

Sem vložte zadanie Vašej práce.

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



Bakalárska práca

Informační systém pro organizaci matematické soutěže

Filip Sakala

Vedúci práce: Ing. Ondřej Guth, Ph.D.

10. mája 2015

Pod'akovanie

Týmto by som chcel vyjadriť pod'akovanie Ing. Ondřejovi Guthovi, Ph.D., ako vedúcemu tejto práce, za možnosť pravidelných stretnutí a všetky inšpiratívne rady ohľadom jej vypracovania. Ďalej by som chcel pod'akovať organizátorom matematického korešpondenčného seminára Riešky za aktívny záujem o vytvorenie informačného systému, ktorý im bude slúžiť. Tiež by som chcel pod'akovať svojej rodine a priateľom za podporu a motiváciu pri štúdiu.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb, autorského zákona, v znení neskorších predpisov. V súlade s ustanovením § 46 odst. 6 tohoto zákona týmto udeľujem bezvýhradné oprávnenie (licenciu) na používanie tejto mojej práce, a to vrátane všetkých počítačových programov ktoré sú jej súčasťou alebo prílohou a tiež všetkej ich dokumentácie (ďalej len „Dielo“), a to všetkým osobám, ktoré si prajú Dielo používať. Tieto osoby sú oprávnené Dielo používať akýmkoľvek spôsobom, ktorý neznižuje hodnotu Diela, ale len pre nezárobové účely. Toto oprávnenie je časovo, územne a množstevne neobmedzené.

V Prahe 10. mája 2015

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2015 Filip Sakala. Všetky práva vyhradené.

Táto práca vznikla ako školské dielo na FIT ČVUT v Prahe. Práca je chránená medzinárodnými predpismi a zmluvami o autorskom práve a právach súvisiacich s autorským právom. Na jej využitie, s výnimkou bezplatných zákonných licencií, je nutný súhlas autora.

Odkaz na túto prácu

Sakala, Filip. *Informační systém pro organizaci matematické soutěže*. Bakalárska práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

Cieľom tejto bakalárskej práce je vytvoriť informačný systém na organizáciu matematického korešpondenčného seminára Riešky. Po dôkladnom preskúmaní súčasného stavu organizácie a analýze požiadaviek organizátorov bol navrhnutý informačný systém ako webová aplikácia vo frameworku Ruby on Rails s využitím databázy MySQL.

Implementovaná webová aplikácia bola ďalej riadne otestovaná jednotkovými testami aj samotnými organizátormi súťaže. Výsledný informačný systém bol nasadený na server organizácie pomocou nástrojov Phusion Passenger a Capistrano. Informačný systém má v budúcnosti uľahčiť a zefektívniť prácu organizátorov.

Kľúčová slova webová aplikácia, informačný systém, matematická súťaž, Ruby on Rails, Ruby

Abstract

The aim of this bachelor's thesis is to create an information system for the organization of a mathematics competition Riešky. After a thorough examination of the current state of the organization and analysis of the requirement specifications, an information system was designed as a web application within the Ruby on Rails framework using a MySQL database.

The implemented web application was subsequently properly tested by unit tests, as well as the organizers themselves. The resultant information system was deployed to the server of the organization with the use of Phusion Passenger and Capistrano tools. The information system should facilitate and optimize the work of the organizers in the future.

Keywords web application, information system, math competition, Ruby on Rails, Ruby

Obsah

Úvod	1
1 Ciele práce	3
1.1 Štruktúra práce	3
2 Analýza	5
2.1 Súčasný stav organizácie	5
2.2 Analýza požiadaviek	14
2.3 Prípady použitia	17
2.4 Analytické triedy	39
3 Návrh	49
3.1 Návrh architektúry	49
3.2 Návrhové triedy	50
3.3 Balíčky návrhových tried	54
3.4 Návrh užívateľského rozhrania	58
4 Implementácia	69
4.1 Implementácia	69
4.2 Problémy a zaujímavosti	70
4.3 Nasadenie	74
5 Testovanie	75
5.1 Automatizované testy	75
5.2 Testovanie použiteľnosti	77
5.3 Testovanie behu vo webových prehliadačoch	79
Záver	81
Rozšíriteľnosť systému	82

Literatúra	83
A Zoznam použitých skratiek	85
B Slovník pojmov	87
C Inštalčná príručka	89
C.1 Verzie	89
C.2 Spustenie aplikácie vo vývojovom prostredí	90
C.3 Postup nasadenia aplikácie	90
C.4 Testovací prístup do aplikácie	90
D Obsah priloženého CD	91

Zoznam obrázkov

2.1	Popis úloh okruhu príklady	6
2.2	Popis úloh okruhu knižka	7
2.3	Popis úloh okruhu obáľkovanie	8
2.4	Popis úloh okruhu propagácia	9
2.5	Štavy organizátora	11
2.6	Časové úseky v seminári	12
2.7	Zoznam účastníkov	18
2.8	UC: Prihlasovanie do systému	19
2.9	UC: Príklady	20
2.10	UC: Knižka	22
2.11	UC: Opravovanie	24
2.12	UC: Obáľkovanie	27
2.13	UC: USP	28
2.14	UC: Propagácia	30
2.15	UC: oRiešok	32
2.16	UC: Akcia	34
2.17	UC: Adresár	36
2.18	UC: Úlohy	38
2.19	Analytický model tried	40
3.1	Príklad použitia kompozície v návrhovom modeli	51
3.2	Príklad použitia agregácie v návrhovom modeli	51
3.3	Príklad rozdelenia analytickej relácie M:N	52
3.4	Návrhový model tried	53
3.5	Balíček oRiešok	54
3.6	Balíček Organizátori	55
3.7	Balíček Úlohy	55
3.8	Balíček Udalosti	56
3.9	Balíček Opravovanie	56
3.10	Balíček Príklady	57

3.11	Balíček Súťaž	57
3.12	Stránky organizátora	59
3.13	Stránky príkladov	60
3.14	Stránky oRieška a úloh	61
3.15	Stránky súťaže	62
3.16	Stránky súťaže	63
3.17	Stránky súťaže	64
3.18	Stránky súťaže	65
3.19	Stránky súťaže	66
3.20	Stránky súťaže	67
4.1	Príklad optimistického zamykania záznamov	73

Zoznam tabuliek

2.1	Klady a zápory <i>mailing listov</i>	13
2.2	Klady a zápory skladu	13
2.3	Klady a zápory interných stránok	13
2.4	Klady a zápory iných riešení	14

Úvod

Občianske združenie Matematická spoločnosť – Mladý génius pôsobí na Slovensku od roku 1999 a okrem iného sa venuje rozvoju mladých talentov v oblasti matematiky. Dobrovoľníci zo združenia preto každoročne organizujú matematický korešpondenčný seminár Riešky [1].

Seminár je určený pre žiakov 5. až 9. ročníka základnej školy alebo gymnázia. V čase písania tejto práce prebieha už 17. ročník súťaže, v ktorej sa každoročne zúčastní približne 100 žiakov. Organizátormi súťaže sú študenti stredných a vysokých škôl. Väčšina organizátorov prešla k dobrovoľníckej činnosti po účasti v tejto súťaži. Oficiálne zastrešenie organizácie poskytuje predmetová komisia matematiky na Gymnáziu Grösslingová v Bratislave.

Organizátor súťaže zmysluplne vyplňa svoj voľný čas. Rozvíja svoje zručnosti, môže si vyskúšať pestrú škálu činností, vyberá si prácu, ktorá ho baví, učí sa dodržiavať dohodnuté termíny a prevziať zodpovednosť za svoju prácu. Okrem toho sa učí pracovať v tíme, v prípade, že získa dostatočné vedomosti, učí mladších organizátorov. V neposlednom rade sa učí, ako pracovať s deťmi, ako byť dobrým vzorom a ako zvládať krízové situácie.

Objem organizácie seminára sa každým rokom zväčšoval. Pôvodnou myšlienkou bolo podporovať matematické myslenie u žiakov. Organizátori teda vymýšľali a zadávali niekoľko príkladov ročne. Riešitelia svoje riešenia poslali poštou alebo ich odovzdali na Gymnáziu Grösslingová v Bratislave. Riešenia boli opravené, organizátori vytvorili vzorové riešenie, aby sa žiaci zo svojich chýb mohli poučiť.

So zväčšujúcim sa počtom organizátorov a rozvojom technológií rástol aj počet nápadov a úloh na organizovanie. Napríklad, najlepším riešiteľom bola po skončení série súťaže poslaná pozvánka na sústreďenie, na ktorom sa okrem získavania nových vedomostí z matematiky zahráli hry a spoznali sa s novými kamarátmi. Sústreďenie teda funguje ako motivačný prvok a odmena pre riešiteľov.

Organizátori spolupracujú aj s podobne zameranými organizáciami na rôznych ďalších súťažiach ako Matboj, Pikopretek alebo Náboj junior.

ÚVOD

Sám autor tejto práce je jedným z organizátorov súťaže, a preto má prehľad vo vykonávaných úlohách a cieľoch tejto organizácie.

Ciele práce

Cieľom tejto práce je poskytnúť organizátorom nástroj na celkovú organizáciu seminára tak, aby ušetrili svoj čas pri vykonávaní úloh bez potreby dlhého zaškoľovania alebo učenia sa s novými programami. Navyše by mal vzniknúť lepší spôsob rozdeľovania a kontroly úloh a chyby spôsobené jednotlivcom by mali byť obmedzené. Hlavné ciele práce sú:

- Vytvoriť systém, ktorý budú vedieť intuitívne používať všetci organizátori bez potreby úvodných školení.
- Zjednotiť postupy vykonávania úloh, aby organizátori nemuseli vymýšľať vlastné spôsoby, ktoré nie je možné kontrolovať.
- Zjednodušiť vykonávanie úloh.
- Obmedziť chyby vzniknuté ľudským faktorom.
- Obmedziť využívanie viacerých programov na jednu úlohu. Všetko by malo prebiehať z jedného miesta.

1.1 Štruktúra práce

Kvôli prehľadnosti je práca rozdelená do základných kapitol: Analýza, návrh, implementácia a testovanie.

V rámci analýzy je podrobne rozobrané súčasné fungovanie organizácie tak, aby čitateľ porozumel jednotlivým vykonávaným úlohám, doteraz používaným technológiám a spôsobom komunikácie medzi organizátormi. Z popísaných úloh vyplynú funkčné a nefunkčné požiadavky na systém, ktoré sú základným zoznamom funkcií systému. Zoznam účastníkov systému a vykonávanie funkcií je podrobne rozobrané v prípadoch použitia, respektíve v scenároch prípadov použitia. Výstupom analýzy je model analytických tried. Popísané sú jednotlivé triedy ako entity vyplývajúce z tejto domény.

1. CIELE PRÁCE

Návrh systému obsahuje časť zameranú na architektúru systému a navrhované technológie. Ďalej návrh vychádza z analytického modelu tried, z ktorého vplynú návrhové triedy. Obsahuje tiež návrh užívateľského rozhrania systému.

Implementácia vychádza z návrhových tried. V tejto kapitole je popísaný spôsob, akým bol informačný systém vytvorený. Obsahuje problémy, s ktorými sa autor tejto práce stretol a spôsoby ich riešenia.

Testovanie systému obsahuje popis vytvorených jednotkových testov, spôsob testovania samotnými organizátormi súťaže a výsledky týchto testov.

V závere bude zhrnutá a zhodnotená celková práca na informačnom systéme. Budú tiež popísané ďalšie možnosti rozšírenia systému.

Autor tejto práce pri tvorbe využíval vedomosti získané v [2], kde sa dozvedel o prechode softvérovým procesom a tvorbe modelov UML.

Analýza

Podľa [2] je cieľom analýzy vytvoriť analytický model tried, na čo je potrebné získať požiadavky na systém. Výsledkom má byť popis, čo má systém vedieť vykonávať. Navyše, analýza tohto systému slúži na ujasnenie obsahu medzi autorom práce a samotnými organizátormi.

Na to, aby vznikol analytický model tried, je potrebné získať prehľad o doméne, jej postupoch a v neposlednom rade jej jazyku. Táto kapitola sa teda v prvej časti zaoberá popisom súčasného fungovania organizácie.

V druhej časti sú popísané požiadavky na systém, ktoré vyplývajú z tejto domény. Na vysvetlenie postupov jednotlivých funkčných požiadaviek sú následne podrobne popísané prípady použitia a ich scenáre.

Na záver bol z týchto informácií vytvorený analytický model tried popisujúci doménu organizácie. Vysvetlený je tiež význam jednotlivých vzniknutých tried, ich vzťahov k ostatným triedam, význam ich atribútov a metód.

2.1 Súčasný stav organizácie

Organizácia sa pri svojom fungovaní zaoberá veľkým množstvom úloh. Táto kapitola poskytuje vyčerpávajúci zoznam úloh a postupov používaných organizátormi. V závere kapitoly sú zhrnuté hlavné používané technológie.

2.1.1 Prehľad organizácie úloh

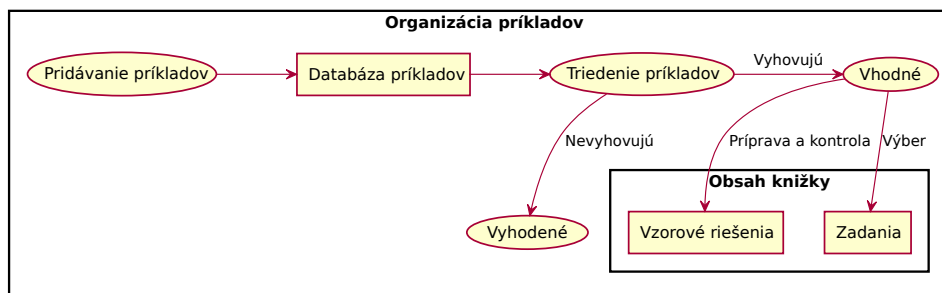
Kapitola ukazuje úlohy, ktoré sú v rámci súťaže vykonávané. K úlohám je popísaný aj spôsob vykonania či použité technológie.

2.1.1.1 Organizácia príkladov

Organizácia príkladov je základným prvkom organizácie. Bez organizácie tohto okruhu by totiž súťaž nemohla prebiehať.

2. ANALÝZA

Okruh sa zaoberá zberom matematických príkladov a ich výberom do zadaní súťaže, kontrolou vzorových riešení príkladov. Okrem organizácie príkladov do seminára organizátori spolupracujú na matematickej súťaži Matboj, do ktorej rovnako pripravujú zadania príkladov. Navyše, vzniká vrámci seminára aj nová kategória príkladov – USP. Príklady sú teda rozdelené do viacerých kategórií. Názorné zobrazenie úloh je v diagrame na obrázku 2.1.



Obr. 2.1: Popis úloh okruhu príklady

Úlohy tohto okruhu sú:

- *Udržiavanie databázy príkladov*: Pridávanie a udržiavanie dostatočného množstva kvalitných príkladov, z ktorých je možné v prípade potreby vyberať. Triedenie do kategórií podľa obtiažnosti a použitia.
- *Výber príkladov do zadaní*: Výber vhodných príkladov do zadaní súťaže.
- *Kontrola vzorových riešení*: Zodpovednosť za prípravu a kontrolu vzorových riešení súťaže.

2.1.1.2 Organizácia knižky

Organizátori pripravujú pravidelne vychádzajúce časopisy so zadaniami príkladov, vzorovými riešeniami, výsledkovými listinami a dodatočnými informáciami. Organizátori sú tiež zodpovední za kontrolu gramatiky a sadzby vo vybraných príkladoch. Názorné zobrazenie úloh je v diagrame na obrázku 2.2.

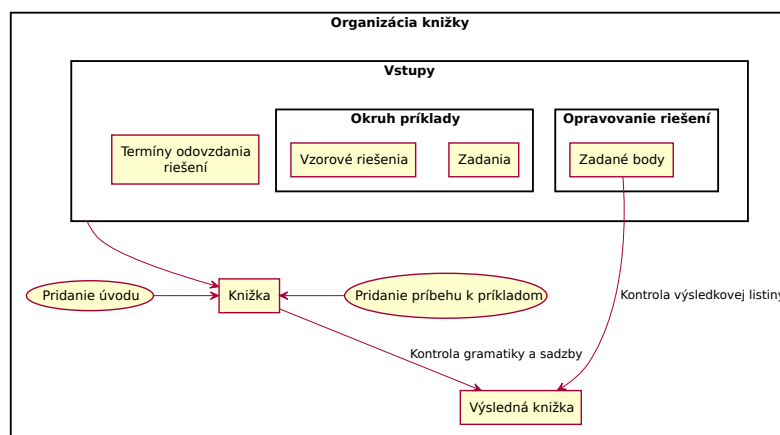
- *Pridávanie obsahu knižky*: Pridanie termínov na odovzdanie riešení, úvodu a príbehu k príkladom.
- *Kontrola gramatiky a sadzby knižky*: Kontrola zadaní, vzorových riešení, výsledkovej listiny a ďalšieho obsahu knižky.

Vrámci každého kola súťaže je pripravovaná knižka s určitým obsahom:

- Knižka 0: Úvod, príklad, ako riešiť Riešky, zadania príkladov 1. a 2. kola, pravidiel súťaže, prihláška do súťaže.

- Knižka 1: Úvod, zadania príkladov 3. kola, vzorové riešenia 1. kola, výsledková listina riešiteľov po 1. kole.
- Knižka 2: Úvod, vzorové riešenia 2. kola, výsledková listina riešiteľov po 2. kole.
- Knižka 3: Úvod, vzorové riešenia 3. kola, výsledková listina riešiteľov po 3. kole.

Toto usporiadanie je najtypickejšie, no nie je úplne striktné. Jednotlivé súčasti sa medzi knižkami podľa potreby presúvajú.



Obr. 2.2: Popis úloh okruhu knižka

2.1.1.3 Organizácia obáľkovania

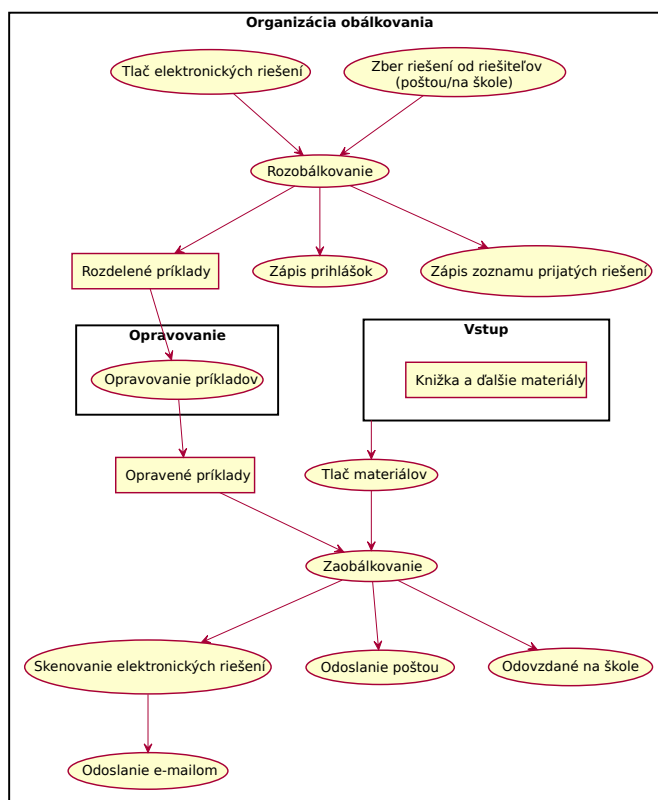
Obáľkovanie sa zaoberá tlačou knižiek, pozvánok, prihlášok a iných materiálov a ich distribúciou medzi riešiteľov. Tiež je zodpovedné za spracovanie prijatých riešení od riešiteľov. Cieľom obáľkovania je, aby sa prichádzajúce riešenia dostali k opravovateľom a opravené riešenia späť k riešiteľom. Názorné zobrazenie úloh je v diagrame na obrázku 2.3.

- *Tlač materiálov*: Úlohou je vytlačiť dostatočné množstvo knižiek, pozvánok, prihlášok, diplomov a iných materiálov podľa potreby.
- *Príjem riešení*: Riešitelia môžu svoje riešenia doručiť troma spôsobmi:
 1. Elektronicky cez stránku alebo e-mail.
 2. Poštou – na papier.
 3. Žiaci Gymnázia Grösslingová 18 a Gymnázia Jura Hronca vo svojej škole – na papier.

2. ANALÝZA

Organizátori vytlačia riešenia posielané elektronickou formou a spoja ich s ostatnými riešeniami.

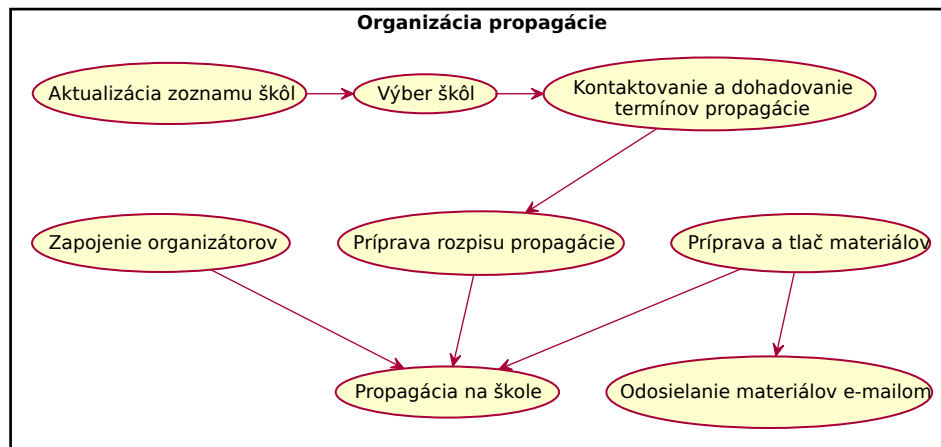
- *Rozobáľkovanie*: Riešenia sú fyzicky rozdeľné na kôpky podľa čísla príkladu a distribuované medzi opravujúcich organizátorov. Organizátori zapisujú zoznam odovzdaných príkladov jednotlivých riešiteľov, ktorý slúži ako kontrola, a tiež prihlášky do súťaže.
- *Zaobáľkovanie*: Opravené riešenia sú zozbierané od opravujúcich organizátorov a manuálne rozdelené na kôpky podľa mena riešiteľa. Riešenia riešiteľov, ktorí súťažia elektronickou formou, sú skenované a odosielané e-mailom. Riešenia ostatných riešiteľov sú odoslané poštou alebo odovzdané v škole.



Obr. 2.3: Popis úloh okruhu obáľkovanie

2.1.1.4 Organizácia propagácie

Propagácia má za úlohu rozširovať povedomie o súťaži medzi deťmi, učiteľmi a školami. Národné zobrazenie úloh je v diagrame na obrázku 2.4.



Obr. 2.4: Popis úloh okruhu propagácia

- *Udržiavanie zoznamu zapojených škôl*: Organizátori ukladajú zoznam škôl, na ktorých sa propaguje spolu s kontaktami na zástupcov škôl.
- *Vytváranie a odosielanie propagačných materiálov*: Vytváranie propagačných materiálov na prezentáciu činností organizácie.
- *Propagácia na školách*: Osobná propagácia na školách je najviac využívaným spôsobom udržiavania riešiteľskej komunity. Z hľadiska organizácie je potrebné dohodnúť sa so školami na termínoch, pripraviť časový plán návštevy škôl, zapojiť organizátorov, vytlačiť materiály a dohodnúť obsah a formu propagácie.

2.1.1.5 Organizácia oRieška

Časopis oRiešok je posielaný spolu s knižkou a obsahuje rôzne populárno-náučné témy. Hlavnou úlohou je vytvorenie článkov a ich gramatická kontrola.

2.1.1.6 Organizácia USP

USP je v čase písania práce novou kategóriou vrámci súťaže. Obsahuje zložitejšie príklady, ktoré môžu riešitelia odovzdávať iba v elektronickej podobe. Úlohy sú podobné ako v prípade organizácie knižky (kapitola 2.1.1.2), teda príprava a kontrola zadaní, vzorových riešení a výsledkových listín.

2.1.1.7 Organizácia opravovania riešení

Pred samotným opravovaním riešení je vytvorený rozpis opravovania, teda prehľad priradenia organizátorov k jednotlivým príkladom. Jeden príklad môže

2. ANALÝZA

opravovať aj viacero organizátorov. V takom prípade opravovanie prebieha formou osobného stretnutia a výsledné body sú udelené na základe dohody týchto organizátorov.

Organizátor zodpovedný za opravovanie príkladu boduje postup riešenia bodmi 0 až 10. Okrem pridelovania bodov je opravujúci povinný napísať vzorové riešenie príkladu.

2.1.1.8 Organizácia sústreďenia

Po skončení série je pre najlepších riešiteľov pripravované sústreďenie. Týmto riešiteľom je poslaná pozvánka a prihláška. Ďalej je vytvorený zoznam účastníkov a program sústreďenia.

Organizácia zabezpečuje okrem sústreďení aj tábory, a to letný tábor a denný tábor. Okrem toho pripravuje výlety.

Organizátori, ktorí sa smú zúčastniť sústreďenia alebo tábora, môžu byť volení na základe ich aktivity v seminári.

2.1.1.9 Ďalšie úlohy

Okrem vyššie spomenutých, existujú mnohé menšie práce, ktoré je možné vykonávať.

2.1.2 Funkcie organizátora

Kapitola popisuje rôzne členenie povinností organizátora a funkcie. Organizátor môže mať funkcie:

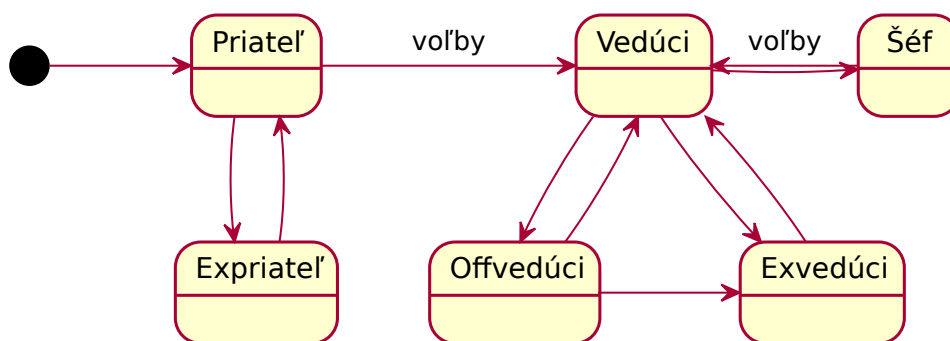
- Práve jednu z hlavných funkcií (kapitola 2.1.2.1).
- Samostatný alebo nesamostatný opravovateľ riešení (kapitola 2.1.2.2).
- Niektoré zo zoznamu okruhov, buď ako zodpovedná osoba za okruh, alebo bežný pomocník (kapitola 2.1.2.3).

2.1.2.1 Hlavná funkcia

V tejto organizácii platí dohodnutá hierarchia podľa získaných skúseností:

- Priateľ: Do tejto kategórie je zaradený každý nový dobrovoľník.
- Vedúci: Zodpovedný priateľ, ktorý má dostatočne veľa skúseností. Je volený ostatnými vedúcimi. Vedúci má právo voliť nových vedúcich a šéfa, voliť organizátorov sústreďenia a podobne.
- Šéf: Hlavný vedúci zodpovedný za chod seminára. Je volený vedúcimi.

Pri neaktivite môže vedúci prejsť do stavu *off*. Predpokladá sa, že takto označený organizátor sa ďalej bude zapájať v chode organizácie, avšak v menšej miere. Stráca tiež hlasovacie právo. Organizátor, ktorý sa už ďalej nechce venovať organizácii, môže prejsť do stavu *ex*. Stavy organizátora sú znázornené na obrázku 2.5.



Obr. 2.5: Stavy organizátora

2.1.2.2 Opravovateľ

Opravujúci organizátor riešení môže mať jeden z dvoch stavov:

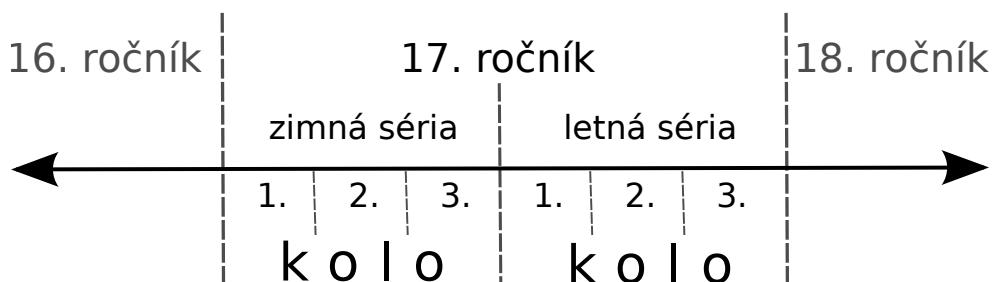
- Nesamostatný: Učí sa opravovať riešenia, zadávať body, písať vzorové riešenia.
- Samostatný: Má skúsenosti s opravovaním, vedie bežných opravovateľov k zodpovednému opravovaniu a radí im.

2.1.2.3 Člen okruhov

Organizátor sa môže zaradiť do určitého okruhu (vedľajšej funkcie) a vykonávať úlohy s ním súvisiace. Jeden alebo viacerí organizátori sú zodpovední za beh daného okruhu, ostatní organizátori sú pomocní. Zoznam okruhov: Knižka, obáľkovanie, príklady, propagácia, oRiešok, komunikácia.

2.1.3 Rozdelenie seminára na časové úseky

Priebeh súťaže typicky vieme rozdeliť na časové úseky. Najväčším z nich je *ročník*, ktorý predstavuje 1 organizačný rok v súťaži a časovo je ohraničený školským rokom. Ročník býva rozdelený na dve *série*, zimnú a letnú. Každá séria má 3 *kolá*. Schématické zobrazenie vidno na obrázku číslo 2.6.



Obr. 2.6: Časové úseky v seminári

2.1.4 Formy organizácie a technológie

Na vnútornú organizáciu súťaže sú momentálne používané nasledujúce formy spracovania informácií:

- *Mailing list* (sekcia 2.1.4.1),
- verzovací systém svn (ďalej tiež „sklad“) (sekcia 2.1.4.2),
- zabezpečená stránka s informáciami a vedomosťami o organizovaní súťaže (ďalej tiež „interné“ – sekcia 2.1.4.3),
- papierová forma, ústna dohoda (sekcia 2.1.4.4).

2.1.4.1 Mailing list

Najviac používaným spôsobom dohadovania úloh je e-mail. E-maily organizátorov sú zjednotené pod spoločný *mailing list*. Výhody a nevýhody sú zhrnuté v tabuľke 2.1. Momentálne existujúce *mailing listy*:

- Priatelia: Hlavný zdroj informácií a rozdeľovanie úloh.
- Vedúci: Riešenie problémov, voľby.
- Šéfovia: Riešenie problémov na úrovni šéfov.
- Tábor: Organizácia letných táborov.
- Víkendovka: Organizácia víkendových stretnutí organizátorov.

2.1.4.2 Sklad

Všetky dôležité súbory súvisiace s organizáciou sú ukladané do skladu. Sklad je vlastne verzovací systém subversion [5]. Výhody a nevýhody sú zhrnuté v tabuľke 2.2.

2.1. Súčasný stav organizácie

Tabuľka 2.1: Klady a zápory *mailing listov*

Výhody	Nevýhody
Prehľadnosť	Príliš veľa správ a ich miešanie pri organizovaní viacerých úloh naraz
Rýchla komunikácia	Závislé od jednotlivcov (frekvencie čítania e-mailov)
Možnosť viacerých príjemcov	Zahlcovanie mailových schránok ľudí, ktorých sa daná úloha netýka

Tabuľka 2.2: Klady a zápory skladu

Výhody	Nevýhody
Verzovanie súborov	Potreba software u každého organizátora
Zobrazenie zmien medzi revíziami (história zmien)	Potrebné zaučenie nových organizátorov
Možnosť návratu k predchádzajúcej verzii	Nástroje na správu verzií síce existujú, no väčšina organizátorov ich nevie (alebo nechce) používať
Automatické generovanie knižky do formátu pdf pri zmene prislúchajúcich súborov (napr. zadaní)	

2.1.4.3 Interné

Interné stránky obsahujú informácie a učebné materiály o organizácii jednotlivých úloh. Výhody a nevýhody sú zhrnuté v tabuľke 2.3.

Tabuľka 2.3: Klady a zápory interných stránok

Výhody	Nevýhody
Dôležité informácie na jednom mieste	Neprehľadnosť
Databáza príkladov	Nepostačujúca funkcionálnosť databázy (napr. nemožnosť pridávania obrázkov k príkladom)
Zadávanie úloh	Príliš jednoduché spracovanie úloh, nemožnosť priradiť úlohu konkrétnym organizátorom.

2.1.4.4 Iné

Množstvo menších úloh je rozdeľovaných ústnou formou a mnohé materiály sú spracovávané či uchovávané v papierovej forme (napríklad riešenia príkladov

od riešiteľov). Výhody a nevýhody tohto riešenia sú zhrnuté v tabuľke 2.4.

Tabuľka 2.4: Klady a zápory iných riešení

Výhody	Nevýhody
Najrýchlejšia forma komunikácie je ústna	Zabúdanie dohodnutých vecí a termínov
Rýchle zbieranie a zápis úloh na papieri.	Strata a následné zabudnutie dohodnutých vecí.

2.2 Analýza požiadaviek

V predchádzajúcich kapitolách boli popísané druhy organizovaných úloh a rôzne formy ich riešenia. Z týchto úloh vyplývajú funkčné požiadavky na systém ako základný zoznam funkcií, ktoré má budúci informačný systém obsahovať a tiež nefunkčné požiadavky, ktoré špecifikujú vlastnosti alebo obmedzujúce podmienky daného systému [2].

2.2.1 Funkčné požiadavky

Funkčné požiadavky sú pre prehľadnosť rozdelené do skupín tak, ako vyplývajú z popísanej domény. [2] tiež odporúča spôsob formulovania požiadaviek ako:

`<id> <systém> bude <funkcia>`,

kde `<id>` predstavuje jedinečné číslovanie požiadaviek, `<systém>` vyjadruje objekt, ktorý *bude* vykonávať vybranú `<funkciu>`. Požiadavky sú tak prehľadnejšie a dostatočne definujúce.

Funkčné požiadavky sú pre prehľadnosť rozdelené do skupín. V popise týchto požiadaviek bude identifikátor zložený z poradového čísla skupiny, názvu a číslovania vrámci tejto skupiny.

1. *Prihlasovanie*

- a) Systém bude umožňovať prihlasovanie a odhlasovanie organizátorov.

2. *Databáza príkladov*

- a) Systém bude umožňovať prídanie nového príkladu.
- b) Systém bude umožňovať úpravu existujúceho príkladu.
- c) Systém bude umožňovať vymazanie príkladu.
- d) Systém bude umožňovať zobrazenie výberovej databázy vo formáte PDF.

- e) Systém bude umožňovať výber a odobranie konkrétneho príkladu v konkrétnej knižke (ročníku, sérii, kole).
- f) Systém bude evidovať štatistiky pridaných príkladov od užívateľov za dané obdobie.

3. *Knižka*

- a) Systém bude evidovať archív doteraz prebehnutých ročníkov.
- b) Systém bude umožňovať prídanie nového ročníka, série či kola.
- c) Systém bude umožňovať pridávanie, úpravu a odoberanie úvodov, zadaní, vzorových riešení, termínov kôl a iných článkov určitého kola do knižky.
- d) Systém bude umožňovať mód kontroly pridaného obsahu knižky.
- e) Systém bude umožňovať zobrazenie výslednej knižky vo formáte PDF.
- f) Systém bude umožňovať zobrazenie výsledkovej listiny riešiteľov v danom kole.

4. *Opravovanie*

- a) Systém bude evidovať rozpis opravovania (teda priradenie organizátorov na príklad v danom kole).
- b) Systém bude evidovať body zadané organizátorom.
- c) Systém bude evidovať jednotlivé opravovania s možnosťou komentárov.

5. *Obáľkovanie*

- a) Systém bude evidovať prihlášky od riešiteľov.
- b) Systém bude umožňovať odosielanie opravených elektronických riešení e-mailom.
- c) Systém bude umožňovať vytvorenie diplomu za sériu.
- d) Systém bude umožňovať prídanie zoznamu prijatých riešení od riešiteľov v danom kole.

6. *USP*

- a) Systém bude evidovať archív doteraz prebehnutých ročníkov.
- b) Systém bude umožňovať prídanie nového ročníka, série či kola.
- c) Systém bude umožňovať pridávanie zadaní, vzorových riešení a termínov určitého kola do knižky.
- d) Systém bude umožňovať mód kontroly pridaného obsahu knižky.

2. ANALÝZA

- e) Systém bude umožňovať zobrazenie výslednej knižky vo formáte PDF.
- f) Systém bude evidovať zoznam riešení na opravenie.
- g) Systém bude umožňovať zadanie bodov a komentáru za riešenie a ich odoslanie riešiteľovi pomocou e-mailu.

7. *oRiešok*

- a) Systém bude evidovať archív minulých čísel časopisu.
- b) Systém bude umožňovať vytvorenie nového čísla časopisu.
- c) Systém bude evidovať ponuku článkov na vytvorenie a ich priradenie organizátorovi.
- d) Systém bude umožňovať prídanie článku a jeho obsahu.
- e) Systém bude umožňovať kontrolu gramatiky článkov.

8. *Adresár*

- a) Systém bude evidovať organizátorov.
- b) Systém bude umožňovať prídanie nového organizátora.
- c) Systém bude umožňovať upravenie údajov.
- d) Systém bude umožňovať vyhľadávanie v adresári podľa kľúčových slov.
- e) Systém bude umožňovať upravenie funkcií organizátorov.

9. *Zoznam úloh*

- a) Systém bude evidovať zoznam pridaných úloh.
- b) Systém bude umožňovať prídanie novej úlohy, jej priradenie konkrétnym užívateľom a prídanie súvisiacich súborov.
- c) Systém bude umožňovať úpravu danej úlohy.
- d) Systém bude umožňovať archiváciu úlohy.

10. *Propagácia*

- a) Systém bude evidovať zoznam škôl s kontaktami.
- b) Systém bude umožňovať vytvorenie programu propagácie na určité dni.
- c) Systém bude umožňovať posielanie propagačných materiálov na vybrané školy zo zoznamu e-mailom.

11. *Akcia*

- a) Systém bude evidovať zoznam minulých akcií v ročníku.

- b) Systém bude umožňovať vytvorenie a úpravu akcie.
- c) Systém bude umožňovať priradenie organizátorov zodpovedných za organizovanie akcie.
- d) Systém bude umožňovať vyhlásenie volieb na konkrétnu akciu, kandidovanie a hlasovanie vo voľbách a vyhodnotenie výsledkov volieb.
- e) Systém bude umožňovať pridanie plánu akcie a podporných súborov.

2.2.2 Nefunkčné požiadavky

1. Dostupnosť na webe – systém bude webová aplikácia zobraziteľná vo webových prehliadačoch.
2. Možnosť prihlasovania do systému pomocou prihlasovacieho mena a hesla.
3. Nasadenie systému na serveri organizácie, použitie databázy *MySQL*.
4. Väčšina funkcií systému má byť prístupná pre všetkých organizátorov. Úlohy, ktoré nemajú byť prístupné všetkým, sú: Zobrazenie jednotlivých opravovaní a komentovanie (samostatný opravovateľ), upravovanie funkcií organizátorov (vedúci/šéf), hlasovanie o vedúcich na sústreďenie (vedúci/šéf).

2.3 Prípady použitia

Táto kapitola obsahuje zoznam účastníkov, diagramy a detaily prípadov použitia. Diagramy prípadov použitia vychádzajú z aktérov (účastníkov systému) a činností, ktoré vykonávajú. Medzi jednotlivými prípadmi použitia môžu vzniknúť relácie.

Dedičnosť medzi účastníkmi systému znamená, že potomok dedí všetky priradené prípady použitia svojho predka. Samozrejme, potomok môže mať navyše priradené ďalšie prípady použitia systému.

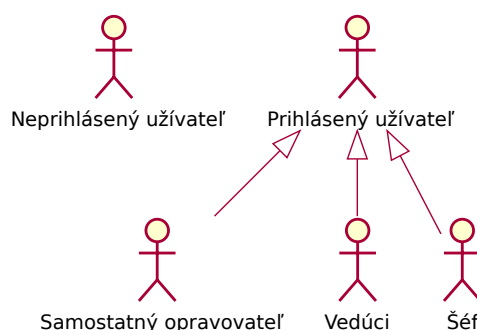
Použitie relácie «*include*» znamená, že správanie dodávateľského prípadu použitia je zahrnuté v klientskom prípade použitia [2]. Použitie tejto relácie ušetrí opakovanie rovnakých krokov, ktoré sú už popísané v inom prípade použitia.

Relácia «*extends*» slúži na rozšírenie prípadu použitia o nové správanie [2]. Tento rozširujúci prípad môže, ale nemusí nastať v každom prípade vykonávania rozširovaného prípadu použitia.

2.3.1 Zoznam účastníkov

Organizátori seminára preferujú otvorenosť systému pre všetkých organizátorov. Nie je veľa funkcií systému, ktoré by mali spravovať iba určité skupiny

organizátorov. Zoznam účastníkov systému vyplýva z bežného fungovania organizácie a jej rolí (obrázok 2.7). Neprihlásený užívateľ sa vie v systéme iba prihlásiť. Prihlásený užívateľ má sprístupnenú väčšinu funkcií systému. Niekoľko vybraných funkcií vie ovládať iba vedúci, šéf alebo samostatný opravovateľ.



Obr. 2.7: Zoznam účastníkov

2.3.2 Scenáre prípadov použitia

2.3.2.1 Prihlasovanie do systému

Základným krokom v systéme bude prihlásenie. Tým sa v systéme sprístupní väčšina funkcií. Diagram je zobrazený na obrázku 2.8.

1. *Prihlásiť sa do systému*

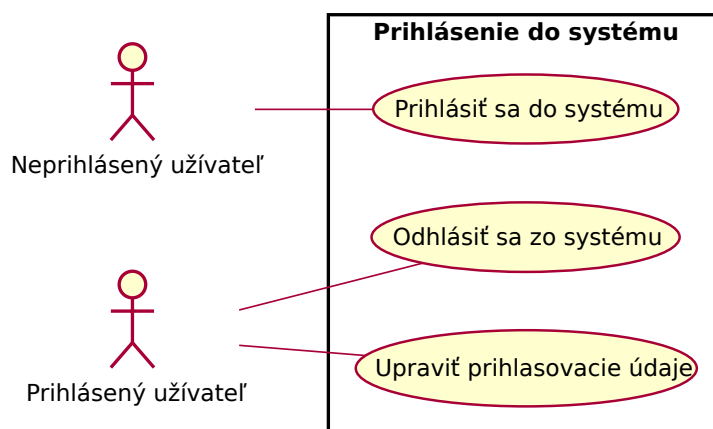
Prípad začína vo chvíli, keď sa neprihlásený užívateľ rozhodne prihlásiť do systému.

1. Systém zobrazí formulár na zadanie používateľského mena a hesla.
2. Užívateľ vyplní údaje.
3. Systém overí správnosť údajov.
4. Ak sú zadané údaje správne, užívateľ je v stave „*prihlásený*“.

2. *Odhlásiť sa zo systému*

Prípad začína vo chvíli, keď sa užívateľ rozhodne odhlásiť zo systému.

1. Užívateľ klikne na odkaz odhlásenia zo systému.
2. Systém sa opýta, či sa užívateľ naozaj chce odhlásiť zo systému.
3. Užívateľ potvrdí odhlásenie.
4. Systém užívateľa odhlási.



Obr. 2.8: UC: Prihlasovanie do systému

3. Upraviť prihlasovacie údaje

Prípád začína vo chvíli, keď sa užívateľ rozhodne zmeniť svoje prihlasovacie údaje.

1. Systém zobrazí formulár s heslom a jeho potvrdením.
2. Prihlásený užívateľ vyplní nové heslo minimálne dĺžky 8 a kvôli potvrdeniu ho napíše znova.
3. Systém skontroluje zhodnosť zadaných hesiel a ich dĺžku. Ak nenastane chyba, heslo je uložené.

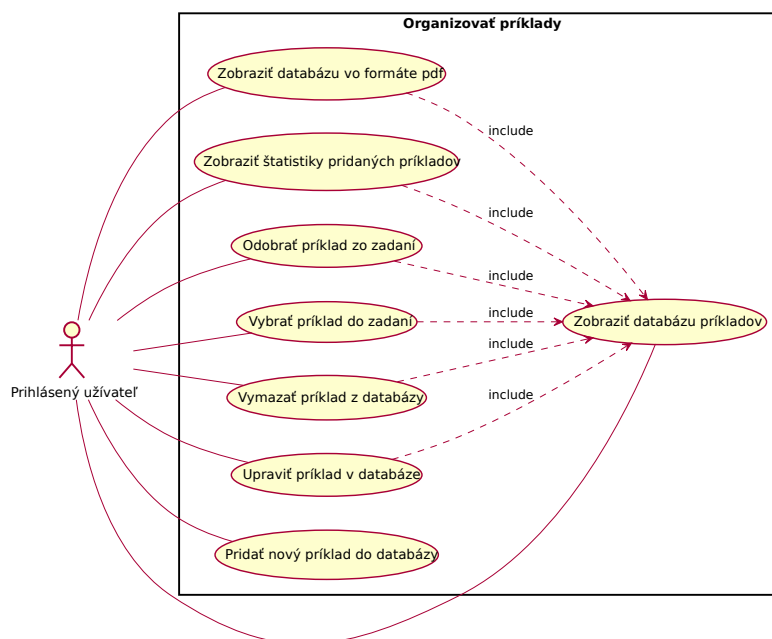
2.3.2.2 Organizovať príklady

Prípady začínajú vo chvíli, keď sa prihlásený užívateľ rozhodne pre organizáciu príkladov. Diagram je zobrazený na obrázku 2.9.

4. Pridať nový príklad do databázy

Prípád začína vo chvíli, keď sa prihlásený užívateľ rozhodne pridať nový príklad.

1. Prihlásený užívateľ zobrazuje stránku vkladania nového príkladu.
2. Systém zobrazí formulár, ktorý obsahuje pole na názov príkladu, text, voľbu kategórie, nahrávanie obrázkov a voľbu autora príkladu.
3. Užívateľ vyplní údaje a vyberie obrázky v počítači. Voľby potvrdí.
4. Systém príklad a obrázky uloží.



Obr. 2.9: UC: Príklady

5. Upraviť príklad v databáze

Prípád začína vo chvíli, keď sa prihlásený užívateľ rozhodne upraviť konkrétny príklad.

1. Prihlásený užívateľ zobrazí úpravu príkladu.
2. Systém zobrazí formulár, ktorý obsahuje pole na názov príkladu, text, voľbu kategórie a voľbu autora príkladu s predvyplnenými uloženými údajmi a pridané obrázky.
3. Užívateľ zmení názov, text alebo kategóriu príkladu.
4. Užívateľ môže pridať alebo odobrať obrázky k príkladu a voľby potvrdí.
5. Systém príklad zmení a upraví obrázky.

6. Vymazať príklad z databázy

Prípád začína vo chvíli, keď sa prihlásený užívateľ rozhodne vymazať konkrétny príklad z databázy.

1. Prihlásený užívateľ vyberá konkrétny príklad a volí vymazanie príkladu.
2. Systém vymaže príklad a obrázky z databázy.

7. Vybrať príklad do zadania

1. Prihlásený užívateľ zvolí výber konkrétneho príkladu do zadania.
2. Systém zvolí ponuku na výber kategórie, ročníka, série, kola a čísla príkladu.
3. Užívateľ vyplní potrebné údaje.
4. Systém voľbu uloží.

8. Odobrať príklad zo zadania

1. Prihlásený užívateľ zvolí výber konkrétneho príkladu, ktorý je zvolený zadaniach.
2. Systém umožní zrušiť výber príkladu v zadaniach.
3. Užívateľ zrušenie potvrdí.
4. Systém voľbu uloží.

9. Zobrazíť databázu príkladov

1. Prihlásený užívateľ zvolí zobrazenie príkladovej databázy.
2. Systém zobrazí všetky pridané príklady s možnosťou vyhľadávania a voľbou kategórie a stavu.
3. Užívateľ vyberá stav a kategóriu príkladov a zadá výraz, podľa ktorého budú príklady vyhladané.
4. Systém zobrazí výsledky prislúchajúce zadaným podmienkam.

10. Zobrazíť databázu vo formáte PDF

1. Systém zobrazí databázu príkladov
2. Prihlásený užívateľ zvolí požadované kategórie a stavy príkladov.
3. Užívateľ volí generovanie príkladovej databázy.
4. Systém spustí kompiláciu do formátu PDF. Zobrazí odkaz na stiahnutie a vymazanie súboru.
5. Užívateľ stiahne súbor zo zobrazeného odkazu.

2. ANALÝZA

11. Zobrazíť štatistiky pridaných príkladov

Prípád začína vo chvíli, keď sa prihlásený užívateľ rozhodne zobrazíť štatistiky pridaných príkladov. Systém generuje formulár na zadanie rozpätia dátumov. Hlavný scenár:

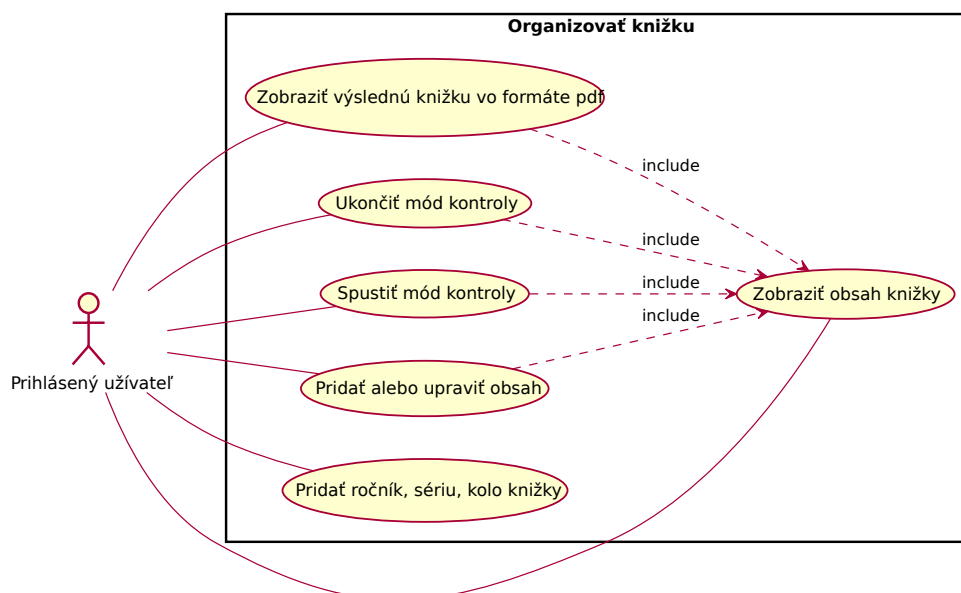
1. Prihlásený užívateľ zvolí dátumy od a do, a formulár odošle.
2. Systém zobrazí zoznam užívateľov a k nim prislúchajúce počty pridaných príkladov za dané obdobie.

Alternatívny scenár:

1. Prihlásený užívateľ nezvolí dátumy od a do.
2. Systém zobrazí zoznam užívateľov a k nim prislúchajúce počty všetkých pridaných príkladov.

2.3.2.3 Organizovať knižku seminára

Prípady začínajú vo chvíli, keď sa prihlásený užívateľ rozhodne pre organizáciu knižky. Diagram je zobrazený na obrázku 2.10.



Obr. 2.10: UC: Knižka

12. Pridať ročník, sériu, kolo knižky

1. Prihlásený užívateľ zobrazí doterajšie ročníky, série či kolá a zvolí pridanie nového.
2. Systém zobrazí formulár na pridanie ročníka, série, či kola.
3. Užívateľ vyplní číslo nového ročníka a rok, názov série či číslo kola.
4. Systém voľby uloží a zobrazí obsah nového ročníka, série či kola.

13. Zobrazíť obsah knižky

1. Prihlásený užívateľ vyberie ročník a sériu.
2. Systém zobrazí zoznam knižiek v danej sérii.
3. Užívateľ vyberá konkrétnu knižku.
4. Systém zobrazí doteraz pridaný obsah knižky.

14. Pridať alebo upraviť obsah

1. Prihlásený užívateľ zvolí ročník, sériu a kolo.
2. Systém zobrazí voľbu obsahu (napríklad vzorové riešenie, zadania, úvod a pod.).
3. Užívateľ zvolí požadovaný obsah knižky.
4. Systém zobrazuje formulár s predvyplneným doteraz uloženým textom a výber súborov z počítača.
5. Užívateľ upraví obsah a vyberá obrázky k danému obsahu z počítača.
6. Systém obsah uloží.

15. Spustiť mód kontroly obsahu

1. Prihlásený užívateľ zobrazí ročník, sériu a kolo.
2. Zvolí spustenie módu kontroly daného obsahu.
3. Systém uloží tento stav a zobrazuje upozornenie na kontrolu obsahu.

2. ANALÝZA

16. Ukončiť mód kontroly obsahu

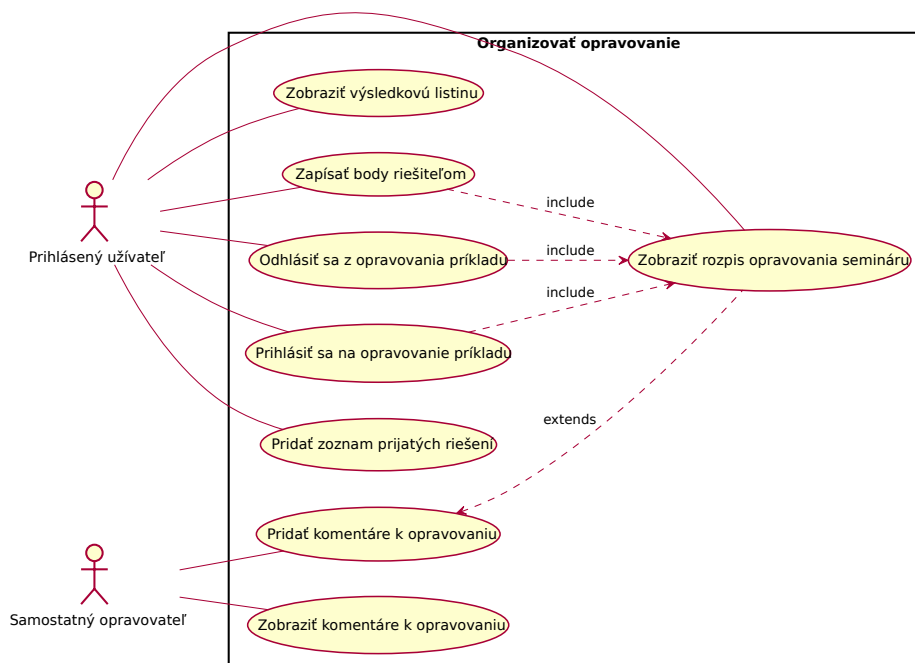
1. Prihlásený užívateľ zobrazí ročník, sériu a kolo.
2. Zvolí ukončenie módu kontroly daného obsahu.
3. Systém voľbu uloží

17. Zobrazíť výslednú knižku vo formáte PDF

1. Prihlásený užívateľ zvolí ročník, sériu a kolo knihy.
2. Zvolí generovanie vo formáte PDF.
3. Systém kompiluje vstupné súbory a generuje z nich súbor vo formáte PDF. Zobrazí odkaz na tento súbor.
4. Užívateľ sťahuje vygenerovaný súbor z odkazu.

2.3.2.4 Organizovať opravovanie

Prípady začínajú vo chvíli, keď sa prihlásený užívateľ rozhodne pre organizáciu opravovania riešení riešiteľov. Diagram je zobrazený na obrázku 2.11.



Obr. 2.11: UC: Opravovanie

18. Zobrazíť rozpis opravovania seminára

1. Prihlásený užívateľ vyberá zobrazenie rozpisu opravovania.
2. Systém zobrazí zoznam príkladov a k nim prihlásených organizátorov.

19. Prihlásiť sa na opravovanie príkladu

1. Prihlásený užívateľ zobrazí rozpis opravovania a vyberá príklad, ktorý chce opravovať.
2. Systém vloží užívateľa k danému opravovaniu príkladu.

20. Odhlásiť sa z opravovania príkladu

1. Prihlásený užívateľ vyberá rozpis opravovania.
2. Systém umožní odobratie užívateľa z opravovania príkladov.
3. Užívateľ potvrdí odobranie užívateľa z opravovania.
4. Systém vymaže užívateľa z opravovania daného príkladu.

21. Pridať zoznam prijatých riešení

1. Prihlásený užívateľ vyberá ročník, sériu a kolo a zvolí pridanie zoznamu prijatých riešení.
2. Systém zobrazí zoznam riešiteľov v danej sérii a polia na zaškrtavanie, ktoré reprezentujú príklady (1 až 9 a prémie).
3. Užívateľ zaškrtnie zaškrtavacie pole, ak riešiteľ s daným menom poslal daný príklad.
4. Systém voľby uloží.

22. Zapísať body riešiteľom

1. Užívateľ zobrazí rozpis opravovania a zvolí číslo príkladu, v ktorom chce zadať body.
2. Systém zobrazí mená riešiteľov, ktorí poslali daný príklad v danom kole a k nim pole na vyplnenie počtu bodov.
3. Užívateľ vyplní body.
4. Systém body uloží.

23. Zobrazíť výsledkovú listinu

1. Užívateľ zobrazí kolo a vyberie zobrazenie výsledkovej listiny.
2. Systém zobrazí výsledky riešiteľov zoradené podľa priezviska.

24. Pridať komentáre k opravovaniu

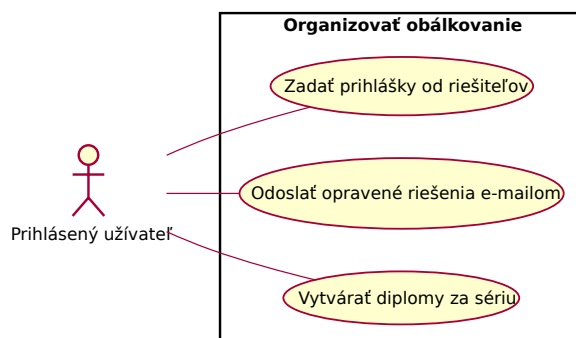
1. Samostatný opravovateľ môže zobrazíť rozpis opravovania a z neho zvolí pridanie komentárov k opravovaniu, alebo zobrazí všetky opravovania, na ktorých sa podieľal a vyberie konkrétne, v ktorom zvolí pridávanie komentárov.
2. Systém zobrazí formulár na pridanie komentárov pre každého z nesamostatných opravovateľov zúčastnených na danom opravovaní.
3. Samostatný opravovateľ vyplní komentáre.
4. Systém komentáre uloží.

25. Zobrazíť komentáre k opravovaniu

1. Samostatný opravovateľ zvolí zobrazenie opravovaní a v ňom vyberá zobrazenie komentárov k nesamostatným opravovateľom.
2. Systém zobrazí výber z nesamostatných opravovateľov.
3. Samostatný opravovateľ vyberie opravovateľov.
4. Systém zobrazí opravovania pre vybraných opravovateľov generované z rozpisov opravovaní a k nim pridané komentáre.

2.3.2.5 Organizovať obáľkovanie

Prípady začínajú vo chvíli, keď sa prihlásený užívateľ rozhodne pre organizáciu obáľkovania. Diagram je zobrazený na obrázku 2.12.



Obr. 2.12: UC: Obáľkovanie

26. Zadať prihlášky od riešiteľov

1. Prihlásený užívateľ zvolí ročník a sériu a v nej zadávanie prihlášok riešiteľov do systému.
2. Systém zobrazí pole na vyplnenie počtu prihlášok, ktorý chce užívateľ pridať.
3. Užívateľ zvolí požadovaný počet prihlášok.
4. Systém vygeneruje formulár na pridávanie požadovaného počtu prihlášok.
5. Užívateľ vyplní údaje riešiteľov z papierovej prihlášky.
6. Systém údaje uloží do databázy.

27. Odoslať opravené riešenia e-mailom

1. Prihlásený užívateľ zvolí odosielanie elektronických riešení.
2. Systém generuje zoznam užívateľov, ktorí v danom kole poslali riešenia a formulár s telom e-mailu.
3. Užívateľ vkladá názov a text e-mailu. Pre každého riešiteľa vyberá súbory s jeho riešeniami a ďalšími súbormi.
4. Zvolí odoslanie e-mailov.
5. Systém odošle e-maily na adresy zadané v databáze riešiteľov pre vybraných riešiteľov so zadaným telom e-mailu a prílohami.

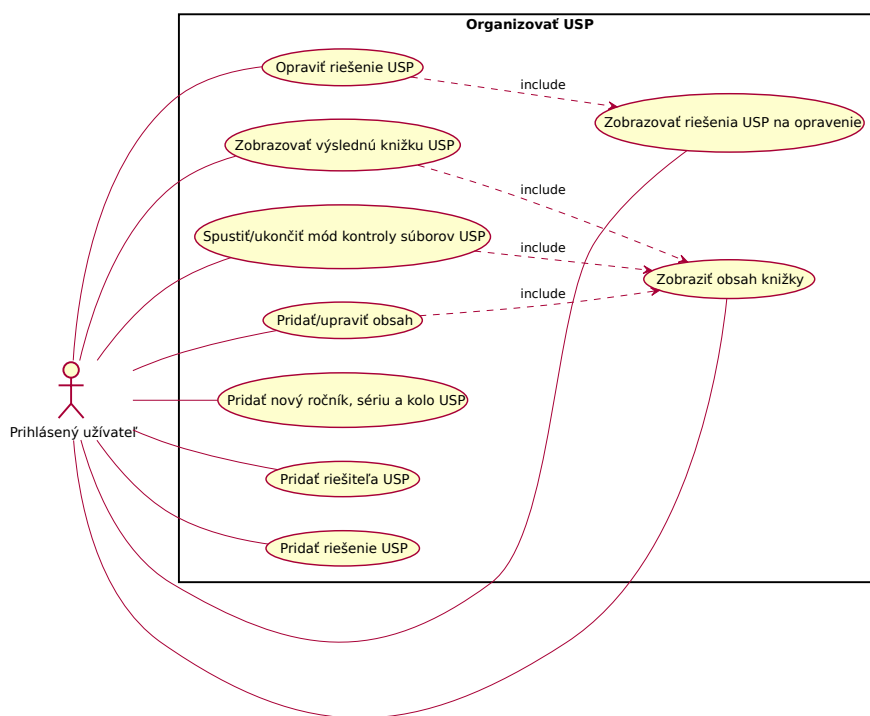
2. ANALÝZA

28. Vytvárať diplomy za sériu

1. Prihlásený užívateľ zvolí vytváranie diplomov za sériu.
2. Systém umožní výber šablóny diplomu a počtu riešiteľov, pre ktorých sa bude diplom generovať.
3. Užívateľ vyberie šablónu alebo vytvorí novú a zvolí počet riešiteľov.
4. Systém generuje zvolený počet diplomov, meno, umiestnenie a body vyplní podľa výsledkovej listiny na určené pozície.
5. Užívateľ môže výsledné diplomy stiahnuť.

2.3.2.6 Organizovať USP

Prípady začínajú vo chvíli, keď sa prihlásený užívateľ rozhodne pre organizáciu USP. Diagram je zobrazený na obrázku 2.13.



Obr. 2.13: UC: USP

29. Pridať nový ročník, sériu a kolo USP

Postup je rovnaký s pridávaním nového ročníka, série a kola seminára.

30. Zobrazit obsah knižky

Postup je rovnaký so zobrazením obsahu knižky seminára.

31. Pridat/upraviť obsah

Postup je rovnaký s pridávaním alebo úpravou obsahu knižky seminára.

32. Spustiť/ukončiť mód kontroly súborov USP

Postup je rovnaký so spustením/ukončením módu kontroly obsahu knižky seminára.

33. Zobrazovať výslednú knižku USP

Postup je rovnaký so zobrazovaním výslednej knižky seminára.

34. Pridat riešiteľa USP

Postup je rovnaký s pridávaním riešiteľa seminára.

35. Pridat riešenie USP

1. Prihlásený užívateľ zvolí pridávanie riešení USP.
2. Systém umožní výber riešiteľa a nahrávanie súborov.
3. Užívateľ vyberie súbory z počítača.
4. Systém výber uloží.

36. Zobrazovať riešenia USP na opravenie

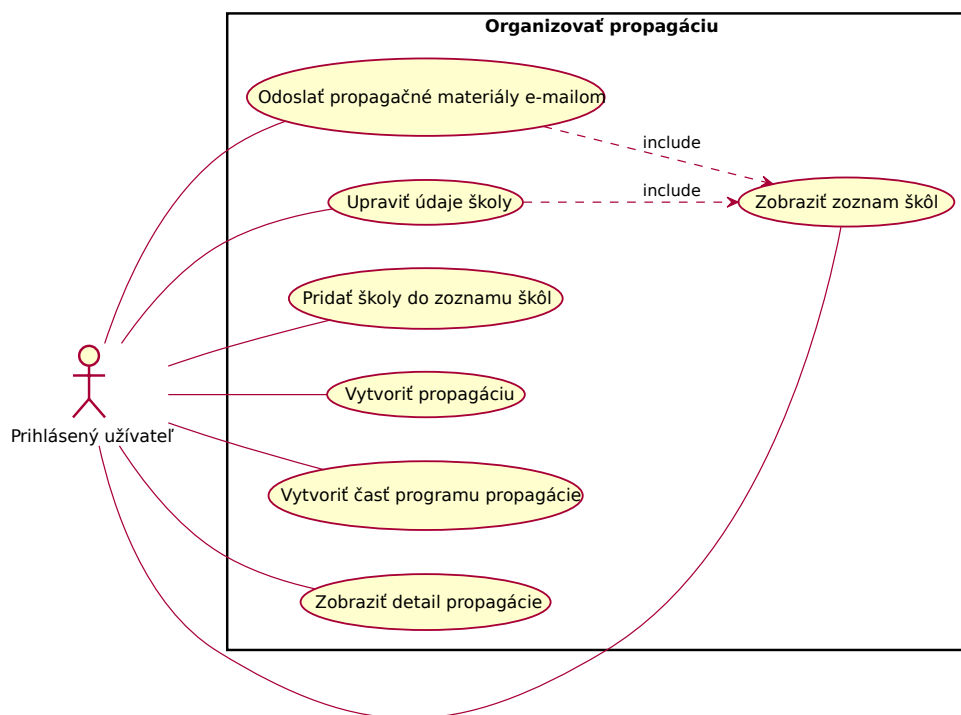
1. Prihlásený užívateľ zobrazuje riešenia od riešiteľov na opravenie.
2. Systém zobrazí riešenia, ktoré ešte neboli opravené.

37. Opraviť riešenie USP

1. Prihlásený užívateľ si zvolí jedno z riešení.
2. Systém zobrazí dané riešenie.
3. Užívateľ riešenie prezrie a zadá počet bodov a komentár.
4. Systém odošle e-mail na adresu riešiteľa, ktorý oznámi, že riešenie daného príkladu bolo ohodnotené na zadaný počet bodov s daným komentárom.

2.3.2.7 Organizovať propagáciu

Prípady začínajú vo chvíli, keď sa prihlásený užívateľ rozhodne pre organizáciu propagácie. Diagram je zobrazený na obrázku 2.14.



Obr. 2.14: UC: Propagácia

38. Pridať školy do zoznamu škôl

1. Prihlásený užívateľ sa rozhodne pridať novú školu do zoznamu.
2. Systém generuje formulár na pridávanie novej školy.
3. Užívateľ zadá názov školy, kontakt a poznámky a určí, či je škola základná/gymnázium.
4. Systém údaje uloží.

39. Zobraziť zoznam škôl

1. Prihlásený užívateľ sa rozhodne zobraziť zoznam pridaných škôl.
2. Systém jednotlivé školy zobrazí.

40. Upraviť údaje školy

1. Prihlásený užívateľ zobrazí zoznam škôl a vyberie si jednu, ktorú sa rozhodne upraviť.
2. Systém zobrazí formulár na vyplnenie údajov o škole s predvyplnenými doteraz uloženými údajmi.
3. Užívateľ upraví názov školy, kontakt a poznámky a určí, či je škola základná/gymnázium.
4. Systém údaje uloží.

41. Vytvoriť propagáciu

1. Prihlásený užívateľ zvolí vytvorenie propagácie.
2. Systém zobrazí formulár na vyplnenie dátumu začiatku a konca propagácie, popisu a nahrávanie súborov.
3. Užívateľ vyplní údaje a nahrá súbory.
4. Systém údaje uloží.

42. Vytvoriť časť programu propagácie

1. Prihlásený užívateľ zvolí vytvorenie programu propagácie.
2. Systém zobrazí formulár na vyplnenie názvu časti, začiatku a konca, popisu, školy, zodpovedných osôb a nahrávania súborov.
3. Užívateľ vyplní údaje a nahrá súbory.
4. Systém údaje uloží.

43. Zobrazíť detail propagácie

1. Prihlásený užívateľ zvolí zobrazenie detailu propagácie.
2. Systém zobrazí program propagácie a ďalšie zadané informácie o propagácii.

2. ANALÝZA

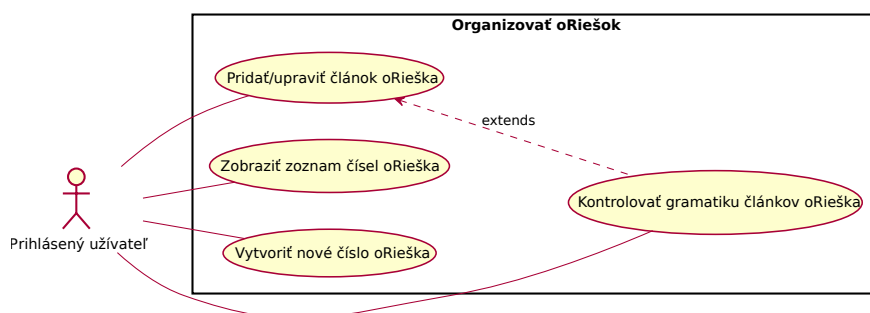
44. *Odoslať propagačné materiály e-mailom*

Prípád začína, keď chce prihlásený užívateľ odoslať propagačné materiály formou e-mailu na vybrané školy.

1. Systém zobrazí zoznam uložených škôl.
2. Užívateľ zvolí školy zo zoznamu škôl.
3. Vyplní hlavičku e-mailu a telo správy.
4. Nahrá materiály, ktoré sa majú odoslať.
5. Zvolí odoslanie správ.
6. Systém odošle e-maily na vybrané školy so zadanou hlavičkou, telom a prílohou.

2.3.2.8 Organizácia oRieška

Prípady začínajú vo chvíli, keď sa prihlásený užívateľ rozhodne pre organizáciu časopisu oRiešok. Diagram je zobrazený na obrázku 2.15.



Obr. 2.15: UC: oRiešok

45. *Zobraziť zoznam čísel oRieška*

1. Prihlásený užívateľ zvolí ročník.
2. Systém zobrazí zoznam čísel časopisu pridaných v danom ročníku.

46. *Vytvoriť nové číslo oRieška*

1. Prihlásený užívateľ zvolí ročník a zvolí pridanie nového čísla.
2. Systém vytvorí nové číslo časopisu.

47. Pridať/upraviť článok oRieška

1. Prihlásený užívateľ zvolí prídanie/úpravu článku oRieška.
2. Systém zobrazí formulár na vyplnenie názvu, textu a autora článku.
3. Užívateľ údaje vyplní.
4. Systém údaje uloží.

48. Kontrolovať gramatiku článkov oRieška

1. Prihlásený užívateľ zvolí ročník a číslo oRieška.
2. Systém zobrazí názvy článkov a autorov.
3. Užívateľ volí zobrazenie konkrétnych článkov.
4. Systém zobrazí vybrané články.
5. Užívateľ môže upraviť text článku a označiť gramatiku za skontrolovanú.
6. Systém údaje uloží.

2.3.2.9 Organizovať akcie

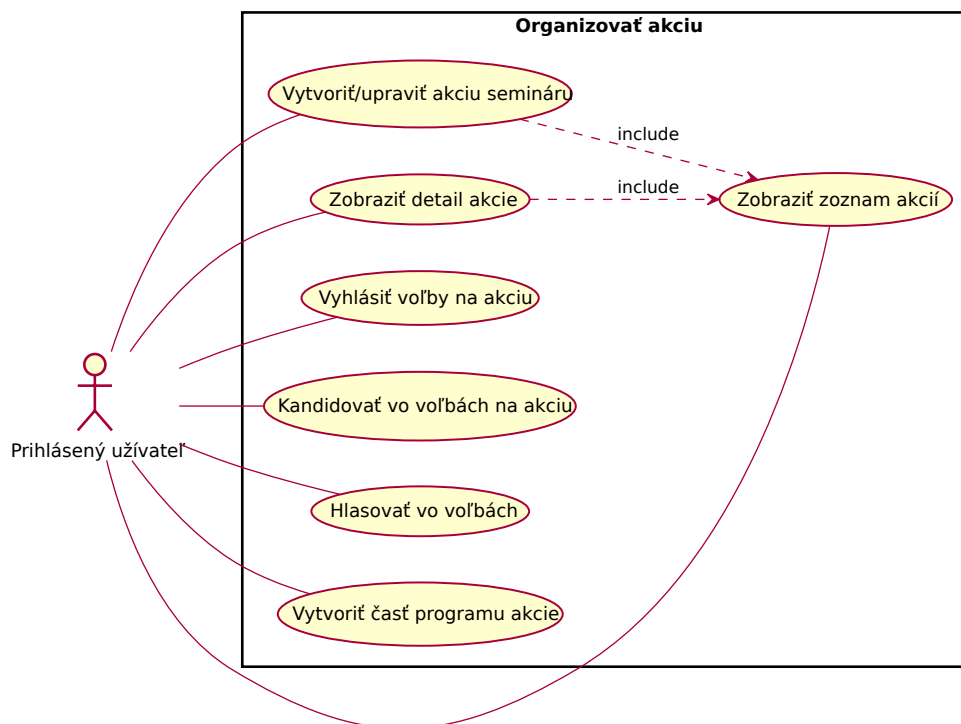
Prípady začínajú vo chvíli, keď sa prihlásený užívateľ rozhodne pre organizáciu akcie seminára. Diagram je zobrazený na obrázku 2.16.

49. Zobrazíť zoznam akcií

1. Prihlásený užívateľ zvolí ročník.
2. Systém zobrazí zoznam akcií priradených k vybranému ročníku.

50. Vytvoriť/upraviť akciu seminára

1. Prihlásený užívateľ vyberie ročník a zvolí vytvorenie novej akcie.
2. Systém zobrazuje formulár na zadanie názvu akcie, trvania od-do, popisu, zoznamu zodpovedných organizátorov, viditeľnosť pre ostatných organizátorov a výber súborov.
3. Užívateľ údaje vyplní.
4. Systém údaje uloží.



Obr. 2.16: UC: Akcia

51. Vyhlásiť voľby na akciu

1. Prihlásený užívateľ zvolí vyhlásenie volieb na akciu.
2. Systém zobrazí formulár na zadanie dátumu prihlasovania kandidátov, dátum hlasovania o kandidátoch, zadanie skupín alebo jednotlivcov, ktorí môžu kandidovať a zadanie skupín alebo jednotlivcov, ktorí môžu hlasovať.
3. Užívateľ vyplní údaje a zvolí vyhlásenie volieb.
4. Systém vytvorí nové hlasovanie pre zadané skupiny a jednotlivcov a umožní kandidovať vo voľbách.
5. Po uplynutí termínu prihlasovania a hlasovania zobrazí výsledky hlasovania.

52. Kandidovať vo voľbách na akciu

1. Prihlásený užívateľ zvolí voľby v konkrétnej akcii.
2. Systém kontroluje, či je daný užívateľ v zozname prípustných kandidátov. Ak áno, zobrazuje možnosť kandidovať vo voľbách.
3. Užívateľ zvolí kandidovanie vo voľbách.
4. Systém overí, že užívateľ je v zozname prípustných kandidátov a že sa prihlásil v zadanom termíne a kandidatúru uloží.

53. Hlasovať vo voľbách

1. Prihlásený užívateľ zvolí voľby v konkrétnej akcii.
2. Systém vypíše zoznam prihlásených kandidátov. Ak je užívateľ v zozname prípustných hlasujúcich, zobrazí formulár na zoradenie kandidátov.
3. Užívateľ zoradí kandidátov od najlepšieho (najviac preferovaného) po najhoršieho (najmenej preferovaného) a svoj hlas potvrdí.
4. Systém overí, že užívateľ je v zozname prípustných hlasujúcich, že hlasuje v zadanom termíne a že zadaný hlas je validný a hlas uloží.

54. Vytvoriť časť programu akcie

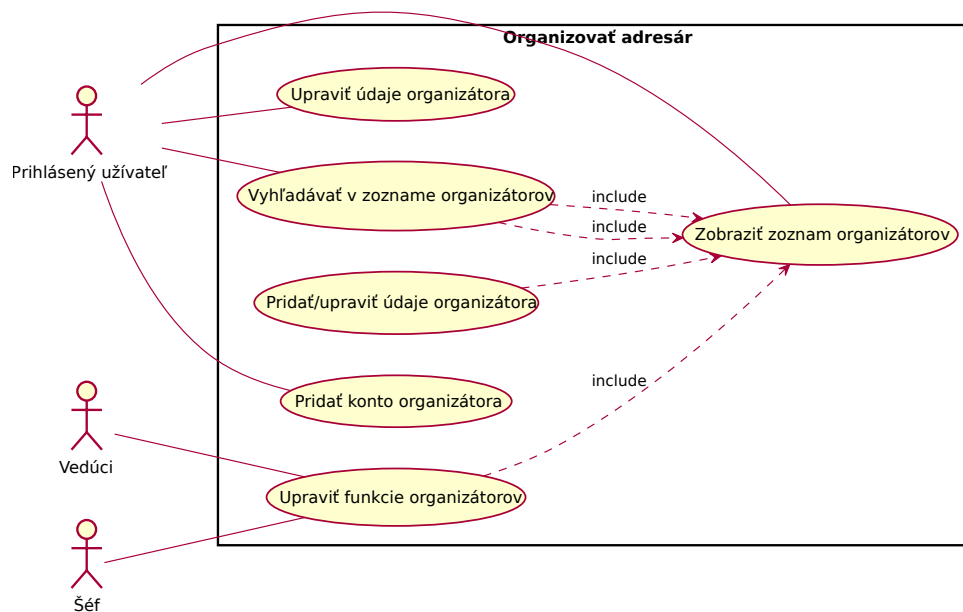
1. Prihlásený užívateľ zvolí vytvorenie programu akcie.
2. Systém zobrazí formulár na vyplnenie názvu časti, dátum začiatku a konca, popisu, zodpovedných osôb a nahrávania súborov.
3. Užívateľ vyplní údaje a nahrá súbory.
4. Systém údaje uloží.

55. Zobrazí detail akcie

1. Prihlásený užívateľ zvolí zobrazenie detailu akcie.
2. Systém zobrazí program akcie a ďalšie zadané informácie.

2.3.2.10 Spravovať adresár organizátorov

Prípady začínajú vo chvíli, keď sa prihlásený užívateľ rozhodne pre spravovanie adresára organizátorov. Diagram je zobrazený na obrázku 2.17.



Obr. 2.17: UC: Adresár

56. *Zobraziť zoznam organizátorov*

1. Prihlásený užívateľ zvolí zobrazenie adresára.
2. Systém zobrazuje údaje o organizátoroch.

57. *Pridať konto organizátora*

1. Prihlásený užívateľ zvolí pridanie konta organizátora.
2. Systém zobrazí formulár na zadanie prihlasovacieho mena a e-mailu organizátora.
3. Užívateľ vyplní prihlasovacie meno a e-mail.
4. Ak je prihlasovacie meno unikátne, systém odošle prihlasovacie údaje (meno a vygenerované heslo) na zadaný e-mail a pridá nového organizátora.

58. Upraviť údaje organizátora

1. Prihlásený užívateľ zvolí pridanie nového organizátora.
2. Systém zobrazí formulár na zadanie mena, priezviska, prezývky, pohlavia, dátumu narodenia, adresy, PSČ, mesta, kontaktného e-mailu, mobilu, telefónu, icq, skype, poznámok, mailu do mailing listov, mailing listu, pôsobiska.
3. Užívateľ údaje vyplní.
4. Systém údaje uloží.

59. Upraviť funkcie organizátorov

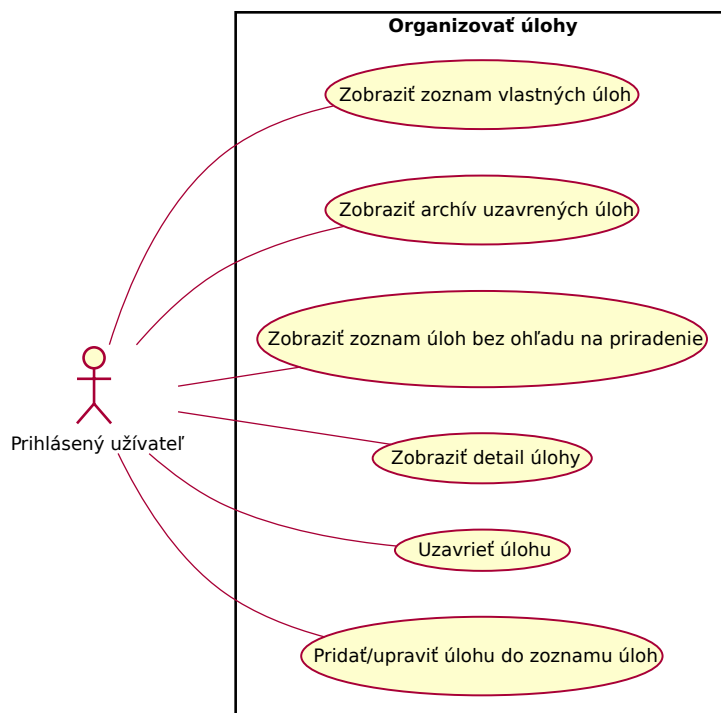
1. Vedúci alebo šéf zobrazí úpravu funkcií konkrétneho organizátora.
2. Systém zobrazí pre vybraného organizátora jeho priradenú hlavnú funkciu, vedľajšie funkcie a či je samostatný opravovateľ.
3. Užívateľ upraví vybrané funkcie.
4. Systém údaje uloží.

60. Vyhľadávať v zozname organizátorov

1. Prihlásený užívateľ zobrazí zoznam organizátorov.
2. Systém zobrazí formulár na zadanie kľúčového slova, podľa ktorého sa bude vyhľadávať a požadované zobrazenie stĺpcov.
3. Užívateľ zadá kľúčové slovo a vyberá stĺpce, ktoré chce vidieť.
4. Systém zobrazí výsledky z údajov organizátorov, v ktorých sa kľúčové slovo vyskytuje.

2.3.2.11 Organizácia úloh

Prípady začínajú vo chvíli, keď sa prihlásený užívateľ rozhodne pre spravovanie úloh organizátorov. Diagram je zobrazený na obrázku 2.18.



Obr. 2.18: UC: Úlohy

61. Zobraziť zoznam vlastných úloh

1. Prihlásený užívateľ zvolí zobrazenie zoznamu úloh.
2. Systém zobrazí úlohy, ktoré nie sú archivované a sú zaradené do osobných (priradené priamo tomuto užívateľovi) a skupinových (priradené skupine, v ktorej sa užívateľ nachádza).

62. Zobraziť archív uzavrených úloh

1. Užívateľ vyberie zobrazenie archívu úloh.
2. Systém zobrazuje archivované úlohy.

63. Zobraziť zoznam úloh bez ohľadu na priradenie

1. Prihlásený užívateľ zvolí zobrazenie zoznamu úloh.
2. Systém zobrazí úlohy, ktoré nie sú archivované.

64. Zobrazí detail úlohy

1. Prihlásený užívateľ vyberá zo zobrazeného zoznamu úloh konkrétnu úlohu.
2. Systém zobrazí detail zadaných údajov.

65. Uzavrieť úlohu

1. Užívateľ vyberie jednu z aktuálnych úloh.
2. Systém zobrazí možnosť archivovať úlohu.
3. Užívateľ zvolí archivovanie úlohy.
4. Systém úlohu vloží do archívu.

66. Pridať/upraviť úlohu do zoznamu úloh

1. Prihlásený užívateľ zvolí prídanie/úpravu úlohy.
2. Systém zobrazí formulár na zadanie názvu, popisu, dátumu konca úlohy, priradenie organizátorom, priradenie skupinám organizátorov a výber súborov.
3. Užívateľ údaje vyplní.
4. Systém úlohu uloží.

2.4 Analytické triedy

Z popísanej domény vyplynuli analytické triedy, ich základné atribúty a metódy. Analytický model tried je zobrazený na obrázku 2.19.

2.4.1 Popis analytických tried

Kapitola vysvetľuje význam jednotlivých analytických tried modelu, ich atribúty, metódy a vzťahy k ostatným triedam.

1. Trieda *Organizator*

Trieda reprezentuje jedného organizátora súťaže.

Atribúty:

- Základné údaje organizátora: meno, priezvisko, pohlavie, datumNarodenia, adresa, psc, mesto, kontaktnyEmail, mobilneCislo, telefonneCislo, icq, skype, poznamky.
- Prezývka: nickname.
- Príznak, či je organizátor samostatným opravovateľom: samostatnyOpravovatel.
- Mail, ktorý organizátor používa do mailing listov: mailDoMailingListov.

Metódy:

- upravFunkcie(): Uprav všetky funkcie organizátora (hlavnú a vedľajšie).
- zobrazZoznamUloh(): Zobraz zoznam úloh priradených organizátorovi.

2. Trieda *Adresar* «*singleton*»

Trieda reprezentuje adresár, teda údaje všetkých organizátorov.

Metódy:

- zobrazAdresar(): Zobraz údaje všetkých organizátorov.
- vyhladajOrganizatora(): Vyhľadaj organizátora podľa zadaného kľúča.

3. Trieda *HlavnaFunkcia*

Trieda reprezentuje hlavnú funkciu organizátora (popísané v kapitole 2.1.2.1).

Atribúty:

- nazovFunkcie

4. Trieda *VedlajsiaFunkcia*

Trieda reprezentuje vedľajšie funkcie organizátora (popísané v kapitole 2.1.2.3).

Atribúty:

- nazovFunkcie

2. ANALÝZA

5. Trieda *MailingList*

Trieda reprezentuje priradenie e-mailu jedného organizátora v jednom mailing liste.

Atribúty:

- `nazov`, `eMail`

6. Trieda *Skola*

Trieda reprezentuje jednu školu. Spojením školy s organizátorom sa myslí pôsobisko organizátora na danej škole. Logicky má teda organizátor 1 školu, môže však existovať organizátor, ktorý už do školy nechodí.

Atribúty:

- Údaje školy: `nazovSkoly`, `adresaSkoly`, `eMail`, `mobilneCislo`, `poznamky`.
- Skrátený názov slúži napríklad pre potreby výsledkovej listiny: `skratenyNazov`.
- Príznak, či je daná škola základnou školou alebo gymnáziom (táto škola bude ponúknutá na výber napríklad pri výbere školy riešiteľa): `jeZS`.

7. Trieda *Uloha*

Trieda reprezentuje jednu zadanú úlohu pre organizátorov. Môže byť zadaná jednotlivcom alebo skupinám organizátorov.

Atribúty:

- Vlastnosti úlohy: `nazov`, `popis`, `stav`.
- Úloha môže mať nastavený konečný termín vykonania: `deadline`.

Metódy:

- `pridajSubory()`: Pridá súbory k danej úlohe.

8. Trieda *SkupinaOrganizatorov*

Trieda reprezentuje skupinu organizátorov, ktorým môže byť priradená úloha.

Atribúty:

- `nazovSkupiny`

9. Trieda *Rocnik*

Trieda reprezentuje jeden ročník súťaže.

Atribúty:

- Poradové číslo ročníka súťaže: `cisloRocnika`.
- Začiatok školského roku: `rok`.

Metódy:

- `zobrazObsah()`: Zobrazí obsah priradený tomuto ročníku.

10. Trieda *Udalost* «*abstract*»

Abstraktná trieda reprezentuje jednu udalosť v ročníku.

Atribúty:

- Vlastnosti udalosti: `nazov`, `popis`, `datumOd`, `datumDo`.

Metódy:

- `pridajSubory()`: Pridá súbory k tejto udalosti.
- `vytvorProgram()`: Vytvorenie programu udalosti.

11. Trieda *Propagacia*

Trieda reprezentuje jednu udalosť v ročníku – propagáciu. Propagácia prebieha na určitých školách.

12. Trieda *Akcia*

Trieda reprezentuje jednu udalosť v ročníku – akciu. Pod akciou sa môže myslieť napríklad sústredenie, denný tábor, letný tábor a podobne.

Metódy:

- `vyhlasVolby()`: Vytvorenie volieb na danú akciu.

13. Trieda *CastProgramu*

Trieda reprezentuje jednu z častí programu udalosti. Môže mať priradených zodpovedných organizátorov.

Atribúty:

- Vlastnosti časti programu: `nazov`, `popis`, `casOd`, `casDo`.

2. ANALÝZA

14. Trieda *Volby*

Trieda reprezentuje voľby k akcii. Zahŕňa organizátorov, ktorí môžu vo voľbách kandidovať (prípustní kandidáti), z nich prihlásených kandidátov a ďalej prípustných hlasujúcich.

Atribúty:

- Časové ohraničenie termínu prihlasovania (kandidatúry) na akciu: `kandidaturaOd`, `kandidaturaDo`.
- Časové ohraničenie termínu hlasovania vo voľbách: `hlasovanieOd`, `hlasovanieDo`.

Metódy:

- `kandiduj()`: Prihlási organizátora do volieb.
- `hlasuj()`: Vytvorí nový hlasovací lístok a uloží hlasovanie organizátora.
- `vyhodnotVolby()`: Vyhodnotí voľby na základe pridaných hlasov.

15. Trieda *Hlas*

Trieda reprezentuje jeden hlas hlasujúceho vo voľbách. Obsahuje zoradených prihlásených organizátorov.

16. Trieda *CisloORieska*

Trieda reprezentuje jedno číslo časopisu oRiešok v danom ročníku.

Atribúty:

- Poradové číslo: `cislo`.

Metódy:

- `zobrazClanky()`: Zobrazí články tohto čísla oRieška.

17. Trieda *ClanokORieska*

Trieda reprezentuje jeden článok v čísle časopisu oRiešok.

Atribúty:

- Vlastnosti článku: `text`, `stav`.

Metódy:

- `oznacZaSkontrolovany()`: Zmení stav článku na skontrolovaný.

18. Trieda *Príklad*

Trieda reprezentuje jedno zadanie príkladu. Príklad môže mať priradeného autora. Je zaradený do určitej kategórie príkladov.

Atribúty:

- Vlastnosti príkladu: `nazovPríkladu`, `text`, `stav`.

Metódy:

- `pridajObrazok()`: Pridá obrázky príkladu.
- `vyberPríkladDoZadani()`: Pridá príklad do vybraných zadanií.
- `odoberPríkladZoZadani()`: Odoberie príklad zo zadanií, v ktorých sa nachádza.
- `zobrazDatabazuPríkladovPDF()`: Vytvorí databázu príkladov vo formáte pdf.
- `zobrazStatistikyPríkladov()`: Zobrazí štatistiky príkladov, teda počet pridaných príkladov organizátormi za zadané obdobie.

19. Trieda *KategóriaPríkladu*

Trieda reprezentuje jednu kategóriu príkladu (popísané v kapitole 2.1.1.1).

Atribúty:

- Názov kategórie: `kategoria`.

20. Trieda *Seria* «*abstract*»

Trieda reprezentuje jednu sériu vrámci ročníka.

Atribúty:

- `nazovSerie`.

Metódy:

- `pridajPrihlasky()`: Pridá prijaté prihlášky riešiteľov.
- `vytvorDiplomy()`: Vytvorí diplomy za sériu pre najlepších riešiteľov.
- `pridajKolo()`: Pridá kolo série.
- `pridajKnizku()`: Pridá knižku v danej sérii.

2. ANALÝZA

21. Trieda *Riesitel*

Trieda reprezentuje jedného riešiteľa. Riešiteľ je žiakom jednej školy.

Atribúty:

- Údaje riešiteľa: meno, priezvisko, adresaDomov, telefonDieta, telefonRodic, datumNarodenia, eMail, poznamky.
- Názov triedy v škole, ktorú riešiteľ navštevuje: `trieda`.
- Kategória, ktorú riešiteľ rieši (vychádza sa z triedy): `kategoria`.
- Príznak, či daný riešiteľ rieši elektronicky: `prijemERieseni`.

22. Trieda *Diplom*

Trieda reprezentuje jeden diplom udelený za umiestnenie sérii.

Atribúty:

- `cestaKSuboru`

23. Trieda *SeriaUSP*

Trieda reprezentuje jednu sériu súťaže v kategórii USP.

24. Trieda *KoloUSP*

Trieda reprezentuje jedno kolo v sérii súťaže v kategórii USP.

Atribúty:

- `cisloKola`

Metódy:

- `pridajRiesenie()`: Pridá riešenie riešiteľa na opravenie.
- `zobrazRieseniaNaOpravenie()`: Zobrazí neopravené riešenia.

25. Trieda *RiesenieUSP*

Trieda reprezentuje jedno riešenie príkladu v kategórii USP poslané riešiteľom.

Atribúty:

- `stav`, `body`, `komentar`

Metódy:

- `pridajBody()`: Pridá riešiteľovi body za dané riešenie.
- `odosliKomentar()`: Odošle riešiteľovi komentár a body e-mailom.

*26. Trieda **SeriaSeria***

Trieda reprezentuje jednu sériu seminára (hlavnej kategórii).

*27. Trieda **KoloSeminar***

Trieda reprezentuje jedno kolo v sérii seminára.

Atribúty:

- **cisloKola**

Metódy:

- **odosliOpraveneERiesenia()**: Odošle opravené riešenia elektronickým riešiteľom e-mailom.

*28. Trieda **RieseniaRiesitela***

Trieda reprezentuje odovzdané riešenia jedného riešiteľa v kole seminára. Zároveň slúži na zapísanie bodov za odovzdané riešenia.

Atribúty:

- Body za odovzdané riešenia: **b1, b2, b3, b4, b5, b6, b7, b8, b9, b10**.

*29. Trieda **Termin***

Trieda reprezentuje termín odovzdania riešení riešiteľmi v kole USP alebo seminára.

Atribúty:

- **datum**

*30. Trieda **Opravovanie***

Trieda reprezentuje jedno opravovanie riešení príkladu. Príklad opravuje aspoň jeden samostatný opravujúci a niekoľko nesamostatných opravujúcich. Samostatný opravujúci môže k opravovaniu pridať komentár.

Atribúty:

- Číslo opravovaného príkladu: **cislo**.

Metódy:

- **prihlasOrganizatoraNaOpravovanie()**: Prihlási konkrétneho organizátora na toto opravovanie.
- **odhlasOrganizatoraNaOpravovanie()**: Odhlási konkrétneho organizátora z tohto opravovania.
- **pridajBody()**: Zadá body riešiteľov za dané číslo príkladu.

2. ANALÝZA

31. Trieda *KomentarkOpravovatelovi*

Trieda reprezentuje jeden komentár k jednému nesamostatnému opravujúcemu riešeniu príkladu zadaný jedným samostatným opravujúcim.

Atribúty:

- `text`

32. Trieda *Knizka*

Trieda reprezentuje jednu knižku vytvorenú v sérii. Knižka má nejaký obsah.

Atribúty:

- Poradové číslo knižky: `cislo`.

33. Trieda *ObsahKnizky* «*abstract*»

Trieda reprezentuje obsah v danej knižke. Obsah sa spája s jedným kolom USP/seminára (napríklad zadania 1. kola zimnej série USP).

Atribúty:

- Vlastnosti obsahu: `nazov`, `text`.
- Poradie obsahu v knižke: `poradie`.
- Príznak zapnutia módu kontroly: `modKontroly`.

Metódy:

- `spustitModKontroly()`: Spustí mód kontroly.
- `ukoncitModKontroly()`: Ukončí mód kontroly.

34. Trieda *Zadania*

Trieda reprezentuje obsah knižky – zadania. Zadania obsahujú niekoľko príkladov.

35. Trieda *VzoroveRiesenie*

Trieda reprezentuje obsah knižky – vzorové riešenie príkladu.

36. Trieda *InyClanok*

Trieda reprezentuje obsah knižky – iný článok.

Návrh

Analýza systému podala odpoveď na otázku, čo má byť obsahom systému. „Smyslom návrhu systému je presná špecifikácia spôsobu, jak také funkcie implementovať“ [2].

Táto kapitola sa bude zaoberať práve touto špecifikáciou systému. Bude zameraná na návrh architektúry a spôsob ukladania dát. Výsledkom návrhu bude návrhový model tried. Kapitola tiež obsahuje návrh rozhrania informačného systému.

3.1 Návrh architektúry

Cieľom voľby architektúry je isté logické rozdelenie systému, zjednodušenie zdrojového kódu a využitie princípov *DRY* (*don't repeat yourself*), teda neopakovať rovnaký kód na viacerých miestach.

Zvolená architektúra systému je *MVC* (*model-view-controller*). Táto architektúra je trojvrstvomá a umožňuje oddelenie dátovej, prezentačnej a aplikačnej vrstvy. Dátová vrstva predstavuje riadenie a základnú prácu s uloženými dátami. Prezentačná vrstva zabezpečuje zobrazovanie vybraných dát z dátovej vrstvy. Aplikačná vrstva môže na základe vstupných údajov z prezentačnej vrstvy dať pokyn dátovej vrstve na výber konkrétnych údajov. Tie potom znova posielajú do prezentačnej vrstvy na zobrazenie.

Výsledný systém je od začiatku plánovaný ako webová aplikácia. Do úvahy teda pripadali rôzne implementačné jazyky či *frameworky*. Uvažované boli jazyky *PHP* a *Ruby* a *frameworky* *Nette*, *Symfony2* a *Ruby on Rails* [7].

Z možností bol nakoniec vybraný *framework Ruby on Rails*, a to kvôli možnosti generovania kódu, ktorý je priamo použiteľný, ďalej kvôli ušetreniu práce s relačným databázovým modelom a možnosti pridania veľkého množstva rozšírení. Dokumentácia tohto *frameworku* je na dobrej úrovni. Nevýhodou je zložitejšie nasadenie aplikácie na server, a to z dôvodu nutnosti použitia nástroja navyše. Programovanie aplikácie tiež vyžaduje rozdielny prístup, na rozdiel

od *PHP* sa niektoré aktivity vykonávajú automaticky, napríklad mapovanie databázových tabuliek podľa mena triedy a podobne.

3.1.1 Spôsob uloženia dát

Vzhľadom na nefunkčné požiadavky bola zvolená databáza MySQL. Výsledný informačný systém bude spustený na serveri organizácie. Všetky súvisiace súbory budú teda uložené na serveri.

3.2 Návrhové triedy

Návrhový model tried vychádza z analytického modelu postupným upresňovaním relácií medzi triedami, smeru jednotlivých asociácií, atribútov a ich typov, metód, ich parametrov, prípadne návratových hodnôt. S ohľadom na výber implementačného jazyka sú názvy metód prispôbené štandardnému pomenovaniu metód v *Ruby on Rails*.

Cielom tohto upresňovania je získať triedy, ktorých špecifikácia je na takom stupni, že ich je možné implementovať [2].

3.2.1 Upresňovanie analytických relácií

Podľa [2] sa upresňovanie analytických relácií skladá z niekoľkých postupných krokov. V tomto systéme boli použité nasledujúce kroky:

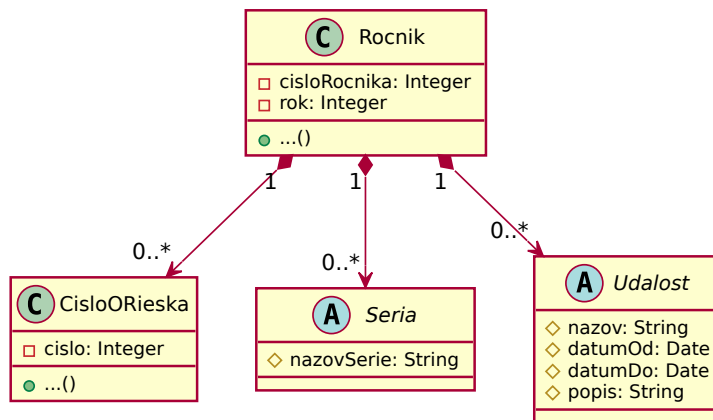
- Upresnenie vhodných asociácií do relácií typu agregácie alebo kompozície,
- upresnenie asociácii typu 1:1, M:N,
- upresnenie smeru asociácií.

Výsledný návrhový model tried je zobrazený na obrázku 3.4. Model bol vytvorený v [6].

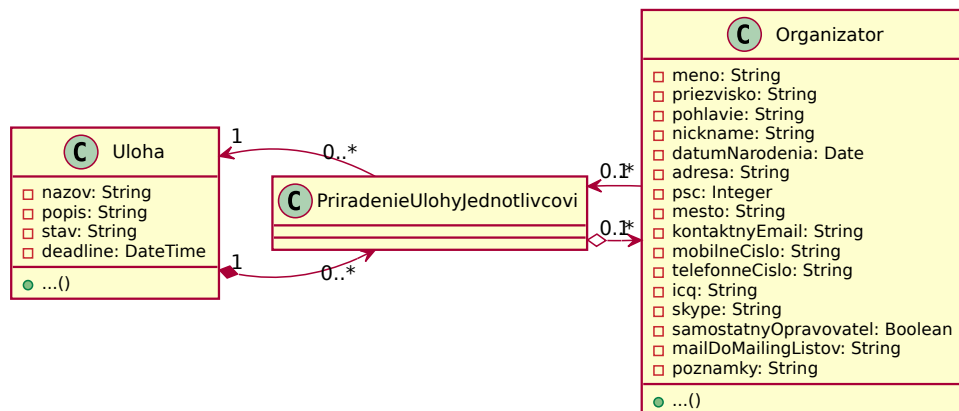
Kompozícia je druh relácie, ktorá môže byť čítaná ako „*A* obsahuje *B*“ alebo „*A* sa skladá z *B*“. Zároveň je *A* zodpovedný za vytváranie a zničenie *B*. Príklad použitia v tejto doméne je na obrázku 3.1. Tento model a modely nižšie boli vytvorené v [23].

Agregácia je použitá napríklad vo vzťahu úloha – organizátor (obrázok 3.2). Úloha zoskupuje určitých organizátorov, avšak úloha existuje aj bez spojenia s organizátormi a organizátori existujú aj bez existencie úlohy. Úloha teda nie je zodpovedná za vznik a zánik organizátorov.

Upresnenie asociácie typu 1:1 prebieha väčšinou formou kompozície, prípadne zlúčením tried. Relácia M:N musí byť rozdelená na dve asociácie 1:M a N:1. V tomto systéme to bolo dosiahnuté pridaním pomocnej triedy.



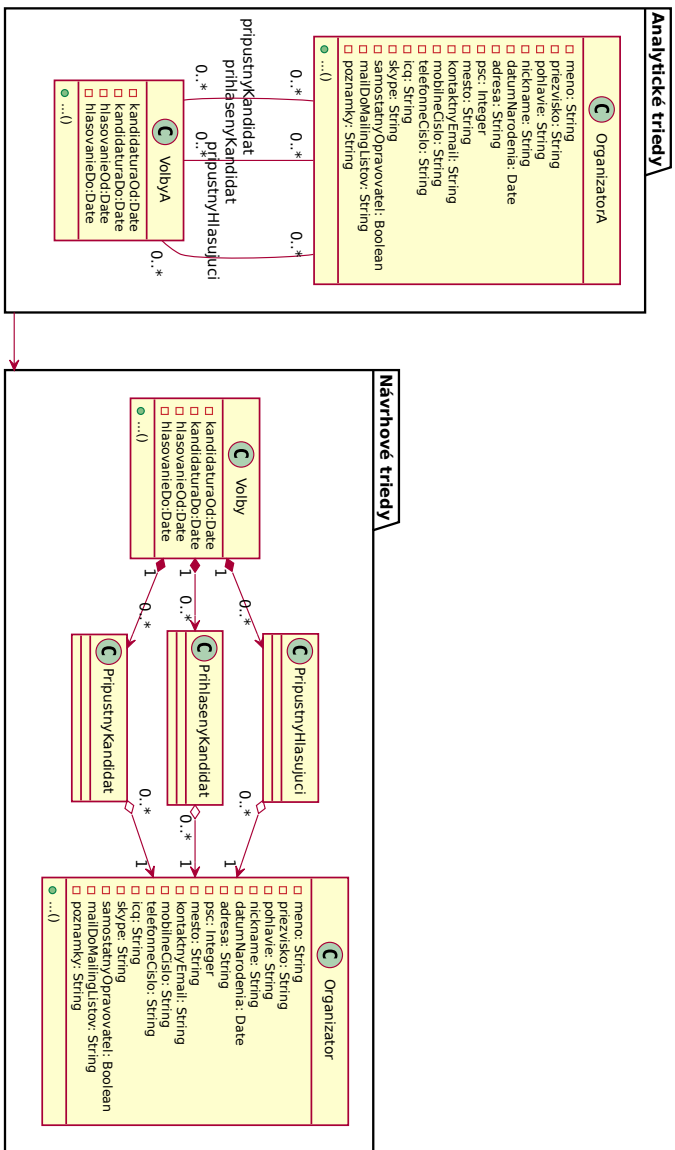
Obr. 3.1: Príklad použitia kompozície v návrhovom modeli



Obr. 3.2: Príklad použitia agregácie v návrhovom modeli

Tieto pomocné triedy môžu mať aj logický význam. Príkladom je analytická asociácia medzi voľbami a organizátorom (obrázok 3.3). Voľby zahŕňajú niekoľko prihlásených kandidátov, prípustných kandidátov a prípustných hlasujúcich. Organizátor môže byť v rôznych úlohách v niekoľkých voľbách. Upresnením vznikajú samostatné triedy pre kandidátov a hlasujúcich. Vo výsledku je teda konkrétny kandidát (hlasujúci) priradený k práve jednému organizátorovi a práve jedným voľbám.

3. NÁVRH



Obr. 3.3: Príklad rozdelenia analytickej relácie M:N

3.3 Balíčky návrhových tried

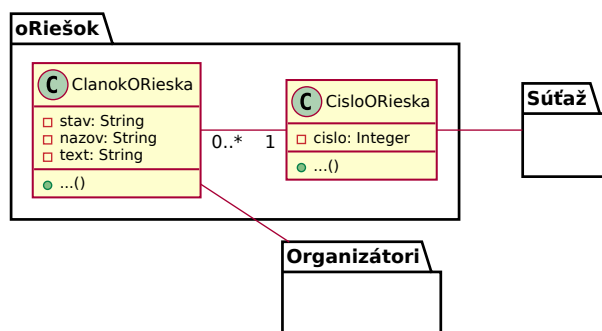
V rámci návrhu vzniká zo všetkých tried návrhového modelu isté logické rozdelenie. Toto rozdelenie zachytávajú balíčky. Ďalej budú slúžiť ako pomôcka pri implementácii, označujú totiž akýsi celok, ktorý je možné implementovať.

V jednotlivých balíčkoch sa nachádzajú úzko súvisiace návrhové triedy. Všetky balíčky budú popísané v tejto kapitole spolu s odkazom na plnenie prípadov použitia. Podľa [2] je balíček univerzálnym mechanizmom usporiadania prvkov a diagramov do skupín.

V rámci tohto systému vznikli nasledovné balíčky: Organizátori, oRiešok, Udalosti, Úlohy, Opravovanie, Príklady a Súťaž. Každý z týchto balíčkov je samostatne popísaný v nasledujúcich kapitolách. Kvôli prehľadnosti v jednotlivých obrázkoch balíčkov chýbajú metódy tried a smer a druh asociácií medzi nimi. Tieto obrázky majú totiž ilustrovať zaradenie tried do balíčkov a prepojenie medzi balíčkami. Ďalšie informácie by ich preto mohli zneprehľadniť. Podrobný návrhový model tried (spolu s vynechanými informáciami z týchto obrázkov) sa nachádza na obrázku 3.4.

3.3.1 Balíček oRiešok

Balíček (obrázok 3.5) pokrýva funkcie spojené s vytváraním časopisu oRiešok. Týka sa to prípadov použitia číslo 45 až 48. Má prepojenie na balíček Organizátori (organizátor je autorom článkov) a Súťaž (časopis je zaradený do ročníka súťaže).

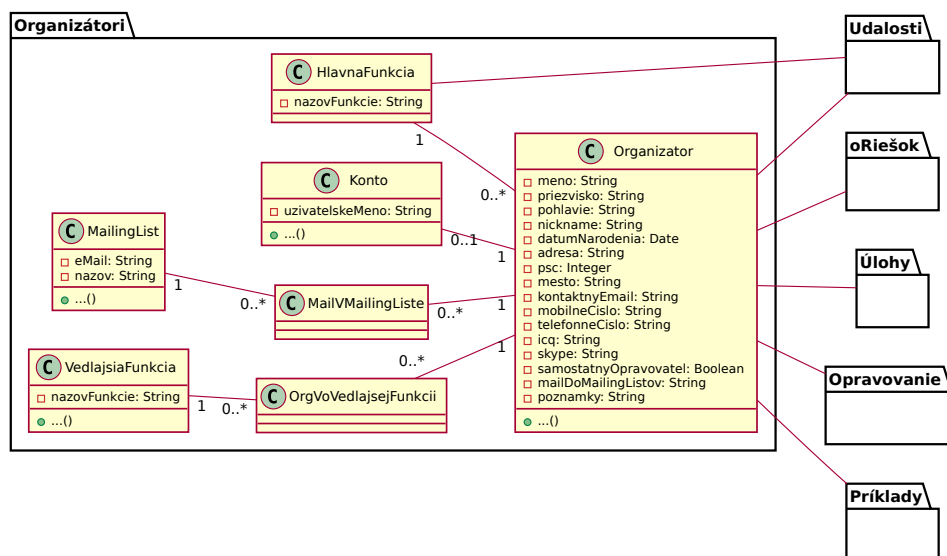


Obr. 3.5: Balíček oRiešok

3.3.2 Balíček Organizátori

Balíček Organizátori (obrázok 3.6) zahŕňa funkcie, ktoré sa týkajú organizátorov, prihlasovania a adresáru. Balíček pokrýva prípady použitia číslo 1 až 3, ktoré sa zaoberajú prihlasovaním a 56 až 60 (adresár).

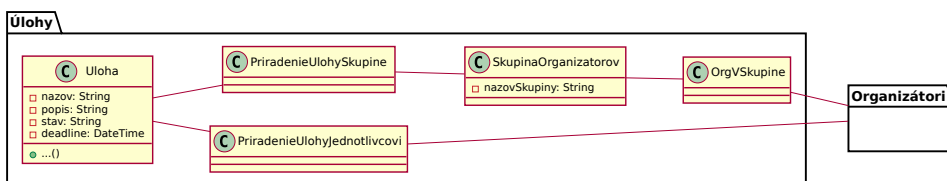
Balíček má prepojenie s väčšinou z ostatných balíčkov (oRiešok, Udalosti, Úlohy, Opravovanie, Príklady).



Obr. 3.6: Balíček Organizátori

3.3.3 Balíček Úlohy

Balíček (obrázok 3.7) sa týka zadávania úloh organizátorom. Pokrýva prípady použitia číslo 61 až 66.

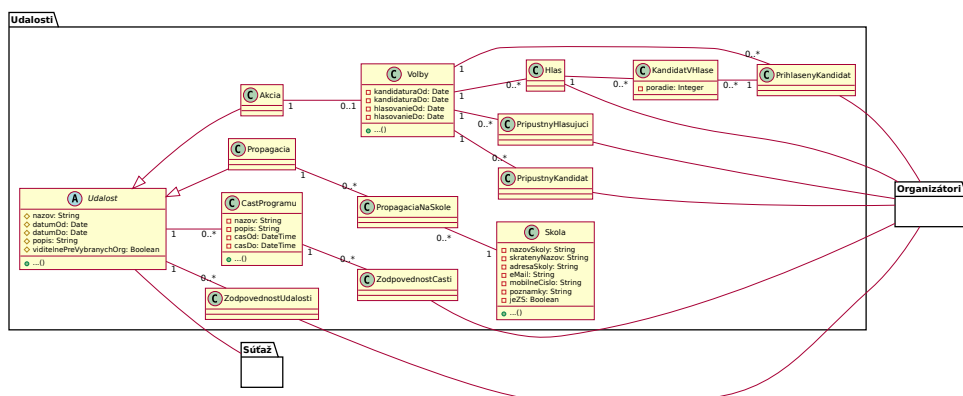


Obr. 3.7: Balíček Úlohy

3.3.4 Balíček Udalosti

Balíček (obrázok 3.8) vykonáva funkcie súvisiace so zadávaním úloh. Má prepojenie na balíček Organizátori (ktorým je udalosť priradená ako zodpovedným osobám) a Súťaž (udalosti sa odohrávajú v ročníkoch súťaže). Balíček pokrýva prípady použitia číslo 38 až 44 (organizácia propagácie) a 49 až 55 (organizácia akcií).

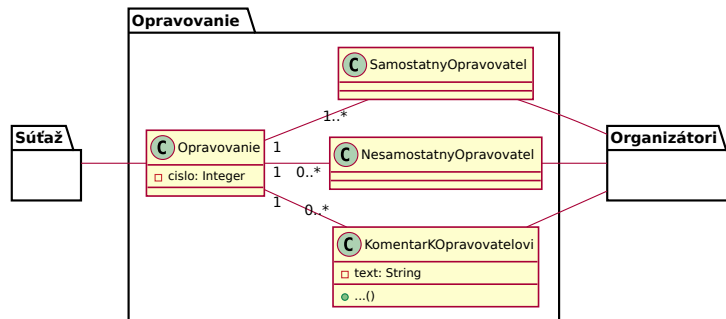
3. NÁVRH



Obr. 3.8: Balíček Udalosti

3.3.5 Balíček Opravovanie

Balíček **Opravovanie** (obrázok 3.9) zahŕňa funkcie opravovania riešení príkladov a písania komentárov k opravovateľom. Je prepojený s balíčkom **Organizatori** (organizátor opravuje riešenie príkladu) a **Súťažou** (riešenia sú zaradené do kola). Balíček pokrýva prípady použitia číslo 18 až 25.



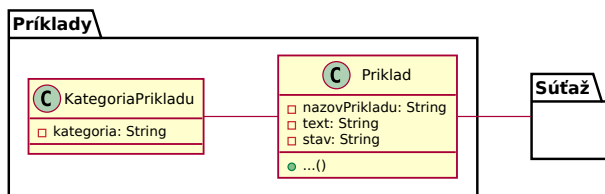
Obr. 3.9: Balíček Opravovanie

3.3.6 Balíček Príklady

Balíček (obrázok 3.10) pokrýva úlohy súvisiace s databázou príkladov a výberom príkladov do zadaní súťaže. Týka sa prípadov použitia číslo 5 až 11.

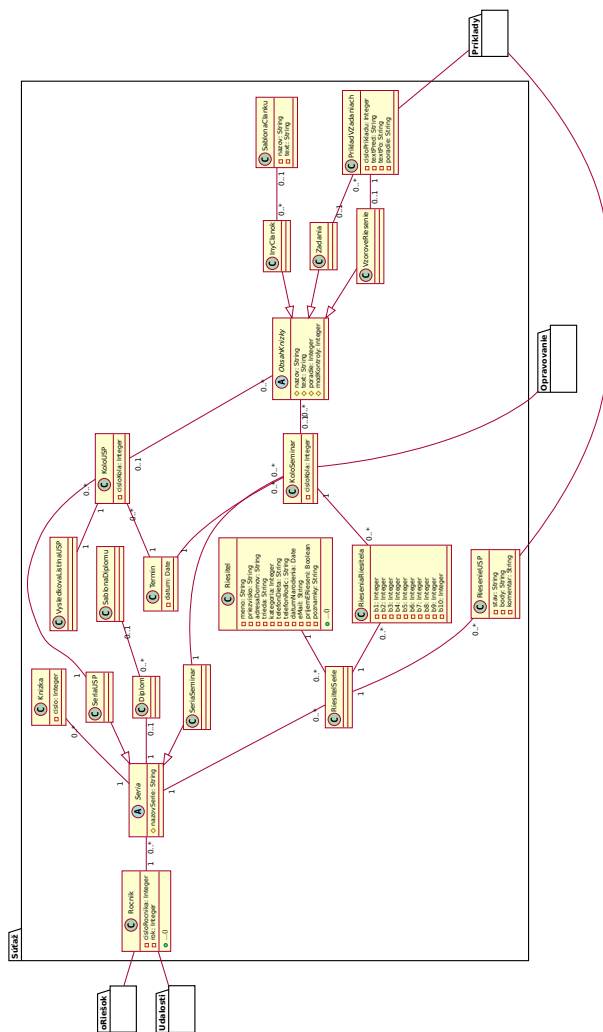
3.3.7 Balíček Súťaž

Balíček (obrázok 3.11) zahŕňa veľkú škálu úloh súvisiacu s organizáciou ročníkov, sérií a kôl seminára a kategórie USP, vytváranie knižiek a diplomov.



Obr. 3.10: Balíček Príklady

Pokrýva prípady použitia číslo 12 až 17 (organizácia knižky), 26 až 28 (organizácia obáľkovania) a 29 až 37 (organizácia USP).



Obr. 3.11: Balíček Súťaž

3.4 Návrh užívateľského rozhrania

Súčasťou návrhu systému je aj návrh užívateľského rozhrania. Cieľom tohto návrhu je vytvoriť predstavu o obsahu jednotlivých zobrazených stránok zo systému, určiť rozmiestnenie a použitie prvkov v systéme (napríklad tlačidlá, odkazy, nadpisy). Výsledok je možné použiť pri implementácii prezentačnej vrstvy systému. Tento návrh vychádza z dohodnutých scenárov prípadov použitia (popísaných v kapitole 2.3.2).

Formou návrhu systému sú *wireframes*, ktoré predstavujú náčrt jednotlivých zobrazených stránok. Na vytvorenie bol použitý nástroj Pencil [13]. Je nutné povedať, že sa jedná skutočne o náčrt stránok. V implementovanom systéme môžu vyzeráť jednotlivé prvky inak, budú mať iný vzhľad, trochu iné rozmiestnenie, veľkosť alebo funkciu (príkladom je zadávanie dátumov).

Kvôli ušetreniu priestoru tiež nie sú v tejto práci ukázané úplne všetky *wireframes*. Veľmi podobné stránky sú zlúčené do jednej spoločnej (napríklad pridávanie a úprava údajov organizátora sa líši len minimálne). Kvôli prehľadnosti sú tiež rozdelené do skupín.

3.4.1 Organizátor

Na obrázku 3.12 sú postupne zobrazené stránky prihlásenia organizátora, pridanie konta, úprava funkcií a údajov organizátora a zobrazenie adresára.

3.4.2 Príklad

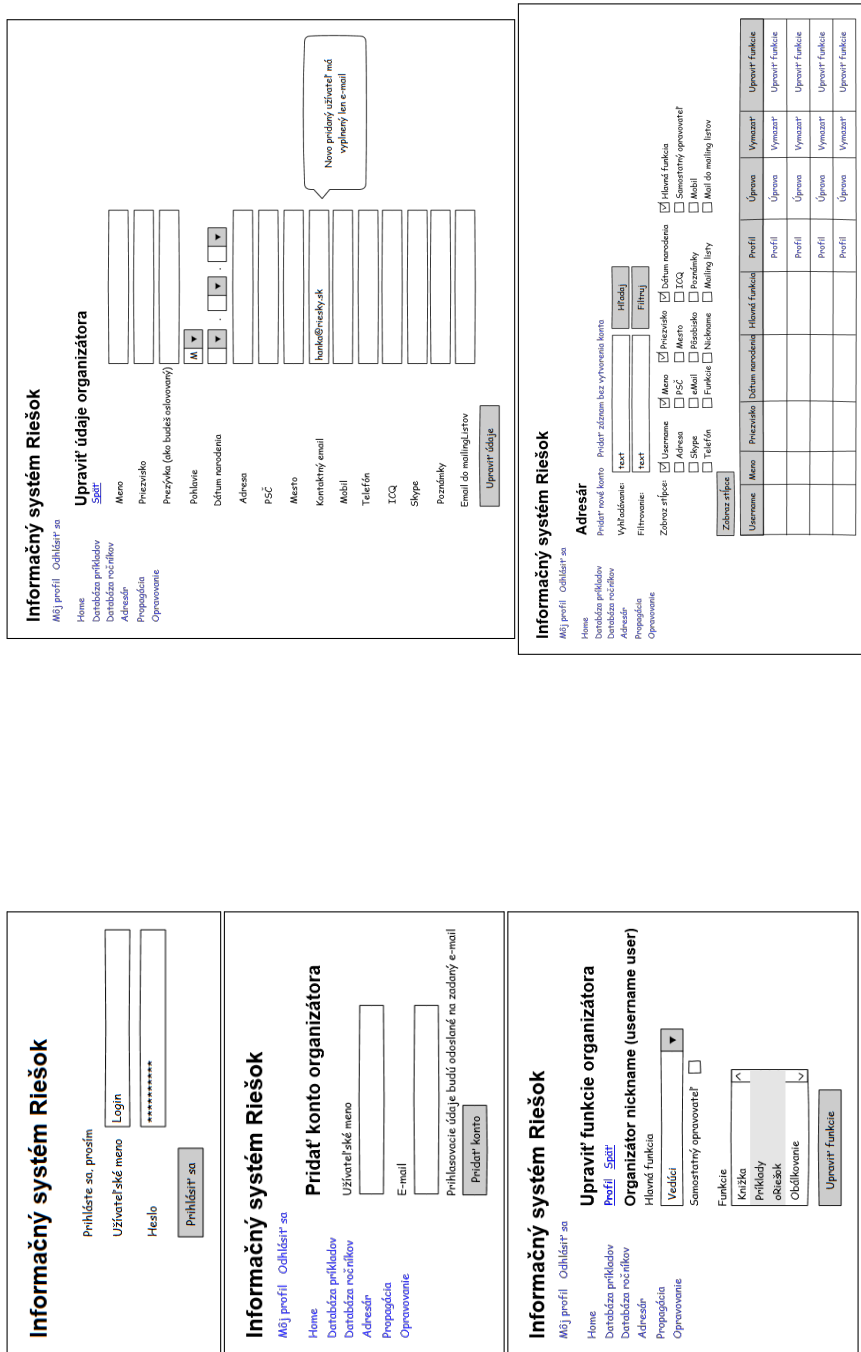
Na obrázku 3.13 sú postupne zobrazené stránky databázy pridaných príkladov, vytvárania, detailu, výberu príkladu do zadania a štatistiky pridaných príkladov.

3.4.3 Úlohy a oRiešok

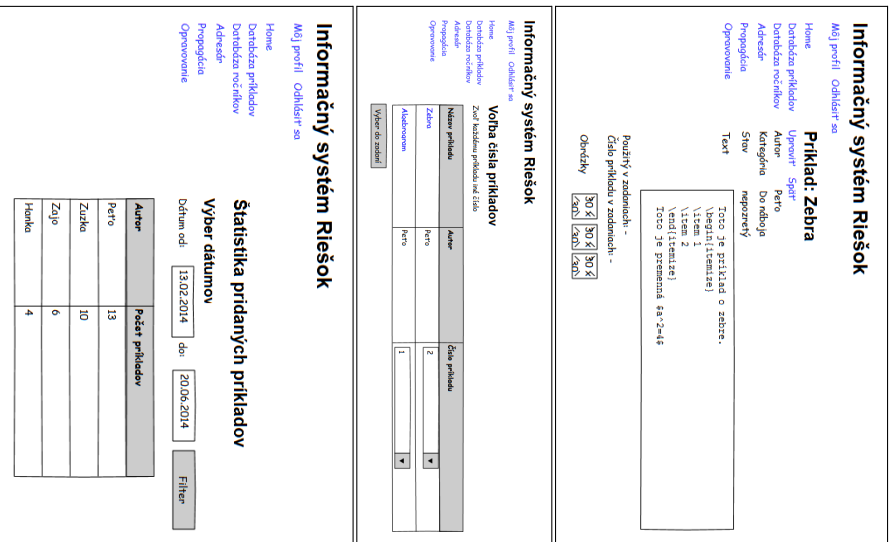
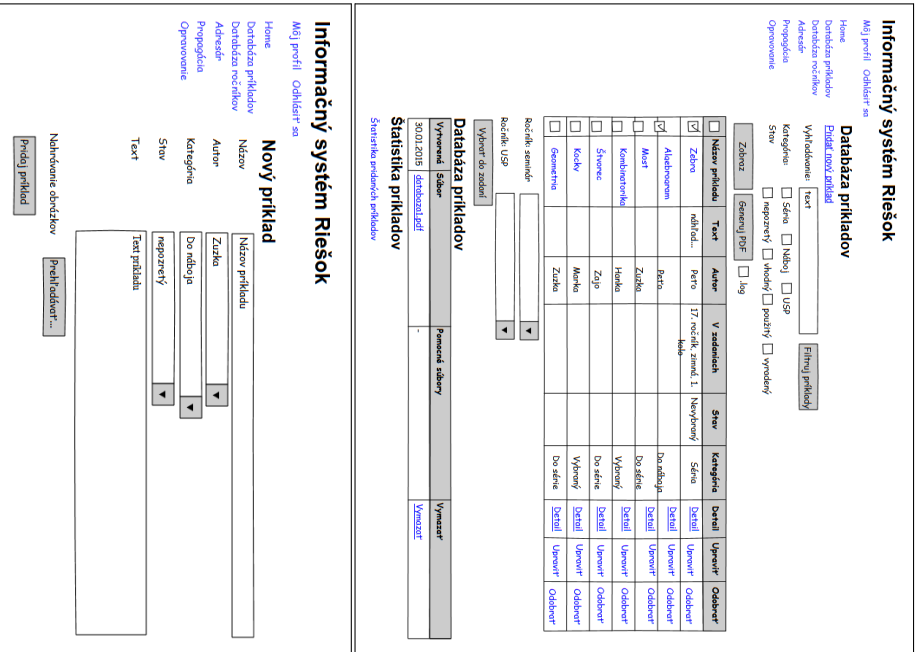
Na obrázku 3.14 sú postupne zobrazené stránky čísla a úpravy článku oRieška, zobrazenie úloh, vytvorenie a zobrazenie úlohy, archív úloh.

3.4.4 Súťaž

Na obrázkoch 3.15 až 3.20 sú postupne zobrazené stránky ročníka súťaže, pridania a obsahu ročníka, obsahu série, pridania prihlášok riešiteľov, databázy riešiteľov, vytvorenia diplomu, úpravy zadania, rozpisu opravovania, príkladov riešiteľov, pridávania bodov, komentárov k opravovaniu, histórie opravovaní, odosielania elektronických riešení, pridávania, zoznamu a opravy riešenia USP, pridávania vzorového riešenia, kontroly výsledkovej listiny, zoznamu škôl, odosielania propagačných materiálov, detailu propagácie, pridania časti programu, vytvorenia akcie, detailu akcie, volieb na akciu, hlasu vo voľbách a detailu programu akcie.

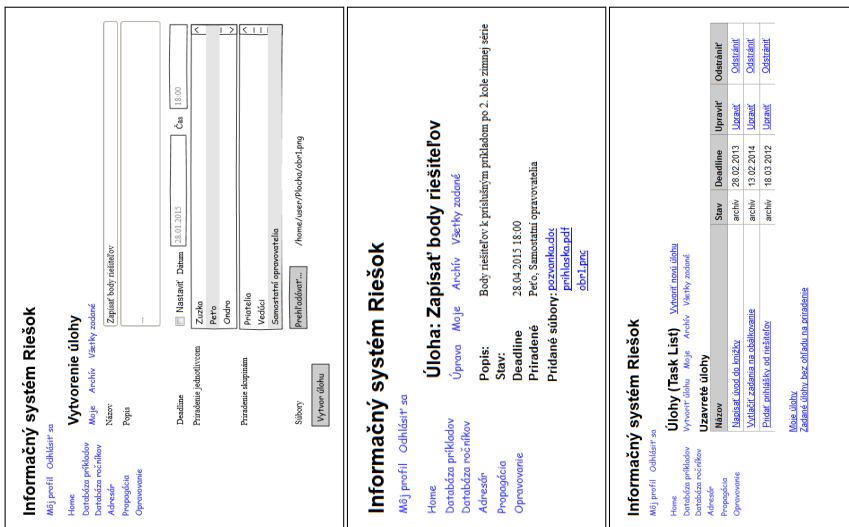


Obr. 3.12: Stránky organizátora

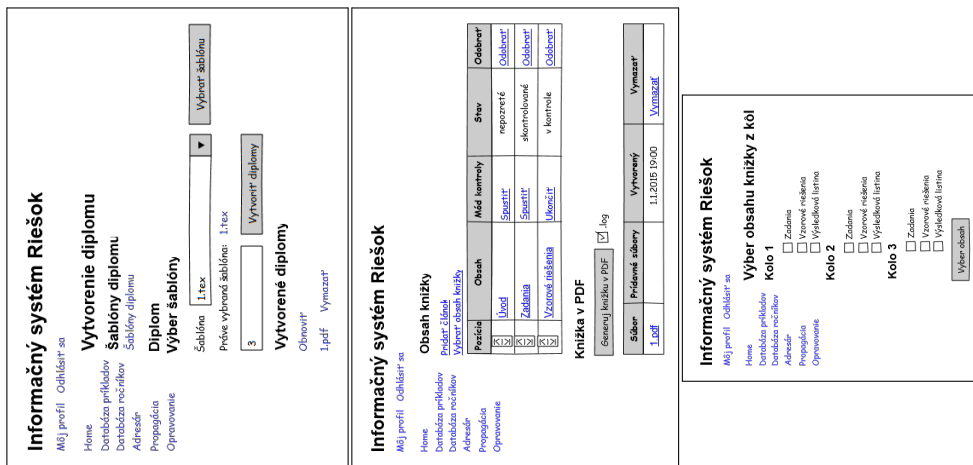
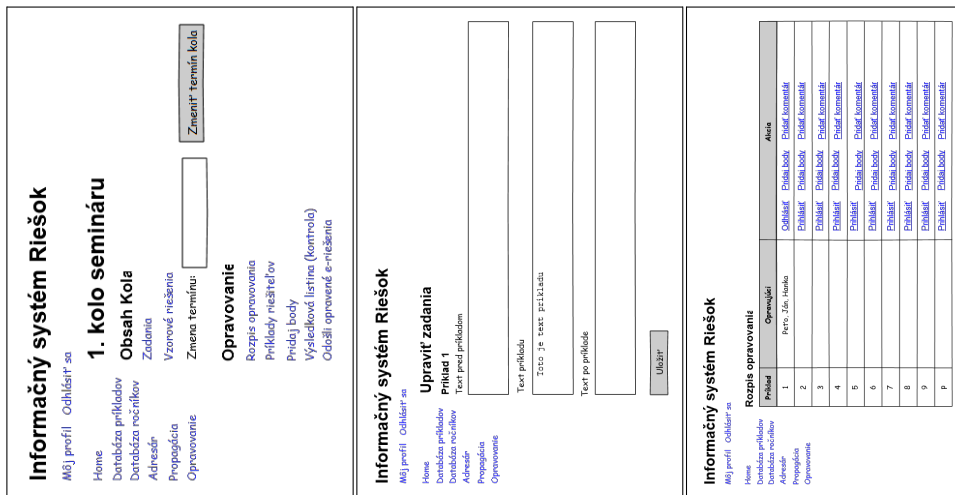


3. NÁVRH

Obr. 3.13: Stránky príkladov



Obr. 3.14: Stránky oRieška a úloh



Obr. 3.16: Stránky súťaže

Informačný systém Riešok

Môj profil Odhlásiť sa

Home
 Databáza príkladov
 Databáza nočníkov
 Adresár
 Propagácia
 Opravovanie

Príklady riešení v kole

Meno	Priezvisko	1	2	3	4	5	6	7	8	9	10
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

[Upraviť príklady riešení](#)

Informačný systém Riešok

Môj profil Odhlásiť sa

Home
 Databáza príkladov
 Databáza nočníkov
 Adresár
 Propagácia
 Opravovanie

Príklad č. 1

Meno	Priezvisko	Body
5		<input type="text"/>
5		<input type="text"/>
5		<input type="text"/>
5		<input type="text"/>
5		<input type="text"/>
5		<input type="text"/>
5		<input type="text"/>

[Pridať body](#)

Informačný systém Riešok

Môj profil Odhlásiť sa

Home
 Databáza príkladov
 Databáza nočníkov
 Adresár
 Propagácia
 Opravovanie

Komentáre k opravovaniu

Spiť na rozpis opravovania Spiť na zoznam mojich opravovaní

Ján

Hanka

[Uložiť komentár](#)

Informačný systém Riešok

Môj profil Odhlásiť sa

Home
 Databáza príkladov
 Databáza nočníkov
 Adresár
 Propagácia
 Opravovanie

Opravovanie

História svojich opravovaní

Kód	Číslo príkladu	Stupeň	Komentár	Učenie
17	nočník, zmes 1, kolo 5	70%	OK	Komentár
17	nočník, zmes 2, kolo 3	100%	Právy švarovky	Komentár

Informačný systém Riešok

Môj profil Odhlásiť sa

Home
 Databáza príkladov
 Databáza nočníkov
 Adresár
 Propagácia
 Opravovanie

E-Riešenia

Meno a priezvisko	Obdržané príklady	Učenie
12		Učenie
5,6,7,8,9		Učenie
1,2,5		Učenie

Obsah e-mailu

Telo e-mailu

[Odslať riešenie](#)

Informačný systém Riešok

Môj profil Odhlásiť sa

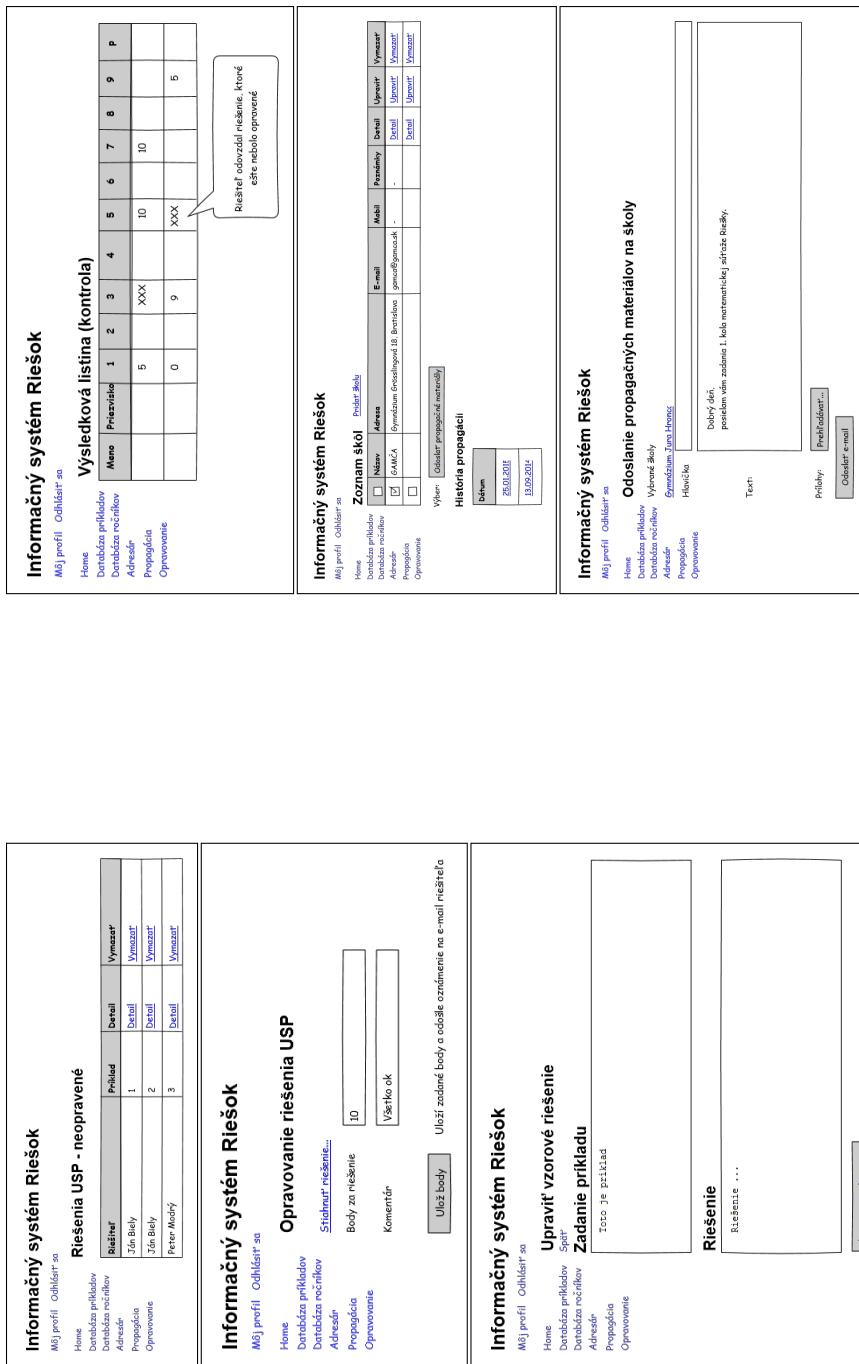
Home
 Databáza príkladov
 Databáza nočníkov
 Adresár
 Propagácia
 Opravovanie

Pridať riešenie USP

Riešiteľ

[Pridať riešenie](#) [Vyber siňbor](#)

Obr. 3.17: Stránky súťaže



Obr. 3.18: Stránky súťaže

Informačný systém Riešok

Môj profil [Odhliadniť sa](#)

Home [Zobraziť aktuálne programy](#) [Pridať časť programu](#)

Databáza príkladov [Zobraziť akciu](#)

Databáza nočníkov [Pridať časť programu](#)

Adresár [Zobraziť akciu](#)

Propagácia [Pridať časť programu](#)

Oprnovanie [Zobraziť akciu](#)

Program propagácie

Id	Na	Škola	Zodpovedná osoba	Popis	Detal	Uprav'	Ostrániť
700	7:30	ESH	Peto, Hanka	-	Detail	Uprav'	Ostrániť
718	7:45	SPMND&S	Marek	-	Detail	Uprav'	Ostrániť
900	11:30	6AM&Z	-	-	Detail	Uprav'	Ostrániť

Pridať akciu

Informačný systém Riešok

Môj profil [Odhliadniť sa](#)

Pridať časť programu

Home [Zobraziť aktuálne programy](#) [Pridať časť programu](#)

Databáza príkladov [Zobraziť akciu](#)

Databáza nočníkov [Pridať časť programu](#)

Adresár [Zobraziť akciu](#)

Propagácia [Pridať časť programu](#)

Oprnovanie [Zobraziť akciu](#)

Názov

Trvanie Od:

Do:

Popis

Škola:

Organizátori:

Súbory:

Informačný systém Riešok

Môj profil [Odhliadniť sa](#)

Nová akcia

Home [Zobraziť aktuálne programy](#) [Pridať časť programu](#)

Databáza príkladov [Zobraziť akciu](#)

Databáza nočníkov [Pridať časť programu](#)

Adresár [Zobraziť akciu](#)

Propagácia [Pridať časť programu](#)

Oprnovanie [Zobraziť akciu](#)

Názov

Dátum od:

Do:

Popis

Vedúci (priamy výber)

Akcia má byť viditeľná iba pre [e] organizátorov

Usporiadať voľby na akciu

Pridať súbory

Informačný systém Riešok

Môj profil [Odhliadniť sa](#)

Zimné sústredenie 25.01.2015 - 30.01.2015

Home [Zobraziť aktuálne programy](#) [Pridať časť programu](#)

Databáza príkladov [Zobraziť akciu](#)

Databáza nočníkov [Pridať časť programu](#)

Adresár [Zobraziť akciu](#)

Propagácia [Pridať časť programu](#)

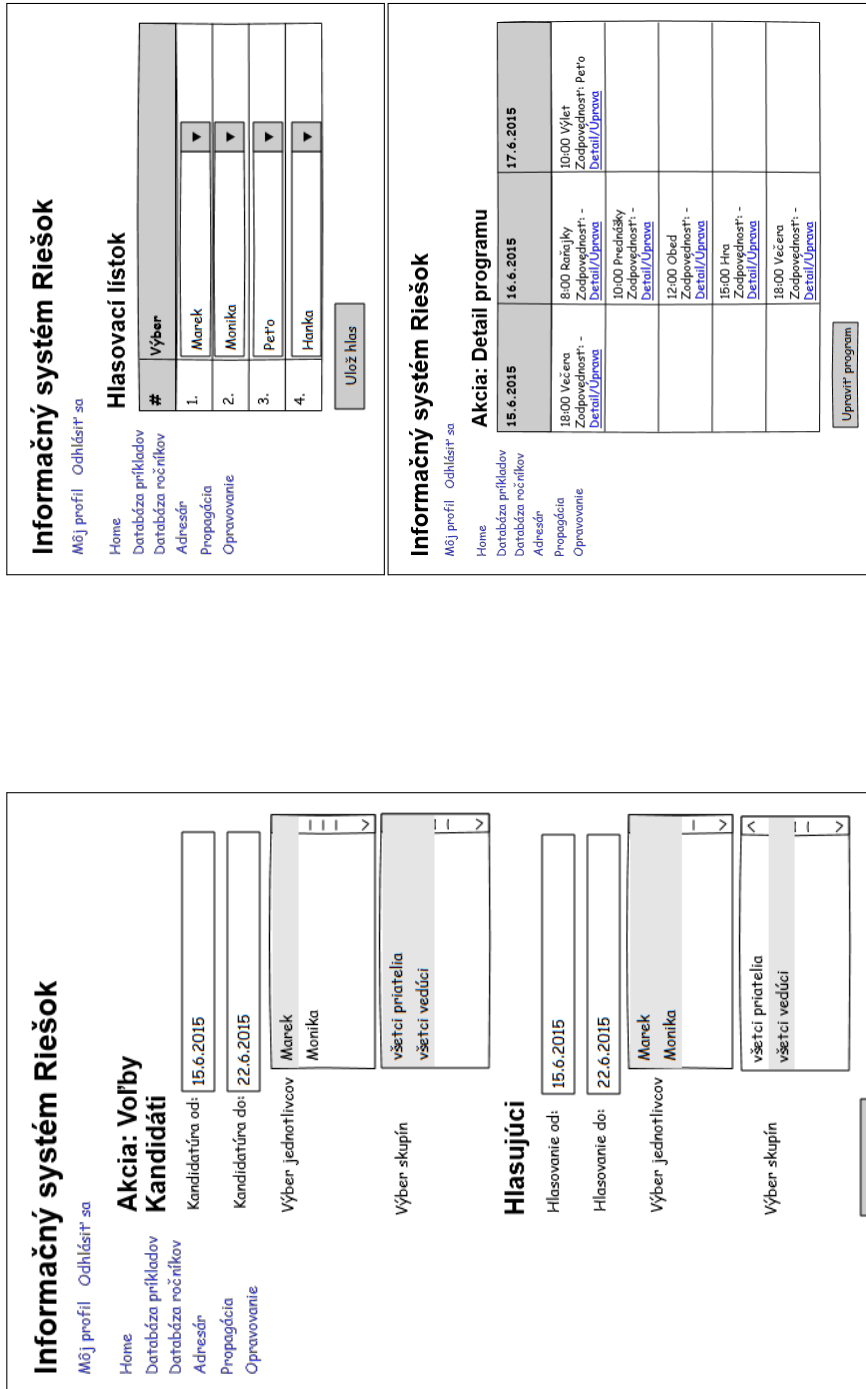
Oprnovanie [Zobraziť akciu](#)

Program akcie

Id	Na	Škola	Zodpovedná osoba	Popis	Detal	Uprav'	Ostrániť
700	7:30	ESH	Peto, Hanka	-	Detail	Uprav'	Ostrániť
718	7:45	SPMND&S	Marek	-	Detail	Uprav'	Ostrániť
900	11:30	6AM&Z	-	-	Detail	Uprav'	Ostrániť

Pridať akciu

Obr. 3.19: Stránky súťaže



Obr. 3.20: Stránky súťaže

Implementácia

4.1 Implementácia

Na implementáciu bol vybraný framework Ruby on Rails [7]. Framework výrazne zjednodušil prácu, keďže je možné generovanie základných súborov jedným príkazom: *Model, view, controller*, súbory pre testy [15]. Okrem toho automaticky mapuje názov triedy s názvom databázovej tabuľky. Nie je preto nutné definovať relačný model a stačí vychádzať z návrhového modelu tried. Nižšie sú popísané využité príkazy:

```
rails new isriesky # Vytvorí novú aplikáciu
rails server # Spustenie aplikácie na lokálnom serveri
rails generate scaffold Konto # Generovanie triedy a jej kostry
rails generate model SamostatnyOpravovatel # Vytvorenie modelu
rails generate controller Organizator # Vytvorenie kontroleru
rails generate mailer KontoMailer # Vytvorenie kostry pre maily
```

Príkaz `generate scaffold` vytvorí *model, view, controller* a testovacie súbory súvisiace s danou triedou. Kontroler má preddefinované základné akcie:

- *index*: Zoznam záznamov danej triedy.
- *show*: Detail konkrétneho záznamu.
- *new, edit*: Vytvorenie/úprava záznamu.
- *create, update*: Metódy vytvoria/upravia záznam podľa údajov z *new, edit*.
- *destroy*: Metóda vymaže záznam.

Tento zoznam môže byť rozšírený alebo upravený nastavením ciest (*routing* [8]).

4.2 Problémy a zaujímavosti

V priebehu implementácie boli riešené rôzne problémy. V tejto kapitole sú popísané spolu s vysvetlením ich riešenia.

4.2.1 Naplnenie databázy testovacími dátami

Na testovanie aplikácie bolo nutné pripraviť si určité testovacie dáta (*seeds*), ktoré mohli byť kedykoľvek nahrané do databázy [16]. Tieto testovacie dáta boli pripravené tak, aby vytvárali určitý príbeh. V našom prípade ide o niekoľkých organizátorov v rôznych funkciách, ktorí ovládajú informačný systém. Dáta môžu byť nahrané do databázy príkazom `rake db:seed`.

4.2.2 Implementácia balíčkov

Balíčky popísané v kapitole 3.3 poskytli určitý postup pri implementácii. Správnym usporiadaním balíčkov totiž vznikla postupnosť krokov, pričom splnenie jedného kroku predchádzalo začiatku plnenia iného kroku.

Balíček Organizátori bol implementovaný ako prvý z dôvodu riešenia prihlasovania organizátora a priradovania funkcií systému jednotlivcom. Hneď druhý v poradí bol implementovaný balíček Úlohy, ktorý má priame napojenie na balíček Organizátori.

Balíčky Udalosti, Príklady a oRiešok boli implementované pred implementáciou balíčka Súťaž. Po implementácii tohto balíčka boli triedy dodatočne prepojené. Nakoniec bol implementovaný balíček Opravovanie. Toto poradie mohlo byť aj iné, každopádne, týmto spôsobom bol systém postupne vytváraný, až kým neboli implementované všetky požadované funkcie.

4.2.3 Prihlasovanie

Prihlasovanie do systému bolo inšpirované riešením v [4]. Konto užívateľa neukladá heslo do databázy tak, ako ho zadáva užívateľ. Podľa [4] „pomocí algoritmu SHA-2 uložíme hashovací otisk veľikosti 256 bitů vytvořený z hesla a další hodnoty, která se v kryptografických kruzích označuje jako sůl (salt).“ Konto teda ukladá výsledok hashovacej funkcie a soľ. Soľ je generovaná ako náhodný reťazec.

Hashovacia funkcia zaisťuje bezpečosť tak, že z vytvoreného reťazca je výpočetne nemožné vytvoriť pôvodný reťazec. Prípadný útočník by teda nemal byť schopný zrekonštruovať heslá užívateľov. Avšak rovnaký reťazec bude vždy prevedený na rovnaký hashovaný reťazec. Táto vlastnosť je využitá pri prihlasovaní užívateľa.

Užívateľ zadá užívateľské meno a heslo do formulára, po odoslaní sa vytvorí hash, ktorá sa porovná s hodnotou pri danom užívateľskom mene v databáze. Ak je hodnota rovnaká, užívateľ pri prihlasovaní uspel. Ak sa hodnota líši, je jasné, že užívateľ zadal nesprávne heslo.

4.2.4 Validácia zadaných dát

Po zadaní údajov z formulárov nastala validácia. Ak zadané údaje neboli správne, bola vypísaná správa o chybách. Validácia je definovaná v modeloch [17]. Najviac používané validácie boli:

- *presence*: Daný údaj je povinný.
- *uniqueness*: Zadaný údaj je v databáze unikátny (napríklad prihlasovacie meno).
- *Vlastné*: Definované pred alebo po uložení záznamu, napríklad výmena opačne zadaných dátumov (dátum začiatku je neskorší ako dátum konca), zápis poradového čísla obsahu knihy, automatické vytvorenie obsahu knihy pri vytvorení kola, prihlasovanie a podobne.

4.2.5 Vytváranie súborov vo formáte PDF

Na vytvorenie knihy boli použité existujúce \LaTeX makrá tak, aby nebolo možné poznať rozdiel medzi knižkou generovanou pred a po použití informačného systému. Na vytvorenie ďalších súborov vo formáte PDF (databáza príkladov, diplomy) boli vytvorené nové makrá.

4.2.6 Generovanie diplomov

Diplomy za sériu pre najlepších riešiteľov majú vždy podobný obsah. Preto boli vytvorené šablóny diplomu, ktoré je možné ľubovoľne meniť podľa požadovaných vlastností. Výhodou je možnosť šablóny opakovane používať v ďalších sériách. Šablóny sú ukladané vo formáte \LaTeX .

Pri vytváraní diplomu sa najprv použijú spoločné makrá pre diplomy. Tie obsahujú pridávanie balíčkov \LaTeX u a úvodné nastavenia dokumentu. Po týchto makrách sú dynamicky generované makrá konkrétneho diplomu, ktoré môžu a nemusia byť použité v šablóne. Preddefinované makrá diplomu sú:

- `\meno` (doplňuje meno a priezvisko riešiteľa)
- `\umiestnenie` (umiestnenie z výsledkovej listiny)
- `\body` (body riešiteľa)
- `\rocnik` (číslo ročníka, napr. 17)
- `\rocnikrok` (školský rok, napr. 2014/2015)
- `\seria` (názov série v základnom tvare, napr. zima)
- `\seriaakuzativ` (vyskloňuje sériu pri použití `\seriaakuzativ{\seria}`, napr. zimnej)

Po definícii makier nasleduje samotná definícia šablóny diplomu. V nej môže užívateľ pridávať vlastné makrá, ktoré budú platiť iba v danej šablóne. Užívateľ môže vygenerovať náhľad diplomu a meniť tak šablónu do požadovanej podoby.

4.2.7 Formát dátumu, času, upozornení

Na zjednotenie formátov dátumu a času boli vytvorené funkcie na ich formátovanie.

Na formátovanie upozornení po vytvorení, úprave a vymazaní záznamov bola vytvorená funkcia, ktorá na vstupe získala číslo, rod a názov funkcie, z ktorej bola volaná a na výsledku vrátila správne vyskloňované upozornenie o vytvorení, úprave alebo vymazaní záznamu.

4.2.8 Optimistické zamykanie

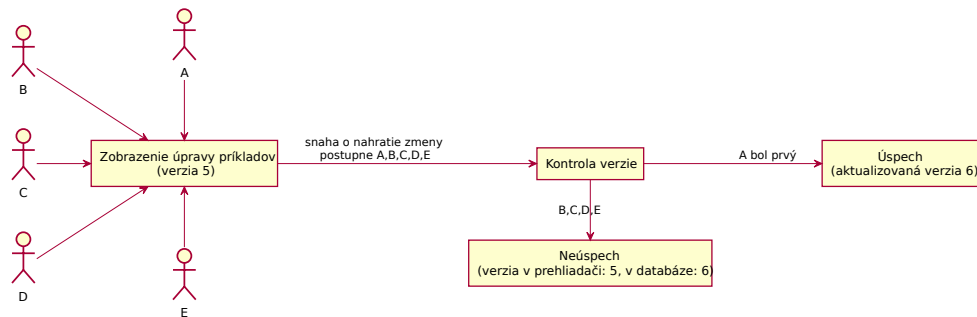
V niektorých prípadoch ukladania záznamov do databázy je predpokladané súbežné nahrávanie zmien (napríklad kontrola vzorových riešení, zadaní a podobne). Aby si organizátori neprepisovali navzájom údaje, je potrebné vybrať nejaký spôsob zamykania. Podľa [3] existuje niekoľko druhov zámkov:

- Podľa úrovne zamykania: Zámky tabuliek, stránok a riadkov.
- Podľa prístupových práv: READ, WRITE.

V tejto práci bol použitý zámok jedného riadku tabuľky, a to konkrétneho záznamu so zámkom WRITE. Teda daný riadok môže byť čítaný naraz viacerými zdrojmi, ale zapisovať môže iba jeden z nich.

Ruby on Rails poskytuje takzvané optimistické zamykanie [18]. Stačí tabuľke priradiť stĺpec `lock_version`. Pri vytvorení záznamu sa priradí hodnota verzie 0. Pri každej zmene záznamu sa hodnota automaticky inkrementuje. Príklad optimistického zamykania popisuje obrázok 4.1 a vysvetľuje nasledujúci priebeh:

- Viacero užívateľov zobrazí formulár na zmenu záznamu v databáze. Každý z nich dostane z databázy číslo verzie záznamu.
- Prvý z užívateľov zmení záznam. Číslo verzie záznamu sa zvýši.
- Druhý a ďalší užívateľ pri snahe o zmenu záznamu kontroluje aktuálnu verziu záznamu z databázy. Zisťuje, že sa verzia zmenila. Užívateľovi je vypísaná hláška o neúspešnom pridaní. Má šancu svoje zmeny uložiť a aktualizovať stránku pre načítanie zmeneného záznamu.



Obr. 4.1: Príklad optimistického zamykania záznamov

Optimistické zamykanie bolo použité len pri triedach, pri ktorých je predpokladaná súbežná práca: Organizator, Skola, Uloha, Udalost (akcia a propagácia), CastProgramu, ClanokORieska, Priklad, Riesitel, RiesenieUSP, ObsahKnizky (vzorové riešenia, iný článok), KomentarKOPravovatelovi, RieseniaRiesitela (pridávanie bodov a príkladov riešiteľov), SablonaClanku, PrikladVZadaniach.

4.2.9 Odosielanie e-mailov zo systému

Na odosielanie e-mailov v *Ruby on Rails* slúžia *mailery* [19]. Bolo potrebné nastaviť ich tak, aby sa e-maily odosielali z požadovanej služby. Nastavenie bolo vykonané v nastavení prostredí.

V konkrétnom *maileri* potom stačilo definovať premenné, ktoré boli zobrazované v konkrétnom e-maile, prípadne prílohy e-mailu. Obsah správy bol nastavený vo formátoch HTML a čistého textu. Samotný príkaz na odoslanie e-mailu je volaný z kontroleru.

E-maily sú odosielané pri vytváraní nového konta, odosielaní propagačných materiálov na školy, opravovaní riešení USP a odosielaní riešení elektronických riešiteľov série.

4.2.10 Štýly tabuliek

Aby boli zobrazované tabuľky prehľadnejšie, bolo potrebné pridať štýly tabuliek. Na ich vytvorenie bol použitý generátor CSS štýlov [14]. Bolo potrebné zvoliť veľkosť oddeľovačov buniek tabuľky a ohraničenia, zarovnanie textu a farby. Vygenerovaný kód bol pridaný do aplikácie medzi štýly.

4.2.11 Upozornenie na blížiaci sa termín úloh

Ak je vo vytvorenej úlohe nastavený termín, jeho farba na stránke zobrazenia úloh sa mení vzhľadom na daný deň. Termíny úloh, ktoré uplynú za viac ako 3 dni, sú zobrazené čiernou farbou. Termíny úloh, ktoré uplynú za menej ako

3 dni, ale ešte neuplynuli, sú zobrazené fialovou farbou. Uplynuté termíny sú zobrazené červenou farbou.

4.2.12 Vytváranie a práca so súbormi

Súbory sú pridávané k udalosti, diplomu, riešeniu USP, príkladu, úlohe, časti programu, šablóne diplomu, knižke alebo databáze príkladov. Súbory sú fyzicky uložené v zložke `public`. Priradenie konkrétneho súboru konkrétnemu záznamu je zapísané v tabuľke súborov.

Pri vymazávaní súboru z databázy (ktoré môže byť vyvolané aj kaskádovo napríklad vymazaním priradenej udalosti či príkladu) sa volá metóda, ktorá súbor fyzicky vymaže aj zo servera. Takto je zabezpečené, že sa na serveri nebudú hromadiť súbory, ktoré nebudú pridané záznamom v databáze.

4.2.13 Zvýrazňovanie syntaxe \LaTeX u

Pri zobrazení zdrojového kódu napísaného v \LaTeX u je použité zvýrazňovanie syntaxe. Na tento účel bol využitý nástroj [25]. Využité bolo napríklad pri zvýraznení zadania príkladu. Použitie vo *view* bolo nasledovné:

```
<pre><code style="tex">
Text, v~ktorom bude zvýraznená \textit{syntax}.
</code></pre>
```

4.3 Nasadenie

Priebeh nasadenia systému na server organizácie bol inšpirovaný [4]. Systém bol nasadený na server Apache s využitím databázy MySQL.

Samotnú obsluhu aplikácie na serveri má na starosti nástroj Phusion Passenger [21]. Vzdialené nasadzovanie aktuálnych zdrojových kódov aplikácie zabezpečuje nástroj Capistrano [22]. Tento nástroj automaticky nahraje zdrojové kódy z repozitára Git [24], do ktorého boli nahrané pri vývoji aplikácie. Nastavenie nasadenia aplikácie je uložené v súbore `config/deploy.rb`. Nasadenie môže byť spustené pomocou príkazu:

```
rvmsudo cap production deploy deploy:restart
```

Tento príkaz preberie aktuálne zdrojové kódy z Git repozitára a vytvorí odkaz `current` na aktuálnu verziu aplikácie. Následne príkaz `deploy:restart` vytvorí súbor `tmp/restart.txt`, ktorý spôsobí reštart aplikácie. Tiež je vytvorený súbor `log/production.log`, do ktorého sú ukladané záznamy aplikácie.

Testovanie

5.1 Automatizované testy

Ako bolo popísané v kapitole 4.1, pri generovaní tried do systému sú zároveň vytvárané súbory súvisiace s testovaním. Sú to napríklad testy súvisiace s funkciami kontrolerov, funkčné testy, integračné testy, testy modelov, výkonu, mailerov, jednotkové testy.

Ruby on Rails na testovanie využíva oddelenú databázu určenú iba testovaciemu prostrediu. Tým je zaručené, že údaje uložené v produkčnom alebo vývojovom prostredí nebudú narušené. Testovanie tejto aplikácie vychádza z [20]. Celkový počet vytvorených testov je 400.

Testy je možné spustiť všetky naraz alebo len niektorú ich časť (testy kontrolerov, modelov a podobne). Môžu byť spustené napríklad pomocou príkazov:

```
rake test # Spustí všetky testy
rake test:controllers # Spustí testy kontrolerov
rake test:models # Spustí testy modelov
```

Pred samotným spustením testu prebehne nahrávanie testovacích dát do databázy. S týmito dátami potom jednotlivé testy pracujú. Napríklad na prácu s kontom je nutné priradiť konto z testovacích dát. Môže to vyzeráť napríklad takto:

```
setup do
  @organizator = organizators(:daniela)
  @konto = kontos(:daniela)
  session[:konto_id] = kontos(:daniela)
end
```

Metóda `setup` nastaví premenné využívané testom. Nastaví premennú `organizator` ako organizátorku Danielu, ktorá bola definovaná v testovacích dátach. Rovnako pomenované bolo aj jej konto.

5.1.1 Testovacie dáta

Podobne, ako pri implementácii, boli vytvorené testovacie dáta tak, aby tvorili príbeh. Dáta sú cielene tvorené tak, aby pokryli čo najviac možností reálneho použitia. V *Ruby on Rails* je možné špecifikovať testovacie dáta názvom.

V prípade tohto systému príbeh začína u organizátorov Daniela, Peter, Michal, Ondrej, Ivan, Jakub, Juraj a Andrej. Daniela je šéfkou Riešok, Peter a Michal vedúcimi, Ivan a Jakub priateľmi. Všetci títo majú vytvorené konto v systéme. Juraj je priateľom, Andrej je vedúcim, obaja konto v systéme nemajú.

Daniela, Peter a Andrej študujú na UK v Bratislave, Ondrej a Ivan na Gymnázium Grösslingová v Bratislave, Jakub a Juraj na Gymnázium Jura Hronca v Bratislave. Michal už neštuduje.

Daniela sa venuje vedľajším funkciám knižka a príklady, Peter je šéfom funkcie príklady a USP, Michal je šéfom funkcie knižka, Ondrej je šéfom obálkovania a členom USP, Ivan je šéfom propagácie a Jakub je členom príkladov je šéfom oRieška a členom okruhu komunikácia.

Zadané sú úlohy na pridávanie a kontrolu príkladov, a kontrolu gramatiky knižky 3. kola. Archivovaná je úloha kontroly gramatiky v knižke 2. kola. Kontrola gramatiky 3. kola je priradená Daniele a Michalovi, pridávanie príkladov je priradené skupine aktívnych vedúcich a kontrola príkladov je určená skupine príklady.

Práve prebieha 17. ročník súťaže. Doteraz každý ročník obsahoval zimnú a letnú sériu seminára. Posledný ročník navyše obsahuje zimnú a letnú sériu USP. V 17. ročníku prebehlo zimné a letné sústredenie, dva týždne denného tábora a letný tábor. Na začiatku zimnej aj letnej série prebehla propagácia na školách.

Za organizáciu oboch propagácií je zodpovedný Ivan. Za zimné sústredenie je zodpovedná Daniela, Peter, Michal a Ondrej. Za letné sústredenie zatiaľ nie je zodpovedný nikto, práve totiž prebiehajú voľby. Vo voľbách môžu kandidovať organizátori zo skupiny priateľov, vedúcich aj šéfov, hlasovať však môžu iba vedúci. Zatiaľ sú prihlásení kandidáti Peter, Ivan a Ondrej.

Pridané časti programu zimného sústredenia sú raňajky a rozevička každý deň. V propagácii zimnej série je priradené propagovanie na Gymnázium Grösslingová v Bratislave a Gymnázium Jura Hronca.

Ďalej sú v poslednom ročníku pridané dve čísla časopisu oRiešok, prvé z nich už obsahuje články úvod, vtipy a príbeh.

Posledná séria seminára obsahuje 3 kolá. Zimnú sériu rieši Ján Nový a Mária Dobrá. Mária navyše rieši aj kategóriu USP. Jánovi sa v 1. kole darilo, za všetky tri odovzdané riešenia získal 10 bodov. Mária odovzdala 4 príklady a získala za ne 0, 5, 7 a 4 body.

Opravovania riešení 1. príkladu sa zúčastnil Peter, Daniela, Ivan a Jakub, 2. príklad opravoval Michal. Peter napísal komentár k opravovaniu o Ivanovi, Daniela o Jakubovi.

Zimná séria obsahuje 2 knižky. Prvá z nich obsahuje zadania 1. kola, vzorové riešenia 2. kola, ďalší článok, ktorému je priradená šablóna článku a článok, ktorý je vyplnený priamo. Zadania 1. kola obsahujú dva geometrické príklady o kocke. Vzorové riešenie bolo pridané zatiaľ k prvému z nich. V jednotlivých kolách sú nastavené termíny odovzdania riešení. Za zimnú sériu bol vytvorený diplom podľa šablóny.

5.1.2 Testy kontrolerov

Cieľom testov bolo skontrolovať, že sa všetky metódy vykonajú podľa predpokladu a že stránka bude presmerovaná na správne miesto, prípadne že záznam bol skutočne pridaný. Tieto testy sú vytvorené pre všetky kontrolery systému.

5.1.3 Testy modelov

Cieľom testov modelov bolo skontrolovať správnosť validácie údajov. Tiež tieto testy kontrolovali, že po vymazaní záznamov triedy boli vymazané aj súvisiace záznamy z tried, ktoré boli v modeli nastavené ako závislé na danej triede. Tieto testy sú vytvorené pre všetky modely systému.

5.1.4 Testy mailov

Testy mailov v tomto informačnom systéme prebiehajú na úrovni zobrazenia vzorovej správy. Správu je možné zobrazit vo formáte HTML alebo čistého textu.

5.2 Testovanie použiteľnosti

Cieľom tohto testovania bolo zistiť, či užívatelia informačného systému budú vedieť vykonávať zadané úlohy, ďalej trvanie a celkovú spokojnosť s rozhraním. Testovania systému sa zúčastnilo celkom 6 organizátorov (z toho 1 s hlavnou funkciou šéf, 3 vedúci a 2 priatelia). Prebiehalo vo forme osobného stretnutia.

Organizátori dostali zoznam 20 úloh. Úlohy na seba naväzovali a boli vyplňané postupne. Niektoré sa týkali iba organizátorov, ktorých hlavná funkcia je vedúci alebo šéf, alebo sú samostatnými opravovateľmi. Ak v úlohe nebolo uvedené inak, organizátori mali vybrať 17. ročník, zimnú sériu, 1. kolo súťaže. Zoznam úloh:

1. Bolo vám vytvorené konto. Skontrolujte svoju e-mailovú schránku a prihláste sa do systému.
2. Zmeňte svoje heslo.
3. Zadaajte sami sebe úlohu na pridanie príkladu s dnešným termínom.
4. Archivujte túto úlohu a zobrazte archív úloh.

5. TESTOVANIE

5. Pridajte príklad do databázy príkladov. Je určený do náboja. Určte sami seba za autora príkladu.
6. Vygenerujte databázu nepozretých príkladov do náboja a zobrazte výsledné pdf. Potom pdf zmažte.
7. Zobrazte štatistiku pridaných príkladov za posledný rok.
8. Zadaťte svoj príklad do zadaní 1. kola.
9. Upravte zadania príkladov a termín 1. kola. Pridajte vzorové riešenie ku svojmu príkladu.
10. Prihláste sa na opravovanie niektorého príkladu.
11. Pridajte prihlášku riešiteľa série, ak viete nasledujúce údaje:
Meno a priezvisko: Peter Nový,
adresa: Biela 5, Bratislava,
škola: GAMČA,
trieda: Sekunda,
riešiteľ rieši elektronicky,
e-mail riešiteľa: `fil.sakala@gmail.com`.
12. Zapište prijaté riešenia tohto riešiteľa. Riešiteľ v 1. kole odoslal všetky príklady.
13. Za príklad, ktorý opravujete, mu udeľujete 5 bodov.
14. Vytvorte jednoduchú šablónu diplomu. Použite aspoň niektoré doporučené makrá.
15. Vytvorte diplom za sériu, použite vytvorenú šablónu.
16. Vytvorte akciu: Letné sústreďenie.
17. Vedúci alebo šéf: Vytvorte voľby na túto akciu.
18. Pridajte článok do 1. čísla oRieška. Po pridaní ho označte za skontrolovaný.
19. Odošlite propagačné materiály e-mailom na školy GAMČA a GJH. Po- nechajte vyplnené e-maily škôl bez zmeny.
20. Samostatný opravovateľ: Pozrite si históriu svojich opravovaní. Skontrolujte, že ste splnili všetky povinnosti samostatného opravovateľa.

Hodnotenie tohto testovania je pozitívne, väčšina úloh bola splnená bez potreby pomoci. Organizátori teda pojmy v odkazoch využívali tak, ako bolo predpokladané. Problémy, ktoré sa vyskytli, možno zhrnúť v nasledujúcich bodoch:

- Chýbajúca navigácia na stránke v každom momente. Užívateľ sa pri zanorení v určitej časti webovej aplikácie (napríklad ročník, séria, kolo, rozpis opravovania) radšej dostal na hlavnú stránku a z nej sa znova dostal o úroveň vyššie.
- Chýbajúce zvýraznenie základných funkcií, napríklad pridávanie príkladov. Hľadanie správneho odkazu, aj keď na správnej stránke, trvalo dlho.
- Nastavenie termínu úlohy je neintuitívne. Užívatelia síce vybrali termín, no nevybrali pole „Nastaviť“. Termín sa tak neuložil.
- Pri vymazaní úlohy z archívu sa stránka nelogicky presmeruje na všetky zadané úlohy.
- Problém s výberom viacerých súborov alebo záznamov (napríklad priradenie úlohy viacerým organizátorom).

Dané problémy sú na úrovni návrhu rozhrania a implementácie. Všetky z týchto problémov sú vyriešiteľné. Riešenie obsahuje:

- Pridanie navigácie do hornej časti stránky tak, aby v každom momente bolo možné zistiť jej pozíciu a vrátiť sa do nadradených stránok.
- Zvýraznenie dôležitých odkazov (napríklad pridaním `css` štýlov).
- Problém s nastavovaním termínu úlohy je možné vyriešiť odstránením poľa „Nastaviť“ a nahradenie prázdny polom na výber dátumov, ktoré značí nezadaný termín.
- Presmerovanie stránky po vymazaní úlohy z archívu môže byť zmenené na archív.
- Organizátor môže byť pri výbere viacerých záznamov naraz sprevádzaný krátkym návodom typu: „Pri výbere viacerých organizátorov držte kláves `Ctrl`.“

5.3 Testovanie behu vo webových prehliadačoch

Funkčnosť výsledného informačného systému bola otestovaná vo viacerých webových prehliadačoch. Zoznam prehliadačov na testovanie bol vybraný z najviac používaných podľa [9]:

- Google Chrome (verzia 42.0.2311.90) [10],
- Internet Explorer (verzia 11.0.9600.17501) [11],
- Mozilla Firefox (veria 37.0.2)[12].

V rámci testovania bola vybraná vzorka úloh organizátorov:

- Prihlásenie do systému,
- pridanie a priradenie úlohy organizátorovi,
- pridanie príkladu a jeho zaradenie do knihy,
- vyhľadávanie v zozname organizátorov a zozname príkladov,
- odosielanie propagačných materiálov na školy,
- pridávanie komentárov o opravovateľoch,
- zadávanie bodov riešiteľom,
- pridávanie akcie a jej časti programu a súborov.

Sledované bolo zobrazenie systému v priebehu vykonávania úlohy a funkčnosť. V priebehu testovania neboli zistené výraznejšie rozdiely vo fungovaní medzi jednotlivými prehliadačmi. Pozorované zmeny boli iba vo vizuálnom zobrazení prvkov stránky (tlačidiel, formulárov, odkazov a podobne).

Záver

Na základe zadania tejto práce bol dôkladne preskúmaný a popísaný aktuálny stav organizácie a jej postupov. V priebehu analýzy boli ďalej vytvorené funkčné a nefunkčné požiadavky na systém a prípady použitia spolu s detailnými scenármi. Na konci analýzy boli vytvorené analytické triedy. Všetky tieto kroky prebiehali po komunikácii s organizátormi súťaže. Samotné výstupy analýzy boli odobrené.

Počas návrhu systému bol vybraný spôsob ukladania dát a implementačný jazyk. Analytické triedy boli upresnené na návrhové a tie boli rozdelené do balíčkov. Tiež bolo navrhnuté užívateľské rozhranie vznikajúcej webovej aplikácie.

Vďaka dôkladne vykonanej analýze a návrhu bola implementácia systému uľahčená, nie však triviálna. V priebehu implementácie bolo riešených mnoho zaujímavých problémov. Výsledná webová aplikácia bola nakoniec testovaná automatizovanými testami, ale aj samotnými organizátormi súťaže.

Stanovené ciele práce boli splnené. Bol vytvorený nástroj na celkovú organizáciu súťaže. Testovanie organizátormi ukázalo, že systém je použiteľný bez úvodného zaškolenia o jeho použití.

Samotní organizátori vyjadrili potešenie, že už viac nebudú musieť na organizáciu používať ďalší softvér a vystačia si s týmto informačným systémom, v ktorom nájdu všetky funkcie na jednom mieste. Systém navyše kontroluje zadávané vstupy. Preto možno hovoriť o budúcej eliminácii chýb spôsobených jednotlivcom.

Organizátori do budúca vyjadrili zvýšený záujem o rozširovanie a zlepšovanie funkcií tohto systému. V priebehu testovania sa navyše o vytvorení systému dozvedeli aj zástupcovia podobných korešpondenčných seminárov a zaujímali sa o jeho využiteľnosť vo svojich organizáciách. Tento záujem otvára dvere ďalšiemu rozvoju a napríklad aj vytvoreniu všeobecného systému na organizáciu takýchto seminárov s použitím funkcií tohto systému.

Rozšíriteľnosť systému

Už v priebehu vytvárania tejto bakalárskej práce prichádzali od organizátorov súťaže nápady na rozšírenie informačného systému. Tie však už v danom rozpracovaní práce nebolo možné zahrnúť do výslednej implementácie. S týmito rozšíreniami je však rátané do budúcnosti. Nápady na ďalšie rozšírenie systému sú:

- Admin panel: Možnosť celkovej administrácie systému, prehľadná správa jednotlivých tried a súborov.
- Rozšírenie zoznamu úloh: Úloha môže byť zadaná automaticky pri vytvorení ľubovoľného termínu v systéme alebo pri inej činnosti (napríklad zodpovednosť za akciu, termín prihlasovania a hlasovania vo voľbách, kontrola časopisu oRiešok alebo knižky a podobne). Blížiaci sa termín úlohy môže odoslať upozornenie na e-mail organizátora.
- Prepojenie systému so skladoom (systémom `subversion`): Možnosť zjednodušenej správy verzií súborov.
- Možnosť vytvorenia obálok, v ktorých majú byť odoslané riešenia poštou.
- Zadávanie prihlášok riešiteľov hromadne nahrávaním z tabuľkových procesorov (napríklad Microsoft Excel [26]).
- Vylepšená správa verzií (optimistické zamykanie): Pri kolízii sa práca užívateľa dočasne uloží a nevymaže. Užívateľ tak nepríde o vykonanú prácu.
- Vytvorenie štýlov na tlač programu propagácie alebo akcie.
- Vytvorenie $\text{T}_{\text{E}}\text{X}$ alebo $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ makier časopisu oRiešok a zavedenie generovania časopisu v systéme.

Literatúra

- [1] *O nás – Riešky* [online]. [vid. 01.05.2015]. Dostupné z: <http://riesky.sk/o-nas/>.
- [2] ARLOW J., NEUSTADT I. *UML2 a unifikovaný proces vývoje aplikáci*. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
- [3] SCHNEIDER, R. D. *MySQL: oficiální průvodce tvorbou, správou a laděním databází*. 1. vyd. Praha: Grada, 2006. ISBN 80-247-1516-3.
- [4] RUBY S., THOMAS D., HANSSON D. H. *Ruby on Rails: průvodce agilním vývojem webových aplikací*. 1. vyd. Brno: Computer Press, 2011. ISBN 978-80-251-3647-8.
- [5] THE APACHE SOFTWARE FOUNDATION. *ApacheTMSubversion[®]* [software]. [přístup 01.05.2015]. Dostupné z: <https://subversion.apache.org/>.
- [6] AUER M., TSCHURTSCHENTHALER T., MEYER L. *UMLet* [software]. [přístup 13.12.2014]. Dostupné z: <http://www.umlet.com/>.
- [7] HANSSON D. H., *Ruby on Rails* [software]. [přístup 21.02.2015]. Dostupné z: <http://rubyonrails.org/>.
- [8] *Rails Routing from the Outside In* [online]. [vid. 21.02.2015]. Dostupné z: <http://guides.rubyonrails.org/routing.html>.
- [9] *Browser Statistics* [online]. [vid. 01.05.2015]. Dostupné z: http://www.w3schools.com/browsers/browsers_stats.asp.
- [10] GOOGLE INC., *Chrome* [software]. [přístup 01.05.2015]. Dostupné z: <https://www.google.cz/chrome/browser/desktop/>.

- [11] MICROSOFT CORPORATION, *Internet Explorer* [software]. [prístup 01.05.2015]. Dostupné z: <https://www.microsoft.com/cs-cz/download/internet-explorer.aspx>.
- [12] MOZILLA FOUNDATION, *Mozilla Firefox* [software]. [prístup 01.05.2015]. Dostupné z: <https://www.mozilla.org/cs/firefox/new/>.
- [13] EVOLUS, *Evolus Pencil* [software]. [prístup 25.03.2015]. Dostupné z: <http://pencil.evolus.vn/>.
- [14] LEVI, I. S., *CSSTableGenerator* [software]. [prístup 01.04.2015]. Dostupné z: <http://www.csstablegenerator.com/>.
- [15] *Getting Started with Rails* [online]. [vid. 21.02.2015]. Dostupné z: http://guides.rubyonrails.org/getting_started.html.
- [16] *Active Record Migrations* [online]. [vid. 21.02.2015]. Dostupné z: http://edgeguides.rubyonrails.org/active_record_migrations.html.
- [17] *Active Record Validations* [online]. [vid. 21.02.2015]. Dostupné z: http://edgeguides.rubyonrails.org/active_record_validations.html.
- [18] *ActiveRecord::Locking::Optimistic* [online]. [vid. 20.04.2015]. Dostupné z: <http://api.rubyonrails.org/classes/ActiveRecord/Locking/Optimistic.html>.
- [19] *Action Mailer Basics* [online]. [vid. 15.04.2015]. Dostupné z: http://edgeguides.rubyonrails.org/action_mailer_basics.html.
- [20] *A Guide to Testing Rails Applications* [online]. [vid. 21.05.2015]. Dostupné z: <http://guides.rubyonrails.org/testing.html>.
- [21] PHUSION, *Phusion Passenger* [software]. [prístup 03.05.2015]. Dostupné z: <https://www.phusionpassenger.com/>.
- [22] CLEMENTS T., HAMBLEY L. *Capistrano* [software]. [prístup 03.05.2015]. Dostupné z: <http://capistranorb.com/>.
- [23] ROQUES A., *PlantUML* [software]. [prístup 29.04.2015]. Dostupné z: <http://plantuml.sourceforge.net/>.
- [24] SOFTWARE FREEDOM CONSERVANCY, *Git* [software]. [prístup 09.02.2015]. Dostupné z: <http://git-scm.com/>.
- [25] SAGALAEV I., *highlight.js* [software]. [prístup 30.04.2015]. Dostupné z: <https://highlightjs.org/>.
- [26] MICROSOFT CORPORATION, *Microsoft Office Excel* [software]. [prístup 01.05.2015]. Dostupné z: <https://products.office.com/cs-cz/excel>.

Zoznam použitých skratiek

CSS Cascading Style Sheets

DRY Don't repeat yourself

HTML HyperText Markup Language

MVC Model-View-Controller

PDF Portable Document Format

SVN Apache Subversion

UML Unified Modeling Language

Slovník pojmov

Korešpondenčný seminár Súťaž prebiehajúca na diaľku formou korešpondencie (poštou, elektronickou formou).

Ročník Časový úsek predstavujúci jeden organizačný rok v súťaži.

Séria Časť ročníka.

Kolo Časť série obsahujúca zadania, vzorové riešenia a výsledkovú listinu.

Knižka Časopis so zadaniami, riešeniami príkladov a podobne.

Sklad Verzovací systém SVN, v ktorom sú fyzicky uložené súbory organizácie.

Interné Stránky, na ktorých sú zapísané informácie a vedomosti o jednotlivých funkciách, databáza príkladov, zadávanie úloh a pod.

oRiešok Časopis s rôznymi populárno-náučnými témami.

Wireframe Znázornenie návrhu užívateľského rozhrania.

Inštalačná príručka

Táto príručka popisuje spôsoby inštalácie systému a použité verzie jednotlivých súčastí.

C.1 Verzie

Pri spúšťaní systému v produkčnom, testovacom aj vývojovom prostredí boli použité nasledujúce verzie programov:

- ruby 2.2.0p0
- Rails 4.2.0
- Verzie balíčkov sú popísané v súbore Gemfile.
- rake, version 10.4.2
- mysql v. 14.14 Distrib 5.6.19
- Bundler version 1.8.2
- Phusion Passenger version 5.0.7
- Capistrano Version: 3.4.0
- gem v. 2.4.6
- rvm 1.26.11
- Apache/2.4.9 (Debian)

C.2 Spustenie aplikácie vo vývojovom prostredí

Nasledujúci postup popisuje spustenie aplikácie vo vývojovom prostredí, vo webovom prehliadači.

1. Umiestniť súbory aplikácie do požadovanej zložky. Nainštalovať potrebné programy a ich súčasti. Nastaviť zložkám `tmp` a `log` prístup na zapisovanie.
2. Vytvoriť databázu MySQL s názvom `isriesky_development` a prístup pre užívateľa `isrieskar` s heslom `riesky`.
3. Spustiť príkaz `bundle install` a riadiť sa zobrazenými pokynmi. Príkaz inštaluje do aplikácie chýbajúce balíčky. Balíčky musia byť nainštalované pomocou `gem install názov`
4. Spustiť príkaz `rake db:migrate db:seed`, ktorý nahraje obsah databázy.
5. Spustiť príkaz `rails s`, ktorý spustí webový server.
6. Zobrazíť vo webovom prehliadači adresu `localhost:3000`.

C.3 Postup nasadenia aplikácie

Na serveri organizácie je nastavená adresa `http://riesky.sk/is` ako adresa informačného systému. Nastavenia nasadenia sú uložené v aplikácii v súbore `config/deploy.rb`. Samotné nasadenie môže byť spustené pomocou príkazu:

```
rvmsudo cap production deploy deploy:restart
```

Tento príkaz si vyžiada viacnásobné zadanie hesla. Príkaz `deploy` začne preberanie z repozitára `git` na server aplikácie, vytvorí najnovšiu verziu od poslednej vytvorenej a nakoniec nastaví odkaz `current` na najnovšiu verziu aplikácie. Tento odkaz je použitý aj v zobrazení informačného systému. Príkaz `deploy:restart` vytvorí alebo prepíše súbor `tmp/restart.txt` a spôsobí, že nástroj Phusion Passenger reštartuje aplikáciu tak, aby bola zobrazená najnovšia verzia.

C.4 Testovací prístup do aplikácie

Na testovacie účely je v systéme vytvorené samostatné konto. Je možné použiť nasledujúce údaje:

Prihlasovanie prebieha cez stránku `http://riesky.sk/is/login`.

Prihlasovacie meno: `test`, heslo: `testtest`.

Obsah priloženého CD

	readme.txt.....	stručný popis obsahu CD
	impl	zdrojové kódy implementácie
	thesis	text práce
	src	zdrojové kódy práce vo formáte \LaTeX
	readme.txt	pokyny na generovanie PDF
	analytickyModel	zdrojový kód analytického modelu tried
	navrhovyModel	zdrojový kód návrhového modelu tried
	output	výstupy generovania a obrázky
	obr2	obrázok priebehu ročníka
	uml	zdrojové kódy a vygenerované diagramy UML
	wf	vygenerované obrázky wireframes
	text	text práce
	BP_Sakala_Filip_2015.pdf	text práce vo formáte PDF