



## ZADÁNÍ BAKALÁ SKÉ PRÁCE

<b>Název:</b>	MobChar - balí ek pro Pána jeskyn v Dra ím doup ti
<b>Student:</b>	Mat j Shán l
<b>Vedoucí:</b>	Ing. Zden k Rybola
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Softwarové inženýrství
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	Do konce letního semestru 2017/18

### Pokyny pro vypracování

Cílem bakalá ské práce je vytvo it balí ek pro Pána jeskyn ve h e Dra í doup v rámci existující aplikace pro správu informací o postavách v r zných hrách na hrdiny MobChar, p ípadn rozší it stávající balí ek pro Dra í doup o podporu agendy Pána jeskyn .

Hlavní funkce balí ku:

- komunikace se za ízeními ostatních hrá - prohlížení jejich postav, p edávání p edm t a soukromých zpráv,
- p íprava dobrodružství - p íprava p edm t , cizích postav a nestv r, správa map,
- import a export dobrodružství z desktopové aplikace.

V souladu se standardy softwarového inženýrství:

- Analyzujte požadavky na modul, a také stávající aplikaci a její moduly v kontextu t chto nových požadavk .
- Vhodným zp sobem navrhn te architekturu nového balí ku a zp sob realizace požadavk .
- Implementujte v souladu s provedeným návrhem.
- Implementaci pat i n zdokumentujte a otestujte.
- P ípravte uživatelskou p íru ku pro používání balí ku.

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

prof. Ing. Pavel Tvrdí k, CSc.  
d kan

V Praze dne 28. listopadu 2016



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

## **MobChar – balíček pro Pána jeskyně v Dračím doupěti**

*Matěj Sháněl*

Vedoucí práce: Ing. Zdeněk Rybola

14. května 2017



---

# Poděkování

Rád bych poděkoval vedoucímu mé bakalářské práce, Ing. Zdeňku Rybolovi, za usměrňování mých roztěkaných myšlenkových pochodů, profesionální názor, pohled z jiné perspektivy, čas strávený čtením mých nekonečných emailů a ochotu se mi věnovat i v pozdních nočních hodinách.

Dále bych chtěl poděkovat Bc. Šárce Sochorové za vytvoření neuvěřitelně krásné grafiky pro mojí aplikaci i přes naprosto neurčitou specifikaci. Také bych rád poděkoval Martinu Navrátilovi, mému prvnímu Pánu jeskyně, a všem členům mé první družiny, kteří ve mně probudili vášeň pro hru Dračí doupě.

Zvláštní poděkování si zaslouží i Bc. Tomáš Nováček, který mi vysvětlil rozdíl mezi pomlčkou a spojovníkem, seznámil mě s poměrem důsledkovým a rozmluvil mi použití uměleckých prvků v mé práci.

Velký dík patří také všem testovacím uživatelům, kteří nám pomohli vyladit výslednou aplikaci a poskytli nám náměty k jejímu dalšímu rozšíření.

Nerad bych zapomněl na své kolegy, Šárku Weberovou a Jana Horáčka. Díky za všechny rady, triky, vzájemné utěšování a především Vaši schopnost se mnou ten napjatý semestr přežít.

Na závěr bych rád poděkoval své rodině a všem svým přátelům, kteří se mě nezřekli, ačkoliv jsem se během posledního semestru změnil v mýtickou bytost, kterou si většina z nich pamatuje pouze z vybledlých fotografií.

Bez Vás všech by toto dílo nikdy nevzniklo.

*Hash 'tar is 'h 'mar*



---

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 14. května 2017

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2017 Matěj Sháněl. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Sháněl, Matěj. *MobChar – balíček pro Pána jeskyně v Dračím doupěti*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.



---

# Abstrakt

Cílem této práce je modernizace hry Dračí doupe vytvořením balíčku aplikace MobChar, jenž se zabývá správou poznámek Pána jeskyně (coby specifického typu hráče).

V první části je vypracována analýza současného stavu aplikace MobChar a požadavků na stav cílový. Následně je vytvořen návrh aplikace, který je přizpůsoben potřebě exportovat data do souborů typu .xml a přenášet tyto soubory mezi zařízeními.

Na základě návrhu je provedena implementace softwaru, která je nakonec otestována nejprve samostatně, následně pak při interakci s jinými komponentami aplikace MobChar.

Vytvořená aplikace umožňuje přenášet data z desktopové aplikace do již existujících balíčků aplikace MobChar a zpět. Nová funkcionální umožňuje Pánům jeskyně jednodušeji kontrolovat průběh hry a dodržování pravidel.

**Klíčová slova** mobilní aplikace, Android, rozšíření, herní scéna, MobChar, Dračí doupe, poznámky Pána jeskyně, Java, SQLite, XML

---

# Abstract

The objective of this thesis is to modernize Czech equivalent of the game Dungeons and Dragons by creating new module for MobChar application, which manages notes of the Game master (specific player type).

In the first part, the analysis of requirements and current state of MobChar application is made. After that, the design adapted for need to export data to .xml files and then send them between two devices is created.

Based on this design, software application is made. Finally, this application is being tested isolated for basic functionality and then in the context of interaction with other components of MobChar application.

The final created application allows transferring data created in desktop application to already created modules of MobChar application and back. New functionality simplifies control of the game progress and rule keeping for Game masters.

**Keywords** mobile application, Android, extension, gaming field, MobChar, Dungeons and Dragons, Game master, Java, SQLite, XML

---

# Obsah

<b>Úvod</b>	<b>1</b>
Cíl práce . . . . .	2
<b>1 Analýza</b>	<b>3</b>
1.1 Hra Dračí doupě . . . . .	3
1.2 Aplikace MobChar . . . . .	6
1.3 Role Pána jeskyně . . . . .	8
1.4 Rozbor domény . . . . .	12
1.5 Vymezení požadavků na nový balíček . . . . .	17
1.6 Interakce uživatele s aplikací – případy užití . . . . .	20
<b>2 Návrh</b>	<b>35</b>
2.1 Architektura balíčku . . . . .	35
2.2 Způsob uložení dat . . . . .	36
2.3 Návrhový model tříd . . . . .	37
2.4 Realizace případů užití . . . . .	39
2.5 Nasazení a závislosti . . . . .	44
<b>3 Implementace</b>	<b>47</b>
3.1 Použité nástroje . . . . .	47
3.2 Použité knihovny . . . . .	48
3.3 Rozdělení na adresáře . . . . .	49
3.4 Ukázky implementace . . . . .	49
<b>4 Testování</b>	<b>55</b>
4.1 Jednotkové testy . . . . .	55
4.2 Uživatelské testy . . . . .	56
<b>Závěr</b>	<b>61</b>

<b>Literatura</b>	<b>63</b>
<b>A Seznam použitých zkratk</b>	<b>65</b>
<b>B Obsah přiloženého CD</b>	<b>67</b>

---

## Seznam obrázků

1.1	Diagram průběhu hry . . . . .	5
1.2	Diagram přípravy dobrodružství . . . . .	9
1.3	Diagram předání předmětu . . . . .	11
1.4	Doménový model předmětů a map . . . . .	14
1.5	Doménový model bytostí . . . . .	17
1.6	Doménový model objektů nezávislých na lokaci . . . . .	18
1.7	Diagram případů užití práce s adresářovou strukturou . . . . .	21
1.8	Wireframe obrazovky adresáře „Hospoda“ . . . . .	22
1.9	Diagram případů užití práce s herními objekty . . . . .	25
1.10	Wireframe obrazovky editace kouzla „Ohnivá koule“ . . . . .	27
1.11	Diagram případů užití práce s externím úložištěm . . . . .	28
1.12	Diagram provázanosti případů užití ohledně komunikace s jiným zařízením . . . . .	33
2.1	Způsob navržení rozhraní . . . . .	36
2.2	Rozdělení datových objektů do balíčků . . . . .	38
2.3	Sekvenční diagram získávání dostupných balíčků . . . . .	40
2.4	Sekvenční diagram získávání dobrodružství a jejich spouštění . . . . .	41
2.5	Sekvenční diagram vytváření XML objektu . . . . .	43
2.6	Sekvenční diagram odesílání XML objektu . . . . .	44
2.7	Diagram nasazení . . . . .	45



---

# Seznam tabulek

1.1	Tabulka pokrytí požadavků případy užití . . . . .	34
-----	---	----





---

# Úvod

V současné době si jen málokdo dokáže představit, jak by vypadal náš život bez mobilních telefonů a způsobů, jimiž nám usnadňují každodenní bytí. Je proto s podivem, že hráči velmi známé české stolní hry na hrdiny, Dračího doupěte, používají stále k hraní bloky a tužky.

Na trhu sice existují desítky mobilních aplikací, které se zabývají podobnou problematikou, bohužel jen minimum z nich je schopné přizpůsobit se Dračímu doupěti a téměř žádné nejsou lokalizované do českého jazyka. Tato skutečnost celkem pochopitelně odrazuje většinu hráčů od jejich používání.

V posledních letech byla na Českém vysokém učení technickém v Praze vyvinuta aplikace MobChar, jež se zabývá správou informací o herních postavách různých her na hrdiny. Každá hra je reprezentována balíčkem a tyto balíčky jsou propojeny jádrem aplikace.

Vhodným řešením tohoto problému by tudíž bylo vytvoření balíčku pro hru Dračí doupě pro aplikaci MobChar. Tento balíček byl vytvořen a během minulého roku rozšířen týmem Alphaprod v rámci předmětů Softwarový týmový projekt 1 a 2.

Současná aplikace ovšem podporuje spravování balíčků pouze pro jeden typ hráčů, což hráčská komunita, která byla požádána o testování nové verze aplikace, zmínila coby návrh pro směr budoucího vývoje aplikace. Po konzultaci s vedoucím projektu a členy týmu Alphaprod bylo rozhodnuto aplikaci rozšířit o nový typ balíčků, který je určen hráčům, jenž se zabývají tvorbou dobrodružství a řízením hry.

Tato práce se v první řadě zaměří na analýzu současného stavu jádra aplikace MobChar, stavu balíčku pro hru Dračí doupě a požadavků hráčské komunity na nový typ balíčků. Na základě této analýzy bude následně navrženo možné řešení, v němž budou vzaty v potaz existující aplikace řešící podobnou problematiku, a následně bude aplikace implementována.

Vytvořená aplikace bude začleněná do systému aplikací pro správu her na hrdiny MobChar coby rozšiřující balíček. V rámci tohoto systému bude schopná komunikovat s aplikací pro členy družiny, jejímž rozšířením se ve své

bakalářské práci zabývá Šárka Weberová, a desktopovou aplikací pro tvorbu dobrodružství, jež je náplní bakalářské práce Jana Horáčka.

Výsledný celek výrazně usnadní přípravu herních dobrodružství, která obecně zabírá zhruba stejný díl času jako samotné hraní.

## Cíl práce

Cílem bakalářské práce je vytvořit balíček pro Pána jeskyně ve hře Dračí doupe v rámci existující aplikace pro správu informací o postavách v různých hrách na hrdiny MobChar, případně rozšířit stávající balíček pro Dračí doupe o podporu agendy Pána jeskyně.

Vytváření musí proběhnout dle standardů softwarového inženýrství. Samotné implementaci bude tudíž předcházet analýza současného stavu balíčku a požadavků na stav nový a návrh řešení. Výsledná implementace bude otestována a zdokumentována.

Mezi hlavní funkce balíčku bude patřit komunikace se zařízeními ostatních hráčů, kterou se rozumí prohlížení jejich postav, předávání předmětů a posílání soukromých zpráv, příprava předmětů, nestvůr a cizích postav a schopnost importovat a exportovat dobrodružství vytvořené v desktopové aplikaci.

---

# Analýza

Tato kapitola se zabývá analýzou současného stavu aplikace MobChar, jejího hráčského balíčku pro hru Dračí doupě a požadavků na nově vytvářený balíček.

Postupně jsou analyzovány typické procesy spojené s hraním hry Dračí doupě, zhodnocen současný stav aplikace MobChar a vytvořen doménový model. Následně jsou na základě výsledků zformulovány požadavky na vytvářený balíček a případy užití, které tyto požadavky pokrývají a vhodným způsobem demonstrují.

## 1.1 Hra Dračí doupě

Dračí doupě je česká stolní papírová hra na hrdiny, kterou vytvořilo nakladatelství Altar v roce 1990 coby obdobu populární americké hry Dungeons & Dragons. Ačkoliv existují oficiální pravidla, hra Dračí doupě je specifická především různými obměnami, kterými si hráči přizpůsobují herní styl své potřebě.

### 1.1.1 Typy hráčů

Hráči Dračího doupě se dělí do dvou kategorií. První skupinu tvoří vybraný hráč, který si předem připraví dobrodružství ve smyšleném světě. Jeho úkolem je připravit celý herní svět, v němž se bude hra odehrávat, od rozlehlých království po nejmenší detaily postav a příšer, které zbytek hráčů bude potkávat. Následně pak interpretuje reakce tohoto světa na akce zbývajících hráčů. Vzhledem k tomu, že se většina prvotních dobrodružství odehrávala především v rozlehlých jeskyních, vžilo se pro tyto hráče označení „Pán jeskyně“, zkráceně PJ.

Do druhé skupiny spadají všichni ostatní hráči. Každý si vytvoří svou postavu, kterou ve světě vytvořeném Pánem jeskyně ovládá a jejíž osud tím určuje. Tento typ hráčů nemá žádné speciální pojmenování, avšak vzhledem k faktu, že ve většině případů jejich postavy putují ve družině, bývají ozna-

čování jednoduše jako „hráči“, jejich postavy pak nesou označení „členové družiny“, potažmo souhrnně jako „družina“.

### 1.1.2 Průběh hry

Samotná hra probíhá formou řízeného dialogu zpravidla u velkého stolu. Pán jeskyně vždy nejprve vyhledá ve svých poznámkách oblast, v níž se postavy hráčů právě nacházejí, zváží podmínky, jež v této oblasti panují, a seznámí hráče se situací, v níž se jejich postavy právě nacházejí.

To může znamenat pouhé popsání nového prvku na herní scéně (např. ze křoví po obou stranách cesty vyskákali lapkové), avšak nastiňování nové herní situace s sebou velmi často nese vyhledávání herních mechanismů v pravidlech[1], složitější výpočty spojené s hody kostkami či kontrolu vlastností postav.

Pokud by například nepozornou postavu zkoušel okrást nepřátelský zloděj, musel by si Pán jeskyně nejprve dohledat v pravidlech[1], jakou šanci má nepřátelský zloděj na úspěšnou krádež. Následně by si hodil kostkami pro určení, zdali se zloději krádež vydařila.

V případě neúspěchu by si Pán jeskyně vypůjčil osobní deník[2] postavy, která měla být okradena, aby mohl přesně určit důsledky zlodějova neúspěchu. Při prohlížení osobního deníku[2] by mohl zjistit, že vak okrádané postavy obsahuje lahvičku s kyselinou.

Tato skutečnost by vyvolala další hod kostkami určující, zdali byl zloděj nešikovný a podařilo se mu se při prohrabávání vaku kyselinou polít.

Po hod, který by určil, že měl zloděj smůlu by musel Pán jeskyně opět v pravidlech[1] vyhledat, jaké účinky bude mít zmíněná kyselina na zloděje, a po zjištění, že kyselina ubere zloději menší počet životů, by musel dalším hodem kostkami určit přesný počet ztracených životů.

Všechny dopady této akce si musí Pán jeskyně navíc zaevidovat do svých poznámek. Navíc celý tento proces probíhá bez jediného zásahu hráčů, a je proto žádoucí, aby trval co nejkratší dobu.

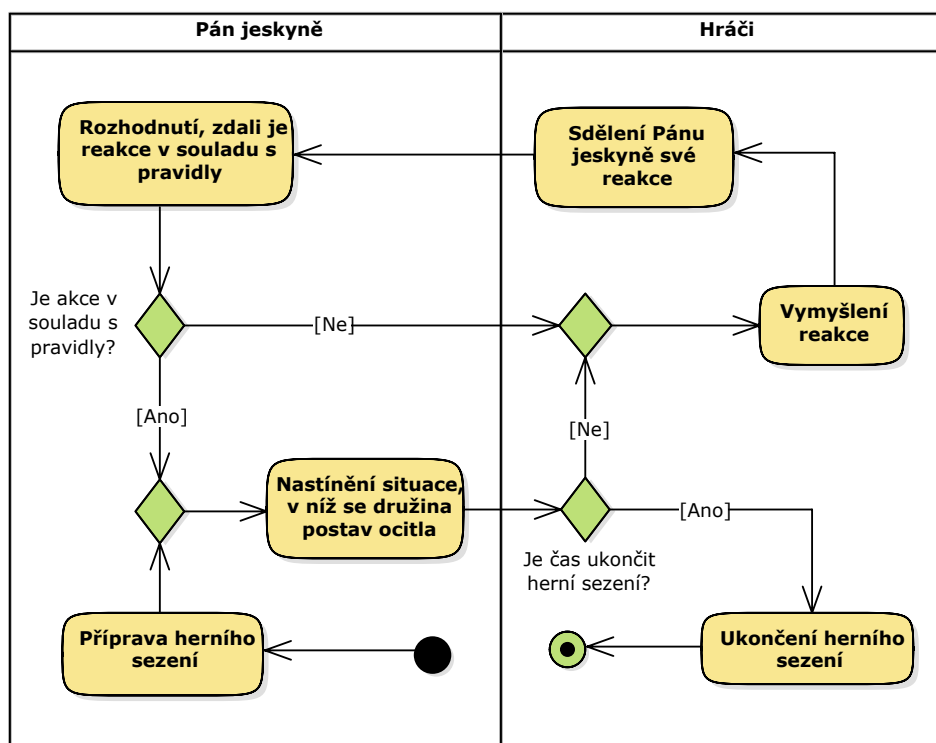
Po nastínění situace nechává Pán jeskyně hráčům prostor, aby mu sdělili, jak si přejí, aby jejich herní postavy na tyto skutečnosti zareagovaly. Pokud jsou dané reakce v souladu s pravidly hry, uváží Pán jeskyně, jaký dopad tyto akce budou mít, a seznámí hráče s následky jejich počínání. Tato akce může vyvolat další pracné prohledávání pravidel[1] a časově náročné hody kostkami.

Reakcí herního světa na hráčské akce vzniká nová situace a celý proces se opakuje, dokud jsou postavy hráčů naživu nebo mají hráči chuť hrát.

Tento proces je názorně zachycen v diagramu 1.1.

### 1.1.3 Boj

Speciálním případem v průběhu hry je situace, při níž se hráči dostanou do boje s nepřítelem. V tomto případě připraví Pán jeskyně papírovou mapku



Obrázek 1.1: Nastínění typického průběhu hry

tvorěnou šestiúhelníkovými políčky (tzv. „hexák“) a rozmístí na ni figurky znázorňující nepřátele a jednotlivé členy družiny.

Boj je rozdělen do tzv. kol. Na začátku každého lichého kola si všichni hráči hodí šestistěnnou kostkou na určení iniciativy v boji. K výsledku hodu kostkou se připočítají případné bonusy či postihy, čímž se určí výsledná iniciativa. Iniciativa určuje pořadí v jakém postavy provádějí své akce.

Postava s nejvyšší iniciativou provede první akci, která se vyhodnotí, následně se pokračuje postavou s druhou nejvyšší iniciativou a tak dále do doby, než všichni účastníci boje neprovedou svou akci. Po provedení poslední akce se pokračuje dalším kolem do doby, než boj skončí.

Boj samotný končí v momentě, kdy jedna z bojujících stran již nemá žádné bojovníky při vědomí, nebo pokud tak neurčí Pán jeskyně z jiných důvodů.

#### 1.1.4 Zlaté pravidlo

Jedním z mála pravidel, které je třeba vždy nezbytně dodržovat pro hladký průběh hry, je tzv. „zlaté pravidlo“, jenž zní: „Hráč není postava a postava

není hráč“. Ač toto pravidlo může znít naprosto samozřejmě a banálně, jedná se o věc, se kterou se především nováčci velmi těžko potýkají.

Hlavní problém při hraní v jedné místnosti vzniká, když je například družina rozdělena do dvou částí, z nichž jedna se dozví klíčovou informaci, která může rozhodovat o přežití druhé. V tomto momentě bývá pro hráče často velmi obtížné pokračovat v hraní svých postav, které danou informaci nemohli nijak zjistit, tak, jako by o ní skutečně nevěděli.

Z tohoto důvodu se někteří Páni jeskyně uchylují k soukromým zprávám či rozhovorům, které bohužel zpomalují průběh hry, a tím kazí herní atmosféru.

Podobně pro některé hráče bývá těžké se oprostít od konfliktů, které celkem přirozeně vznikají mezi jejich herními postavami (například při tahanici o vydobytý poklad). Následně pak přenášejí tyto konflikty do reálného života zapominaje, že se jednalo pouze o situaci ve hře. Tyto situace zpravidla končívájí rozpadem herní skupiny a zánikem dobrodružství.

### 1.1.5 Uchovávání informací

Každý hráč by měl znát základní charakteristiky své postavy (sílu, obratnost, ...), vybavení, které má postava na sobě, efekty, které na ni působí, kouzla a schopnosti, kterým se naučila, a mnohé další detaily. Nakladatelství Altar pro účel uchovávání těchto informací vydalo tzv. „osobní deník“ [2], který tuto práci výrazně ulehčuje.

Podobně jako hráči i Páni jeskyně musí řešit otázku uchovávání informací o hře. Na rozdíl od hráčů však musí mít přehled o celém zbytku herního světa. To zahrnuje naprosto vše od počtu životů příšer v nejbližší jeskyni přes mapky královského hradu až po obsah truhly ve sklepe hokynářství. Toto množství informací je nesrovnatelně větší, takže Páni jeskyně zpravidla během jednoho dobrodružství popíší až několik bloků.

Mimo to musí mít Páni jeskyně ve svých poznámkách kvalitní systém, neboť reakce hráčů jsou jen málokdy předvídatelné a změny herního kontextu (oblasti, reakce bytostí, výsledky bojů) se dějí každých několik minut.

## 1.2 Aplikace MobChar

Aplikace MobChar se zabývá správou herních údajů pro různé hry na hrdiny. Z počátku vznikla v rámci předmětů Softwarový týmový projekt 1 a 2 coby náhrada za „osobní deník“ [2] pro hru Dračí doupe a v průběhu dalšího běhu těchto předmětů byla rozšířena o další funkcionalitu představující především inventář postav (založený na principu adresářů) a systém dopočítávání ovlivnění základních charakteristik postavy způsobeného aktivními efekty.

Zlomovým okamžikem ve vývoji aplikace MobChar bylo vytvoření bakalářské práce Bc. Jakuba Nižaradzeho, který na základě poznatku, že většina

her na hrdiny má jistou množinu společných rysů, tuto množinu zobecnil a vyčlenil ji z původního balíčku do společné části. Zároveň vytvořil část aplikace, která umožňuje integrovat balíčky popisující další podobné hry na hrdiny, a ze zbytku původní aplikace udělal právě takový balíček. Tato společná část se nazývá jádrem aplikace a společně s knihovnou, jež vznikla z vyčleněných funkcí, tvoří páteř současné aplikace MobChar.

V následujících letech byl balíček pro Dračí doupě velmi výrazně rozšířen a vznikly balíčky pro hry Dračí doupě plus a Dungeons & Dragons. I přes svůj velký potenciál nebyla aplikace MobChar zpřístupněna veřejnosti a k funkční verzi mají přístup pouze vývojáři a testéři.

### 1.2.1 Současný stav

V této části bude rozebrán současný stav jednotlivých komponent existující aplikace MobChar a jejich potenciál pro komunikaci s nově vytvářeným balíčkem.

#### 1.2.1.1 Jádro aplikace MobChar a knihovní funkce

V současné době je jádro aplikace schopné hostovat pouze typ balíčků pro hráče (správa postav), jejichž rozpoznávání je založeno na speciálním atributu, který se musí přidělit spouštěným aktivitám v manifestu balíčku. Po spuštění aplikace je zobrazen seznam již vytvořených postav, případně text oznamující uživateli, že prozatím nemá vytvořené žádné postavy.

Jádro dále umožňuje uživateli vytvořit novou postavu, exportovat vytvořenou postavu do souboru či provést import dříve exportované postavy. Mimo to může uživatel jednoduše vyhledávat pomocí části jména v seznamu svých postav.

Knihovní funkce poskytují širokou paletu základních typů stavebních kamenů uživatelského rozhraní OS Android (speciální typy fragmentů a aktivit), jež půjdou bezesporu použít při tvorbě zástupce nového typu balíčku. Zbylé položky jsou však spíše funkcionalitou přizpůsobeny existujícímu typu balíčku a o jejich využitelnosti v rámci této práce mám silné pochybnosti.

Začlenění nového typu balíčku tudíž bude vyžadovat markantní předělání stavby úvodní obrazovky, neboť bude potřeba rozlišit dobrodružství od samotných postav. Zároveň musí být rozšířen seznam speciálních atributů, aby bylo možné spustit jiný typ balíčků.

#### 1.2.1.2 Balíček pro hru Dračí doupě

Aktuální stav hráčského balíčku pro Dračí doupě nabízí hráčům možnost vidět základní informace o jejich zvolené postavě, její inventář, naučená kouzla a schopnosti, působící efekty, vytvořené sety a záznam provedených aktivit a změn. Jednotlivé přehledy jsou uspořádány do záložek a seřazeny vedle sebe.

Mimo to je implementováno rozhraní pro načítání šablon kouzel, schopností, efektů a předmětů ze souborů typu XML.

Vzhledem k tomu, že nově vytvářený balíček pracuje s naprosto totožnou doménou, nabízí se využít objekty vytvořené v hráčském balíčku místo jejich opětovné implementace. Kvůli dodržení požadavků na univerzálnost knihovny a dodržení DRY principu bude s největší pravděpodobností vytvořena závislost nově vyvíjeného balíčku na balíčku stávajícím, čímž se docílí jejich provázání a možnosti využití zmíněných funkcionalit bez jejich zbytečného výskytu na dvou místech zároveň.

### 1.3 Role Pána jeskyně

Jak již bylo nastíněno v sekci zabývající se samotnou hrou Dračí doupě, Pán jeskyně se zabývá především přípravou dobrodružství a řízením průběhu hraní. Tyto dvě činnosti nyní budou rozebrány v pořadí, v němž za normálních okolností probíhají.

#### 1.3.1 Příprava dobrodružství

První činností, kterou každý Pán jeskyně musí vykonat, je příprava dobrodružství pro zbytek hráčů. V této fázi Pán jeskyně vymyslí příběh, zápletku a klíčové okamžiky dobrodružství. Následně se pustí do větších detailů vytvořením klíčových postav, nestvůr, jejich pokladů a pastí, které na postavy budou po cestě za cílem čekat. Celý proces prolíná vytváření map.

Jednou z předností hry Dračí doupě je schopnost reagovat na libovolné reakce hráčů. Z tohoto důvodu by kvalitní dobrodružství mělo být připravené na jakoukoliv situaci. Nemělo by se proto stát, že při odbočení družiny z hlavní cesty stráví Pán jeskyně další hodinu a půl vytvářením situace, do níž se tímto odbočením družina dostala.

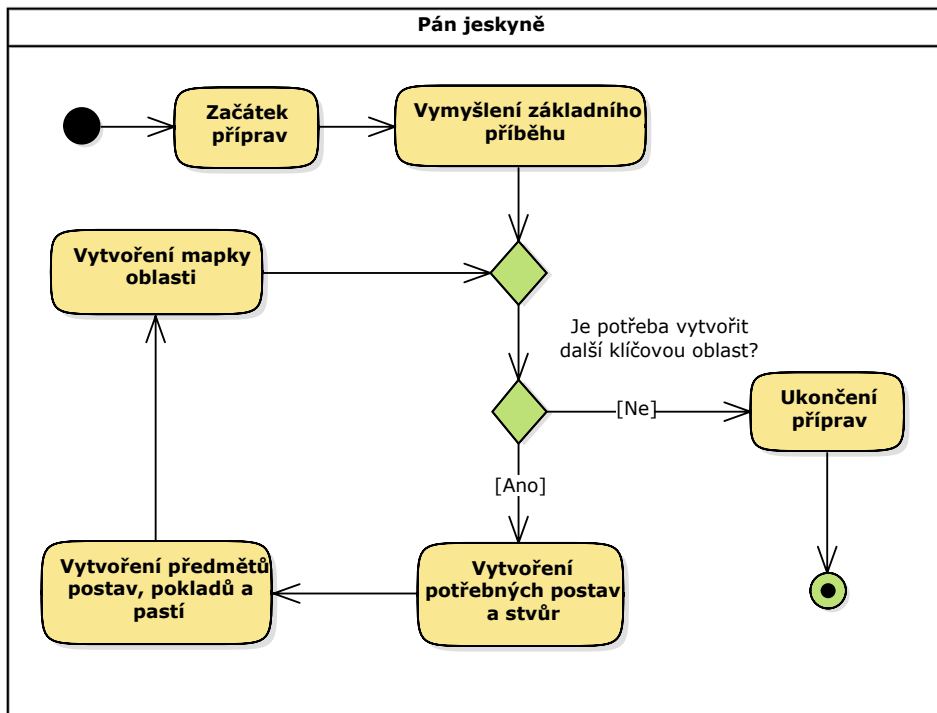
Vytvoření kvalitního dobrodružství tudíž může zabrat velké kvantum času, a i tak se mnohdy stane, že se postavám podaří dostat do situace, kterou Pán jeskyně nepředpokládal. V takovém případě je na obratnosti, výřečnosti a fantasii Pána jeskyně, jak se mu podaří situaci vyřešit.

Celý proces vytváření dobrodružství je nastíněn v diagramu 1.2 a podstatné dílčí činnosti jsou rozebrány v následujících podkapitolách.

##### 1.3.1.1 Vytváření nestvůr a cizích postav

Při vytváření nestvůr využívá Pán jeskyně zpravidla bestiáře, který je součástí pravidel[1]. V něm najde popis nestvůry, velice často doplněný o obrázek a potřebné informace pro její vytvoření, především pak tzv. „životaschopnost“. Jedná se o formuli, která oznamuje Pánům jeskyně, kolika kostkami kterého typu je třeba hodit, pro určení počtu životů nestvůry. Ten se spočte jako





Obrázek 1.2: Nastínění typické přípravy dobrodružství

součet výsledků hodu těchto kostek. Některé typy nestvůr může Pán jeskyně vybavit zbraněmi (zpravidla pokud jim to jejich tělesná stavba dovoluje).

Cizí postavy vytváří Pán jeskyně podobně, jako by hráč vytvářel svou postavu. V tomto případě si však může vybrat jaké podrobnosti zohlední či jakými předměty své cizí postavy vybaví.

V obou případech se může Pán jeskyně rozhodnout, že mu pravidla[1] nevyhovují, a vymyslet si postavu či nestvůru vlastní. V tomto ohledu nemá žádné zábrany, měl by však zohlednit zasazení těchto bytostí do příběhu.

### 1.3.1.2 Vytváření předmětů

Na rozdíl od vytváření nestvůr si Pán jeskyně jen málokdy vystačí s pravidly[1], když vytváří předměty. Je v nich sice nemalá kolekce zbraní, brnění a jiných typických předmětů, ovšem většina dobrodružství potřebuje předměty, které do nich přímo zapadají.

Do této sekce spadá také vytváření pastí. Pastí mají oproti obyčejným předmětům navíc dvě charakteristiky – tzv. „past“ a efekt. „Pastí“ se rozumí formule, která určuje počet a typ kostek, kterými si hráč ovládající postavu, jež do pasti vstoupila, musí hodit. Dále pak určuje hranici, kterou musí součet

hodu kostek překročit pro vyhnutí se efektu. Pakliže hráč v hodu proti „pasti“ neuspěje, je jeho postava postižena efektem.

Tímto efektem může být například krvácení, které bude způsobovat pravidelnou ztrátu menšího počtu životů do doby, než bude vyléčeno, či zlomená kost, která bude postavu omezovat při cestování a boji.

### 1.3.1.3 Tvorba map

Pro nastínění situací hráčům se používají mapy. Ať už se jedná o jednoduchou malůvku nebo o složitou mapu s vrstevnicemi, vždy je lepší mít přehled o oblasti, v níž se družina postav nachází. Tyto mapy jsou pak zpravidla slovně zprostředkovávány hráčům, kteří si je na volné archy papírů překreslují.

Mapy bývají dvojího typu. Jedny zobrazují rozsáhlejší geografickou oblast, kterou může být například údolí, horský průsmyk, město či celé království. V těchto mapách se Pán jeskyně zaměřuje především na rozložení klíčových oblastí. Detailní mapy těchto oblastí představují druhý typ map. Tyto mapy zobrazují přesné půdorysy místností v klíčových oblastech, rozložení pastí, rozmístění nepřátel a umístění předmětů a tajných chodeb.

Mapy jako takové jsou spíš pomocným prostředkem, a nejsou tudíž nezbytné k samotnému hraní. Přesto se velmi doporučuje jejich používání, neboť pomáhají zdokonalit herní atmosféru.

### 1.3.2 Řízení hry

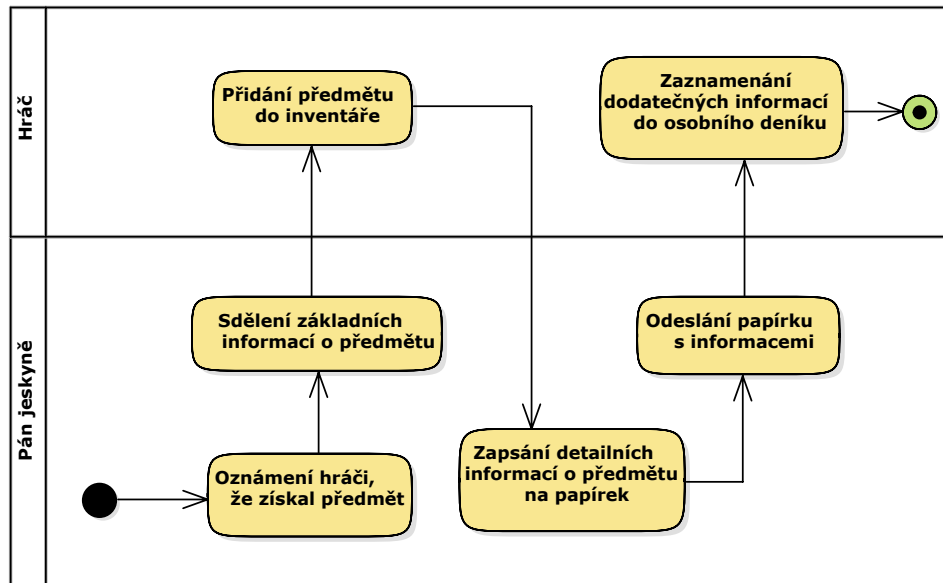
Jakmile je dobrodružství připraveno a všichni hráči sedí připraveni u jednoho stolu, začíná samotné hraní. To by mělo probíhat formou diskuze, v níž vždy jeden hovoří a zbytek hráčů čeká na projevení svého názoru. Většinou se ovšem hráči překřikují, zapomínají na „zlaté pravidlo“ (viz sekci 1.1.4) či se nenápadně snaží nahlédnout do poznámek Pána jeskyně. Pán jeskyně má tudíž kromě popisování událostí také za úkol udržet přátelskou atmosféru mezi hráči.

V následujících podkapitolách budou rozebrány specifické situace, se kterými se Pán jeskyně velmi často setkává při řízení hry a které se vymykají obecnému průběhu hry (viz sekci 1.1.2).

#### 1.3.2.1 Předání předmětů

Jednou z typických situací při řízení hry je předání nově nalezeného předmětu některému z hráčů. Pán jeskyně oznámí hráči, že jeho postava získala určitý předmět, a hráč si následně tento předmět připiše do inventáře. Jakmile má hráč předmět zapsaný, napíše Pán jeskyně detaily předmětu, jež má znát postava, ovšem ne zbytek hráčů, na papírek, který hráči pošle. Tento hráč si pak detaily předmětu zaznamená do svého osobního deníku [2].

Tento proces je zachycen na diagramu 1.3.



Obrázek 1.3: Nastínění průběhu předání předmětu hráči

### 1.3.2.2 Popisování nových kouzel a schopností

V průběhu dobrodružství získávají postavy hráčů zkušenosti a společně s nimi nové úrovně. Některé postavy se mohou při výcviku na novou úroveň naučit i nové schopnosti či kouzlu, které jsou popsány v pravidlech[1]. Mimo tyto případy se mohou postavy novým kouzlům a schopnostem naučit během dobrodružství (například v ruinách starobylého chrámu). V takovémto případě musí Pán jeskyně hráči novou schopnost či nové kouzlo popsát.

Průběh popisování nového kouzla či nové schopnosti je velmi podobný procesu předání předmětu (viz sekci 1.3.2.1), až na to, že nové kouzlo či novou schopnost si hráč zapíše do jiného místa osobního deníku[2] a detaily tohoto kouzla či schopnosti jsou skoro vždy tajné.

### 1.3.2.3 Popisování nově působících efektů

Na svých cestách za dobrodružstvím se členové družiny setkávají s vlivy, které mohou být jak příznivé (požehnání, extrémní štěstí, dobrá kondice, ...), tak nepříznivé (hlad, prochladnutí, krvácení, ...). Tyto vlivy pak mohou mít na jednotlivé postavy efekty, které upravují některé z jejich vlastností.

Pán jeskyně při vytvoření tohoto efektu oznámí hráčům pouze informaci, že na jejich postavy působí některý vliv. O existenci efektu uvědomuje Pán jeskyně hráče zpravidla až v okamžiku, kdy se projeví ovlivnění nějaké vlastnosti

jejich postavy. V tomto okamžiku si může hráč poznamenat, jakým způsobem je jeho postava ovlivněna.

### 1.3.2.4 Posílání tajných vzkazů

Během hraní se často stává, že Pán jeskyně potřebuje některému z hráčů sdělit něco, co se má dozvědět výhradně jeho postava. Samozřejmě může informaci sdělit všem s poznámkou, že tato informace je určena pouze pro postavu daného hráče, a zároveň je požádat, aby hráli dále tak, jako by o ní nevěděli.

Tento přístup si ovšem většina Pánů jeskyně dovolit nemůže, neb hráči mívají problém s dodržováním tzv. „zlatého pravidla“ (viz sekci 1.1.4). Z tohoto důvodu se mnoho Pánů jeskyně uchyluje ke vzkazům napsaným na kousek papírku, který nenápadně pošlou cílovému hráči.

### 1.3.2.5 Udržování přehledu o postavách

Jak již bylo zmíněno v předchozích kapitolách, Pán jeskyně by měl mít přehled o celém svém vytvořeném dobrodružství. To zahrnuje kromě připraveného dobrodružství a několika doplňujících poznámek i samotnou družinu postav hráčů, což je opravdu mnoho informací. Informace o postavách si ale evidují sami hráči, a proto si většina Pánů jeskyně v rámci šetření místa a zabránění nekonzistencí informace o postavách hráčů neudržuje.

V momentě, když nějakou informaci o postavě potřebují, požádají hráče o jeho osobní deník[2] a z něho tyto informace vyčtou. Druhou alternativou získávání informací o postavě je daného hráče se na tuto informaci přímo zeptat.

## 1.4 Rozbor domény

Při zkoumání role Pána jeskyně byly pro popis využívány části herní domény, která bude v této sekci podrobněji rozebrána společně se vztahy mezi souvisejícími entitami.

Základním objektem v doméně z pohledu Pána jeskyně je dobrodružství, neboť kolem něj se soustřeďuje celý proces hraní. Každé dobrodružství má svůj název a stručný popis osvětlující základní herní kontext a cíle. Mimo to bývá u některých dobrodružství zvykem evidovat i aktuální herní čas pro lepší nastínění atmosféry.

Ostatní objekty domény se dají rozdělit do dvou skupin na základě toho, zdali přímo souvisejí s určitou lokací, či jsou využívány spíše obecně na více místech, a s konkrétním místem tudíž nesouvisí.

Na konci každé sekce věnované určitému typu objektů je diagram znázorňující vlastnosti jednotlivých objektů a vztahy mezi těmito objekty.

### 1.4.1 Objekty spojené s lokací

Do této skupiny objektů spadají takové objekty, které úzce souvisejí s určitou lokací tím, že se v ní buď nacházejí, nebo ji popisují. Klíčovým objektem této skupiny je lokace. Každé dobrodružství může být děleno na vícero lokací, minimálně by však mělo obsahovat alespoň jednu. Lokace bývá charakterizována svým názvem, který by měl být v rámci přehlednosti unikátní, a případným popisem. Kvůli zvýšení přehlednosti mohou být lokace děleny na další lokace.

Jedním z nejpřehlednějších způsobů, jak zachytit velikost, charakteristiku a rozložení dalších herních objektů v rámci lokace, je použití mapy. Mapa má svůj název a doplňující popis, který vysvětluje skutečnosti, jež z mapy nemusejí být zřejmé, či obsahuje vysvětlení legendy. Každá lokace může být znázorněna na více mapách, většinou to ale není nutné.

Nejdůležitějšími objekty nacházejícími se v lokaci jsou předměty a bytosti.

#### 1.4.1.1 Předměty

Předměty se dělí na několik typů, avšak každý předmět je charakterizován názvem, popisem, cenou a váhou.

**název** Název předmětu se používá pro jeho identifikaci (v rámci typu i případné jedinečnosti).

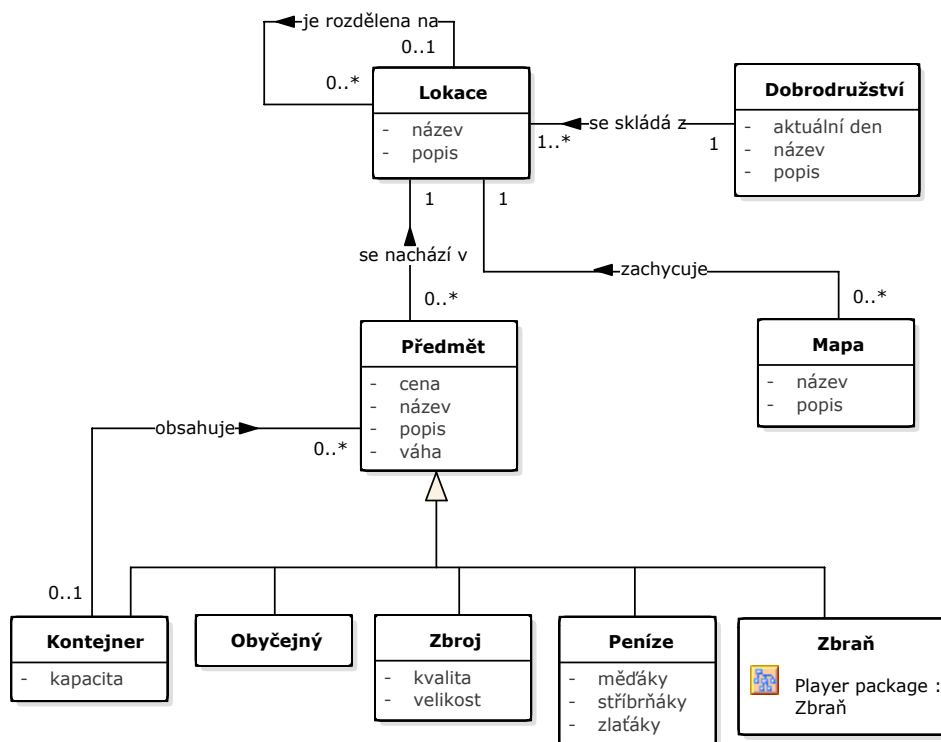
**popis** Popis pomáhá při dotvoření si představy o předmětu. Mohou zde být textově zachyceny jednotlivé vizuální detaily či jeho stručná historie (pakliže se jedná o známý předmět).

**cena** Cenou předmětu se rozumí průměrná cena, za níž lze konkrétní předmět koupit u průměrného obchodníka. Jedná se proto o údaj čistě orientační, neb celkem přirozeně za meč nabídne kovář pravděpodobně mnohem více než pekař.

**váha** Váha předmětu slouží k výpočtu zatížení postavy a jejího případného snížení pohyblivosti. Tato metrika se udává v „mincích“. „Mince“ je jednotka váhy a její hodnota je stejná jako váha jedné mince. Tato jednotka je ryze praktická, neboť mince jsou nejhojnějším a (až na pár výjimek) nejlehčím předmětem ve hře Dračí doupě.

Základními typy předmětů jsou: kontejner, zbroj, peníze a zbraň. V rámci dělení pak vznikla ještě jedna skupina – ostatní –, kam spadají všechny předměty, jež nejdou zařadit do žádného ze základních typů. Rozborem jednotlivých vlastností specifických typů předmětů a jejich případných podtypů se ve své bakalářské práci zabývá Šárka Weberová [3]. Tyto podrobnosti však nejsou pro pochopení domény z pohledu Pána jeskyně klíčové, a nebudou zde proto rozebrány.

Vztahy předmětů k dalším objektům domény jsou znázorněny na diagramu 1.4.



Obrázek 1.4: Doménový model popisující předměty a mapy

#### 1.4.1.2 Bytosti

Bytosti jsou komplexními objekty, k jejichž kompletnímu popisu je potřeba následujících patnáct vlastností:

**název** Název se používá pro identifikaci typu bytosti (např. drak).

**popis** Popis obsahuje dodatečné informace o konkrétní bytosti (např. s páskou přes oko, pahýl místo levé nohy, ...).

**velikost** Ve hře Dračí doupě je vymezeno šest skupin dle velikosti. Od nejmenší po největší jsou to A0, A, B, C, D a E s tím, že průměrný člověk spadá do skupiny B. Tato vlastnost bytosti určuje, do které skupiny bytost svou velikostí náleží.

**životaschopnost** Životaschopnost je dvojice čísel, která udává způsob, jakým lze určit životy bytosti. První číslo určuje počet hodů šestistěnnou kostkou a je vždy celé a nezáporné. Druhé číslo představuje tzv. „opravu“, tj. hodnotu, která se k výsledku hodu šestistěnnými kostkami přičte.

Toto číslo může být i záporné, avšak nesmí se stát, že by vytvořená bytost měla méně než jeden život.

**životy** Tato hodnota udává, kolik má bytost ve skutečnosti životů.

**útočné číslo** Útočné číslo popisuje způsoby, jakými může bytost útočit, a jejich sílu. Jedná se zpravidla o soubor dvojic čísel, ve kterých první číslo představuje přímo útočné číslo, které bývá doplněno o útočnost daného typu útoku.

**útočnost** Útočnost je úprava, která se provede s výsledkem hodu, pokud rozdíl součtu hodu na útok a útočného čísla byl větší nebo roven součtu hodu na obranu a obranného čísla. Pokud by výsledek byl po úpravě menší než jedna, je zaokrouhlen právě na jedna.

**obránné číslo** Obranné číslo se přičítá k hodu na obranu, čímž snižuje poškození, které by bylo bytosti uděleno útokem.

**bojovnost** Bojovnost bytosti se používá pouze u méně inteligentních bytostí a symbolizuje jejich touhu po boji. Čím nižší je tato hodnota, s tím větší pravděpodobností se bude bytost chovat agresivně a bude mít tendence družinu napadat. Bytosti s inteligencí devět a výše se vždy chovají logicky a družinu nenapadnou bezdůvodně.

**pohyblivost** Pohyblivost udává, jak rychle se bytost může pohybovat při pronásledování a boji. Pohyblivost může být upravena speciální typem bytosti (například šelmy mají při pronásledování dvojnásobný bonus za pohyblivost).

**vytrvalost** Tato vlastnost ovlivňuje dobu, jakou bude bytost schopná pronásledovat postavy či před nimi utíkat. Podobně jako pohyblivost může být vytrvalost upravena speciálním typem bytosti (například draky a magické tvory nelze unavit).

**odolnost, intelligence a charisma** Tyto tři vlastnosti slouží pro rozhodování, zdali bytost uspěje v hodu proti „pasti“ založené na dané vlastnosti.

**přesvědčení** Přesvědčení bytosti udává její základní charakterové rysy. Dělí se do pěti kategorií, kterými jsou „Zákonné dobro“, „Zmatené dobro“, „Neutrální“, „Zmatené zlo“ a „Zákonné zlo“.

**poklady** Tato informace pomáhá Pánu jeskyně určit, kolik věcí má bytost u sebe, potažmo ve svém brlohu, a jakou tyto věci mají hodnotu. Jedná se o dvě čísla, z nichž první vyjadřuje celkovou cenu věcí ve zlatácích, druhé celkovou magenergii kouzelných předmětů.

Tato hodnota je ovšem pouze orientační a Pán jeskyně se může rozhodnout ji ignorovat.

**zkušenost** Zkušenost udává počet zkušenostních bodů, které by měl Pán jeskyně rozdělit mezi postavy, které nestvůru porazily. Poměr rozdělení zkušenostních bodů je čistě na Pánu jeskyně, ale většinou se řídí dle zásluh, jaké jednotlivé postavy měly na poražení bytosti.

Kromě těchto základních vlastností mohou bytosti umět sesílat kouzla, ovládat speciální schopnosti a být ovlivňovány rozličnými efekty.

Návaznost bytostí na další objekty domény je patrná z diagramu 1.5.

### 1.4.1.3 Cizí postavy

Speciálním typem bytostí jsou cizí postavy. Tyto bytosti plní v příběhu většinou klíčové role a složením vlastností se velmi podobají hráčským postavám. Pán jeskyně si proto tyto postavy eviduje podobným stylem jako hráči svoje postavy.

Na rozdíl od zbylých druhů bytostí, které zpravidla neopouštějí lokaci, k níž náleží, může družina na cizí postavy narazit na více místech v dobrodružství. Přesto toto migrování většinou nepřevládá nad rolí v jednotlivých lokacích, a proto byly cizí postavy zařazeny do této kategorie.

## 1.4.2 Objekty nezávislé na lokaci

Druhou skupinu doménových objektů tvoří takové objekty, které nejsou přímo spojeny s jednou určitou lokací. Jedná se tedy o objekty nehmotné a migrující.

Diagram 1.6 obsahuje přehled objektů nezávislých na lokaci a jejich vztah vůči dalším objektům domény.

### 1.4.2.1 Nehmotné objekty

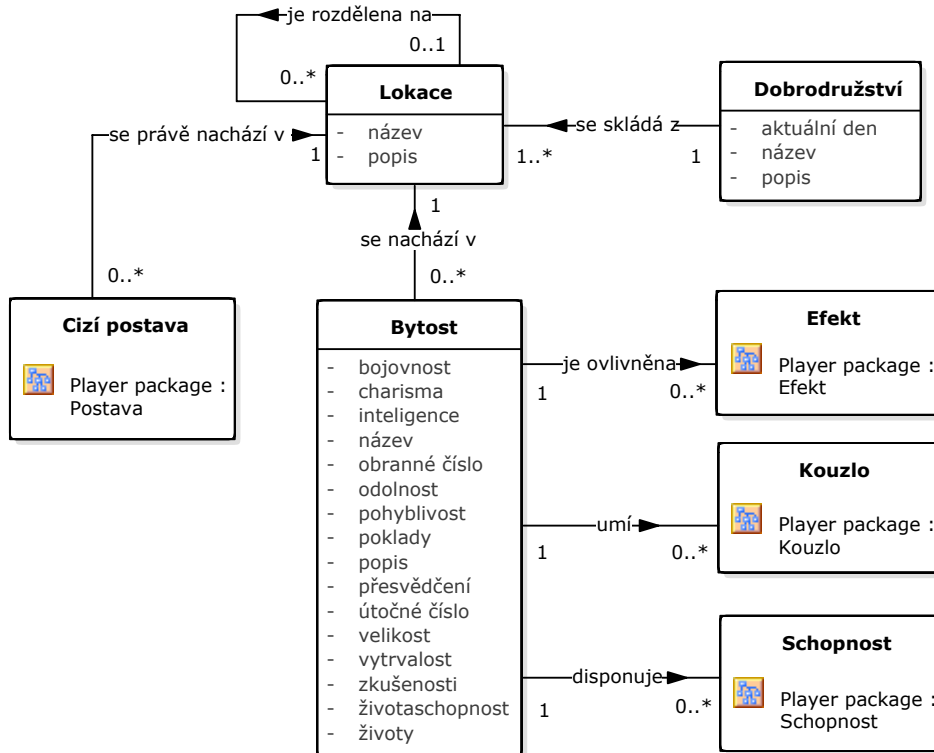
Do této skupiny spadají nehmotné objekty domény, kterými jsou kouzla, schopnosti a efekty. Množství nehmotných objektů v dobrodružství není nijak omezeno, tj. Pán jeskyně jich může vytvořit libovolně mnoho či nemusí vytvořit žádné (ačkoliv tento scénář není příliš pravděpodobný).

Podrobnosti jednotlivých nehmotných objektů a jejich vazby na další objekty nejsou pro pochopení role Pána jeskyně ani přidružených procesů potřeba. Pokud by přesto čtenář chtěl jejich rozbor vidět, je k nalezení v bakalářské práci Šárky Weberové[3].

### 1.4.2.2 Migrující objekty

Hlavním zástupcem migrujících objektů je družina postav. Každého dobrodružství se musí účastnit alespoň jedna družina a každá družina by se měla účastnit alespoň jednoho dobrodružství (jinak celkem postrádají smysl). Teoreticky se jednoho dobrodružství může účastnit naráz více družin a konkrétní družina může prožít více dobrodružství, avšak tyto scénáře nejsou velmi časté.





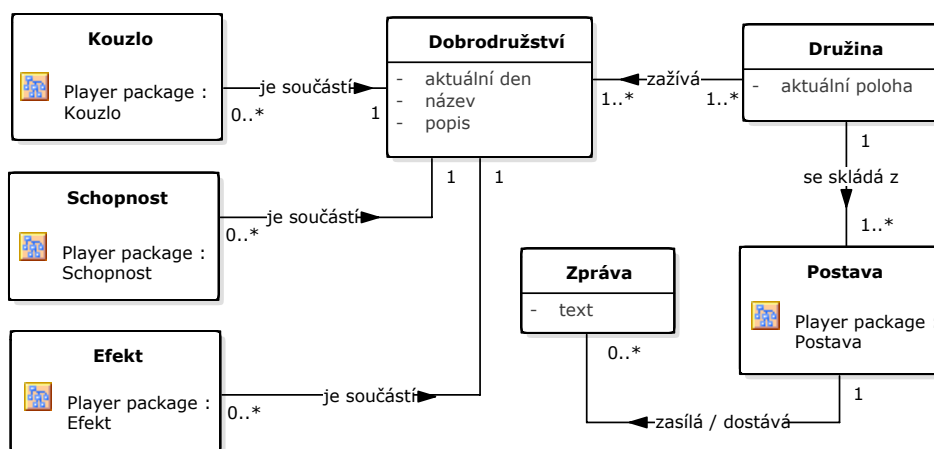
Obrázek 1.5: Doménový model popisující bytosti

U družiny si Pán jeskyně zpravidla eviduje její současnou polohu, zbytek informací je evidován u jednotlivých postav.

Družina se skládá alespoň z jedné postavy. Vlastnosti o postavě si eviduje vždy hráč, který za danou postavu hraje. Jedinou výjimkou jsou tajné zprávy, které si Pán jeskyně s daným hráčem posílá. Těchto zpráv může být neomezeně mnoho či nemusí existovat pro daného hráče žádná, záleží čistě na situaci. Zbytek informací a vazeb je pro Pána jeskyně zprostředkován skrze hráčův „osobní deník“ a nebude proto rozebírán zde, nýbrž v bakalářské práci Šárky Weberové[3], která se zabývá rozšířením hráčského balíčku, a tedy i touto problematikou.

## 1.5 Vymezení požadavků na nový balíček

Na základě zpětné vazby testerů hráčského balíčku pro hru Dračí doupě a rozboru typických herních procesů byly vymezeny požadavky na funkcionalitu nového balíčku. Druhá skupina požadavků je spojena s potřebou integrace



Obrázek 1.6: Doménový model popisující objekty, jež nejsou závislé na lokaci

nového balíčku do aplikace MobChar.

Požadavky z obou těchto skupin budou v následujících sekcích v bodech vypsány a podrobněji vysvětleny.

### 1.5.1 Funkční požadavky

Do této podkategorie spadají takové požadavky, které se vztahují k funkcionálně novému balíčku. Na základě analýzy herních procesů byly vymezeny tyto požadavky:

**Adresářová struktura ukládání** Nový balíček bude vytvořené předměty, nestvůry, cizí postavy a importované mapy ukládat v adresářové struktuře, kterou si budou moci uživatelé upravovat (vytváření nových adresářů, úprava jejich detailů, mazání adresářů a přesun položek mezi adresáři).

**Příprava herních objektů** Nový balíček bude umožňovat uživatelům vytvářet nové herní objekty (předměty, kouzla, schopnosti a efekty), vytvořené herní objekty upravovat a mazat. Z těchto objektů budou uloženy v adresářové struktuře dobrodružství pouze předměty. Zbytek objektů bude mít speciální obrazovky pro interakci.

**Příprava nestvůr a cizích postav** V rámci nového balíčku budou moci uživatelé vytvářet nové nestvůry a cizí postavy, tyto nestvůry a postavy upravovat a mazat a v rámci editace přidávat další volitelné vlastnosti. Vytvořené nestvůry a cizí postavy budou uloženy v adresářové struktuře dobrodružství.

**Správa map** Uživatelé budou mít možnost si prohlížet a mazat mapy importované z desktopové aplikace pro Pány jeskyně DeskChar. Mapy budou uloženy v adresářové struktuře dobrodružství.

**Předávání předmětů** Nový balíček bude umožňovat posílání vytvořených předmětů do balíčku pro hráče pro hru Dračí doupě na jiném zařízení. Zařízení budou muset být před přenosem předmětu spárována.

**Sdílení kouzel a schopností** Vytvořený balíček bude uživatelům nabízet možnost přeposlat detaily vytvořeného kouzla či schopnosti do balíčku pro hru Dračí doupě na jiném zařízení. Tato zařízení budou muset být před přenosem souborů spárována.

**Sdílení efektů** V rámci nového balíčku bude uživatelům umožněno odeslání detailů vytvořeného efektu do balíčku pro hru Dračí doupě na jiném zařízení. Tato zařízení budou muset být před přenosem souborů spárována.

**Posílání zpráv** Uživatelé budou moci při používání nového balíčku posílat zprávy do balíčku pro hru Dračí doupě na jiném zařízení a zprávy z tohoto balíčku přijímat. Balíček nebude podporovat skupinové konverzace a posílat bude možno pouze obyčejný text. Konverzace si bude udržovat pouze omezený počet zpráv v historii.

**Získávání přehledu o postavách** Ve vytvořeném balíčku budou mít uživatelé možnost prohlédnout si postavy hráčů, s jejichž zařízeními jsou spárováni. Tento náhled bude probíhat pomocí exportování nahlížené postavy do souboru, přeposlání tohoto souboru a následné nahrání do balíčku.

**Import a export dobrodružství** Nový balíček bude umožňovat importování dobrodružství z vytvářené desktopové aplikace pro Pány jeskyně DeskChar a vyexportování současného dobrodružství do takového stavu, aby bylo možno toto dobrodružství ve zmíněné desktopové aplikaci otevřít. Importování a exportování bude probíhat pomocí souborů.

### 1.5.2 Nefunkční požadavky

Nefunkční požadavky se zaměřují na nároky na hardware, software či výkonnostní specifikace vyvíjeného softwaru. V závislosti na zpětné vazbě testerů hráčského balíčku pro Dračí doupě a podmínek pro možnost začlenění nového balíčku do systému aplikací MobChar byly vymezeny tyto požadavky:

**Podpora pro Android** Nový balíček bude podporován na zařízeních, které mají OS Android s verzí 4.1 (API 16 – JELLY\_BEAN) a vyšší.

**Jazyková podpora** Balíček bude podporovat lokalizaci do českého a anglického jazyka. Pokud nastaveným jazykem na zařízení nebude ani jeden z těchto jazyků, bude použita lokalizace do anglického jazyka.

**Rozlišení** Cílovým rozlišením nového balíčku je 1080x1920 (mdpi), pro zbylé typy rozlišení nebude vytvářena speciální verze. Balíček bude podporovat portrait i landscape zobrazení u všech obrazovek. Speciální layout pro tablet a AndroidTV nebude implementován, na těchto zařízeních bude použit layout aplikace.

**Přístup k externímu uložení** Balíček bude umožňovat importování a exportování dobrodružství pro aplikaci DeskChar skrze externí uložení zařízení. Z tohoto důvodu bude vyžadovat k tomuto uložení přístup a bez něj nebude podporovat všechnu funkcionalitu.

**Komunikace s jinými zařízeními** Balíček bude umožňovat komunikaci s jinými zařízeními, která budou muset mít nainstalovaný hráček balíček pro hru Dračí doupe a bude komunikovat výhradně s ním. Komunikace bude probíhat přes službu Bluetooth přenosem specifických souborů. Obě zařízení budou muset mít zapnutou službu Bluetooth a budou muset být skrze balíčky spárována.

**Ukládání dat** Nový balíček bude ukládat data pomocí služby SQLite, která je nativně dostupná na zařízeních s OS Android. Pro využití této služby bude balíček vyžadovat speciální povolení.

## 1.6 Interakce uživatele s aplikací – případy užití

Vzhledem k vysoké úrovni abstrakce u funkčních požadavků budou v této sekci vypsány případy užití, které popisují, jakým způsobem bude probíhat interakce uživatele (Pána jeskyně) s aplikací.

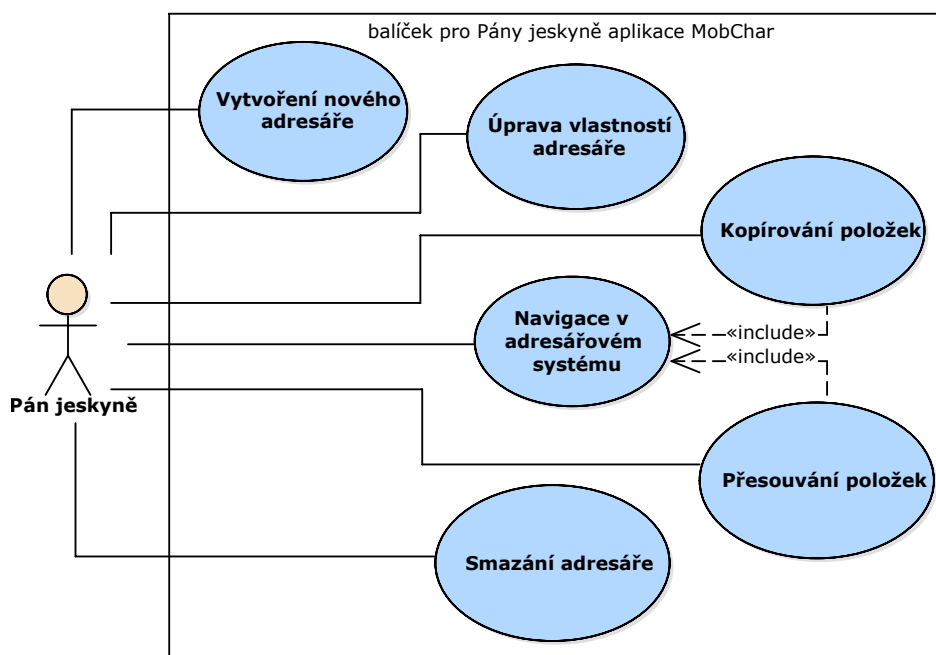
Tyto případy užití jsou rozděleny do skupin tak, aby v rámci každé skupiny byly případy užití, které mají podobné zaměření. Tímto rozdělením vznikly čtyři skupiny, které budou dále rozebrány.

Na konci této sekce je uvedena tabulka pokrytí požadavků případy užití (viz tab. 1.1).

### 1.6.1 Práce s adresářovou strukturou

Případy užití v této skupině se zaměřují na práci s adresářovou strukturou. Postupně budou pokryty všechny části korespondujícího požadavku (viz první požadavek „Adresářová struktura ukládání“ 1.5.1).

Případy užití v této skupině jsou zachyceny na diagramu 1.7.



Obrázek 1.7: Zobrazení případů užití spojených s adresářovou strukturou

### 1.6.1.1 Navigace v adresářovém systému

Jedním ze základních požadavků pro práci s libovolným adresářovým systémem je možnost navigace v něm. Aplikace nabízí Pánům jeskyně jednoduchý způsob, jak se mohou pohybovat směrem od kořene systému i směrem k němu, který je rozebrán v následujícím scénáři.

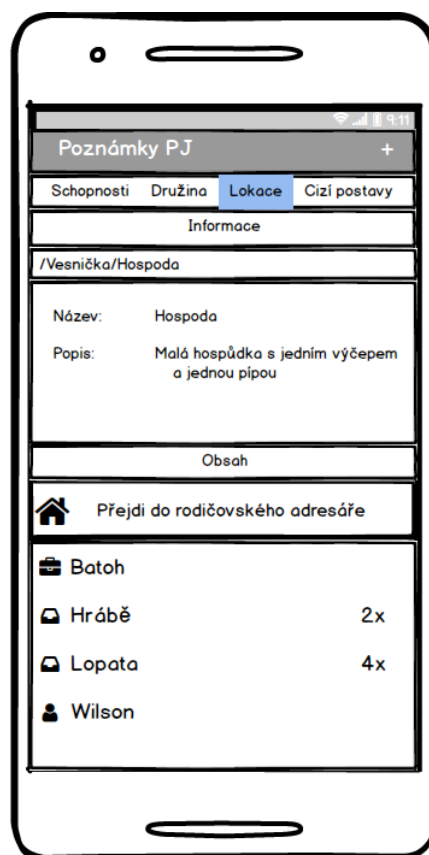
Začátek scénáře je v libovolném adresáři obrazovky „Lokace“. Aplikace zobrazí Pánu jeskyně obsah současného adresáře. Pokud cesta k cílovému adresáři vede ze současného adresáře směrem ke kořeni systému, zvolí Pán jeskyně možnost „Přejít do rodičovského adresáře“.

Naopak pokud na cestě k cílovému adresáři je třeba projít přes některý podadresář současného adresáře, zvolí Pán jeskyně tento podadresář krátkým kliknutím na jeho položku v zobrazeném seznamu obsahu současného adresáře.

Celý tento cyklus se opakuje, dokud Pán jeskyně nedospěje k cílovému adresáři. Příklad obrazovky adresáře zachycuje wireframe 1.8.

### 1.6.1.2 Vytvoření nového adresáře

Pro zvýšení členění adresářového systému umožňuje aplikace Pánům jeskyně zakládat vlastní adresáře. Tím si mohou podstatně zpřehlednit vyhledávání vytvořených herních objektů. Scénář popisující proces vytvoření nového adre-



Obrázek 1.8: Wireframe obrazovky adresáře „Hospoda“

sáře začíná na obrazovce s obsahem adresáře, v němž chce Pán jeskyně vytvořit nový adresář.

Pán jeskyně zvolí možnost „Vytvořit“, na což aplikace zareaguje zobrazením dialogu se všemi typy objektů, které je možné v tomto adresáři vytvořit. Pán jeskyně vybere možnost „Lokace“. Aplikace zobrazí obrazovku s poli pro vyplnění základních vlastností vytvářeného adresáře, které Pán jeskyně následně vyplní. Jakmile je Pán jeskyně hotov, zvolí možnost „Uložit změny“.

Aplikace nový adresář uloží do databáze a zobrazí obrazovku s obsahem původního adresáře. Nyní již bude obsahovat nově vytvořený adresář.

### 1.6.1.3 Úprava vlastností adresáře

Při vytváření návrhu rozdělení adresářů Páni jeskyně často opomínají detaily, které pak mohou vyžadovat drobné úpravy ve vlastnostech jednotlivých adre-

sářů. Případ, v němž by Pán jeskyně musel kvůli úpravě názvu adresáře tento adresář smazat a znovu vytvořit, je nemyslitelný, a proto aplikace umožňuje Pánům jeskyně vlastnosti vytvořených adresářů snadno upravit.

Způsob, jakým lze tyto úpravy provést, vyžaduje, aby se Pán jeskyně nacházel na obrazovce s obsahem adresáře, který si přeje upravit, a probíhá následovně.

Na začátku zvolí Pán jeskyně možnost „Zobrazit detail lokace“ krátkým kliknutím na hlavičku adresáře. Aplikace následně zobrazí obrazovku s detaily adresáře. Na této obrazovce zvolí Pán jeskyně možnost „Editovat lokaci“. Aplikace zareaguje zobrazením obrazovky s poli pro vyplnění základních vlastností upravovaného adresáře, v nichž budou předvyplněny současné hodnoty těchto vlastností.

Pán jeskyně tyto hodnoty upraví a zvolí možnost „Uložit změny“. Aplikace následně změní hodnoty adresáře v databázi a zobrazí obrazovku s aktuálními detaily tohoto adresáře.

### 1.6.1.4 Smazání adresáře

Během přípravy dobrodružství a samotného hraní se mohou vyskytnout adresáře, které jsou nadbytečné nebo byly vytvořeny omylem. Pro řešení této situace je nutným protikladem k možnosti vytvářet adresáře možnost vytvořené adresáře mazat. Proces, jakým lze smazání adresáře provést, začíná v rodičovském adresáři adresáře, který se Pán jeskyně chystá smazat.

Pán jeskyně nejdříve dlouhým kliknutím na danou položku v zobrazeném obsahu současného adresáře zvolí adresář, který chce smazat. Aplikace zobrazí dialog s možnostmi interakce se zvoleným adresářem. Pán jeskyně zvolí z nabídky možnost „Smazat položku“. Následně aplikace tuto položku odstraní z databáze a aktualizuje současnou obrazovku, takže zmizí položka smazaného adresáře.

### 1.6.1.5 Přesouvání položek

Dalším užitečným nástrojem pro reorganizaci adresářového systému, který aplikace nabízí, je přesouvání položek mezi adresáři. Pán jeskyně takto může přesouvat jednotlivé položky nebo adresáře včetně jejich obsahu. Scénář tohoto případu užití začíná na obrazovce s obsahem adresáře, ve kterém se nachází položka, kterou si Pán jeskyně přeje přesunout.

Pán jeskyně tuto položku zvolí dlouhým kliknutím na ni. Aplikace zobrazí dialog s možnostmi interakce s vybranou položkou. Pán jeskyně zvolí z nabídky možnost „Vyjmout“. Aplikace odstraní přesouvanou položku, uloží si ji do schránky a zobrazí v menu možnost „Vložit“. Pokud již schránka nějakou položku obsahuje, nebude možnost „Vyjmout“ viditelná.

Pán jeskyně se přesune do adresáře, do něž chce zvolenou položku přesunout (viz sekci „Navigace v adresářovém systému“ 1.6.1.1).

Jakmile se Pán jeskyně nachází v cílovém adresáři, zvolí možnost „Vložit“ z menu a aplikace vloží položku, která se nachází ve schránce, do tohoto adresáře. Zároveň tuto položku smaže ze schránky.

Pokud by Pán jeskyně opustil obrazovku před vložením vyjmuté položky, zůstane tato položka ve schránce, dokud nebude vložena.

### 1.6.1.6 Kopírování položek

Ač bývají dobrodružství velmi rozmanitá, většinou se najde skupina herních objektů, které se vyskytují ve více lokacích. Aby Páni jeskyně nemuseli tyto totožné položky znovu vytvářet, nabízí jim aplikace možnost jednotlivé položky kopírovat. Pokud Pán jeskyně chce položku zkopírovat, přesune se nejprve na obrazovku s obsahem adresáře, který danou položku obsahuje, a provede následující kroky.

Pán jeskyně zvolí ze zobrazeného seznamu dlouhým kliknutím položku, kterou chce zkopírovat. Aplikace zobrazí dialog s možnostmi interakce s vybranou položkou, z něhož si následně Pán jeskyně vybere možnost „Kopírovat“. Aplikace si zvolenou položku přidá do schránky a zobrazí možnost „Vložit“ v menu. Pokud již schránka nějakou položku obsahuje, nebude možnost „Kopírovat“ viditelná.

Pán jeskyně se přesune do adresáře, do něž chce zvolenou položku zkopírovat (viz sekci „Navigace v adresářovém systému“ 1.6.1.1).

Až se Pán jeskyně ocitne v adresáři, do něhož chce zkopírovat položku, učiní tak zvolením možnosti „Vložit“ z menu. Aplikace zareaguje vložením položky do současného adresáře a zároveň ji smaže ze schránky.

## 1.6.2 Práce s herními objekty

Druhou skupinu tvoří případy užití popisující práci s herními objekty (jmenovitě předměty, bytostmi, kouzly, schopnostmi a efekty), které pokrývají druhý, třetí a čtvrtý funkční požadavek (viz funkční požadavky 1.5.1). Vzhledem k tomu, že by tyto scénáře byly až na umístění skoro totožné, pokud by byly rozepsány jednotlivě, a přidaná hodnota by byla minimální, budou zde nastíněny obecněji.

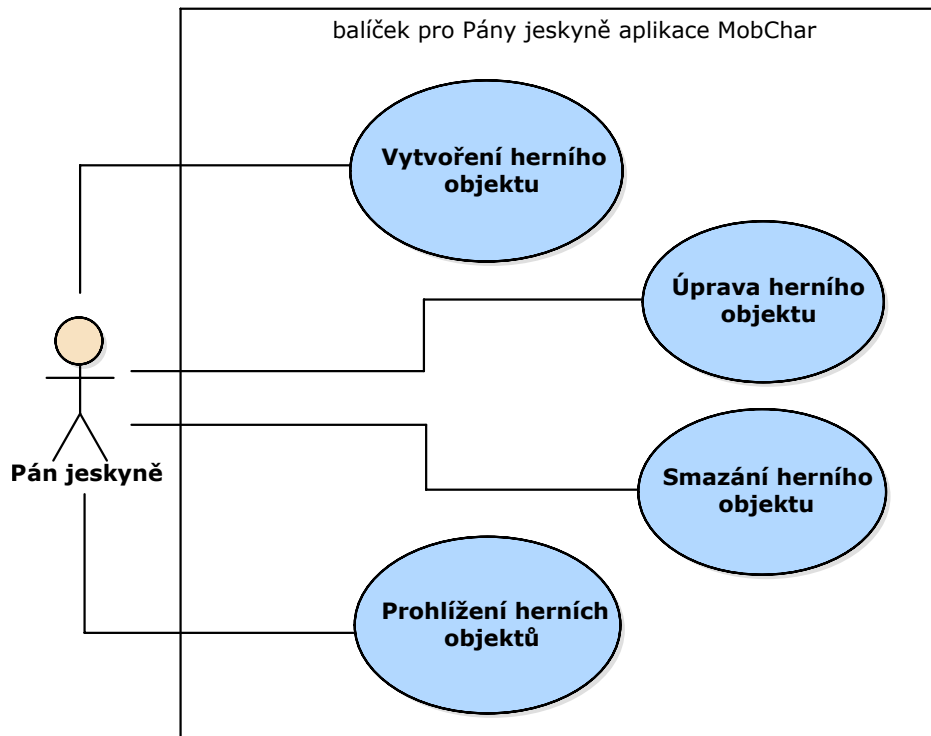
Jednotlivé typy případů užití jsou zachyceny na diagramu 1.9.

### 1.6.2.1 Prohlížení herních objektů

Základní operací pro práci s herními objekty je možnost prohlížet si podrobnosti těchto objektů. Způsob, jakým toho lze v aplikaci dosáhnout, je nastíněn ve zbytku sekce a vyžaduje, aby se Pán jeskyně nacházel na obrazovce se seznamem herních objektů, mezi nimiž se nachází i ten, který si chce Pán jeskyně prohlédnout.

Pán jeskyně tento objekt zvolí dlouhým kliknutím na něj. Aplikace pak zobrazí obrazovku s detaily daného herního objektu.





Obrázek 1.9: Zobrazení případů užití spojených s herními objekty

### 1.6.2.2 Vytvoření herního objektu

Druhou ze základních funkcí při práci s herními objekty je jejich vytváření. Páni jeskyně mají možnost herní objekty vytvářet na příslušné obrazovce zobrazující přehled daných objektů.

Na začátku zvolí Pán jeskyně možnost „Vytvořit“ z horního horizontálního menu. V případě předmětů a bytostí zobrazí aplikace dialog s různými typy objektů, které je možné vytvořit, a Pán jeskyně vybraný typ zvolí. Tento krok platí pouze u předmětů, v ostatních případech se přeskakuje.

Aplikace následně zobrazí obrazovku s poli pro vyplnění vlastností vytvářeného herního objektu. Pán jeskyně tato pole vyplní a zvolí možnost „Uložit změny“. Pokud by Pán jeskyně zapomněl vyplnit některá povinná pole či je vyplnil špatně, aplikace odmítne objekt vytvořit a chybně vyplněná pole zvýrazní.

Aplikace vytvoří herní objekt dle specifikací, uloží jej do databáze a zobrazí původní obrazovku s přehledem objektů daného typu, nyní již s nově vytvořeným objektem.

### 1.6.2.3 Úprava herního objektu

Aplikace také umožňuje Pánům jeskyně upravovat vlastnosti existujících herních objektů. To se může hodit v případě, že Pán jeskyně udělal při vytváření chybu nebo potřebuje vytvořit podobný objekt. V druhém případě tedy může objekt zkopírovat (viz sekci „Kopírování položek“ 1.6.1.6) a pak upravit.

Scénář tohoto případu užití začíná vždy na obrazovce s detailem herního objektu, který bude cílem úprav.

Pán jeskyně zvolí možnost „Upravit“ z horního menu. Aplikace zobrazí obrazovku s poli pro vyplnění vlastností herního objektu, v nichž budou předvyplněny současné hodnoty jednotlivých vlastností. Pán jeskyně upraví vlastnosti objektu a zvolí možnost „Uložit změny“. Pokud by Pán jeskyně zapomněl vyplnit některá povinná pole či je vyplnil špatně, aplikace odmítne objekt vytvořit a chybně vyplněná pole zvýrazní.

Aplikace následně uloží nové hodnoty vlastností objektu do databáze a zobrazí původní obrazovku s aktuálními detaily daného objektu.

Pro získání lepší představy o vzhledu této obrazovky slouží wireframe 1.10. Tento wireframe zachycuje obrazovku editace kouzla „Ohnivá koule“, v níž Pán jeskyně zapomněl vyplnit pole *Udržovací cena magenergie* a *Dosah*. Tato pole jsou proto zvýrazněna.

### 1.6.2.4 Smazání herního objektu

Poslední klíčovou operací pro práci s herními objekty je jejich mazání. Na rozdíl od případů užití popisující vytváření a úpravu objektů se tento případ vztahuje i na importované mapy. Smazání herního objektu může Pán jeskyně docílit z obrazovky se seznamem objektů daného typu následujícím způsobem.

Nejprve Pán jeskyně označí herní objekt, který má být smazán, dlouhým kliknutím na jeho položku v seznamu objektů. Aplikace zvýrazní vybraný herní objekt a zobrazí kontextové menu, z něhož Pán jeskyně zvolí možnost „Smazat“. Aplikace tento herní objekt smaže z databáze, zobrazí původní obrazovku a aktualizuje ji, aby již nezobrazovala smazaný objekt.

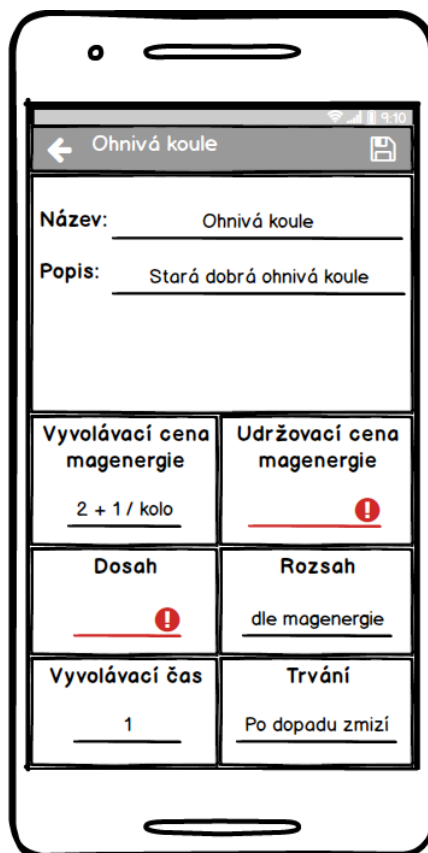
## 1.6.3 Práce s externím úložištěm

Do následující skupiny spadají případy užití spojené s využitím externího úložiště. Společně pokrývají požadavek „Import a export dobrodružství“ (viz funkční požadavky 1.5.1).

Výčet těchto případů užití je zachycen na diagramu 1.11.

### 1.6.3.1 Importování dobrodružství

Vytváření dobrodružství je dlouhodobý proces, který vyžaduje mnoho psaní a plánování, které může být na obrazovce mobilního zařízení velmi namáhavé,



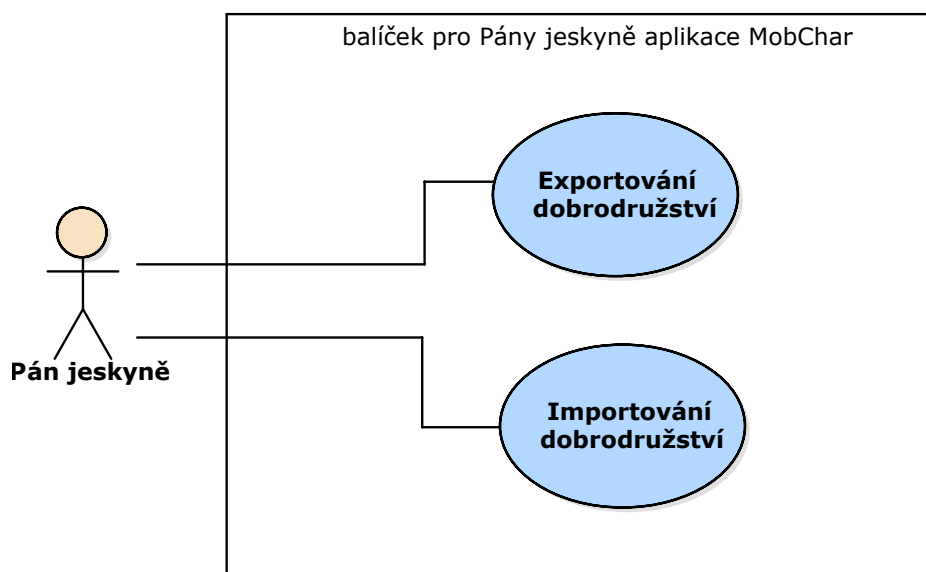
Obrázek 1.10: Wireframe obrazovky editace kouzla „Ohnivá koule“

zdlouhavé a nepřehledné. Páni jeskyně proto mají možnost dobrodružství vytvořit v desktopové aplikaci DeskChar.

Na druhou stranu k samotnému hraní je nepříjemné s sebou muset brát celý počítač. Aplikace proto umožňuje importování dobrodružství, která byla navržena v desktopové aplikaci. Proces importování začíná na úvodní obrazovce jádra aplikace v záložce „Dobrodružství“ poté, co Pán jeskyně přesunul dobrodružství, které chce importovat, do patřičné složky.

Pán jeskyně zvolí možnost „Importovat dobrodružství“. Aplikace zobrazí obrazovku se seznamem souborů, které se jí podařilo načíst, a dotáže se Pána jeskyně, který z nich si přeje nahrát. Pán jeskyně si daný soubor vybere krátkým kliknutím na jeho položku v seznamu.

Aplikace nahraje zvolený soubor a zobrazí dobrodružství, které obsahuje, na obrazovce společně s ostatními dobrodružstvími. Pokud by soubor s dobrodružstvím nebyl validní, nahrávání selže a aplikace o tom informuje Pána



Obrázek 1.11: Zobrazení případů užití spojených s externím úložištěm

jeskyně.

### 1.6.3.2 Exportování dobrodružství

Někdy se stává, že Páni jeskyně nestihnou připravit celé dobrodružství před prvním herním setkáním. Připraví proto část, jdou hrát a při hraní si zaevidují nemalé množství informací.

Nyní nastává problém, pokud je dobrodružství připravené v aplikaci DeskChar, neb jeho doplnění o evidované poznámky z hraní by bylo časově náročné a příprava další části dobrodružství na mobilním zařízení by byla nepohodlná.

Aplikace proto nabízí možnost aktuální dobrodružství exportovat do souboru, který může být otevřen v aplikaci DeskChar a dále upravován, čímž vzniklý problém elegantně řeší. Toto exportování je možné provést ze záložky „Dobrodružství“ hlavní aktivity jádra aplikace.

Nejprve zvolí Pán jeskyně dobrodružství, které chce exportovat, dlouhým kliknutím na jeho položku v seznamu. Aplikace zobrazí dialog s možnostmi interakce s vybraným dobrodružstvím, z nichž Pán jeskyně zvolí možnost „Exportovat“. Aplikace následně exportuje zvolené dobrodružství do souboru, který uloží ve specifické složce na externím úložišti. Jakmile s tímto procesem skončí, informuje o tom Pána jeskyně.

#### 1.6.4 Komunikace s jiným zařízením

Pojícím prvkem případů užití, které spadají do této čtvrté skupiny, je komunikace s jiným zařízením. Tyto případy užití pokrývají zbytek funkčních požadavků (tj. pátý až devátý požadavek, viz sekci 1.5.1).

První tři zmíněné případy užití figurují pouze jako dílčí součásti jiných případů užití a byly vyčleněny vzhledem k jejich opakovanému použití v různých případech užití. Samy však nejsou Pánu jiskyně přímo přístupné.

Na rozdíl od ostatních skupin mají tyto případy užití mezi sebou netriviální vazby, a jsou proto přehledně zachyceny na diagramu na konci této sekce (viz obr. 1.12). Jednotlivé případy užití budou rozebrány v takovém pořadí, aby v momentě použití reference měl čtenář již odkazovanou sekci přečtenou.

##### 1.6.4.1 Ověření dostupnosti cizího zařízení

Předtím, než je možné začít s cizím zařízením komunikovat, je potřeba se ujistit, zdali je toto zařízení dostupné, a v případě, že není, tuto skutečnost Pánu jiskyně oznámit. Proces ověřování dostupnosti bude probíhat následovně.

Aplikace se nejprve pokusí spojit s vybraným zařízením. Jakmile je zařízení dostupné, pokračuje aplikace v průběhu případu užití, který tento případ užití využívá.

Pakliže aplikace zjistí, že cizí zařízení není dostupné, zobrazí Pánu jiskyně text informující ho o této skutečnosti. Pán jiskyně pak požádá majitele vybraného zařízení, aby své zařízení zpřístupnil.

##### 1.6.4.2 Ověření dostupnosti služby Bluetooth

Jednou z nutných podmínek komunikace přes službu Bluetooth je, aby tato služba byla na zařízení Pána jiskyně aktivována. To se zdá jako samozřejmost, ovšem Páni jiskyně občas na aktivaci služby zapomenou. Toto ověření bude probíhat podle následujícího scénáře.

Aplikace pomocí služeb poskytovaných operačním systémem zjistí, zdali je služba Bluetooth dostupná a aktivovaná. Pakliže tomu tak je, pokračuje aplikace v průběhu případu užití, který tento případ užití využívá.

V opačném případě zobrazí aplikace dialog nabízející Pánu jiskyně možnost službu aktivovat. Pokud se Pán jiskyně rozhodne službu aktivovat, aplikace službu aktivuje a bude pokračovat v průběhu nadřazeného případu užití.

Pokud Pán jiskyně zvolí možnost „Zrušit“, skončí tento i nadřazený případ užití neúspěchem.

##### 1.6.4.3 Přeposlání souborů

Primární náplní komunikace přes službu Bluetooth bude posílání souborů obsahující herní informace mezi zařízeními. Při posílání souborů neprojevuje Pán

jeskyně žádnou činnost, a kromě zobrazení výsledku odeslání se celý proces děje na pozadí.

Aplikace nejprve ověří dostupnost služby Bluetooth (viz sekci 1.6.4.2) a dostupnost cílového zařízení (viz sekci 1.6.4.1). Pokud oba případy užití dopadly úspěšně, pokusí se aplikace o přenos vybraného souboru. Po dokončení této operace zobrazí Pánu jeskyně informaci o jejím výsledku.

### 1.6.4.4 Spárování zařízení

Operační systém Android vyžaduje před začátkem komunikace mezi zařízeními, aby tato zařízení byla spárovaná. Pán jeskyně proto bude potřebovat spárovat své zařízení se zařízením hráče, s nímž chce přes službu Bluetooth komunikovat. K dosažení spárování svého a cizího zařízení musí Pán jeskyně následovat tento postup.

Pán jeskyně nejprve zvolí záložku „Družina“ v horním horizontálním menu a aplikace zobrazí obrazovku se seznamem spárovaných zařízení. Pán jeskyně následně zvolí možnost „Spárovat další zařízení“. Aplikace před začátkem párování zkontroluje dostupnost služby Bluetooth (viz sekci 1.6.4.2).

Jakmile je služba dostupná, prohledá aplikace okolí a zobrazí pomocí dialogu seznam dostupných zařízení. Pán jeskyně zvolí zařízení, s nímž se chce spárovat, krátkým kliknutím na jeho položku v seznamu. Pokud Pán jeskyně v seznamu cílové zařízení nevidí, zvolí možnost „Zavřít“ a případně zkusí vyhledávání zopakovat.

Pokud si Pán jeskyně vybral zařízení, aplikace tomuto zařízení odešle žádost o spárování. Jakmile cizí zařízení přijme požadavek, aplikace si uloží detaily o tomto zařízení a postavě, která je na zařízení spuštěna. V případě, že cílové zařízení odmítne žádost o spárování, končí případ užití neúspěchem.

V obou případech aplikace informuje Pána jeskyně o výsledku procesu.

### 1.6.4.5 Odeslání informací o herním objektu

První možností komunikace, kterou aplikace nabízí, je odeslání informací o herním objektu (předmět, kouzlo, schopnost či efekt) jinému zařízení. To umožňuje Pánům jeskyně podstatně snáze předat informace o daném objektu, které by musel nejprve hráči ústně popsat a dle kterých by si pak musel daný hráč objekt sám vytvořit.

Nejprve Pán jeskyně spáruje své zařízení se zařízením cílovým (viz sekci 1.6.4.4) a následně se přepne na obrazovku se seznamem postav spárovaných zařízení.

Pán jeskyně zvolí postavu, jejímuž hráči chce herní objekt přeposlat, dlouhým stiskem na položku seznamu s jejím jménem. Aplikace zobrazí dialog s možnostmi interakcí („Přeposlat objekt“, „Otevřít konverzaci“, „Upravit přezdívku“ a „Smazat“). Pán jeskyně zvolí možnost „Přeposlat objekt“, na-

čez aplikace zobrazí obrazovku se všemi vytvořenými herními předměty (v adresářovém systému).

Pán jeskyně vybere objekt, který chce odeslat, dlouhým kliknutím na jeho položku v seznamu. Pokud by se v aktuálním adresáři předmět nenacházel, Pán jeskyně k němu donaviguje (viz sekci 1.6.1.1) a teprve pak ho zvolí.

Aplikace vybraný předmět zvýrazní a zobrazí v horním menu možnost „Odeslat“, kterou Pán jeskyně následně zvolí. Aplikace tento zvolený předmět odešle (viz sekci 1.6.4.3).

### 1.6.4.6 Získávání přehledu o postavě

V průběhu dobrodružství nastávají situace, v nichž Pán jeskyně potřebuje znát aktuální informace o některé z postav. To se může například dít, pokud se nepozorné postavě probírá věcmi pouliční zloděj. Aplikace v takovém případě Pánovi jeskyně umožňuje si od hráče vyžádat informace o jeho postavě a nechat si je poslat, čímž se vyhne možnosti, že se hráč dovtípí, že je okrádán.

Neboť aplikace k této operaci potřebuje přenášet soubory, musí být před přenosem daná dvě zařízení spárována (viz sekci 1.6.4.4). Samotný přenos probíhá podle následujícího scénáře, který předpokládá, že se Pán jeskyně nachází na obrazovce se seznamem postav spárovaných zařízení.

Nejprve si Pán jeskyně vybere ze seznamu postavu, jejíž podrobnosti si chce prohlédnout. Aplikace zobrazí dialog s možnostmi „Vyžádat obnovení“ a „Použít současný stav“. Pokud Pán jeskyně vybere možnost „Použít současný stav“, aplikace přímo spustí obrazovku se zvolenou postavou.

Při vybrání možnosti „Vyžádat obnovení“ vyšle aplikace požadavek zařízení majitele zvolené postavy a vyčká na odpověď (viz sekci 1.6.4.3). Pakliže nastane chyba při přenosu dat, aplikace tuto skutečnost Pánu jeskyně oznámí a případ užití skončí neúspěchem.

V opačném případě přijme aplikace jako odpověď vyexportovanou postavu a naimportuje ji do balíčku. Následně spustí obrazovku s právě importovanou postavou a Pán jeskyně si dohledá požadované informace o postavě.

### 1.6.4.7 Odesílání zpráv

Další z užitečných funkcí nabízených aplikací je posílání textových zpráv spárovanému zařízení. Pomocí této funkce může Pán jeskyně vynutit dodržování „zlatého pravidla“ (viz sekci 1.1.4) předáváním určitých informací pouze těm hráčům, jejichž postav se týkají.

Pakliže Pán jeskyně chce odeslat hráči zprávu, přepne se na obrazovku se seznamem postav spárovaných zařízení, mezi nimiž musí být zařízení cílového hráče (viz sekci 1.6.4.4).

Pán jeskyně poté zvolí postavu, jejímuž hráči chce poslat zprávu, dlouhým kliknutím na položku seznamu s jejím jménem. Aplikace zobrazí dialog

s možnostmi interakcí („Přeposlat objekt“, „Otevřít konverzaci“, „Upravit přezdívku“ a „Smazat“) a Pán jeskyně zvolí možnost „Otevřít konverzaci“.

Aplikace zobrazí obrazovku s konverzací mezi Pánem jeskyně a cílovým hráčem společně s případnou historií jejich konverzace. Pán jeskyně zvolí možnost zadání textu klepnutím na pole pro možnost zadání textu, načež aplikace toto pole zvětší a zobrazí klávesnici.

Pán jeskyně napíše zprávu, kterou chce odeslat, a stiskne tlačítko „Odeslat“. Aplikace vytvoří soubor se zprávou a ten cílovému hráči odešle (viz sekci 1.6.4.3).

### 1.6.4.8 Přijímání zpráv

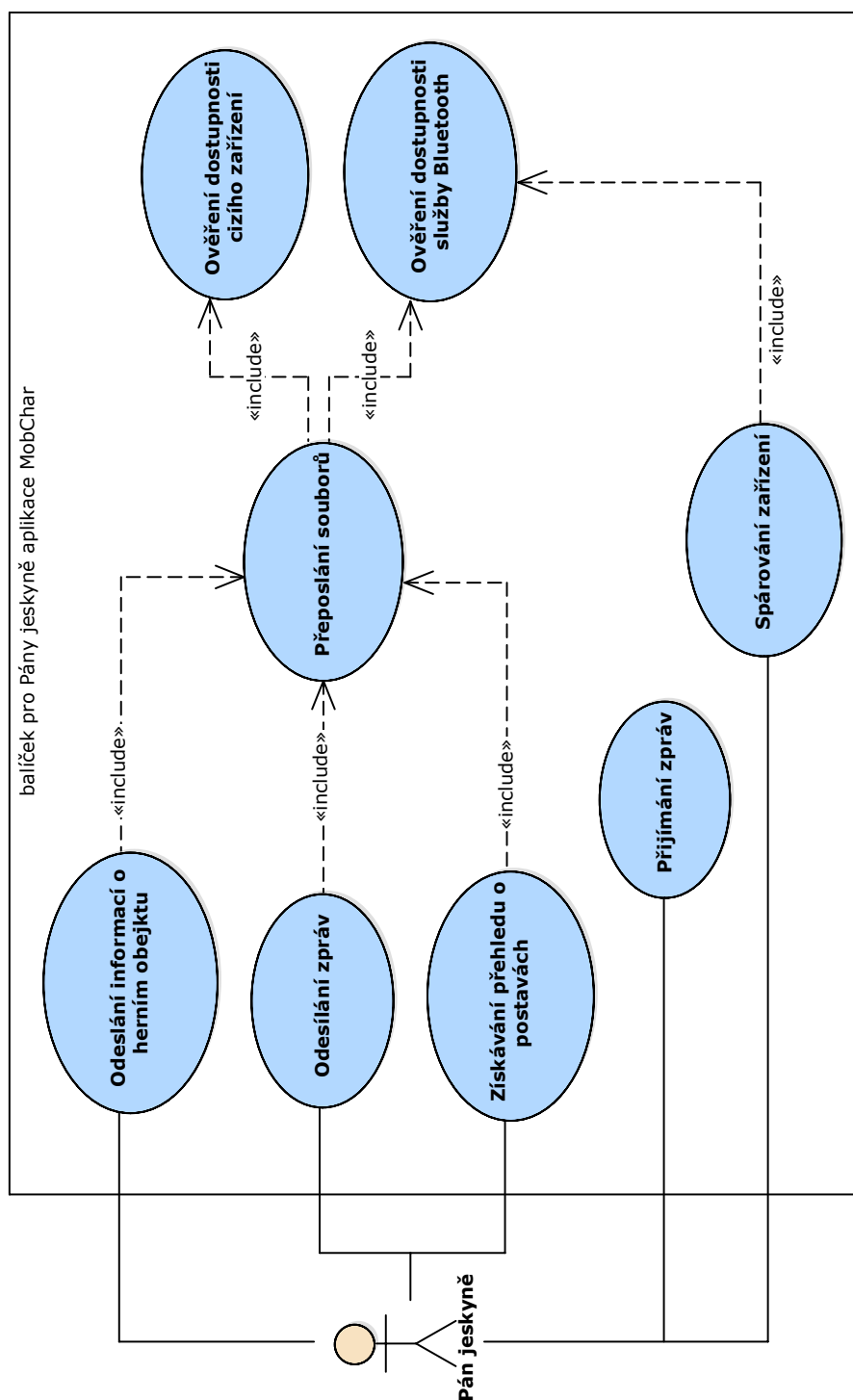
V rámci dodržování „zlatého pravidla“ (viz sekci 1.1.4) se může hráč rozhodnout, že se Pána jeskyně na něco zeptá tak, aby o tom zbtek hráčů nevěděl. Aplikace proto umožňuje jako doplnění procesu odesílání zpráv (viz sekci 1.6.4.7) zprávy od hráčů také přijímat. Chování aplikace je závislé na situaci, v níž se Pán jeskyně nachází, když zpráva od hráče přijde.

Aplikace zaznamená, že některý hráč odeslal Pánu jeskyně zprávu, a vyhodnotí, zdali se Pán jeskyně nachází na obrazovce se zprávami. Pokud se Pán jeskyně nachází na obrazovce se zprávami od daného hráče, zobrazí aplikace příchozí zprávu nejbliže k poli pro zadávání textu.

Pokud se Pán jeskyně nachází v přehledu družiny, aplikace zvýrazní položku s hráčovou postavou. Pán jeskyně se ke konverzaci s daným hráčem, a tudíž i k přijaté zprávě dostane podobně, jako ve scénáři „Odesílání zpráv“ (viz sekci 1.6.4.7).

V případě, že se Pán jeskyně nenachází ani na jednom z předchozích míst, zobrazí aplikace klasické systémové upozornění, oznamující Pánu jeskyně, že mu přišla nová zpráva.





Obrázek 1.12: Zobrazení závislostí mezi případy užití spojenými s komunikací s jiným zařízením

## 1. ANALÝZA

	Adresářová struktura ukládání	Příprava herních objektů	Příprava nestvůr a cizích postav	Správa map	Import a export dobrodružství	Předávání předmětů	Sdílení kouzel a schopností	Sdílení efektů	Získávání přehledu o postavách	Posílání zpráv
Případy ze sekce "Práce s adresářovou strukturou"	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Prohlížení herních objektů	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗
Vytvoření herního objektu	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗
Úprava herního objektu	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗
Smazání herního objektu	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗
Importování dobrodružství	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
Exportování dobrodružství	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
Spárování zařízení	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓
Odeslání informací o herním objektu	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗
Získání přehledu o postavě	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Odesílání zpráv	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Přijímání zpráv	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

Tabulka 1.1: Tabulka pokrytí požadavků případy užití

---

# Návrh

V této části práce je popsán návrh vytvářeného balíčku. V první části této kapitoly je rozebrán model architektury a způsob, jakým budou ukládána v navrhovaném balíčku data. Následně je detailně vysvětlen způsob přenášení dat pomocí datových objektů mezi jednotlivými vrstvami aplikace a jejich rozdělení do balíčků.

Součástí této kapitoly jsou také diagramy komunikace popisující klíčové operace v aplikaci a diagram nasazení popisující vztahy a závislosti mezi jednotlivými komponentami aplikace MobChar.

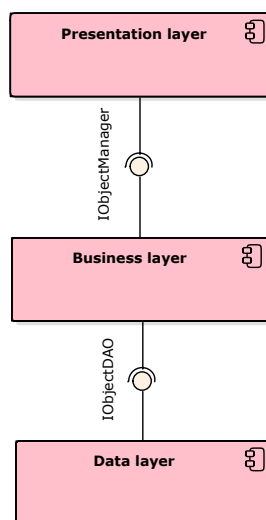
## 2.1 Architektura balíčku

Pro balíček Pána jeskyně byla zvolena třívrstvá architektura skládající se z prezentační, logické a datové vrstvy. Vzhledem ke skutečnosti, že data přijatá na prezentační vrstvě bude potřeba skoro vždy před uložením zpracovat, a tudíž nebude docházet k přímému ukládání do databáze, byla zvolena striktní varianta třívrstvé architektury, která na rozdíl od relaxované varianty neumožňuje přímou komunikaci prezentační a datové vrstvy.

Komunikace mezi vrstvami probíhá přes předem definovaná rozhraní. Pro přístup k jednotlivým rozhraním byl zvolen implementační vzor factory[4], který poskytuje specifickou implementaci požadovaného rozhraní.

Každému datovému objektu, který se v aplikaci využívá, je přiřazeno právě jedno rozhraní mezi datovou a logickou vrstvou a jedno mezi vrstvou logickou a prezentační. Způsob navržení rozhraní je znázorněn na diagramu 2.1.

**Prezentační vrstva** Prezentační vrstva zprostředkovává komunikaci uživatele s aplikací. Skládá se především z objektů spojených s uživatelským rozhraním, kterými jsou aktivity, fragmenty, podpůrné adaptéry a objekty pro zobrazování položek seznamů. V prezentační vrstvě by se mělo vyskytovat pouze minimum práce s daty a logických operací.



Obrázek 2.1: Způsob navržení rozhraní v balíčku

**Logická vrstva** Logická vrstva představuje spojení mezi prezentační a datovou vrstvou. Vzhledem k tomu, že by ani jedna z těchto dvou vrstev neměla provádět žádné logické operace s objekty, jsou tyto operace uskutečněny zde. Jedná se především o řazení seznamů a přiřazování kolekcí datovým objektům. Součástí logické vrstvy jsou také veškeré jmenné typy.

**Datová vrstva** Úkolem datové vrstvy je zajistit persistenci dat. Tato vrstva ukládá datové struktury do databáze a je naopak schopná z databáze tyto struktury získávat a poskytovat je vrstvě logické.

## 2.2 Způsob uložení dat

Nativním nástrojem pro ukládání dat v aplikacích operačního systému Android je databáze typu SQLite, která byla použita i pro balíček pro Pána jeskyně. Většina tabulek databáze vytvářeného balíčku je založena na podobném principu, a proto zde bude popsán jenom tento princip a případné výjimky z něj. Kompletní relační datový model je k nalezení na příloženém CD.

### 2.2.1 Princip databázových tabulek

Všechny tabulky v databázi balíčku pro Pána jeskyně mají sloupec *ID*, který slouží jako primární klíč. Sloupce v tabulkách jsou pojmenovány podtržítkovou

notací. Každá tabulka je zapsána do centrální třídy zabývající se správou databáze, která se stará o jejich vytváření a případnou úpravu.

### 2.2.2 Cizí klíče

Aplikace obsahuje několik objektů, které mohou záviset na různých typech jiných objektů. V takovém případě mají tabulky sloupec pro každý typ, na němž mohou záviset, který slouží jako cizí klíč do tabulky tohoto objektu. Z těchto sloupců vždy právě jeden obsahuje informaci o identifikátoru nadřazeného objektu, sloupce pro zbylé typy nadřazených objektů obsahují *null* hodnotu.

### 2.2.3 Předměty

Aplikace obsahuje jedinou ISA hierarchii – předměty. Tato ISA hierarchie je implementována pomocí tabulky se společnými vlastnostmi předmětů, která je rozšířena o sloupec *type* určující typ předmětu na základě odpovídajícího jmenného typu. Jednotlivé typy předmětů pak mají každý svou tabulku, jejíž primární klíč je zároveň cizím klíčem do tabulky se společnými vlastnostmi předmětů.

## 2.3 Návrhový model tříd

V této sekci je popsáno, jaké podoby a využití mají datové objekty v jednotlivých vrstvách. Rozdělení datových objektů v rámci jednotlivých vrstev je zachyceno na diagramu 2.2. Kompletní model celého balíčku obsahující podrobnější dělení je k nalezení na přiloženém CD. Součástí tohoto modelu ovšem nejsou primitivní metody (getter, setter a konstruktory).

### 2.3.1 Základní datové objekty

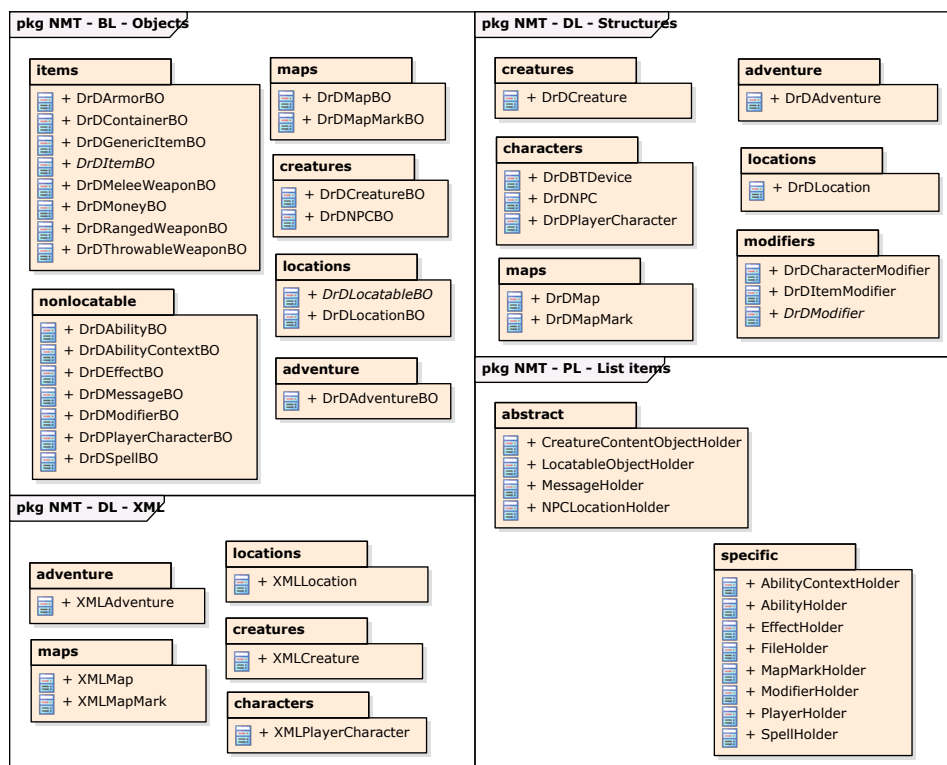
Tyto datové objekty se používají především na datové vrstvě k uchování atributů, které jsou uloženy v databázové tabulce daného typu objektu. Těmito atributy jsou především primitivní datové typy a jmenné typy, které jsou v databázi uloženy pomocí svých identifikátorů a na konkrétní typ se při vytváření na základě těchto identifikátorů převádějí.

Základní datové objekty obsahují pouze primitivní metody (getter, setter a konstruktory).

### 2.3.2 Logické objekty

Logické objekty jsou základní strukturou pro předávání dat na logické vrstvě. Skládají se ze základního datového objektu a kolekcí logických objektů, které tomuto typu objektu náleží (například efekt bude mít kolekci modifikátorů). Dále mohou obsahovat další atributy, které se využívají pouze dočasně při běhu

## 2. NÁVRH



Obrázek 2.2: Rozdělení datových objektů do balíčků na jednotlivých vrstvách

aplikace, ale do databáze se neukládají, a nejsou proto součástí základních datových objektů.

Tento typ datových objektů je vytvářen na základě základního datového typu a umožňuje převod na objekt pro konverzi do XML či položku seznamu.

### 2.3.3 Objekty pro konverzi do XML

Speciálním typem jsou objekty pro konverzi do XML. Tyto objekty mají velmi podobnou hierarchii jako logické objekty, každý atribut je však doplněn o způsob, jakým má být zapsán do souboru formátu XML. Jedná se především o jméno jeho štítku, typ dat, která obsahuje, a informaci, zdali se v dané XML struktuře musí vyskytovat, či nikoliv.

Tento objekt umožňuje oboustranný převod pouze se základním datovým typem. Všechny kolekce náležících objektů musí být vytvořeny mimo XML objekt a tomuto objektu následně přiřazeny. V balíčku pro Pána jeskyně tento proces zajišťují logické objekty ve svých metodách pro převod na XML objekty.

### 2.3.4 Položky seznamů

Na prezentační vrstvě je potřeba zobrazovat datové objekty dvěma způsoby.

Prvním je zobrazení všech základních detailů o daném datovém objektu. Toho se dosáhne vyžádáním základního datového objektu z adekvátního logického objektu a následným dotazováním na jeho jednotlivé atributy.

Druhým způsobem je zobrazování seznamu datových objektů (například efekty na zvoleném předmětu). V takovém případě velmi často z těchto základních atributů objektu postačí jedna či dvě identifikující informace (zpravidla identifikátor a název). Využívání logických objektů pro tyto účely by bylo především pro obsáhlé seznamy neefektivní a nepraktické.

Z tohoto důvodu byly vytvořeny speciální objekty pro udržování informací o těchto položkách seznamů. Kromě základních identifikačních informací velmi často také nesou typ zobrazovaného předmětu, na základě něhož adaptéry dané položce přiřadí odpovídající ikonu.

## 2.4 Realizace případů užití

Tato sekce obsahuje diagramy komunikace pro nastínění průběhu klíčových procesů vytvářeného balíčku. V této práci bude podrobně rozebrán způsob, jakým jádro aplikace spouští balíček pro Pána jeskyně, a procesy spojené s posíláním dat přes službu Bluetooth. Diagramy v této sekci jsou pro přehlednost rozděleny na více částí (kompletní diagram by kvůli přeškalování nebyl čitelný).

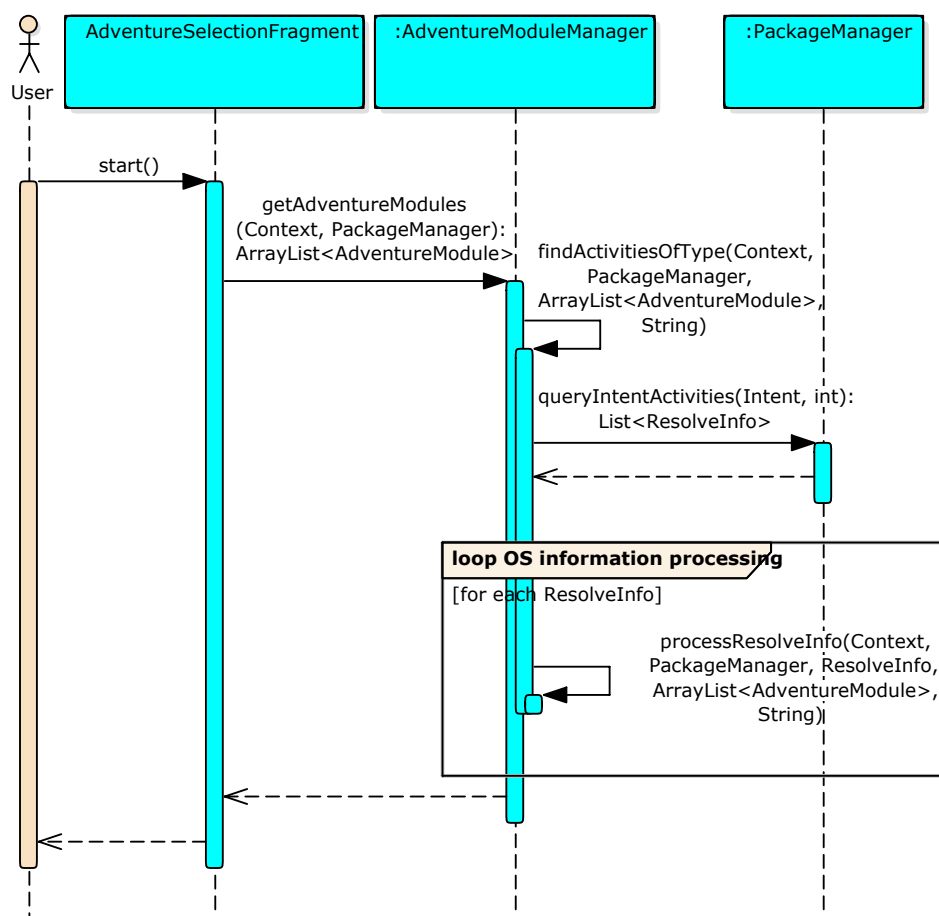
### 2.4.1 Spouštění balíčku pro Pána jeskyně jádrem aplikace

Pro integraci vytvářeného balíčku do systému aplikace MobChar musí být zajištěno, aby jádro aplikace bylo schopné skrze operační systém balíček vyhledat a zobrazit seznam dobrodružství, která byla v rámci tohoto balíčku vytvořena.

Jednotlivé kroky procesu, jakým se jádro dotazuje operačního systému na dostupné balíčky, jsou zachyceny na diagramu 2.3, na který navazuje diagram 2.4 popisující způsob získávání seznamu dobrodružství z nalezených balíčků a následného spuštění odpovídajícího balíčku se zvoleným dobrodružstvím. Nyní následuje slovní popis obou diagramů.

Proces získávání dostupných balíčků v rámci zařízení začíná, jakmile uživatel zvolí záložku *Dobrodružství* z horního horizontálního menu na základní obrazovce jádra aplikace, čímž si vyžádá instanci `AdventureSelectionFragment`. Tento objekt v rámci své inicializace volá metodu `getAdventureModules` třídy `AdventureModuleManager` s cílem získat kolekci dostupných balíčků.

Tato třída v metodě `findActivitiesOfType` připraví datové struktury pro prohledání operačního systému za účelem nalezení aktivit v rámci nain-



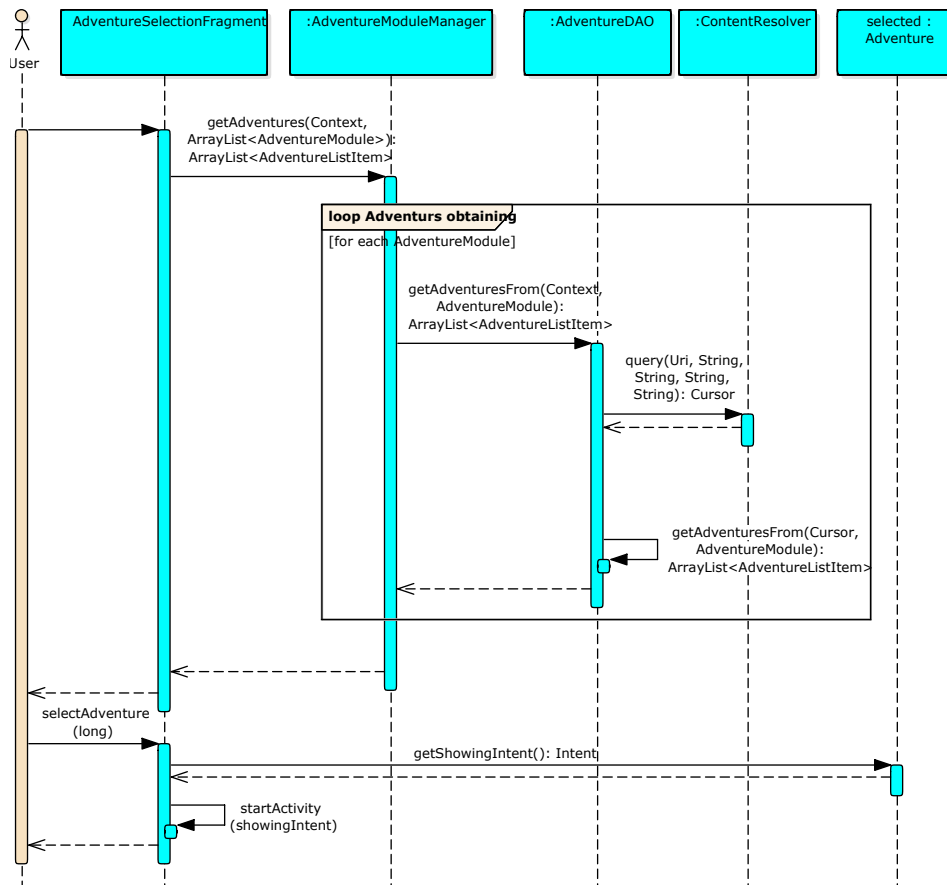
Obrázek 2.3: Sekvenční diagram získávání dostupných balíčků pro tvorbu dobrodružství

stalovaných aplikací (potažmo balíčků) a zavoláním metody `queryIntentActivities` objektu operačního systému `PackageManager` prohledání zahájí.

Operační systém následně prohledá aplikační manifesty všech nainstalovaných aplikací. V každém manifestu otestuje u každé definované aktivity, zda má zaregistrovaný speciální atribut, který je definován pro balíčky pro Pána jeskyně. Pokud aktivita má tento atribut definovaný, operační systém přidá její informace ve formě struktury `ResolveInfo` do své kolekce, kterou po skončení prohledání vrátí.

Po obdržení kolekce s informacemi o nalezených balíčcích `AdventureModuleManager` převede tyto informace ve své metodě `processResolveInfo` na objekt typu `AdventureModule`, jejichž kolekci po převedení všech nalezených aktivit vrátí instanci `AdventureSelectionFragment`.





Obrázek 2.4: Sekvenční diagram získávání dobrodružství a jejich spouštění

Jakmile má `AdventureSelectionFragment` kolekci dostupných balíčků, vyžádá si od `AdventureModuleManager` kolekci obsahující všechna dobrodružství ze všech dostupných balíčků voláním metody `getAdventures`, které předá jako jeden z parametrů kolekci dostupných balíčků.

`AdventureModuleManager` projde kolekci dostupných balíčků a pro každý balíček zavolá metodu `getAdventuresFrom(Context)` třídy `AdventureDAO`. Tato metoda zavolá systémovou třídu `ContentResolver`, která na základě unikátní adresy obsažené ve struktuře `AdventureModule` vytvoří objekt obsahující záznamy z databáze o dobrodružstvích daného balíčku.

Na základě tohoto objektu `AdventureDAO` v metodě `getAdventuresFrom(Cursor)` vytvoří seznam dobrodružství, který vrátí instanci `AdventureSelectionFragment`, jež je zobrazí uživateli.

Uživatel si následně vybere dobrodružství, které chce zobrazit. `AdventureSelectionFragment` zareaguje vyhledáním příslušného dobrodružství v se-

znamu dobrodružství a získáním datové struktury, jež je nutná pro spuštění balíčku, zavoláním metody `getShowingIntent` na zvoleném dobrodružství.

Získanou strukturu použije jako parametr funkce `startActivity`, která spustí odpovídající balíček se zvoleným dobrodružstvím.

### 2.4.2 Posílání herních objektů přes službu Bluetooth

Jednou z klíčových funkcí vytvářeného balíčku je možnost posílat herní objekty jiným zařízením pomocí služby Bluetooth. Vzhledem k požadavku služby Bluetooth na serializaci odesílaných dat bylo rozhodnuto, že herní objekty budou převedeny na XML objekty, které mají nadefinovaný způsob serializace.

Proces odeslání herního objektu se skládá ze tří kroků. Nejprve je potřeba získat kompletní herní objekt (ve formě naplněného logického objektu), ten následně převést na XML objekt schopný serializace, který bude nakonec serializován a předán službě Bluetooth k odeslání.

Způsob, jakým je v aplikaci vytvářen kompletní herní objekt, není pro posílání těchto objektů podstatný, a nebude proto v této práci rozebrán podrobně. Pokud by tento způsob čtenáře zajímal, může diagram zachycující jeho průběh nalézt na příloženém CD.

Nyní bude podrobně rozebrán způsob, jakým jsou vytvářeny XML objekty z herních objektů (zachycen na diagramu 2.5), a proces odesílání těchto XML objektů pomocí služby Bluetooth (diagram 2.6).

Po úspěšném vytvoření kompletního logického objektu, který má být odeslán (na diagramu znázorněn jako `ObjectB0`), zavolá proces, který má za úkol odeslání tohoto objektu, metodu `getXMLObject` tohoto logického objektu.

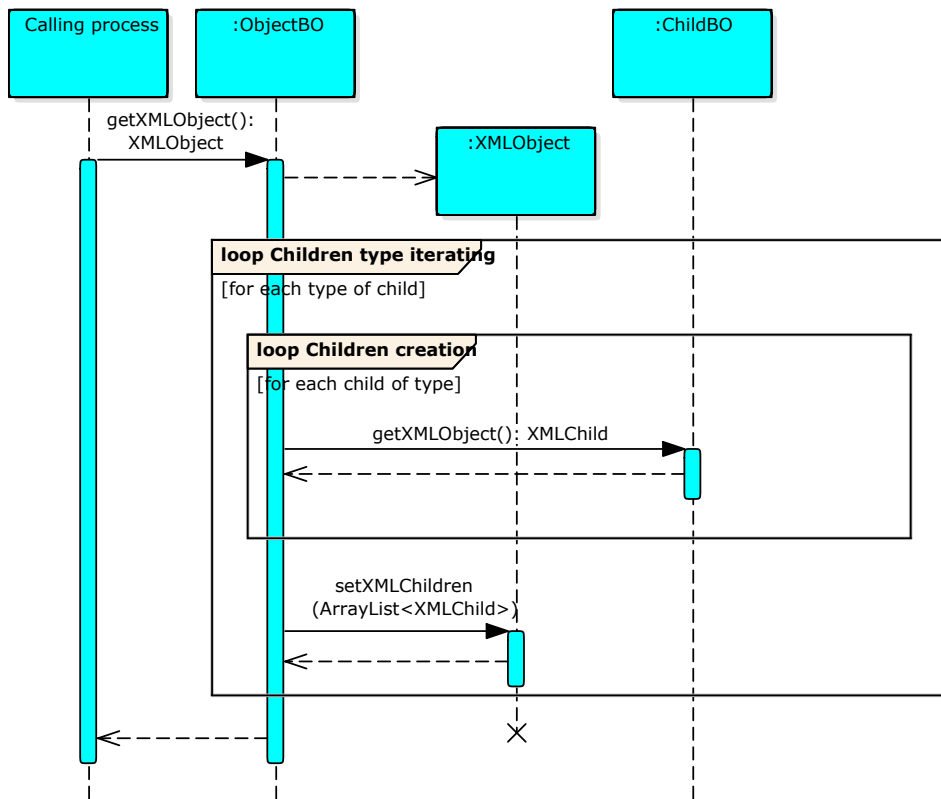
Logický objekt nejprve vytvoří adekvátní základní XML objekt pomocí svého základního datového objektu (tj. pokud by logickým objektem byl `DrDEffectB0`, pak by vytvářel `XMLEffect` pomocí `DrDEffect`).

Následně logický objekt projde všechny typy kolekcí svých potomků a pro každý typ kolekce z každého jejího prvku vytvoří XML objekt. Takto vytvořené kolekce XML objektů nakonec přiřadí dříve vytvořenému XML objektu. Vzhledem k tomu, že i potomci mohou mít další potomky, je celý tento proces volán rekurzivně, dokud potomci mají další potomky.

Vzhledem k velké úrovni abstrakce bude tento princip vysvětlen na následujícím příkladu. Pokud by odesílaným objektem byl `DrDEffectB0`, pak by se nejprve vytvořil `XMLEffect`. Následně by `DrDEffectB0` měl projít všechny typy kolekcí svých potomků, což je v tomto případě pouze kolekce modifikátorů – `DrDModifierB0`.

Pro každý modifikátor by vytvořil adekvátní `XMLModifier`. Ten již žádné potomky nemá, a rekurze by tudíž skončila. Vzniklá kolekce XML modifikátorů by byla přiřazena původnímu `XMLEffect` objektu a proces by skončil.

Jakmile volající proces obdrží připravený XML objekt, může započít třetí fáze odesílání herního objektu. Proces použije metodu `sendXMLObject` třídy `BluetoothManager`. Tato třída nejprve získá pomocí metody `getCommuni-`

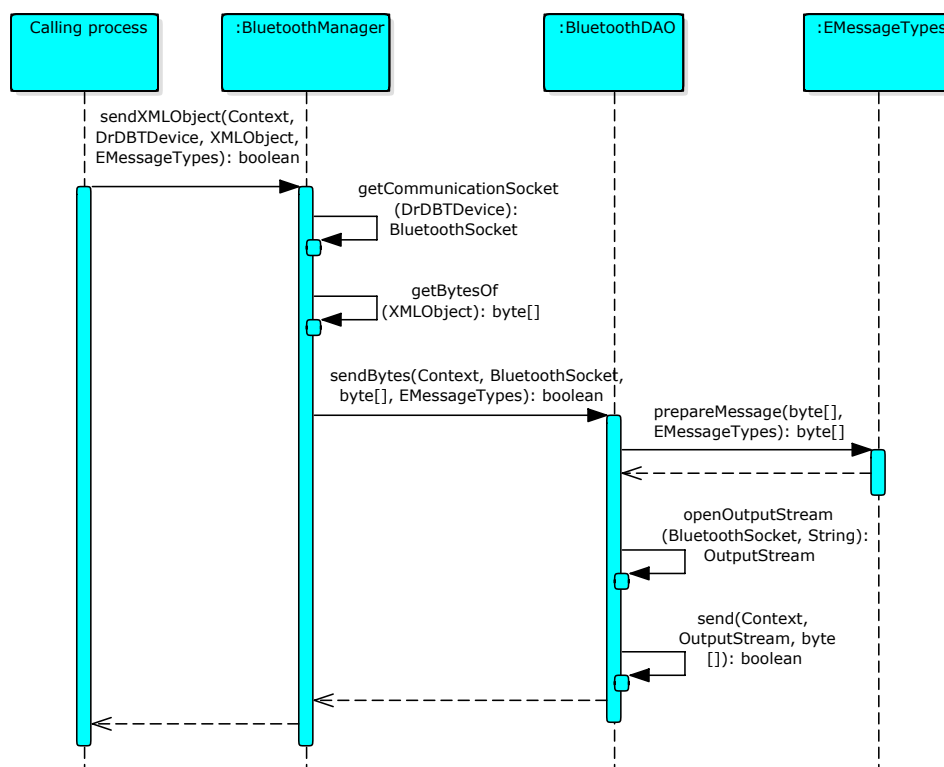


Obrázek 2.5: Sekvenční diagram vytváření objektu pro konverzi do formátu XML

ationSocket socket pro komunikaci se zařízením, kterému má být herní objekt přeposlán. Následně serializuje XML objekt, který má být odeslán, metodou `getBytesOf`.

Po dokončení příprav deleguje odesílání na třídu `BluetoothDAO` pomocí metody `sendBytes`. Třída `BluetoothDAO` nejprve připraví data, která mají být odeslána zavoláním statické metody `prepareMessage` na objektu `EMessageTypes`.

Následně použije komunikační socket pro otevření kanálu pro přenos dat svou metodou `openOutputStream` a pomocí tohoto kanálu data odešle svou metodou `send`. Po dokončení běhu metody `send` vrátí její výsledek zpět procesu, který tuto operaci zahájil.



Obrázek 2.6: Sekvenční diagram odesílání XML objektu přes službu Bluetooth

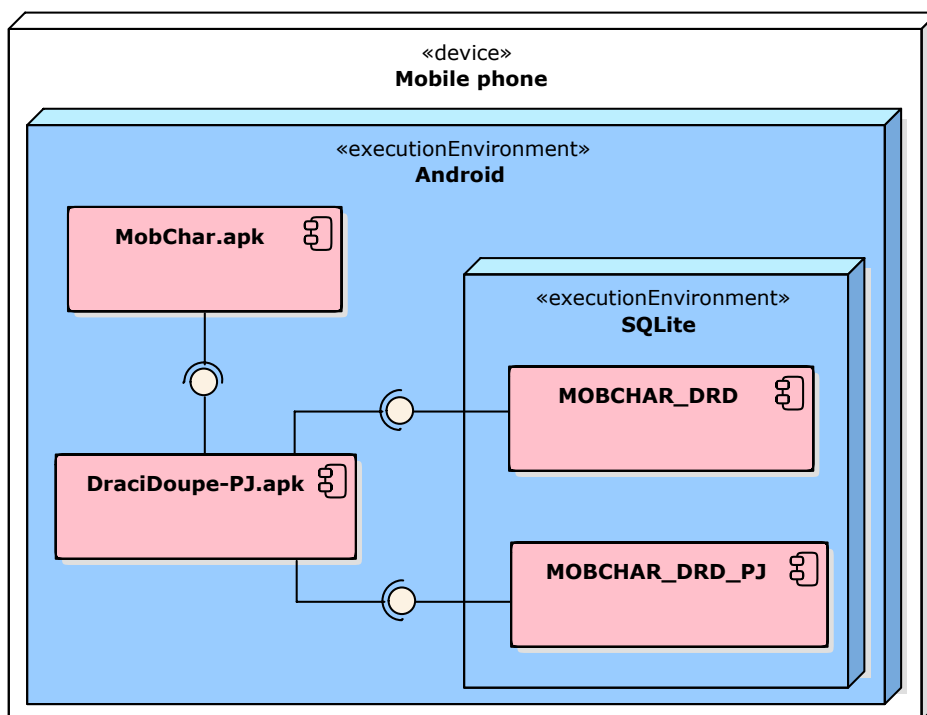
## 2.5 Nasazení a závislosti

V této sekci je popsán způsob, jakým je balíček pro Pána jeskyně začleněn do systému aplikace MobChar a jaké komponenty systému využívá. Vazby mezi jednotlivými komponentami jsou znázorněny na diagramu 2.7.

Aplikace MobChar je určena pro mobilní telefony s operačním systémem Android (verze je specifikována v nefunkčních požadavcích 1.5.2). Jádro aplikace slouží pouze ke správě jednotlivých balíčků, a nepotřebuje proto žádná speciální oprávnění ani přístup k databázi.

Samotný balíček pro Pána jeskyně je funkčně závislý na jádru aplikace MobChar. Balíček je tedy možné nainstalovat bez ohledu na to, zdali je na zařízení jádro přítomno, avšak přístup k balíčku je možný pouze přes jádro aplikace MobChar. Balíček pro Pána jeskyně vyžaduje oprávnění pro přístup k externímu úložišti a službě Bluetooth (viz nefunkční požadavky 1.5.2).

Balíček pro Pána jeskyně využívá podobně jako ostatní balíčky databáze typu SQLite, neboť jsou nativně podporovány systémem Android. Tyto databáze jsou v balíčku využity dvě – jedna pro postavy a herní objekty s nimi



Obrázek 2.7: Diagram nasazení balíčku a jeho závislostí

spojené, která je součástí knihovny pro hru Dračí doupe, a druhá pro dobrodružství a jejich související objekty vyjma postav.

Tímto bohužel dochází k ukládání objektů, které mohou souviset s postavami i dobrodružstvími (předměty, schopnosti, kouzla a efekty), do různých databází na základě toho, zdali jsou součástí postavy či dobrodružství. Tento problém by se dal vyřešit buď rozšířením knihovny databáze o dodatečné tabulky a sloupce specifické pro balíček pro Pána jeskyně nebo využitím pouze databáze balíčku pro Pána jeskyně (na diagramu MOBCHAR\_DRD\_PJ).

První řešení bylo označeno za nevhodné, neboť databáze používaná knihovnou (na diagramu MOBCHAR\_DRD) by měla obsahovat pouze objekty a vazby společné pro oba balíčky zabývající se hrou Dračí doupe (hráčský a pro Pána jeskyně). Z tohoto důvodu bylo rozhodnuto, že knihovny databáze nebude rozšířena o vazby specifické pouze pro balíček pro Pána jeskyně.

Naproti tomu nevyužitím knihovny databáze by ztratilo využití knihovny smysl a došlo by k narušení konzistence dat, která jsou pro oba balíčky společná (přeci jenom, pokud dojde ke změně postavy, bude se měnit stejně z pohledu hráčů i Pána jeskyně). Kvůli těmto skutečnostem byl tento návrh pro použití dvou databází určen jako nejlepší možný v současné situaci.



---

# Implementace

Tato kapitola je zaměřena na popis implementačních podrobností, které přímo nesouvisí s návrhem vytvářeného balíčku.

Nejprve jsou zde zmíněny nástroje a externí knihovny, které byly použity při vytváření balíčku. V další části je rozebrán způsob, jakým jsou zdrojové kódy jednotlivých tříd rozděleny do balíčků. Na závěr této kapitoly jsou uvedeny ukázky implementace některých netriviálních mechanismů vytvářeného balíčku.

## 3.1 Použité nástroje

V této sekci jsou vypsány nástroje, které byly použity za účelem vytvoření této bakalářské práce.

**Enterprise Architect** Tento program slouží k vytváření diagramů popisujících vyvíjenou aplikaci a následné generování komplexní analytické a návrhové dokumentace. Všechny diagramy v této práci byly vytvořeny v jazyce UML pomocí tohoto nástroje. Pro vytváření diagramů byl použit Enterprise Architect ve verzi 12.0.1215.

**Android Studio** Vývojové prostředí Android Studio bylo využito k vytváření zdrojových kódů vyvíjeného balíčku. Dále umožňuje napojení na verzovací systém, spouštění jednotkových a integračních testů, přístup k výpisu záznamů při běhu aplikace a přímé nasazení testovací verze na fyzické zařízení. V průběhu vývoje balíčku docházelo k průběžné aktualizaci programu Android Studio, poslední použitou verzí byla verze 2.3.1.

**Verzovací nástroje** K udržení konzistence byly pro verzování souborů použity konvence aplikace MobChar. Zdrojové kódy byly verzovány pomocí systému Git ve verzi 2.8.0 skrze Android Studio, vytvořené diagramy byly ukládány pomocí SVN verze 1.9.3.

**TeXMaker** Pro sazbu textu bakalářské práce bylo použito prostředí TeX-Maker ve verzi 4.5.

## 3.2 Použité knihovny

Tato sekce je věnována netypickým knihovnám, které jsou ve vytvářeném balíčku použity. U každé knihovny je vysvětleno, z jakého důvodu byla zahrnuta do projektu.

### 3.2.1 Apache Commons Lang

Při vytváření jednotkových testů vznikla potřeba přetížít základní metody `equals(Object)` a `hashCode()` u všech datových a logických objektů. Vzhledem k tomu, že se především pro objekty obsahující kolekce objektů jedná o netriviální úkol, byla využita část *Lang* knihovny Apache Commons [5].

Tato část knihovny poskytuje objekty `EqualsBuilder` a `HashCodeBuilder`, které pomocí svých metod `append` přetěžování těchto metod výrazně usnadňují a zpřehledňují.

### 3.2.2 Simple XML Parser

Jedním z požadavků na vytvářený balíček je možnost exportovat a importovat dobrodružství. V rámci hráčského balíčku se pro podobný proces využívá Simple XML Parser [6], který vyžaduje pouze využití specifických anotací pro všechny proměnné objektu, který má být serializován. Z důvodu zachování konzistence a možnosti využití knihovny byl tento překladač začleněn do vytvářeného balíčku.

Kvůli schopnosti serializace byl Simple XML Parser [6] využit i pro přípravu herních objektů pro přenos pomocí služby Bluetooth.

### 3.2.3 Touch Image View

Při vývoji balíčku vznikl požadavek na možnost přiblížení prohlížené mapy a přesouvání tohoto přiblíženého pohledu po základním obrázku. Tím ovšem přestal stačit nativní nástroj operačního systému Android pro prohlížení obrázků, `ImageView`.

Po přezkoumání dostupných alternativ byl zvolen projekt Mika Ortize, `TouchImageView` [7], který potřebnou funkcionalitu poskytuje a doplňuje o plynulé animace.

Vzhledem k tomu, že tento projekt obsahuje několik ukázek použití objektu (v podobě spustitelných aplikací), byl do vytvářeného balíčku začleněn pouze samotný objekt.



### 3.3 Rozdělení na adresáře

Balíček je rozčleněn na dvě základní kategorie – zdrojové kódy tříd (soubory s koncovkou `.java`) a soubory obsahující rozvržení grafických prvků aplikace (soubory s koncovkou `.xml`).

Zdrojové kódy jednotlivých tříd jsou následně rozděleny do adresářů podle toho, ve které ze tří základních vrstev se nacházejí.

Datová a logická vrstva se dále dělí na adresář s rozhraním pro komunikaci s danou vrstvou, adresář se třídami, které tato rozhraní implementují, adresář s datovými objekty náležícími dané vrstvě a adresář se zbylými třídami.

Soubory v prezentační vrstvě jsou do adresářů rozděleny podle funkcionality, kterou uživatelům zpřístupňují. V jednom adresáři budou například všechny třídy související se zobrazováním kouzel, v dalším třídy volané z jádra aplikace `MobChar`, atd.

### 3.4 Ukázky implementace

V rámci této sekce budou předvedeny ukázky implementace zabývající se vytváření notifikací při přijetí zprávy od hráče a způsobem, jakým je ve vytvořeném balíčku dosaženo oznamování dokončení práce zahájené na pracovních vláknech.

#### 3.4.1 Oznamování dokončení práce pracovního vlákna

Vyvíjený balíček umožňuje Pánu jiskyně navázat spojení pomocí služby Bluetooth se zařízeními v blízkém okolí, která mají zapnutou viditelnost v rámci hráčského balíčku pro hru `Dračí doupě`. Pomocí tohoto spojení může Pán jiskyně posílat jednotlivým hráčům herní objekty, zprávy a požadavky o zaslání informací o postavě. Přes navázané spojení mohou také hráči zasílat Pánu jiskyně zprávy.

V rámci zvýšení uživatelské přívětivosti není spojení mezi zařízeními ukončováno okamžitě po přijetí dat, ale přetrvává, aby mohly být zprávy posílány bez neustálého navazování nových spojení. Jednotlivá spojení jsou proto obsluhována pracovními vlákny, která vyčkávají na přijetí dat, jež po přijetí zpracují.

V okamžiku dokončení vzniká potřeba informovat uživatele o skutečnosti, že je daná práce dokončena, a výsledku, který měla. Zde ovšem vzniká problém, neboť pro zobrazení této informace potřebuje pracovní vlákno referenci na aktuálně prohlíženou aktivitu. Od zahájení činnosti pracovního vlákna se však může uživatel přesunout na jinou obrazovku, čímž pracovní vlákno ztratí o současné aktivitě informaci. To vede k neschopnosti zobrazit oznámení o dokončení práce.

Z tohoto důvodu byl vytvořen singleton `UIStash`, jež udržuje referenci na aktuálně prohlíženou aktivitu, aktuální fragment, který je v ní umístěn

### 3. IMPLEMENTACE

---

a kontext aplikace (pro přístup k ikonám a lokalizovaným textům). Každá aktivita v rámci přetížené první metody svého životního cyklu (`onCreate`) upraví hodnotu aktuální aktivity v rámci `UIStash` na sebe, čímž se tato reference udržuje neustále aktuální.

Nyní již mohou pracovní vlákna přistupovat k aktuální aktivitě a skrze ni zobrazovat vyskakovací okno se zprávou uživateli. V některých případech však dochází ke změně údajů, které jsou v aktuální aktivitě (potažmo jejím fragmentu) zobrazeny. V takových případech by bylo vhodné, aby se daná aktivita společně se svým aktuálním fragmentem aktualizovala a zobrazila nový stav svého obsahu.

Z tohoto důvodu byly základní typy aktivit (`DrDSingleFragmentActivity` a `DrDSLidingTabsActivity`) rozšířeny o rozhraní `IAReloadable` (viz ukázkou kódu 1). Toto rozhraní umožňuje pracovním vláknům na aktuální aktivitě zavolat metodu `reload()`.

```
public interface IAReloadable {  
  
    /**  
     * This method will reload activity's content.  
     */  
    void reload();  
  
    /**  
     * This method will in some way show message passed as argument.  
     *  
     * @param message Message, which should be shown.  
     */  
    void showMessage(String message);  
}
```

Ukázka kódu 1: Rozhraní pro aktualizaci aktivit

Aktivity většinou mají implementovanou metodu `showMessage(String)` tak, aby došlo k vytvoření dočasného textu se zprávou, která je této metodě předána jako argument. Vzhledem k principu *AUF* vymezeném v rámci příručky programování pro Android [8] má každá aktivita svůj fragment a veškeré prvky obrazovky jsou umístěny v něm.

Proto bylo vytvořeno rozhraní `IFreloadable` (viz ukázkou kódu 2), které implementuje každý fragment použitý ve vytvářeném balíčku. Chování této metody povětšinou spočívá v opětovném načtení obsahu volaného fragmentu, jsou ovšem fragmenty, které toto volání záměrně ignorují.

Výsledné volání daných metod na pracovních vláknech pak vypadá jako na ukázce kódu 3 ze třídy `DataTransferThread`.

```
public interface IFReloadable {

    /**
     * This method will reload fragment's content.
     */
    void reload();
}
```

Ukázka kódu 2: Rozhraní pro aktualizaci fragmentů

```
{
    ...
    UIStash stash = UIStash.getInstance();
    stash.getCurrentReloadable().reload();
    ...
    stash.getCurrentReloadable().showMessage(message);
    ...
}
```

Ukázka kódu 3: Volání metod pro aktualizaci aktuální aktivity

### 3.4.2 Notifikace

Předchozí část této sekce pokrývá většinu případů přijímání dat z hráčského zařízení. Jedinou výjimkou jsou zprávy, neboť u nich pouhá aktualizace současné aktivity (potažmo fragmentu) nestačí. Přijetí zprávy musí být viditelné na všech místech aplikace i mimo ni. Navíc se nedá použít řešení v podobě dočasného textu oznamujícího tuto skutečnost, neboť zpráv může přijít naráz více a tyto texty by se pak překrývaly.

Elegantním řešením tohoto problému je využití systémových notifikací. Notifikace se zobrazí ve speciálním systémovém menu, kde navíc po uplynutí doby jejich zobrazení zůstanou. Tím pokryjí všechny důvody, proč se nedal využít dočasný text.

Vytvoření a odeslání notifikace systému zajišťuje třída `Notification-Launcher` pomocí své statické metody `launchNewMessageBroadcast(long, String)`, jejíž klíčové části jsou zachyceny na ukázce kódu 4.

Jedinou nevýhodou tohoto přístupu je, že v tomto stavu by se notifikace zobrazovaly vždy, tzn. i v momentě, pokud by měl Pán jeskyně zobrazenou konverzaci s daným hráčem a aktivně s ním komunikoval. V tomto případě by bylo lepší, pokud by místo notifikace došlo k zobrazení nově příchozí zprávy na spodním okraji historie konverzace.

Podobně by bylo lepší v záložce s přehledem hráčů, *Družina*, namísto notifikace zobrazit u hráče, od něhož zpráva přišla, pouze ikonu, která by Pána jeskyně na tuto skutečnost upozornila. Tohoto chování lze docílit vytvořením

### 3. IMPLEMENTACE

---

```
public static void launchNewMessageBroadcast(long playerCharacterID,
                                             String messageText) {

    ...

    Notification notification = new NotificationCompat.Builder(context)
        .setTicker(notificationTitle)
        .setSmallIcon(R.drawable.ic_envelope)
        .setContentTitle(notificationTitle)
        .setContentText(notificationText)
        .setAutoCancel(true)
        .build();

    Intent intent = new Intent(NEW_MESSAGE);
    intent.putExtra(EXTRA_NOTIFICATION, notification);
    intent.putExtra(EXTRA_PLAYER_ID, playerCharacterID);

    context.sendOrderedBroadcast(intent, PERM_PRIVATE, null,
                                null, Activity.RESULT_OK, null, null);
}
```

Ukázka kódu 4: Vytváření a odesílání systémové notifikace

tzv. „příjímače“ (angl. receiver) a „zachytitele“ (angl. interceptor).

Úkolem „příjímače“ je zachytit systémové vysílání (angl. broadcast) a podle jeho obsahu se zachovat. V základním případě se jedná o zobrazení notifikace, která je předávána společně se systémovým vysíláním.

„Zachytitel“ pracuje velmi podobně jako „příjímač“. Nejprve zachytí systémové vysílání a vykoná příslušnou akci, avšak následně na rozdíl od „příjímače“ systémové vysílání pozmění tak, aby každý další příjímač vyhodnotil toto vysílání jako neplatné a nereagoval na něj.

Operační systém umožňuje přiřadit jednotlivým „příjímačům“ a „zachytitelům“ prioritu, s níž budou systémová vysílání přijímat. Korektního chování se tedy dá docílit, pokud se „příjímačům“ nastaví nižší priorita než „zachytitelům“. Tím dojde k jevu, kdy nejprve vysílání obdrží „zachytitel“, který vykoná svou akci, pozmění vysílání a pošle ho dál. Následně bude vysílání zachyceno „příjímačem“, který ho ovšem vyhodnotí jako neplatné a neprovede žádnou akci.

Pro tento případ využití vysílání se „příjímač“ přiřadí všem aktivitám v balíčku (přiřazením jedné ze základních aktivit, `DrDSingleFragmentActivity` nebo `DrDSlidingTabsActivity`). Následně se aktivitám, jež nemají zobrazit notifikaci, přidá „zachytitel“, jehož chování se implementuje tak, aby odpovídalo specifikaci.

Pro lepší představu je v této práci uveřejněna ukázka 5 zachycující třídu `NewMessageReceiver`, která slouží jako „příjímač“, a ukázka 6 popisující abstraktní definici třídy `NewMessageInterceptor`, jež slouží jako předpis pro konkrétní „zachytitele“.

```
public class NewMessageReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        if (getResultCode() != Activity.RESULT_OK) {
            return;
        }

        int requestCode = 0;
        Notification notification = intent.getParcelableExtra(
            NotificationLauncher.EXTRA_NOTIFICATION);

        NotificationManagerCompat manager =
            NotificationManagerCompat.from(context);
        manager.notify(requestCode, notification);
    }
}
```

Ukázka kódu 5: Příjímač pro přenosy obsahující notifikace

```
public abstract class NewMessageInterceptor extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        setResultCode(Activity.RESULT_CANCELED);
    }

    public static IntentFilter prepareFilter() {
        return new IntentFilter(NotificationLauncher.NEW_MESSAGE);
    }
}
```

Ukázka kódu 6: Abstraktní předpis pro zachytitele přenosů obsahujících notifikace

Celý koncept je převzatý z příručky programování pro Android [8], kde je podrobněji rozepsán pro obecné použití.



---

# Testování

Předmětem této kapitoly je testování vyvinutého balíčku. Pro účely testování byly vytvořeny jednotkové a uživatelské testy, které dohromady svým rozsahem pokrývají všechny případy užití (a tím i požadavky na vytvářený balíček). U obou skupin testů je popsán jejich účel, průběh a výsledky, které testování přineslo.

## 4.1 Jednotkové testy

Tato skupina testů se zaměřuje na funkčnost metod jedné konkrétní třídy. Jednotkové testy zpravidla nepotřebují přístup k databázi, znát aplikační kontext ani jiné proměnné spojené s mobilním zařízením, a mohou být tedy spouštěny přímo na virtuálním stroji Javy (zkráceně JVM).

### 4.1.1 Účel

V rámci vyvíjeného balíčku byly vytvořeny jednotkové testy, jež kontrolují korektní průběh převodu z logického objektu na XML objekt a nazpět. Tento princip se využívá při odesílání herních objektů a importování a exportování dobrodružství. Z tohoto důvodu je žádoucí, aby tento převod proběhl vždy správně.

Pro otestování aplikace v tomto ohledu bylo vytvořeno 19 sad testů (pro každý logický objekt jedna) skládajících se celkem ze 164 testů.

### 4.1.2 Průběh

Všechny vytvořené sady jednotkových testů probíhají velmi podobně. Před spuštěním jednotlivých testů v rámci sady je nejprve vytvořen logický objekt se základními testovacími údaji a kolekcemi logických objektů (pakliže nějaké může obsahovat). Každá taková kolekce obsahuje právě jeden logický objekt se základními testovacími údaji, avšak bez případných kolekcí logických objektů.

Po inicializaci probíhají nejprve testy, jejichž úkolem je zjistit, zdali jsou všechny klíčové atributy logického objektu přeneseny do XML objektu a nazpět. Každý test si nejdřív vytvoří hlubokou kopii logického objektu vytvořeného při inicializaci a pozmění hodnotu testovaného atributu. Následně vytvoří z logického objektu XML objekt a na základě tohoto vytvořeného XML objektu vytvoří nový logický objekt.

Tento produkt obousměrného převodu nakonec porovná s původním objektem vytvořeným při inicializaci. Test předpokládá, že se objekty nebudou shodovat. Pokud by se shodovaly, znamenalo by to, že informace o změně daného atributu nebyla vzata při konverzi v potaz, a tudíž se při převodu ztratila.

Po dokončení všech negativních testů je proveden jeden test pozitivní. Jeho cílem je zkontrolovat, zdali pouhým převodem na XML objekt a zpět nedošlo k nežádoucí změně objektu. Proveďte tedy převod logického objektu z inicializace na XML objekt a zpět a nově vytvořený objekt porovná s původním. Očekávaným výsledkem je shoda objektů.

### 4.1.3 Výsledky

Po proběhnutí jednotkových testů skončily pozitivní testy v pěti sadách neúspěchem. Podrobnějším zkoumáním byly odhaleny přehlédnuté chyby v převodu těchto pěti typů logických objektů. Jednalo se o špatnou rekonstrukci typu rodového bonusu u všech typů zbraní, kvůli nevhodně zvolenému getteru při konstrukci XML objektů, a nejednotnou inicializaci nevyužitých atributů kontejnerů a peněz.

Všechny problémy byly ohlášeny osobě, která se stará o jediný další balíček využívající knihovnu pro hru Dračí doupě, v níž se zmíněné chyby nacházely, Šárce Weberové. Po konzultaci se Šárkou byly chyby v knihovně opraveny a oba závisující balíčky znovu otestovány, aby opravou nedošlo k poškození jiné části balíčků.

Po opravě skončily všechny testy v obou balíčcích úspěchem, a byly proto začleněny do knihovny.

## 4.2 Uživatelské testy

Uživatelské testy mají simulovat praktické využití vyvíjených aplikací. Pro účely tohoto typu testování je vytvořena testovací verze aplikace, která se předá testovacím uživatelům společně se scénáři, jež testovací uživatelé instruuji, jaké činnosti mají vykonávat a jaká je předpokládaná reakce aplikace na tyto akce.

Pokud skutečné chování aplikace odpovídá chování stanovenému ve scénáři, označí testovací uživatelé tento test za „splněný“. V opačném případě je test označen za „nesplněný“ a do poznámek je připsáno chování, jež se neshodovalo s chováním stanoveným ve scénáři.



### 4.2.1 Účel

Hlavním účelem uživatelských testů, jež byly vytvořeny pro vyvíjený balíček, je postupné projití případů užití 1.6 a vyzkoušení, zdali odpovídají vyvinutému balíčku. Vzhledem k tomu, že případy užití pokrývají funkční požadavky (viz tab. 1.1), dokáže úspěšné absolvování, že vyvinutý balíček splňuje vymezené funkční požadavky.

### 4.2.2 Průběh

Balíček pro Pána jeskyně představuje jistý druh „komunikačního mostu“ mezi hráčským balíčkem a desktopovou aplikací DeskChar a jako takový by měl být testován společně s oběma aplikacemi. Z tohoto důvodu probíhá uživatelské testování všech tří aplikací současně formou napodobení herní relace. Pro tuto relaci byl vytvořený hrubý scénář, který popisuje postup, v jakém mají být prováděny jednotlivé uživatelské testy.

Kvůli velkému množství testovacích scénářů, jejichž uveřejnění v kompletním rozsahu by nemělo velkou přidanou hodnotu, bude v této práci popsána pouze struktura scénáře testovací herní relace a její části, které přímo souvisejí s vyvíjeným balíčkem. Zbylé části jsou k nalezení v pracích Šárky Weberové [3] a Jana Horáčka [9] podle toho, ke které aplikaci se vztahují.

Testovací herní relace začíná přípravou dobrodružství v desktopové aplikaci DeskChar. Průběh této části relace je popsán podrobněji v práci Jana Horáčka [9]. Po dokončení přípravy je dobrodružství exportováno do souboru typu XML a přesunuto do příslušné složky v mobilním zařízení Pána jeskyně.

Pán jeskyně následně spustí aplikaci MobChar, přepne se na záložku *Dobrodružství* a importuje toto dobrodružství do aplikace. Další průběh relace se (až na závěrečné exportování dobrodružství) dělí na testování interních procesů a testování komunikace se zařízeními hráčů.

#### 4.2.2.1 Interní procesy

Na hlavní obrazovce balíčku pro Pána jeskyně postupně vytvoří alespoň dvě kouzla, jedno z vytvořených upraví a následně jedno smaže. Tento postup zopakuje i pro schopnosti a efekty, u nichž navíc vytvoří alespoň jeden závislý objekt (modifikátor u efektů, kontext u schopností).

Po dokončení těchto úkonů se Pán jeskyně přepne do záložky *Lokace*, kde vytvoří libovolnou adresářovou strukturu tak, aby výsledná struktura měla hloubku alespoň dva (tj. základní lokace bude obsahovat lokaci, která bude obsahovat další lokaci). Dále v jedné z vytvořených lokací vytvoří alespoň jednu bytost. Této bytosti následně vytvoří jeden libovolný předmět, jedno kouzlo a jednu schopnost.

Dále vytvoří alespoň po jednom předmětu od každého typu předmětů v libovolných vytvořených lokacích. Uvnitř vytvořeného kontejneru vytvoří další libovolný předmět a alespoň jednomu z vytvořených předmětů vytvoří efekt

## 4. TESTOVÁNÍ

---

s modifikátorem. Po vytvoření potřebných předmětů vytvoří Pán jeskyně ještě alespoň jednu cizí postavu. Vyzkouší u několika vybraných herních objektů v rámci vytváření jejich úpravu a mazání.

Následně vyzkouší funkce „Vyjmout“, „Kopírovat“ a „Vložit“ v rámci adresářového systému. Tyto operace budou vyzkoušeny také na lokacích a kontejnerech. Pomocí těchto operací také vytvoří kopii jedné cizí postavy v jiné lokaci, než se nachází její originál.

Na závěr této části vyhledá mapu vytvořenou v aplikaci DeskChar a prohlédne si její podrobnosti.

### 4.2.2.2 Komunikace se zařízeními hráčů

Po dokončení testování interních procesů vyhledá Pán jeskyně zařízení dalších dvou testovacích uživatelů (kteří používají testovací verzi hráčského balíčku) a vytvoří s nimi spojení.

Jakmile dojde k navázání spojení, vyžádá si od obou testovacích uživatelů informace o jejich postavách. Po obdržení těchto informací si obě postavy prohlédne a společně s testovacími uživateli ověří, že se informace shodují.

Následně Pán jeskyně vyzkouší odesílání herních objektů ze záložek hlavní obrazovky a adresářového systému (včetně detailů bytostí a kontejnerů).

Další v pořadí je testování odesílání a přijímání zpráv. Pán jeskyně nejprve odešle několik zpráv různým hráčům dle instrukcí podrobněji popsanych v práci Šárky Weberové [3] a ověří, že se zprávy korektně zobrazují na obrazovce s historií konverzace. Následně si nechá od jednoho z hráčů poslat čtyři zprávy, mezi jejichž přijetím se vždy přesune na specifické místo v aplikaci.

Po dokončení odesílání zpráv zopakuje celý proces vyžádání postav od hráčů a kontroly jejich údajů. Jakmile je tento krok hotový, vrátí se Pán jeskyně do obrazovky jádra aplikace MobChar, z níž testované dobrodružství exportuje do souboru formátu XML. Tento soubor následně importuje do desktopové aplikace DeskChar, v níž provede závěrečné testy a ověření, která jsou detailněji popsána v práci Jana Horáčka [9].

### 4.2.3 Výsledky

Pro uživatelské testy byly domluveny dvě kompletní skupiny hráčů hry Dračí doupe a tři Páni jeskyně, kteří v současné době aktivně nehrají. Těmto testovacím uživatelům byl odeslán balíček obsahující pokyny, scénáře a soubory potřebné pro instalaci testovacích verzí.

První odezva se dostavila již po několika minutách, kdy většina hráčů jedné skupiny a jeden z oslovených Pánů jeskyně hlásili, že při instalaci dochází k restartování mobilního zařízení. Na základě této zprávy byl kontaktován zbytek testovacích uživatelů, kteří se ovšem s podobným problémem nesetkali a byli již v různých stádiích testování.

Při hledání chyby bylo zjištěno na základě dodaných specifikací použitých zařízení, že se tento problém vyskytuje výhradně u zařízení značky Huawei. Vzhledem k této skutečnosti byl vyvozen závěr, že se bude nejspíše jednat o chybu v implementaci Android API, a tento problém byl nahlášen společnosti Huawei, jejichž vývojáři ho v současné době řeší.

Kvůli tomuto problému bylo množství testovacích uživatelů výrazně redukováno. Druhá testovací skupina prošla celou testovací relací a reportovala zpět úspěch ve všech scénářích. Zbylí Páni jeskyně vyzkoušeli jednotlivé aplikace odděleně (vynuli komunikační části) a také nenašli žádnou chybu.

Všichni testovací uživatelé byli s aplikacemi spokojeni a zaslali několik nápadů na vylepšení. Jednalo se především o rozšíření aplikace o další herní objekty jako například ochočená zvířata a vytvoření šablon často využívaných herních objektů (na základě pravidel [1]).

Současně při rozšíření povědomí o aplikaci byl projevem zájem o verze pro tablety, zařízení s novějšími verzemi systému Android a zařízení s operačním systémem iOS.

Tyto nápady byly zaevidovány, avšak nebudou již realizovány v rámci této práce.



---

## Závěr

Cílem této práce bylo vytvořit balíček aplikace MobChar pro Pána jeskyně pro hru Dračí doupě v souladu se standardy softwarového inženýrství.

V rámci práce byla vytvořena analýza současného stavu jádra aplikace MobChar, hráčského balíčku pro hru Dračí doupě a požadavků na vytvářený balíček. Na základě této analýzy byl navržen způsob, jakým bude vytvářený balíček implementován, aby splňoval vytyčené požadavky. Součástí návrhové části práce je i popis způsobu, jakým je vytvářený balíček začleněn do aplikace MobChar.

Po vytvoření návrhu byl balíček na základě tohoto návrhu implementován a výsledná implementace byla kompletně zdokumentována. Pomocí série jednotkových a uživatelských testů bylo ověřeno, že vytvořený balíček splňuje veškeré vytyčené požadavky. Pro vytvořený balíček byla sepsána uživatelská příručka, která má usnadnit uživatelům práci s vytvořeným balíčkem a je spolu s kompletní dokumentací k nalezení na přiloženém CD.

Vytvořením tohoto balíčku došlo ke splnění cíle této bakalářské práce.

Další směr vývoje vytvořeného balíčku byl identifikován na základě postřehů testovacích uživatelů ve vytvoření několika dalších herních objektů, rozšíření balíčku o šablony často využívaných herních objektů a vytvoření verzí balíčku pro tablety, zařízení s novějšími verzemi systému Android a zařízení s operačním systémem iOS.



---

## Literatura

- [1] Benda, M.: *Dračí doupě: pravidla pro začátečníky, edice B.* verze 1.6, nakladatelství Altar, 2001, ISBN 80-85979-34-9.
- [2] nakladatelství Altar: *Osobní deník pro Dračí doupě.* [online], červenec 2011, [cit. 2017-02-28]. Dostupné z: [http://www.altar.cz/download/drd/osobni\\_denik.pdf](http://www.altar.cz/download/drd/osobni_denik.pdf)
- [3] Weberová, Š.: *MobChar - balíček pro Dračí doupě 3.0.* Bakalářská práce, Praha: České vysoké učení technické, Fakulta informačních technologií, 2017.
- [4] OODesign: *Object Oriented Design.* [online], duben 2008, [cit. 2017-04-28]. Dostupné z: <http://www.oodesign.com/factory-method-pattern.html>
- [5] apache commons documentation team: *Apache Commons Lang.* [online], květen 2013, [cit. 2017-05-02]. Dostupné z: <https://commons.apache.org/proper/commons-lang/>
- [6] Gallagher, N.: *Simple 2.7.1.* [online], červen 2004, [cit. 2017-05-02]. Dostupné z: <http://simple.sourceforge.net/>
- [7] Ortiz, M.: *Touch Image View.* [online], duben 2014, [cit. 2017-05-02]. Dostupné z: <https://github.com/MikeOrtiz/TouchImageView>
- [8] Phillips, B.; Hardy, B.; Stewart, C.; aj.: *Android Programming: The Big Nerd Ranch Guide.* Createspace Independent Publishing Platform, druhé vydání, 2016, ISBN 978-0134171500.
- [9] Horáček, J.: *MobChar - desktopová aplikace pro Pána jeskyně pro Dračí doupě.* Bakalářská práce, Praha: České vysoké učení technické, Fakulta informačních technologií, 2017.





## Seznam použitých zkratk

**OS** Operační Systém

**API** Application Programming Interface

**XML** eXtensible markup language

**DRY** Do not Repeat Yourself

**DAO** Data Access Object

**UML** Unified Modeling Language

**AUF** Always Use Fragments

**JVM** Java Virtual Machine



---

## Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
apks .....	adresář s instalačními soubory aplikací
src	
_ impl .....	adresář se zdrojovými kódy implementace
_ thesis.....	adresář se zdrojovou formou práce ve formátu L <sup>A</sup> T <sub>E</sub> X
_ user_guide.....	adresář se zdrojovou formou uživatelské příručky
doc	
_ javadoc.....	adresář s javadoc zdrojových kódů
_ documentation.pdf .....	analytická a návrhová dokumentace
text	
_ thesis.pdf .....	text práce ve formátu PDF
_ user_guide.pdf .....	uživatelská příručka ve formátu PDF