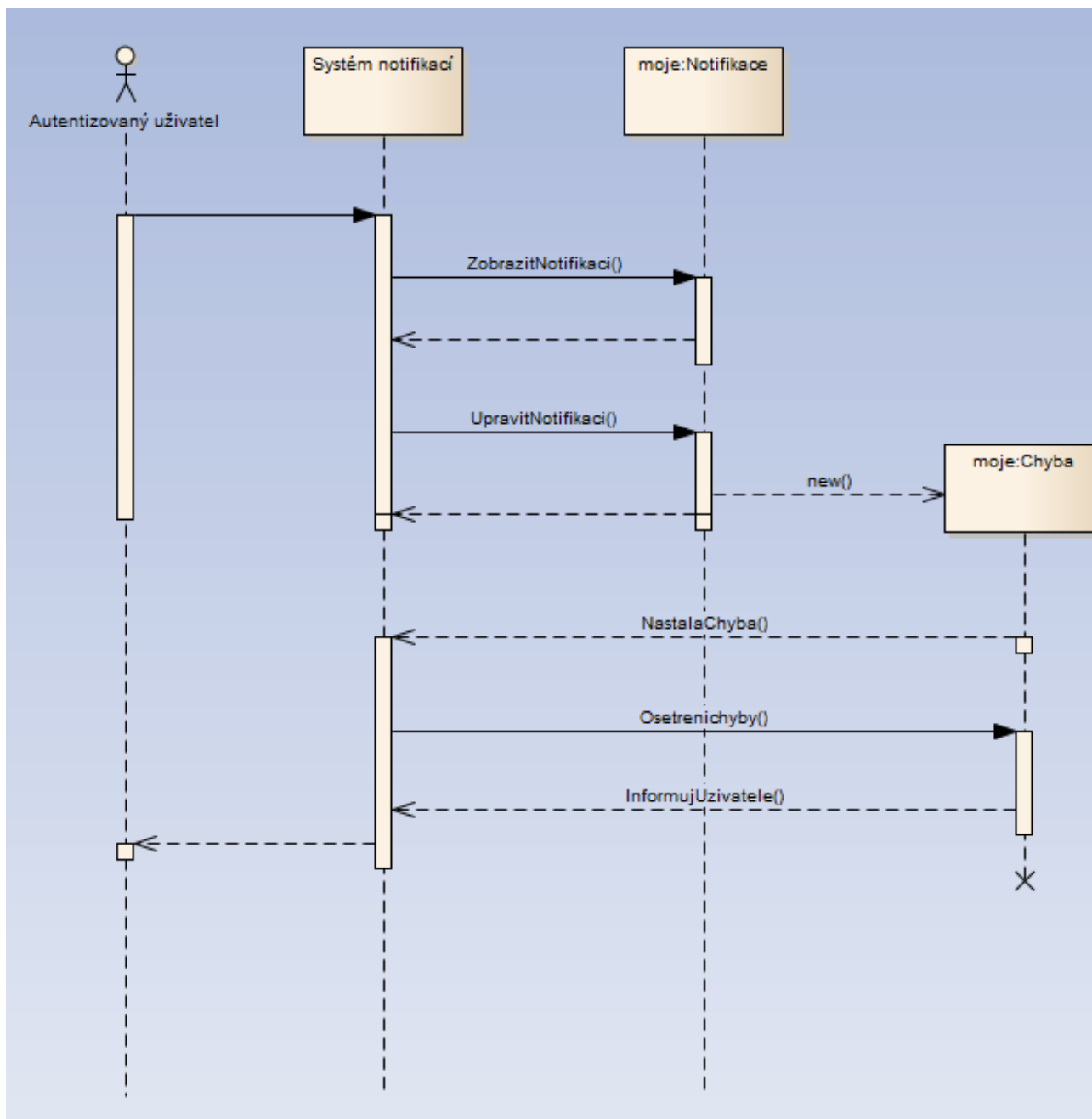
Vytvoření nové notifikace může provést pouze autentizovaný uživatel. Při vytváření notifikace je nejprve zavolána metoda *New()*, která alokuje nový objekt. Dále je zavolána metoda *VytvorNotifikaci()*, která tento objekt inicializuje. Pokud v této chvíli nastane nějaká chyba, je následně alokován objekt *Chyba*, který upozorní systém notifikací o nově vzniklé chybě. Systém notifikací se pokusí ošetřit vzniklou chybu zavoláním metody *OsetreniChyby()* a výsledek této akce je zahrnut ve zprávě *InformujUzivatele()*. Všechny neoznačené vracející se šipky symbolizují návratové hodnoty.



Obrázek 13: Sekvenční diagram - vytvoření notifikace

4.3.2 Modifikace notifikace

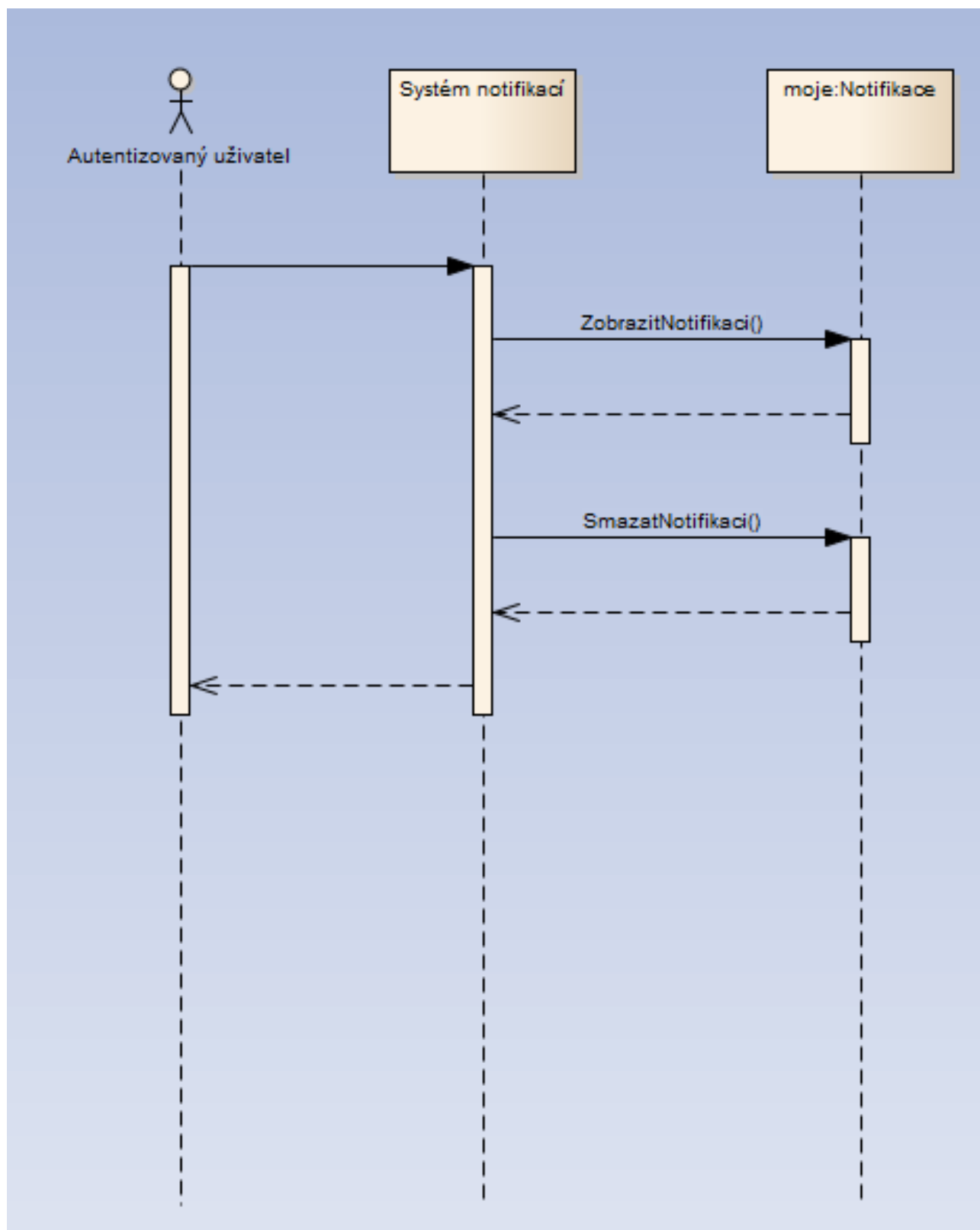
Modifikace notifikace může provést pouze autentizovaný uživatel. Při modifikaci notifikace je nejprve zavolána metoda *ZobrazitNotifikaci()*, která zobrazí zadanou již vytvořenou notifikaci. Dále je zavolána metoda *UpravitNotifikaci()*, která tento objekt otevře. Pokud v této chvíli nastane nějaká chyba, je následně alokován objekt *Chyba*, který upozorní systém notifikací o nově vzniklé chybě. Systém notifikací se pokusí ošetřit vzniklou chybu zavoláním metody *Osetrenichyby()* a výsledek této akce je zahrnut ve zprávě *InformujUzivatele()*. Všechny neozačené vracející se šipky symbolizují návratové hodnoty.



Obrázek 14: Sekvenční diagram - modifikace notifikace

4.3.3 Vymazání notifikace

Při mazání notifikace je důležité ji nejprve vyhledat zavoláním metody *ZobrazitNotifikaci()*, a následně je možné ji vymazat zavoláním metody *SmazatNotifikaci()*.



Obrázek 15: Sekvenční diagram - vymazání notifikace

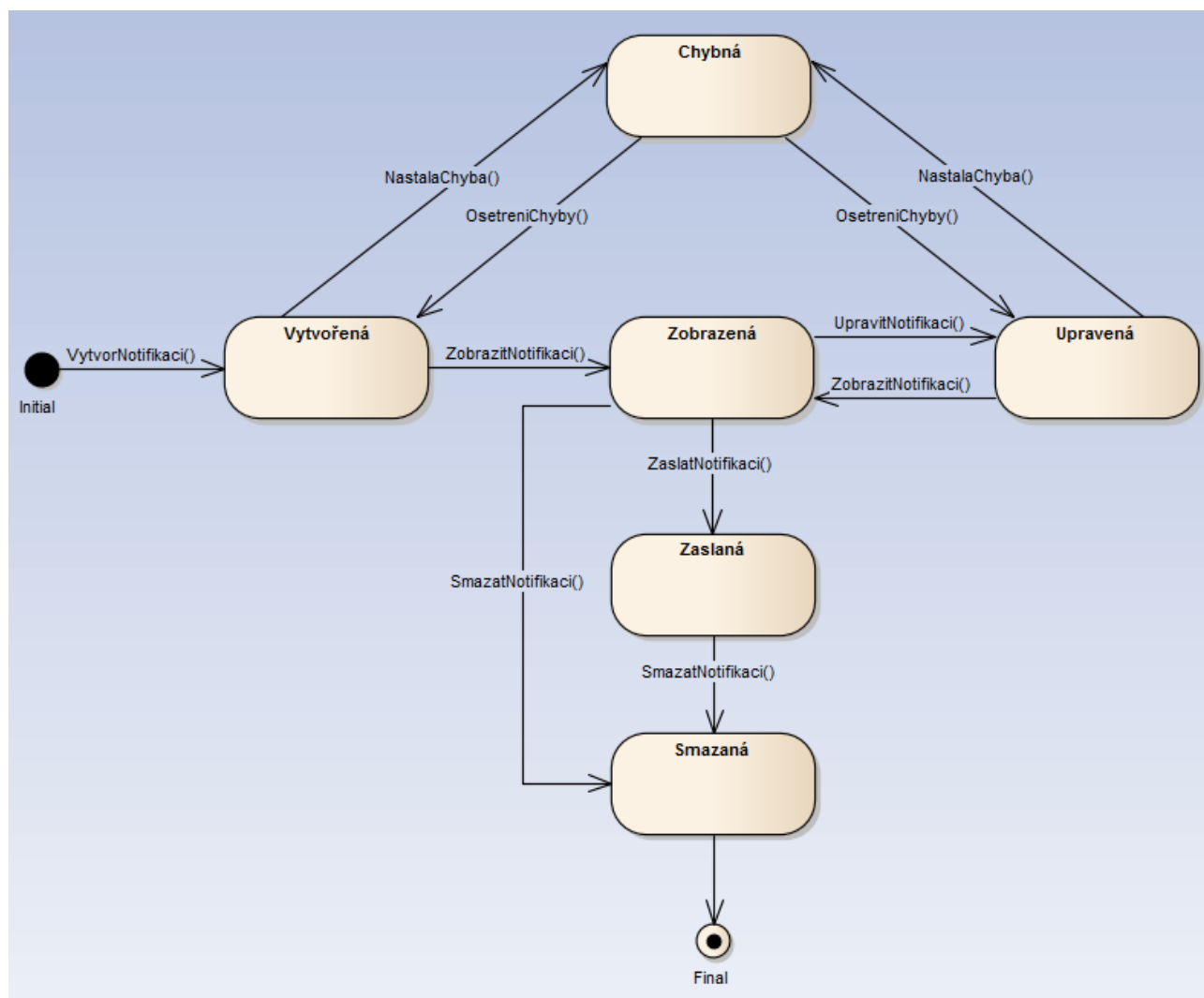
4.4 Stavový diagram

Posledním navrženým diagramem je stavový diagram, který rovněž slouží k ověření validity dynamického systému. Na následujícím diagramu jsou zobrazeny stavy a přechody mezi nimi. V této fázi již neschází doimplementovat žádnou metodu, a tudíž návrh systému je úspěšně dokončen.

Notifikace začíná ve stavu *Vytvořená*, dokud není *Zobrazena*. Při vytváření notifikace může vzniknout chyba, která změní stav notifikace na *Chybná*, dokud není chyba ošetřena.

Zobrazená notifikace může být *Upravená*, čímž se stane znovu nezobrazenou, dokud to nezmění uživatel systému. Díky tomu je zajištěno, že uživatel bude upozorněn na každou změnu již zobrazené notifikace. Pokud při úpravě notifikace dojde k chybě, změní se stav na *Chybná*, dokud není chyba ošetřena.

Zobrazená může být také *Zaslána* jinému uživateli. Notifikace se tak zkopíruje a pro nového uživatele bude nově *Vytvořená*. Ať již zobrazená nebo zasláná notifikace může být systémem *Smazána*, čímž opouští systém.



Obrázek 16: Stavový diagram

5 Závěr

Cílem tohoto semestrálního projektu bylo navržení systému pro správu tisku na školní 3D tiskárně. Nejprve byl navržen celkový systém pomocí Eriksson-Penkersových diagramů, který se skládá z pěti samostatných modulů. Jednotlivé moduly byly rozděleny mezi studijní skupiny a naším úkolem bylo podrobné zpracování modulu správy uživatelů a následně modulu notifikací. Analýza těchto jednotlivých částí je popsána pomocí diagramů z jazyka UML. Na základě analýzy a návrhu systému a jeho částí lze konstatovat, že tento systém je realizovatelný a je tedy připraven pro další fáze jeho vývoje.