



Pedagogická
fakulta
Faculty
of Education

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Pedagogická fakulta

Mobilní soutěžní aplikace Bobřík informatiky

Bakalářská práce

Vypracoval: Jan Hamerník

Vedoucí práce: Mgr. Václav Šimandl, Ph.D.

České Budějovice 2016

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Fakulta pedagogická

Akademický rok: 2014/2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan HAMERNÍK**
Osobní číslo: **P13100**
Studijní program: **B7507 Specializace v pedagogice**
Studijní obor: **Informační technologie a e-learning**
Název tématu: **Mobilní soutěžní aplikace pro OS Android**
Zadávající katedra: **Katedra informatiky**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je navrhnout a vytvořit aplikaci pro soutěž Bobřík informatiky, která bude určena pro mobilní zařízení s operačním systémem Android. Aplikace bude umožňovat soutěžení prostřednictvím mobilního telefonu, přičemž potřebná data budou načítána z databáze soutěže, kam budou také ukládána data vzniklá během soutěžení. Student navrhne a realizuje automatickou úpravu obrázků, které jsou součástí úloh, tak, aby je bylo možno pohodlně prohlížet na mobilním zařízení. Student dále otestuje možnosti zobrazování úloh naprogramovaných pomocí HTML5 a případně navrhne jejich optimalizaci pro využití v mobilní soutěžní aplikaci. Po dokončení bude tato aplikace umístěna na Google Play a následně bude její funkčnost otestována na vybrané škole. V teoretické části práce student provede analýzu několika existujících soutěžních mobilních aplikací a dále popíše aspekty fungování soutěže Bobřík informatiky, které jsou významné pro tvorbu aplikace.

Rozsah grafických prací: CD ROM

Rozsah pracovní zprávy: 40

Forma zpracování bakalářské práce: tištěná

Seznam odborné literatury:

1. LACKO, L'. Vývoj aplikací pro Android. 1. vyd. Brno: Computer Press, 2015, 472 s. ISBN 978-80-251-4347-6.
2. VÁVRŮ, J. a M. UJBÁNYAI. Programujeme pro Android. 2., rozš. vyd. Praha: Grada, 2013, 250 s. Průvodce (Grada). ISBN 978-80-247-4863-4.
3. LACKO, L'. 1001 tipů a triků pro SQL. Vyd. 1. Brno: Computer Press, 2011, 416 s. ISBN 978-80-251-3010-0.
4. TOMCSÁNYI, P. a J. VANÍČEK. Implementation of informatics contest Bebras in Czechia and Slovakia. In: Information and communication technology in education. Proceedings. Ostrava: Ostravská univerzita, 2009. s. 214-218. ISBN 80-7368-459-4.
5. VANÍČEK, J. Rozvoj informatických kompetencí žáků pomocí soutěže: kauza Bobřík informatiky. In: Sborník konference Počítač ve škole. Nové Město na Moravě: Gymnázium Vincence Makovského, 7.-9. 4. 2009
6. VANÍČEK, J. Potenciální a skutečný dopad informatické soutěže do změn kurikula ICT v České republice. In: DidInfo 2012 [CD-ROM]. Banská Bystrica: Univerzita Mateja Bela, Fakulta přírodních vied v Banskej Bystrici, 2012. s. 15-24. ISBN 978-80-557-0342-8.

Vedoucí bakalářské práce:

Mgr. Václav ŠIMANDL

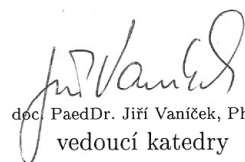
Katedra informatiky

Datum zadání bakalářské práce: 29. dubna 2015

Termín odevzdání bakalářské práce: 29. dubna 2016



Mgr. Michal Vančura, Ph.D.
děkan



doc. PaedDr. Jiří Vaníček, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 29. dubna 2015

Anotace

Tato bakalářská práce pojednává o vytvoření soutěžní aplikace pro zařízení s operačním systémem Android. Aplikace umožňuje zúčastnit se soutěže Bobřík informatiky. Aplikace je zaměřena hlavně na uživatele mobilních telefonů. Všechna potřebná data jsou načítána z databáze soutěže a do této databáze jsou také ukládána data vzniklá během soutěžení. Zobrazení otázek a úprava velikosti obrázků, které jsou součástí úloh tak, aby je bylo možno pohodlně prohlížet na mobilním zařízení. Tato úprava byla vytvořena pomocí kaskádových stylů. Byl proveden test možnosti zobrazování úloh naprogramovaných pomocí HTML5 a taktéž návrh na úpravy řešení pro budoucí použití. Po dokončení je tato aplikace ke stažení na Google Play. Funkčnost byla otestována na vybrané škole. V teoretické části práce jsou popsány aspekty významné pro tvorbu aplikace.

Klíčová slova

Android, Mobilní technologie, Soutěž, Bobřík informatiky, JAVA, PHP

Anotace

The aim of this bachelor thesis is design and develop application contest Bobřík informatics, which is developed for mobile devices with operation system Android. Application allow compete with mobile devices. Data of contest are downloading and uploading to database of contest. In this contest was designed and developed process for automatic editing images for displays of mobile devices. Next task was testing HTML 5. It was designed and tested process of optimization for use in mobile version of contest. When this application was completed it was uploaded to Google Play and it was tested at school. In theoretical part of this thesis are described important aspects for developing of application.

Keywords

Android, Mobile technology, Competition, JAVA, PHP

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 23. prosince 2016

Jan Hamerník

Poděkování

Na tomto místě bych rád poděkoval panu doktoru Václavu Šimandlovi za vedení mé práce, odbornou pomoc a cenné připomínky.

Obsah

1	Úvod	10
1.1	Cíle práce	10
1.2	Metody práce	11
2	Teoretická část	12
2.1	Soutěž Bobřík informatiky	12
2.2	Podvádění v soutěžích	13
2.3	Jiné soutěžní aplikace	14
2.4	Historie operačního systému Android	16
2.5	Využití operačního systému Android	16
2.6	Programování pro Android aneb odlišnosti od jiných programova- cích jazyků	17
2.7	Příprava pro zahájení vývoje aplikace	18
2.8	Volání komponent	19
2.9	Testování - virtuální stroj	20
2.10	Použitelné aktivity a stavy aktivit	21
2.10.1	Použitelné aktivity	21
2.10.2	Stavy aktivit	22
2.11	API	23
2.12	Publikování softwaru přes Google Play	24
3	Praktická část	26
3.1	Vytvoření návrhu	26
3.1.1	Úvodní obrazovka	26
3.1.2	Výběr kategorie a registrace	27
3.1.3	Obrazovka testu	29
3.2	Vytvoření stabilního připojení k databázi	30
3.3	Vytvoření modulu pro zobrazování dat z databáze	33
3.4	Otestování úloh	33
3.5	Změna velikosti obrázků	34
3.6	Technologie Flash	35
3.7	Podpora HTML5	35

3.8	Grafické rozhraní	36
3.9	Otestování	37
4	Závěr	39
	Literatura.	41

1 Úvod

Pro vypracování této práce jsem se rozhodl z důvodu mého zájmu o mobilní technologie. Dále jsem si chtěl prohloubit znalosti s programováním pro operační systém Android a zaujala mě možnost vytvářet software, který bude používán pro soutěžení v soutěži Bobřík informatiky, kterou provozuje Jihočeská univerzita v Českých Budějovicích. Univerzita nyní provozuje soutěž jako webovou aplikaci zejména tedy pro počítače.

Mobilní aplikace jsou v dnešní době velice oblíbené, a to zejména kvůli tomu, že je má uživatel připravené ve svém mobilním telefonu a po otevření aplikace může ihned začít pracovat, aniž by musel něco hledat. Další důvod, proč vyvíjet mobilní aplikace, je úspora dat, které telefon musí přenášet. Zatímco u webových aplikací se v některých případech přenáší některá data opakovaně, u mobilních aplikací se přenáší jen minimum dat a zbytek je uložen v telefonu. Pokud se zaměříme na soutěžní aplikaci Bobřík informatiky, přenáší se vždy pouze otázka a její odpovědi, zatímco ostatní komponenty jako například ovládací prvky, čísla otázek, tlačítka, barva pozadí jsou uloženy přímo v mobilní aplikaci. Další výhodou mobilní aplikace je ta, že pokud se uživateli při práci třeba vybije mobilní telefon nebo se mu restartuje z nedostatku operační paměti, tak nemusí přes webový prohlížeč znova hledat webovou stránku, ale pouze se otevře soutěžní aplikaci ve svém telefonu a po zadání identifikačních údajů může ihned pokračovat v práci.

1.1 Cíle práce

Cílem bakalářské práce je navrhnout a vytvořit aplikaci soutěž Bobřík informatiky, která bude určena pro mobilní zařízení s operačním systémem Android.

Dílní cíle práce jsou:

- Vytvořit modul pro komunikaci s databází.
- Zajistit zobrazení obrázkových úloh na mobilních obrazovkách, které jsou součástí úloh tak, aby je bylo možno pohodlně prohlížet na mobilním zařízení.
- Otestovat a zhodnotit možnosti zobrazování úloh naprogramovaných pomocí HTML5 a případně navrhnout proces optimalizace pro využití v mobilní

soutěžní aplikaci.

- Po dokončení všech prací umístit aplikaci na Google Play ke stažení pro uživatele
- Otestovat soutěžní aplikaci při soutěži na škole a vyhodnotit nasbíranou zpětnou vazbu.

1.2 Metody práce

Nejprve jsem nastudoval, jak efektivně programovat pro mobilní zařízení s operačním systémem Android. Následně se seznámil blíže s programovacím jazykem Androidu a jeho příkazy. Další na řadě bylo vytvořit první spustitelnou verzi aplikace a otestování stability. Hned jak to bylo hotové, začal jsem vytvářet modul pro komunikaci s databází. Nejprve bylo otestováno čtení a později i zápis. Dále jsem vyvíjel grafické rozhraní a testoval v něm zobrazení otázek bez obrázků. Navrhl a realizoval jsem řešení, jak efektivně zobrazit obrázkové úlohy na mobilním displeji a vyřešil problémy různého rázu, které se vyskytly. Celkový test aplikace, zda všechno funguje tak, jak má a následně vyladil nedostatky z hlediska funkčnosti, tak i grafického rozhraní. V neposlední řadě byla aplikace otestována na vybrané škole, kde jsem získal cennou zpětnou vazbu od několika studentů. Všechny připomínky a různé nejasnosti jsou důležitou zkušeností a pomohly vylepšit fungování aplikace. Aplikaci jsem publikoval, tedy její finální verzi na Google Play ke snadnému stažení a používání pro veřejnost.

2 Teoretická část

2.1 Soutěž Bobřík informatiky

Soutěž Bobřík informatiky pořádá Jihočeská univerzita v Českých Budějovicích, a to vždy na podzim. Soutěž je určena pro všechny základní a střední školy, které se do soutěže přihlásí. Registrace není nijak omezená, proto se může přihlásit opravdu jakákoli škola. Soutěž je zaměřena na digitální gramotnost a celkově obor informačních technologií. Soutěžící narazí na otázky z algoritmizace, které mu přiblíží a rozšíří představu pod pojmem programování. Dále soutěž obsahuje různé úlohy, kde je potřeba se zamyslet a porozumět informacím. Jedná se o úlohy, kde informace jsou kódované a soutěžící má za úkol tyto informace rozkódovat a následně s nimi pracovat a vyřešit úkol. Další kategorií jsou logické úkoly, kde soutěžící musí najít určitou strategii, jak se s úkolem vypořádat. Tyto úkoly jsou přípravou na matematické základy informatiky.[1]

V současné době soutěž probíhá tak, že soutěžící otevře soutěžní webovou aplikaci, ve které vyplní údaje nutné pro registraci a vybere si kategorii, ve které bude soutěžit. Pro každou kategorii je určen časový interval, kdy lze soutěžit a kolik je času na vyplnění. Pokud soutěžící zvolí kategorii Mini, která je určená pro první stupeň základní školy, zobrazí se mu v testu 12 otázek a na vypracování testu bude mít 30 minut. Pokud zvolí soutěžící jakoukoliv jinou kategorii, zobrazí se mu 15 otázek a časový limit je 40 minut. U každé otázky se nachází ve spodní polovině obrazovky buď textové pole, do kterého zapíše odpověď a následně stiskne tlačítko uložit, nebo je dole připraven box s odpověďmi ABCD, které stačí zaškrtnout a odpověď bude ihned uložena. Každá otázka má jinou obtížnost, a k tomu se vztahuje i bodové ohodnocení. Pokud soutěžící odpoví správně, bude mu určitý počet bodů přičtený, ale pokud bude odpověď špatná, body se mu naopak odečtou. Platí tedy, že čím obtížnější otázka, tím více bodů soutěžící může získat, pokud odpoví správně, ale i o více bodů přijde, pokud odpoví špatně. Pokud uživatel nechce odpovídat, zvolí odpověď „nechci odpovědět”. Pokud zaškrtně tuto možnost, nebudou mu žádné body přičteny ani strženy. Každý soutěžící začíná se startovním počtem 60 bodů. Maximální počet bodů, které je možné v testu získat, je 240 bodů a úspěšným řešitelem soutěže se stane ten, kdo získá alespoň 150 bodů. Výjimkou

je opět kategorie Mini, ve které se nachází pouze 12 otázek, a proto startovní počet bodů je 48. V testu je tedy možné získat až 192 bodů, přičemž úspěšný řešitel je ten, kdo dosáhl alespoň 120 bodů. Test je možné kdykoliv ukončit stisknutím tlačítka pro ukončení testu. Za dřívější ukončení soutěžního testu se však žádné body nepřidávají, protože v této soutěži nejde o čas, ale o to, aby se soutěžící zamyslel a pokusil se vyřešit všechny úlohy správně. Jedinou výjimkou je krajské kolo, ve kterém když nastane situace, že vícero soutěžících získá stejný počet bodů, určuje se jejich pořadí podle rychlejšího soutěžního času. Při soutěžení není povolena žádná nápověda, což znamená, že soutěžící nesmí hledat žádné informace na internetu a ani nesmí komunikovat s okolím. Navíc soutěžící nesmí soutěžit opakovaně, protože má na test jen jeden pokus. [2]

2.2 Podvádění v soutěžích

Podvádění v elektronických soutěžích je pro soutěžícího daleko jednodušší než při vyplňování klasického papírového testu. Pokud soutěžící vyplňuje klasický papírový test, jediným rizikem, které hrozí, je opsání odpovědi od svého spolužáka. Elektronická verze soutěže má svoje výhody i nevýhody. Mezi velkou výhodou patří to, že aplikace dokáže odpovědi otázek různě proházet, a tudíž opisování mezi spolužáky není tak jednoduché a navíc se zvyšuje šance, že opsaná odpověď bude spíše špatná. Na druhou stranu nevýhodou je, že má soutěžící v ruce nástroj na získávání informací. At' už žák sedí u počítače nebo má v ruce mobilní telefon či tablet, je pro vyučujícího o dost náročnější uhlídat, zda si informace nehledá někde na internetu nebo že si přes messenger nedopisuje s někým jiným. V tomto ohledu by měla mít mobilní aplikace výhodu oproti webové aplikaci. Ve webové aplikaci je jednoduché otevřít další záložku, ve které si uživatel může hledat informace nebo komunikovat s ostatními. Mobilní aplikaci by musel uživatel nejprve zavřít a takto zavřenou aplikaci operační systém Android po nějakém časovém limitu ukončí. Než by uživatel zapnul webový prohlížeč a vyhledal data, Android by pravděpodobně aplikaci ukončil a uživatel by se musel do ní přihlásit opětovným zadáním příjmení a soutěžního kódu. Tento způsob podvádění byl pro soutěžícího nepraktický a časově náročný. Soutěžní aplikace Bobřík informatiky v dnešní době obsahuje software, jenž se snaží různé podvody odhalit. Software sleduje u soutěžících na-

příklad čas vyplňování testu at' už nad hranici průměru či pod hranici průměru.[3] Mezi další podvody, které by byly možné, patří spuštění testu, následné uzavření prohlížeče, vrácení času počítače a opětovné otevření testu. Takto by uživatel získal na test více času, než bylo povoleno. Tím, že spustil test a vrátil čas například o dvě hodiny zpět, měl najednou na test místo 40 minut dvě hodiny a 40 minut. Těmto podvodům je zabráněno jak ve webové aplikaci, tak i v zamýšlené mobilní aplikaci Bobříka informatiky.

2.3 Jiné soutěžní aplikace

Před zahájením samotného programování nebo-li spíše vytváření grafického designu jsem se chtěl inspirovat jinde. Na doporučení mého vedoucího práce jsem se pokusil prozkoumat aplikaci Bebras, která je obdobná jako soutěž Bobřík informatiky, ale v Německu. Bohužel jsem si tuto aplikaci mohl prohlédnout pouze podle screenshotů a obrázků, protože když jsem se pokusil vyhledat aplikaci na Google Storu tedy na Google Play, instalace byla zamítnuta z důvodu, že se nenacházím v Německu, ale že jsem na území České republiky. Dostal jsem se ještě k dalšímu softwaru, sice nešlo úplně tak o soutěžní software jako software na získávání informací. Jednalo se o software výzkumné agentury STEM/MARK, která působí na území celé republiky a získává informace od široké veřejnosti. Se softwarem firmy STEM/MARK se lze setkat buďto při setkání se zaměstnancem této firmy a vyplní s námi dotazník přímo v terénu nebo se dotazník vyplňuje ve webové aplikaci z domova. Setkal jsem se s oběma verzemi softwaru této firmy, a i když to není soutěžní software, tak se mechanismem soutěžním aplikacím podobá. Na začátku po uživateli aplikace požaduje základní informace jako je jméno, region bydliště a popřípadě město. Následně po potvrzení registrace se zobrazí obrazovka s dotazníkem. V levé části je ukazatel času, jak dlouho už na dotazníku pracujete a níže je zobrazena otázka, u které aktuálně jste. V horní části je zobrazen nadpis a otázka. Pod tímto blokem se nachází okno s odpověďmi. V tomto okně jsou buď předpřipravené odpovědi ve stylu zaškrťovacích okének, popřípadě je zobrazen prostor pro zapsání textové odpovědi. V úplně spodní části obrazovky je zobrazeno, kolik procent již máte hotovo, což je znázorněno i graficky. Při dokončení poslední otázky se zobrazí okno s poděkováním a informací, že data byla odeslána.

nevýhodou tohoto řešení je, že pokud uživatel bude vyplňovat soutěžní test na velice malém displeji, tak se mu ještě o tento prostor zmenší místo pro zadání.

2.4 Historie operačního systému Android

Operační systém Android je jeden z nejmladších operačních systémů. Vznikl v roce 2003 ve společnosti Android Inc. ve Spojených státech amerických. Tato společnost začala vyvíjet software pro mobilní telefony. V roce 2005 společnost Google koupila stoprocentní podíl, a tak se společnost Android Inc. stala dceřinou společností Googlu. Tímto krokem Google udělal razantní krok směrem vpřed do vývoje mobilních technologií. Google koupí společnosti měl najednou hodně specialistů zabývajících se mobilními technologiemi, kteří předtím Googlu chyběli. Dne 5. listopadu 2007 byla vydána první verze operačního systému, čímž tímto dnem začal Google konkurovat na trhu ostatním operačním systémům. První verze systému obsahovala základní funkce. Postupně byly vydávány další verze, které přidávaly nové funkce a zároveň opravovaly chyby předchozích verzí. 5. listopadu 2007 také vzniklo sdružení firem Open handset Alliance. Toto sdružení zahrnovalo v době založení více než 30 významných firem, které se zabývaly výrobou telefonů či komponent do telefonu nebo softwarem. Cílem tohoto sdružení bylo vytvořit otevřené standardy pro mobilní zařízení. [4]

2.5 Využití operačního systému Android

Android je open source, což znamená, že za předpokladu dodržení pravidel softwaru open source je možné používat software zcela zdarma a v případě potřeby lze software i editovat. Operační systém Android je založen na jádře operačního systému Linux. Android zajišťuje zabezpečení systému jako celku, což znamená zajištění přístupu ke všem sensorům a komponentám telefonu, správce paměti, správa procesů, zajištění přístupu k síti. Pokud vývojář potřebuje později přistoupit například k sensorům, nepřistupuje přímo k funkcím jádra, ale přes API. Android byl primárně vyvinut pro mobilní zařízení a tablety. Pro výrobce mobilních telefonů je velice výhodné, že si mohou operační systém upravit podle svých představ a následně tento systém zdarma umístit a nainstalovat na své zařízení.

Možná právě tohle je důvod, proč se tento operační systém uchytil na trhu tak rychle.[4]

2.6 Programování pro Android aneb odlišnosti od jiných programovacích jazyků

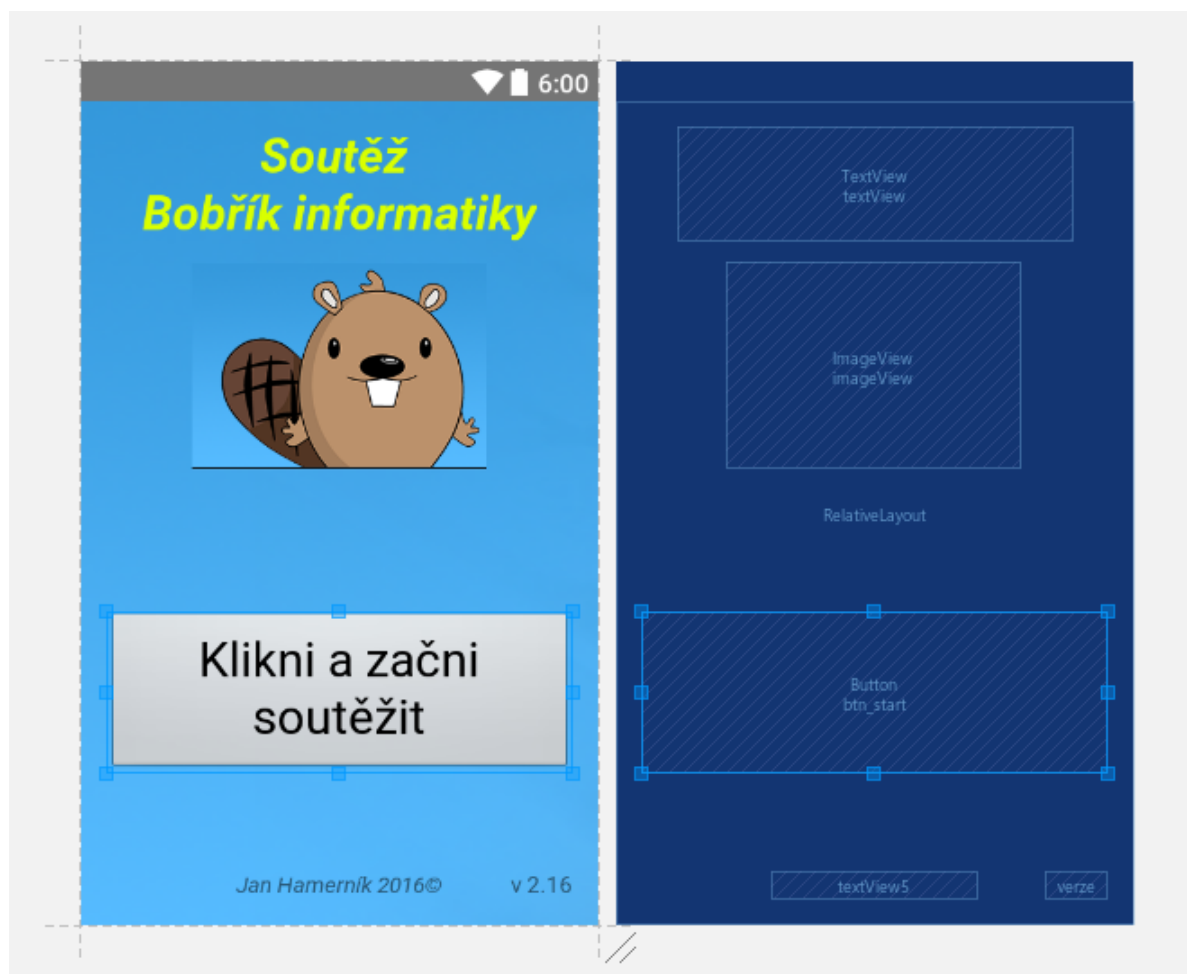
Při prvním pohledu na programovací jazyk a vývoj aplikací pro Android to vypadá, že vývojáři postačí znalost jazyka Java pro hlavní část zdrojového kódu. Dále je nutná znalost XML, právě v tomto jazyce se zapisuje umístění komponent a celkově vzhled aplikace. V neposlední řadě pomyslí vývojář na komunikaci s databází a jazyk SQL. Problémem však je, že Android neumí komunikovat s databází nebo alespoň ne přímo. Co to znamená v praxi? Operační systém Android má svou vlastní databázi v zařízení, do které může vývojář bez problémů ukládat informace a data. Tato data pak může jednoduše z databáze opět vytáhnout a dále s nimi pracovat stejně jako tomu je u jiných programovacích jazyků. Pokud potřebujeme přistupovat s Androidem k databázi, která je umístěna mimo zařízení, tedy přes internet, musí vývojář vyvinout vlastní komunikační prostředí. K tomu se používá komponenta WebView, která dokáže přes webový prohlížeč telefonu odeslat či přijmout data. Jak už napovídá použití webového prohlížeče, na druhé straně musí být rozhraní v jazyce PHP, které zpracuje požadavek a vrátí hodnoty pro zpětné zpracování do Androidu. Android nechá načíst webovou stránku, náš soubor PHP na webu s určitými parametry. V praxi to vypadá takto „<http://www.janhamernik.cz/zpracovani.php?data=xyz>”. Webový prohlížeč zpracuje data a použije je dle kódu zapsaného v PHP souboru. Pokud potřebujeme dostat data opačným směrem, komponenta WebView dokáže načíst ze stránky vrácenou hodnotu a uložit jí v Androidu do proměnné. Tato hodnota se uloží jako String a je potřeba jí rozparsovat nebo přetypovat pro další použití. Při vývoji rychle dochází ke zjištění, že kromě hojně používané Javy budete velice často muset použít jazyk PHP.

```
Intent intent = new Intent(Student.this, Core.class);
intent.putExtra("kategorie", kategorie);
intent.putExtra("sadaID_otazky", sadaID_otazky);
intent.putExtra("casSpustenehoTestu", casSpustenehoTestu);
intent.putExtra("idSouteziciho", idSouteziciho);
startActivity(intent);
```

Obrázek 2: Náhled kódu: přenos dat mezi aktivitami v OS Android

2.7 Příprava pro zahájení vývoje aplikace

Vývoj aplikace je ve většině případů dlouhá a náročná cesta. Proto je potřeba porovnat různé vývojové prostředí a vybrat to nejvhodnější. Volil jsem mezi dobře známým NetBeans a tehdy novým softwarem od Google, Android Studiím. NetBeans je stabilní vývojové prostředí a hodí na vývoj softwaru téměř ve všech programovacích jazycích. Nevýhodou vývoje aplikace pro Android v NetBeans je složitá instalace virtuálního stroje pro testování funkcí budoucí aplikace a složitá instalace debuggeru. Vývojové prostředí Android studio bylo na začátku vývoje nové a ne úplně stabilní. Navíc se tento software od Google velice rychle vyvíjel. Občas se tedy stalo, že kompilátor nechtěl kód zkompileovat, protože část kódu byla zastaralá. Nyní je Android Studio ve verzi dvě a dá se říci, že je to nejjednodušší a celkově nejefektivnější software na vývoj pro Android. Jak už je asi z textu zřejmé, zvolil jsem a doporučil bych vývoj v tomto nástroji od Google. Od této nové verze je po ruce vše, co vývojář potřebuje k práci. Od grafického návrhu bez nutnosti psaní kódu, přes definici názvů a není tak tedy problém navrhnout graficky aplikaci a pak jen vybrat, v jakém jazyce má být vše zobrazeno. V posledních verzích Android studia je grafický návrh aplikací ještě vylepšen tak, že vlevo vývojář vidí, jak bude aplikace vypadat a vpravo vidí návrh v režimu komponent.



Obrázek 3: Náhled návrhu aplikace v Android Studiu

2.8 Volání komponent

V jiných programovacích jazycích je obvyklé, že pokud si vložíš komponentu `textView` a následně jí zavoláš, tak s ní můžeš ihned pracovat. Za předpokladu, že náš `textView` má id `testovacíTextView`, tak by tedy mělo být možné použít zápis „`testovacíTextView.setText(„rezezecTextu“)`“, ale v Androidu tomu tak není. Android obsahuje třídu `R`, ve které je potřeba komponentu nejprve najít. Pokud tedy v Androidu budeme chtít použít naši komponentu `testovacíTextView`, musíme ji nejprve najít ve třídě `R`. Jako první si vytvoříme proměnnou s příslušným datovým typem, pokud tedy máme `textView`, tak voláme třídu `TextView` a vytvoříme

proměnnou `text`. Výsledek je `TextView text`. Nyní do této proměnné uložíme naší komponentu, abychom s ní mohli pracovat. Komponentu budeme hledat ve třídě `R` podle `id` a následně zapíšeme název hledané komponenty. Příkaz pro hledání komponenty je `findViewById()` a je nutné před něj do závorek napsat hledaný datový typ, v našem případě `TextView`. Celkový zápis vypadá následovně „`TextView text = (TextView)findViewById(R.id.testovacíTextView)`“. Teprve teď máme v proměnné `text` naší komponentu a můžeme s ní pracovat. Zápis pro změnu textu v komponentě tedy vypadá „`text.setText(„nový text“)`“. Toto je nejvýznamnější změna oproti jiných programovacím jazykům.

2.9 Testování - virtuální stroj

Jak už k programování patří, každý vývojář potřebuje svou aplikaci průběžně testovat. Velice důležité je zvolit správný nástroj, aby vývojáři usnadnil práci a nikoli naopak. Pro aplikace pro Android existuje různý vývojový software, ale u většiny si uživatel musí emulátor doinstalovat a nastavit sám. Při využití vývojového prostředí Android Studio, vyvíjené společností Google, si s touto problematikou ušetříte práci. Android Studio má vestavěný emulátor. Tento emulátor má různé výhody oproti ostatním emulátorům. Díky tomu, že je vyvíjen společně s Android studiem, má funkce, které dokáží urychlit práci. Pokud například testujete aplikaci a něco není podle vašich představ, je možné udělat okamžitou úpravu kódu a následně odeslat tento update do zařízení. Pokud Android Studio vyhodnotí úpravu jako minimální, pouze se obnoví aktuálně zobrazená aktivita. Pokud uděláte zásah větších rozměrů, spustí se vám celá aplikace od začátku.

Dříve měli uživatelé problém s instalací Android studia a konkrétně s instalací emulátoru, který nešlo nainstalovat kvůli chybě při instalaci (Intel HAXM installation error - This computer does not support Intel Virtualization Technology (VT-x)). Problém byl způsoben antivirovým programem Avast, kde bylo zapotřebí v nastavení programu nastavit parametr „Enable Hardware-assisted Virtualization“ na stav vypnuto[5]. Není jisté, jestli už je problém vyřešen, ale zajisté potrápil spoustu uživatelů. Pokud se však tento nástroj povede nainstalovat, tak velice ušetří práci. Pokud nepoužijeme virtuální stroj, je možné připojit reálné zařízení s Androidem a vše odesílat a testovat na něm.

Další problém nastal, když jsem se rozhodl pokračovat ve vývoji na jiném výkonnějším počítači. Hlavní rozdíl byl ten, že původně jsem aplikaci vyvíjel na počítači s procesorem od společnosti Intel a na novém zařízení, na kterém jsem později pracoval, byl osazen procesorem společnosti AMD. Při vývoji v Android Studiu na novém počítači bylo vše bez problémů a vše fungovalo tak jak má do té doby, než jsem se pokusil spustit aplikaci na emulátoru. Android Studio mi oznámilo, že emulátor nelze spustit, protože mi chybí hardwarová akcelerace. Při kliknutí na tlačítko troubleshooting neboli řešení problémů se zobrazila nabídka, ve které bylo zobrazeno několik řešení. Zaujalo mě hlavně řešení číslo dvě, kde bylo napsáno přesuňte si projekt na jiný počítač, který je osazen procesorem Intel. Pokud by vývojář rovnou začínal na počítači s procesorem AMD a byl to jeho jediný počítač, tak je to opravdu rada nad zlato. Na druhou stranu vývojář může využít k testování místo virtuálního stroje reálné zařízení, ale nebude to tak rychlé. Největší výhoda emulátoru zabudovaného v Android studiu je rychlost, protože dokáže do emulátoru po úpravě kódu odeslat pouze upravenou část a v některých případech se aplikace ani nespouští úplně od znova. Nebýt těchto zádrhelů s instalací a problému s procesory, byl by tento zabudovaný emulátor v Android studiu bezkonkurenčním nástrojem.

2.10 Použitelné aktivity a stavy aktivit

2.10.1 Použitelné aktivity

Aktivita je samostatně fungující celek a je spuštěná ihned po spuštění aplikace. Jedna aplikace se skládá z více aktivit a aktivity se mezi sebou přepínají a navzájem si předávají informace. Každá aktivita má vlastní grafické rozhraní, během kterého je uživatel může ovládat. Aktivita je navrhována tak, aby byla přehledná a pomohla uživateli obsloužit činnost, pro kterou je určena a poté předala informaci další aktivitě. Aktivity lze graficky nastavit tak, aby se přizpůsobovaly, ať už jsou zobrazeny na mobilním telefonu či na tabletu. Programování na Android se liší hlavně tím, že nemá žádnou hlavní metodu jako ostatní programovací jazyky, které mají metodu `Main()`. U Androidu definujeme hlavní aktivitu v souboru `manifest.xml`. Hlavní spuštěná aktivita pak může volat a spouštět jiné aktivity a předávat jim informace. Operační systém Android se sám stará o to, jaká aktivita

bude pouze odsunuta do pozadí a která bude zničena. Pokud chceme, aby se na aktivitu už nedalo v žádném případě vrátit, můžeme po zavolání jiné aktivity původní aktivitu zničit zavoláním příkazů `finish()`. Protože každá aktivita by měla fungovat samostatně, pokud je správně navržená, tak zničení předchodí aktivity by nemělo zapříčinit žádné potíže. Pokud po přechodu na jinou aktivitu nezavoláme metodu `finish()` a operační systém má k dispozici dostatek operační paměti, aktivita zůstane v paměťové oblasti, která se nazývá `BackStack`. Díky tomu, že aktivita zůstala v paměťové oblasti, tak pokud se na ní budeme přepínat v budoucnu zpátky, přepnutí nastane okamžitě bez prodlevy. Aktivitu můžeme rozdělit do tří skupin. [6]

Aktivita na popředí značí aktivitu, kterou právě uživatel vidí na displeji a může s ní pracovat. Pokud je tato aktivita spuštěná, může mít stav `running`, pokud bude obnovena z pozadí, bude mít stav `resumed`. [6]

Dalším typem je aktivita pozastavená. Tuto aktivitu uživatel pořád může vidět na displeji, ale je překrytá jiným dialogovým oknem. V soutěžní aplikaci Bobřík informatiky tato situace nastává v případě, když se soutěžící snaží ukončit test, aniž by měl vyplněné veškeré otázky. V tuto chvíli mu vyskočí vyskakovací okno, kde je hláška, zda chce opravdu ukončit test, i když nejsou vyplněné všechny otázky. Pod tímto dialogovým oknem i nadále vidí spuštěný test. [6]

Poslední kategorie je aktivita zastavená. Tuto aktivitu uživatel již na displeji nevidí, protože má na displeji jinou aktivní aktivitu. Zastavená aktivita však pořád zůstává v paměti, a to v již zmíněné oblasti `BackStack`. [6]

Narozdíl od aktivit můžeme ještě využít takzvané služby. Služba je část aplikace, která nemá grafické rozhraní a běží v pozadí aplikace, aniž by to uživatele zatěžovalo. Službu využijeme například, když uživatel potřebuje pracovat s aplikací, zatímco aplikace pracuje na nějakém složitém výpočtu nebo stahuje data z internetu.[6]

2.10.2 Stavy aktivit

Každá aktivita má svůj aktuální stav. Při vytvoření aktivity se volá metoda `onCreate()`. Do této metody může vývojář zapsat veškeré příkazy, volání jiných metod, nastavení aplikace a celkově vše, co se má stát po spuštění aktivity. Metody `on-`

Start() nebo onResume() jsou volány, když aktivita přechází do popředí. Metoda onStart() je vyvolána, pokud se aktivita spouští poprvé a metoda onResume() se volá, pokud je spouštěná aktivita v pozadí a pouze se přesouvá do popředí. Metoda onPause() se volá, pokud aktivita přechází do pozadí a je překrytá jinou aktivitou. Aktivita, jež má stav onPause(), může být operačním systémem Android po čase ukončena. Android takto ukončuje automaticky aktivity, které se dlouho nepoužívají za účelem úspory operační paměti a baterie zařízení. Metoda onStop() je volána v případě, když se aktivita zastaví, a to z jiného důvodu než nedostatku paměti. Aktivita je tedy pořád v paměťové oblasti BackStack a pokud se uživatel rozhodne znovu zobrazit, je volána metoda onRestart(), která sputí aktivitu znova. Poslední metodou je onDestroy() a volá se při úplném ukončení aktivity. Do každé této metody můžeme dopsat vlastní kód. To znamená, že pokud například budeme chtít, aby se něco vykonalo při ukončení aplikace například tedy při metodě onDestroy(), zapíšeme přímo do ní, at' už chceme, aby vyskočilo vyskakovací okno či se přehrál zvuk.[6]

2.11 API

Android je rozdělen do různých verzí s různými API (Application Programming Interface).

První verze operačního systému Android se objevila na trhu v září roku 2008. O rok později, v únoru 2009, vyšel Linux Android 1.1 a později v dubnu Android ve verzi 1. 5 Cupcake. Tato verze podporovala nahrávání a přehrávání videa v různých formátech i přehrávání videí na YouTube. Virtuální klávesnice třetích stran s podporou vlastních slov a možnosti kopírování a vkládání ze schránky. Dalším Androidem byl Android 1.6 Donut a vyšel 1. září 2009, byl vylepšen obchod s aplikacemi o obrázky a hodnocení uživatelů. Zvládal práci s více soubory a měl vyšší rozlišení obrazovky a integrovanou navigaci od Googlu. Android 2.0 a 2.1 Eclair vyšel 26. října 2009 a disponoval novým designem uživatelského prostředí. Podporoval standard HTML5, Bluetooth 2.1 a podporoval velké množství rozlišení displeje. Další Android 2.2 Froyo z 28. května 2010 obsahoval nový webový prohlížeč s podporou JavaScriptu či možnost vytvořit WiFi hotspot. V této verzi proběhla významná optimalizace paměti a celkového výkonu. Ve verzi Android

2.3 Gingerbread vydané 6. prosince 2010 přibyla možnost komunikace přes NFC a podpora dalších typů senzorů jako gyroskop a barometr. Následoval Android 3.0 Honeycomb z 21. února 2011 a první verze byla určena pouze pro tablety. V této verzi bylo možné přizpůsobit domovskou obrazovku a přibyla podpora USB příslušenství. Další Android 4.0 IceCream Sandwich, vydána 19. října 2011, odstraňuje speciální verzi pro telefony a pro tablety a místo nich vytváří jednotnou verzi pro oba typy zařízení. Přidány aplikace na obrazovce uzamčení a podpora full HD videa. Android 4.2 Jelly Bean, vydaný 9. července 2012, měl vylepšené notifikace a zlepšené aktualizace aplikací. Další Android 4.4 KitKat se začal používat 3. září 2013. Měl vyšší výkon, rychlejší multitasking a systém pro vypínání nepotřebných procesů na pozadí. Následník byl Android 5.0 Lollipop, vydaný 25. června 2014, jenž začal podporovat nové senzory jako měřič srdečního rytmu a podporoval vylepšenou grafiku. Aktuálně předposlední verzí systému Android je verze 6.0 Marshmallow, který výrazně zlepšuje práci s textem, označování a kopírování a upravuje používání záložek webových prohlížečů. Úplně poslední a nejnovější verze Android je verze 7.0 Nougat, který vylepšuje práce s baterií, paměti a má optimalizaci procesů na pozadí. [6, 7, 8, 9]

API tedy application programming interface je velice důležitý pro vývojáře již před započatím vývoje softwaru. Chceme-li, aby software běžel na Androidu 3.0, je zapotřebí, aby vývojář před zahájením zvolil API Android 3.0 nebo nižší. Takový software bude použitelný na Androidu 3.0 a všech vyšších verzích. Důležité je si ověřit, na jaké nejnižší API se vyplatí vyvíjet, tedy kolik uživatelů danou verzí systému ještě používá. Čím novější API vývojář zvolí, tím jednodušší vývoj bude, protože vyšší API obsahuje nové metody a komponenty. Cenou za nejnovější příkazy a komponenty daného API zaplatí vývojář tím, že uživatelé s nižším API software nespustí.

2.12 Publikování softwaru přes Google Play

Google Play, dříve nazývaný Android Market, slouží k distribuci softwaru. Kdokoli se může registrovat v Google Play jako vývojář. Registrace probíhá přes web a je nutné zaplatit registrační poplatek. Po registraci může vývojář nahrát jakoukoli aplikaci, kterou vytvořil. Po nahrání na Google Play není možné aplikaci ihned

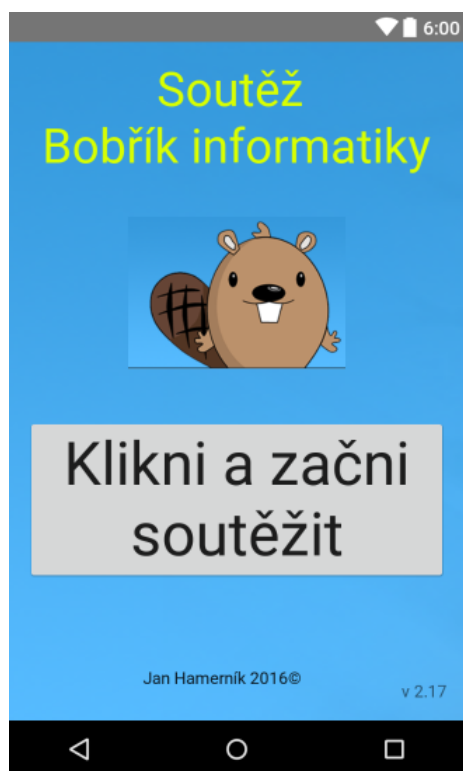
stahovat a používat. Vývojář musí počkat, až dojde ke schválení jeho aplikace. Až když je aplikace schválena jako vhodná a bezpečná, je zveřejněna pro širokou veřejnost. Aplikace na Google Play jsou buď zdarma nebo placené. Pokud se vývojář rozhodne umístit aplikaci jako placenou, tak mu Google nařizuje povinnost poskytovat podporu k jeho softwaru. Pokud si tedy uživatel stáhne a zakoupí placenou aplikaci z Google Play a následně má s touto aplikací problémy, vývojář je povinný problém vyřešit do tří dnů od oznámení. Pokud Google označí problém jako urgentní, vývojář má na reakci pouze 24 hodin.[10]

3 Praktická část

3.1 Vytvoření návrhu

Před vývojem jsem udělal návrh aplikace a jejího životního cyklu. V teoretické části jsem rozvrhl rozbor fungování Bobříka informatiky a podle toho sestavoval kostru aplikace. Návrh jsem několikrát upravoval, aby bylo následné programování co nejefektivnější.

3.1.1 Úvodní obrazovka



Obrázek 4: Úvodní obrazovka

V návrhu je v první části aplikace zahrnut splash screen neboli uvítací obrazovka. Taková obrazovka by měla obsahovat uvítací zprávu a přivětivou grafiku. Dále by celá aplikace měla být user-friendly, tedy uživatelsky přívětivá a uzpůsobená tak, aby ji uživatel dokázal používat, aniž by musel číst manuál či hledat návody

k použití. Ovládání musí být intuitivní. Proto je na displeji zobrazeno vždy jen nezbytně nutné množství informací a ovládacích komponent. Na další obrazovce si soutěžící volí, zda se chce registrovat nebo pokračovat v již rozpracovaném testu, či zobrazit výsledky hotového

3.1.2 Výběr kategorie a registrace



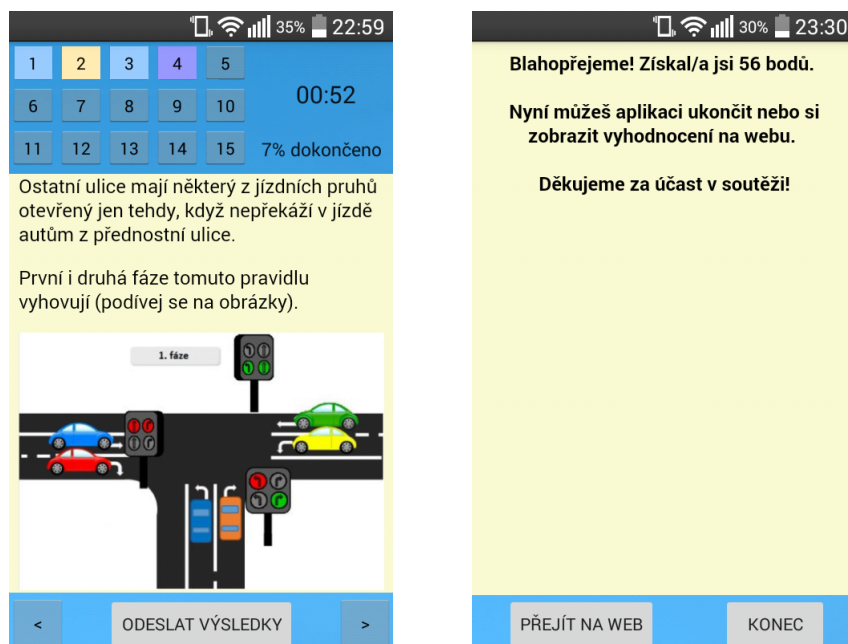
Obrázek 5: Obrazovka s výběrem kategorie a registrace

Následuje obrazovka s možností zvolit si soutěžní kategorii. Tato obrazovka se skládá z nadpisu, tedy výzvy k výběru kategorie, série radio buttonů k zakliknutí vybrané možnosti a na spodní straně displeje nalezneme tlačítko pro pokračování.

Nyní se nacházíme na stránce registrace, kde jsou připravena pole pro vyplnění uživatelem. Uživatel zde vyplní jméno, příjmení, rok narození. Následně narazí na

radio button, kde vybere pohlaví. Další kolonkou pro vyplnění je kód školy. Tento kód je použit pro následující přiřazení soutěžícího ke správné škole. A nejnižší je usazené opět tlačítko pro pokračování. Textová pole jsou nastavena tak, aby se klávesnice sama přepnula podle toho, jestli očekává text nebo číselnou hodnotu. Tímto se zároveň zabrání tomu, aby se do proměnné dostal špatný datový typ. Není tedy možné, aby do roku narození šel napsat text, a tím vznikl problém s ukládáním String do Integer. Pokud uživatel zapomene některé pole vyplnit nebo ho nevyplní a stiskne tlačítko pokračovat, bude upozorněn, že některá pole nejsou vyplněna. Pokud uživatel pole doplní nebo už při prvním pokusu jsou vyplněna správně, aplikace zobrazí další obrazovku. Nastane vyhodnocení dat z registrace a pokud vše proběhne správně, tak je zobrazena obrazovka s upozorněním, že je test připraven. Dále je na této obrazovce umístěna výzva s tím, aby si uživatel opsal svůj soutěžní kód. Tento kód slouží k opětovnému připojení do aplikace pro případ, že by si uživatel zavřel rozpracovaný test nebo kdyby požadoval opětovné zobrazení výsledků testu. Po opsání soutěžního kódu soutěžícím je připraveno tlačítko pro pokračování. Stisknutím tohoto tlačítka se spustí hlavní aktivita CORE, která zařizuje funkcionalitu celého testu.

3.1.3 Obrazovka testu



Obrázek 6: Obrazovka testu a zobrazení hodnocení

V levém horním rohu této obrazovky soutěžící nalezne jednotlivá čísla otázek. Pokud zvolil kategorii Mini, nalezne čísla 1 až 12. V kategoriích ostatních nalezne čísla 1 až 15. Čísla jsou interaktivní. Pokud soutěžící klikne na číslo otázky, příslušná otázka se mu zobrazí. Na začátku testu nejsou čísla otázek podbarvená. Nepodbarvené otázky jsou otázky, které uživatel ještě nenavštívil. Otázka, kterou uživatel aktuálně prohlíží, je podbarvena světle žlutou barvou. Po spuštění testu je takto žlutě označena otázka 1 a je také zobrazeno její zadání. Otázky, na které uživatel odpoví ať už správně nebo špatně, jsou podbarveny fialovou barvou, aby uživatel věděl, že tuto otázku už viděl a odpovídal na ní a tudíž na ní už nemusí myslet. Pokud však otázku navštíví, ale neodpoví žádnou odpověď, číslo otázky se obarví na světle modrou. Pokud soutěžící odpoví možnost „nechci odpovídat“, otázka bude podbarvena fialově, protože uživatel odpovídal.

V pravé horní části obrazovky běží čas od začátku testu, aby měl uživatel přehled, jakou dobu už u testu strávil. Hned pod časomírou jsou zobrazena procenta hotového testu. Na začátku testu ukazatel ukazuje 0 %. Procenta se přepočtou při

každém přepnutí otázky, a tak má uživatel přehled, kolik práce má již hotové.

Celá prostřední část je využita pro zobrazení otázky a odpovědí. V horní polovině je zobrazena otázka a v dolní polovině je možnost pro odpověď a to v různých formách. Buď je například zobrazen box s radio buttony tedy typ ABCD, nebo textové pole pro zápis textové odpovědi.

V úplně spodní části obrazovky je ovládací lišta, která obsahuje šipku doleva a doprava. Mezi nimi pak nalezneme tlačítko na ukončení testu. Šipky slouží pro přepínání otázek, buď o jednu dále nebo o jednu zpět. Z tohoto vyplývá, že soutěžící má dvě možnosti ovládání a to buď přepínání otázek šipkami nebo kliknutím na číslo otázky. Tlačítko pro ukončení testu po stisknutí zkontroluje, zda má uživatel vyplněno 100 % testu. Pokud je test celý vyplněný, přejde aplikace na další obrazovku. Pokud ne, vyskočí upozornění, zda soutěžící chce test opravdu ukončit, i když neodpověděl na všechny otázky. Na další obrazovku se uživatel dostane taktéž pokud mu při testu vyprší čas.

Další obrazovka, obrazovka s výsledky testu zobrazuje počet bodů, které soutěžící získal. Výpočet je proveden při přechodu na tuto obrazovku dle definovaných vzorců. Soutěžící má nyní už jen dvě možnosti, tlačítko pro ukončení aplikace nebo pro zobrazení výsledků na webu.

3.2 Vytvoření stabilního připojení k databázi

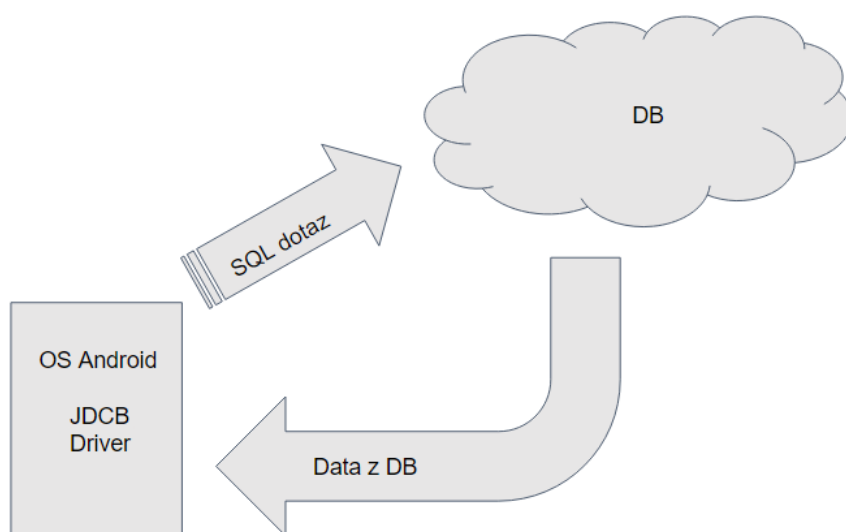
Vytvoření spojení s databází se zdálo jako jednoduchý úkol, a protože jsem to už dříve v minulosti v Javě či C++ dělal, nevěnoval jsem této části zvýšenou pozornost. Problémy se začaly vyskytovat až při programování této části, kdy příkazy nefungovaly či je Android vůbec neznal. V této chvíli jsem začal pátrat na webu, v čem vězí problém. Nalezl jsem, že Android počítá s tím, že Aplikace si budou ukládat a stahovat data z databáze, která je připravena Googlem a je k ní jednoduchý přístup. Bohužel pokud se rozhodnete vyvíjet aplikaci, která se potřebuje připojit k existující databázi na vzdáleném serveru, už to tak jednoduché není. Našel jsem knihovnu JDBC neboli Java Database Controller. Skoro ihned jsem našel i informaci, že tato knihovna má s Androidem velice špatnou kompatibilitu a lze ji použít pouze v malém procentu případů. Druhá varianta byla naprogramovat vlastní komunikační software v PHP, avšak tato možnost se tvářila jako složitá a

náročná. Zvolil jsem tedy první variantu a pokusil se zprovoznit JDBC driver.

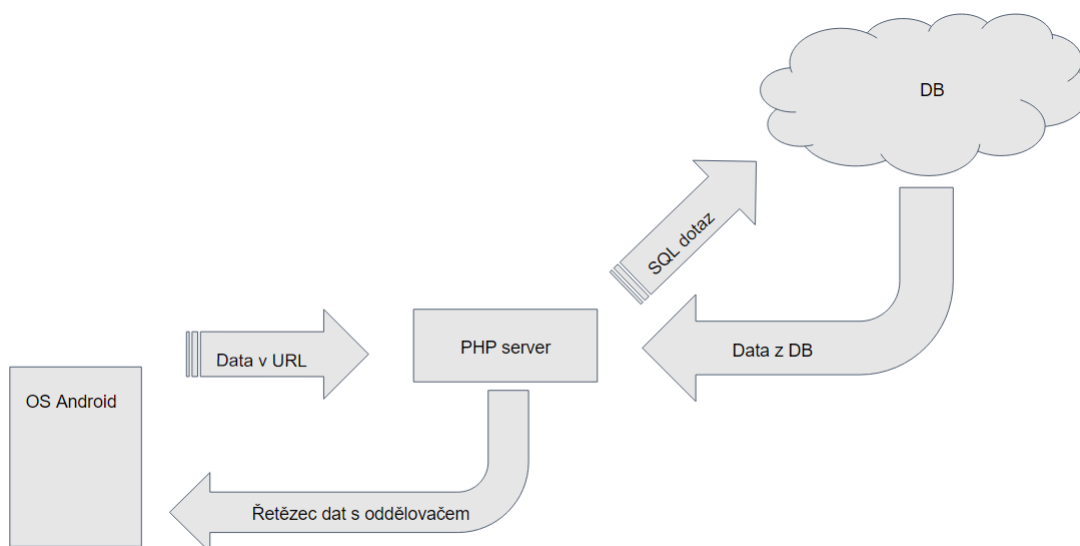
Po importu JDBC driveru do mého projektu vyskočilo několik chybových hlášení. Jednalo se o různé chyby verzí mezi JDBC a Android studiem. Po dlouhém bádání se podařilo chyby odstranit, ale vyskytl se další problém. Celý projekt nebylo možné zkompileovat, protože se kompilátor měl potíže s JDBC driverem. Některé chyby se podařilo vyřešit, ale Android hlásil další problémy s příkazy, které jsou už zastaralé a nepoužitelné, proto jsem se rozhodl JDBC driver z projektu odstranit.

Začal jsem pracovat na druhé variantě, a to komunikaci s databází přes PHP rozhraní. Začal jsem pročtením návodů a rad, aby tentokrát byl výsledek použitelný a plně funkční. Vytvořil jsem PHP soubor a protože PHP soubory jsou zpracovávány serverem, byla tedy na místě volba na jaký server soubor uložit. Zvolil jsem nejjednodušší možnost uložit tyto soubory společně na server se soutěžní aplikací. Naprogramoval jsem systém, který očekává vstupy a z těchto vstupů vytvoří SQL dotaz, jenž je následně odeslán do databáze. Odpověď z databáze uloží do proměnné a následně vygeneruje HTML stránku.

Bylo potřeba vyřešit, jak dotaz odeslat přímo z mobilní aplikace a následně do ní stáhnout data. Android obsahuje komponentu WebView, která umožňuje načíst libovolnou webovou stránku. Aby WebView fungoval, musí vývojář nejprve povolit v Android Manifest projektu přístup k internetu. Pokud povolíme toto nastavení, ovlivní to i instalaci aplikace tím způsobem, že při instalaci bude v seznamu potřebných oprávnění uvedeno, že aplikace požaduje plný přístup k internetu. Dalším krokem bylo už samotné zprovoznění WebView. WebView má metodu „load”, která zavolá webovou stránku nebo webový script. V tomto případě je volán PHP soubor, jemuž jsou předány parametry pro zpracování. Až PHP soubor dokončí svou práci, tak WebView v Androidu z vygenerované stránky stáhne data a přijme je jako textový řetěz. Tento řetězec je následně párován a data se uloží do proměnných, které se použijí tam, kde jsou potřeba. Ve finále vzniklo několik PHP souborů a každý z nich se stará o něco jiného, jeden například hlídá čas, a druhý, zajišťující zobrazení otázek, bude rozebrán níže.



Obrázek 7: Původní návrh komunikace s databází



Obrázek 8: Aktuální řešení komunikace s databází

3.3 Vytvoření modulu pro zobrazování dat z databáze

Původně jsem vyvíjel třídu, která se bude starat o stažení dat do telefonu a následně s nimi pracovat. Cílem tedy bylo, aby po registraci uživatele aplikace stáhla veškeré data a následně už nebyla závislá na připojení k internetu. Z počátku bylo vše bez problémů, stažení otázek a přiřazení id fungovalo správně. První problém nastal až při pokusu stáhnout obrázek a uložit v aplikaci. Pro stahování obrázků jsem použil knihovnu JSOUP. Obrázky se po nějaké době povedlo stahovat, ale práce s nimi byla obtížnější. Další věcí, která mi vadila, byl rozsah a velikost knihovny, ze které jsem potřeboval jen několik málo funkcí. Dalším problémem bylo ukládání dat. Data se ukládala přímo do aplikace a byla odeslána na server až po ukončení testu. Pokud uživatel v průběhu testu změnil svou odpověď, nezatěžoval tím server. Obrovskou nevýhodou by byl náhodný pád aplikace způsobený ať už vybitím baterie zařízení nebo nedostatkem operační paměti zařízení. Při tomto pádu by uživatel přišel o všechny neodeslaná a rozpracovaná data.

Celý systém, jenž byl založen na tomto postupu stažení dat a jejich odeslání až na konci, byl veden jako verze jedna. Po této úvaze a hrozbě spojené se ztrátou dat jsem vývoj této verze ukončil.

Vývoj v celé této oblasti započal znova a byl založen na průběžném přenosu dat po menších částech. Tato verze je označena jako verze dvě a je používána v současné verzi hotové aplikace. Podle nového schématu si po výběru kategorie aplikace stáhne odkazy na otázky a později stahuje jednotlivé otázky až při žádosti o zobrazení. Ukládání otázek oproti první verzi je také vylepšeno. Ukládání řeší JavaScript a metoda `onClick`, výsledkem je uložení odpovědi okamžitě po zakliknutí jedné z pěti možností u úloh typu ABCD. Pokud se jedná o úlohu, na kterou se odpovídá textovým řetězcem, je nutné po zapsání odpovědi kliknout na tlačítko uložit. Pokud aplikace spadne, uživateli nic nebrání otevřít si aplikaci znovu, kdy se po ověření načtou všechna data, která do aplikace už zadal. Veškeré otázky a odpovědi jsou zobrazovány v komponentě `WebView`.

3.4 Otestování úloh

Zobrazení úloh bez obrázků nebyl v podstatě žádný problém. Z databáze soutěže se stáhne zadání, možné odpovědi a typ otázky, tedy zda se jedna to ABCD nebo

otázku s textovou odpovědí. Nejprve je zobrazeno zadání následně obrázky k zadání, pokud jsou k dispozici, nejnižší se nachází modul pro odpovědi. Odpovědi typu ABCD jsou ohraničeny tabulkou, aby vše bylo přehledné a i na menším mobilním displeji neměl soutěžící problém se trefit na příslušný radioButton. U úloh, kde se odpověď zadává jako textový řetězec, je připraveno pole pro zápis textu. Ihned po kliknutí na tuto komponentu se zobrazí klávesnice na dotykovém zařízení. Při kliknutí na radioButton, popřípadě na tlačítko uložit u textových odpovědí, dojde k okamžitému uložení. Pokud uživatel přejde na jinou otázku a následně se vrátí zpět na již vyplněnou otázku, příslušný radioButton bude již zaškrtnutý. Opět v případě otázky s textovou odpovědí bude textové pole již vyplněno odpovědí, kterou uživatel předtím zadal. Pokud uživatel odpověď přepíše nebo zaškrtně jiný radioButton, odpověď se změní.

3.5 Změna velikosti obrázků

V cílech práce bylo stanoveno, že se obrázky mají přizpůsobit mobilní obrazovce. První varianta byla změnit velikost na straně serveru a přenést minimum dat. Vzhledem k tomu, že by server musel změnit velikost obrázku podle velikosti displeje a uložit je do mezipaměti serveru, jsem vyhodnotil, že by zatížení serveru bylo neúnosné. Pokud by na druhou stranu byl na serveru každý obrázek například v pěti rozlišeních, nemuselo by být zobrazení optimální. Vzhledem k rychlému vývoji displejů by tato metoda nebyla efektivní, navíc nelze předem určit, jestli bude soutěžící používat aplikaci na mobilu či na tabletu, proto jsem přistoupil k řešení pomocí kaskádových stylů, kdy se obrázek upraví automaticky podle šířky displeje. Tímto bylo docíleno efektu, že uživatel má vždy obrázek zobrazen ideálně na jeho obrazovce, ať už aplikaci používá na mobilním telefonu s minimální úhlopříčkou nebo na tabletu s úhlopříčkou dohánějící obrazovky notebooků. Otestoval jsem knihovnu JSOUP a problematiku jsem řešil systémem, který stáhne obrázek do telefonu a následně zobrazí. Avšak knihovna pro stahování obrázku byla složitá, proto jsem od tohoto postupu po čase opustil. Řešení přes kaskádové styly je vhodné i z toho důvodu, že existuje nespočet velikostí mobilních displejů a každý z nich má jiné rozlišení.

3.6 Technologie Flash

Některé úlohy v soutěži Bobřík informatiky jsou vytvořené a napsané v programovacím jazyce Flash. V posledních letech se od této technologie ustupuje a předpokládá se, že do budoucna tato technologie zanikne. Jedním z cílů bylo otestovat podporu technologie Flash na operačním systému Android. Na webových stránkách a na různých fórech existuje spousta návodu, jak technologii Flash na operačním systému Android zprovoznit. Já, jakožto vývojář aplikace, bych možná po nějakém čase, dle návodu, dokázal Flash na Androidu spustit. Ovšem z pohledu uživatele, který má vyhrazený čas na to, aby si stáhnul soutěžní aplikaci Bobřík informatiky a pustil se do soutěžního testu, už to tak jednoduché není. Nemůžeme po uživateli chtít, aby si stáhl mobilní aplikaci Bobříka informatiky a následně z webových stránek stahoval různé balíčky či jiné webové prohlížeče, aby mohl obsah zobrazit. Lze předpokládat, že by uživatel v takové situaci raději vyhledal počítač a spustil si klasickou verzi soutěže, než aby zprovožňoval soutěžní aplikaci na telefonu. Výhoda a smysl aplikací pro Android je ta, že uživatel si na Google Play vybere aplikaci, kde klikne na tlačítko instalovat a následně už se nemusí o nic starat. Aplikace se sama nainstaluje a zprovozní a uživatel ji po instalaci pouze spustí a může začít pracovat. Spousta odborníků označuje technologie Flash za zastaralou, a proto aplikace Bobřík informatiky technologii Flash a tedy i otázky napsané v jazyce Flash podporovat nebude. Protože soutěžní aplikace zobrazuje velké množství dat přes komponentů ve WebView, jež pro zobrazení obsahu používá zabudovaný prohlížeč operačního systému Android, záleží vždy na tom, jaký standard podporuje daný webový prohlížeč. Webový prohlížeč, který je v základu v operačním systému Android, se odvíjí od toho, jaká verze Androidu v telefonu je. Platí tedy, že čím novější telefon, tím vyšší verze Androidu, a tím lepší služby nabízí.

3.7 Podpora HTML5

Dalším cílem bylo otestovat HTML5 podporu v prohlížeči na Androidu. Testoval jsem podporu HTML5 na Androidu 4.4 KitKat a dle výsledku testu jsem zjistil, že většina funkcí téměř 100% je podporována. Znepokojujícím problémem je skutečnost, že jsou nepodporované dvě klíčové funkce, a to jsou Drag and Drop. Problém

to je hlavně z hlediska toho, že Bobřík informatiky obsahuje některé otázky, které jsou napsané v HTML5 a cílem je přetáhnout určitý obrázek z bodů A do bodu B. Aby tyto otázky byly použitelné v operačním systému Android, bylo by nutné algoritmus generování otázek trochu upravit, a to tak, aby po kliknutí na daný obrázek se obrázek označil jako zachycený a následně uživatel klikl, kam chce tento obrázek přesunout.

Dalším řešením, jak zprovoznit Drag and Drop na Androidu, je použití knihovny Hammer. Jde o to, že funkce Drag and Drop čeká na kliknutí a přetažení kurzorem myši. Díky knihovně Hammer lze přeložit dotyk na dotykovém displeji na kliknutí myši a plně tak využívat funkci Drag and Drop i na dotykových zařízeních. Toto se jeví jako nejjednodušší a nejefektivnější řešení tohoto problému. Bude však nutná úprava systému generování otázek HTML5, která by neměla být nějak složitá a vyřeší to podporu důležité funkce pro soutěž Bobřík informatiky.[11]

Podle informací z internetu funkci Drag and Drop samostatně nepodporuje žádný ze současných verzí prohlížečů. Pokud by se ovšem u otázek HTML5 upravilo přesouvání obrázků z místa na místo, lze s jistotou říci, že vyhodnocení zda je odpověď správná nebo špatná a následné přičtení či odečtení bodů by již nebyl problém. Jak se bude situace vyvíjet do budoucna zatím nelze jednoznačně určit, avšak opět by bylo potřeba, aby technologie Drag and Drop podporoval přímo webový prohlížeč Androidu, protože nemůžeme chtít po uživatelích, aby se kvůli vyplnění soutěžního testu instalovali speciální webový prohlížeč.

3.8 Grafické rozhraní

Aplikace dostala celkově jednoduchý design, aby se soutěžící bez problémů orientoval a soustředil se pouze na vypracování testu. Grafické rozhraní je navrženo tak, aby soutěžícího nijak nerozptylovalo a aby bylo přehledné. Uživatel by tak neměl ztrácet čas hledáním různých polí či jiných ovládacích prvků a aby se v celé aplikaci bez problémů zorientoval bez pročítání návodu. Úvodní obrazovka je graficky jednoduchá a obsahuje pouze tlačítko pro pokračování. Na další obrazovce je zobrazeno několik možných kategorií a tlačítko pro pokračování. Obrazovka s výběrem typu zahájení testu obsahuje dvě tlačítka a to buď pro spuštění nového testu nebo tlačítko pro pokračování v již rozpracovaném. V okně registrace jsou

zobrazena pole pro vyplnění. Pole se skládají z nadpisu a kolonky pro zápis hodnoty. Samostatná obrazovka testu je opět vyřešena v jednoduchém stylu. Nahoře obrazovky jsou zobrazené otázky a vedle nich se nachází čas od spuštění testu a hned pod ním procenta hotového testu. Otázky v horní části displeje buď nejsou podbarvené vůbec, nebo jsou podbarveny různými barvami, aby měl soutěžící přehled o testu. V hlavní části displeje je přes celý střed zobrazené zadání otázky a pod zadáním možné odpovědi. V nejnižší části displeje se nachází ovládací panel, na kterém je šipka doleva, šipka doprava a uprostřed tlačítko pro odeslání dat. Poslední obrazovka je rozdělena tak, že v horní části jsou zobrazeny získané body z testu a ve spodní části se nachází tlačítko pro ukončení aplikace a tlačítko pro zobrazení výsledků na webu.

Protože platí, že čím vyšší verzi Androidu použijeme, tím lepší funkce a grafické rozhraní můžeme použít, tak jsem při vývoji využil kódu, který zajistí, aby se na nových zařízeních vykreslila grafika lépe.

```
/**
 * obarvení aktualni otazky
 */
public void obarveniAktOtazky(int otazka) {
    for (int tlacitka = 0; tlacitka < btns.length; tlacitka++) {
        if (otazka == tlacitka) {
            // android 6.0

            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
                btns[tlacitka].setBackgroundTintList(ColorStateList.valueOf(Color.rgb(255, 236, 179)));
            } else {
                btns[tlacitka].setBackgroundColor(Color.rgb(255, 236, 179));
            }
        }
    }
}
```

Obrázek 9: Náhled kódu: automatická úprava grafiky

3.9 Otestování

Mobilní soutěžní aplikace Bobřík informatiky pro operační systém Android byla otestována malou skupinou žáků vybrané základní školy. Protože při testování ještě tato aplikace nebyla uložena v obchodě Google Play, bylo žákům propůjčeno několik zařízení s operačním systémem Android. Žáci měli čas si vyzkoušet průchod

testem od spuštění aplikace přes výběr kategorie, následné vyplnění registrace až po spuštění samotného testu. Taktéž dostali čas na to, aby si aplikaci vyzkoušeli, konkrétně přepínání otázek, zaškrťování odpovědí a celkové zobrazení a práci s testem. Následně test standardně ukončili a nastal čas k diskuzi. Na základě tohoto otestování bylo opraveno několik funkcionalit aplikace. V prvním kroku byla upravena obrazovka registrace, aby políčka, do kterých se zapisuje text, byla přehlednější. V některých případech nebylo zřejmé, kam text přesně zapsat. Další věci, jež musela být upravena u otázek, na které se odpovídá zaškrtnutím možnosti, byly doplněny rozestupy, aby se i na menších displejích soutěžící bez problémů trefil na požadované políčko. Následně jsem provedl rozbor u vybraných soutěžících, zda byly body za jejich zodpovězené otázky přičteny a odečteny správně. Po vypočítání bodů a projití otázek jsem testování aplikace ukončil. Aplikace vypočítává body správně a celkově je plně funkční.

4 Závěr

Současná webová verze soutěže Bobřík informatiky obsahuje nespočet úloh, které pro své fungování potřebují technologii Flash. Tato technologie není v našem případě využitelná, protože nemůžeme chtít po soutěžícím, aby si před testem stahoval doplňky pro systém či webový prohlížeč. Proto mobilní aplikace Bobřík informatiky úlohy s technologií Flash nepodporuje. Soutěž s tímto typem úloh do budoucna nepočítá a nahradí je úlohy v HTML5.

Byla otestována podpora HTML5 v operačním systému Android se závěrem, že technologie je v Androidu plně podporována až na funkci Drag and Drop. Tato funkce je klíčová pro soutěžní úlohy. Bude nutné provést úpravu kódu, který generuje HTML5 úlohy pomocí některé z knihoven, které dokáží kliknutí na displej na kliknutí myši. V době tvorby této práce téměř všechny soutěžní testy Bobříka informatiky obsahovaly HTML5 úlohy, které nejdou na Androidu využívat. Proto byly vytvořeny speciální testy pro mobily, které slouží jako demo testy. Nicméně je aplikace funkční pro zobrazení soutěžního testu, mimo HTML5 úloh. Po úpravě generování HTML5 úloh je možné aplikaci využívat jako aplikaci v ostrém testu. Aplikace nezobrazuje, jaké odpovědi měl soutěžící správně či špatně, ale jen počet bodů. Je tomu tak z toho důvodu, že při ostrém testu soutěžící tyto informace dostat nesmí. Aby měl soutěžící zpětnou vazbu, může po ukončení soutěžního období např. využít odkaz pro zobrazení výsledků, který se tam zobrazuje.

Připojení k databázi soutěž Bobřík informatiky potřebuje kvůli stahování zadání otázek a následné ukládání odpovědí zpět do databáze. Původně se tento úkol a cíl zdál jako jednoduchý, ale jen do té doby než nastaly první problémy s kompatibilitou. Android počítá s tím, že uživatel bude data ukládat a následně si je vytahovat z databáze, kterou Android obsahuje. V tomto případě však bylo nutné se připojit k existující databázi na vzdáleném serveru, a proto bylo nutné problém začít řešit. Neúspěšné řešení pomocí JDBC driveru jsem brzy opustil. Následně jsem vytvořil PHP platformu, která se stará o přenos dat oběma směry a o zobrazování dat, otázek a odpovědí. Tento způsob nebyl úplně nejrychlejší, ale rozhodně nejefektivnější. Protože jsem celou PHP platformu naprogramoval sám, mám dokonalý přehled, jak celý systém pracuje a po vytvoření této platformy již bylo opravdu jednoduché přenášet data na server a zpátky. Komunikace mobilní

aplikace mezi soutěžní aplikací a serverem je kompletně splněna a funkční.

Závěrem bych chtěl ještě zmínit informace o literatuře. Pokud se rozhodnete pustit do nějakého velkého projektu, který se bude týkat programování v nějakém novém programovacím jazyce, je dost možné, že narazíte na stejný problém, který zpočátku trápil mě. Vývoj aplikací pro Android je pořád tak nové téma a tak rychle se vyvíjí, že je těžké sehnat aktuální informace. Zpočátku jsem si půjčil z knihovny knihu, která byla asi rok stará a popisovala, jak vyvíjet aplikace pro Android, například prostředí NetBeans a jak doinstalovat celé prostředí pro vývoj aplikace pro Android. Později jsem konečně narazil na publikaci, která byla pouhé dva měsíce po vydání. V této knize už byly informace aktuálnější a popisovaly vývoj v Android studiu od Googlu. V knize jsem se dočetl některé základní informace a různé programovací postupy, ale než jsem se však dostal k samotnému programování, některé věci už byly opět aktualizované a postupy v ní uvedené se nedaly použít. Jako jediný zdroj informací, kde jsou vždycky informace aktuální, je webová stránka www.developer.android.com. Tato stránka je v podstatě oficiální dokumentace k vývoji aplikací pro Android a právě zde je vše vždy aktuální.

Aplikace je umístěna na Google Play pod názvem Bobřík informatiky a je ke stažení zdarma.

Použitá literatura a zdroje

- [1] Soutěžní otázky. *Bobřík informatiky* [online]. 2016 [cit. 2016-12-08]. Dostupné z: <http://www.ibobr.cz/o-soutezi/156-soutezni-otazky>
- [2] Pravidla soutěže. *Bobřík informatiky* [online]. 2016 [cit. 2016-12-08]. Dostupné z: <http://www.ibobr.cz/o-soutezi/23-pravidla-souteze>
- [3] CANDRA, Martin. *Online aplikace pro odhalování podvádění v soutěži Bobřík informatiky*. České Budějovice, 2015.
- [4] UJBÁNYAI, Miroslav. *Programujeme pro Android*. Praha: Grada, 2012. Průvodce (Grada). ISBN 978-80-247-3995-3.
- [5] Error during installing HAXM, VT-X not working. *Stack Overflow* [online]. 2014 [cit. 2016-12-02]. Dostupné z: <http://stackoverflow.com/questions/21635504/error-during-installing-haxm-vt-x-not-working>
- [6] LACKO, Ľuboslav. *Vývoj aplikací pro Android*. Brno: Computer Press, 2015. ISBN 978-80-251-4347-6.
- [7] Android Lollipop. *Developers* [online]. 2014 [cit. 2016-12-09]. Dostupné z: <https://developer.android.com/about/versions/lollipop.html>
- [8] Android 6.0 Changes. *Developers* [online]. 2015 [cit. 2016-12-09]. Dostupné z: <https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html>
- [9] Android 7.0 Behavior Changes. *Developers* [online]. 2016 [cit. 2016-12-09]. Dostupné z: <https://developer.android.com/about/versions/nougat/android-7.0-changes.html>
- [10] Distribuční smlouva pro vývojáře Google Play. *Google Play* [online]. 2016 [cit. 2016-12-18]. Dostupné z: <https://play.google.com/about/developer-distribution-agreement.html>

- [11] Tvorba moderního e-shopu: HTML5 drag & drop a kategorie. *Zdroják.cz* [online]. 2013 [cit. 2016-12-08]. Dostupné z: <https://www.zdrojak.cz/clanky/tvorba-moderniho-e-shopu-html5-drag-drop-a-kategorie>

Seznam obrázků

1	Mobilní aplikace Bebras	15
2	Náhled kódu: přenos dat mezi aktivitami v OS Android	18
3	Náhled návrhu aplikace v Android Studiu	19
4	Úvodní obrazovka	26
5	Obrazovka s výběrem kategorie a registrace	27
6	Obrazovka testu a zobrazení hodnocení	29
7	Původní návrh komunikace s databází	32
8	Aktuální řešení komunikace s databází	32
9	Náhled kódu: automatická úprava grafiky	37