Заглавна страница

Съдържание

**Увод**

С развиването на модерните технология и интернет в последните двадесет години устройства лаптопи и смарт-телефони стават все по-голяма част от живота на средностатистическия човек, до степен до която тяхната липса е вече немислима възможност. Прогресирането на ИТ и софтуерната сфера е толкова силно, че буквално ежедневно се създават нови приложения, които да улеснят живота на човека в модерния свят, замествайки вече остарелите и ненужни технология на вчерашния свят. Google Maps и подобни приложения за локализация и изготвяне на маршрути, отдавна замениха GPS-сите, които всички ползвахме преди десетина години. Хартиените карти са демоде, което е станало трудно за намиране. Човек няма нужда да чете вестници или телевизия, когато всички новини са на един клик разстояние и могат да бъде наблюдавани в реално време. Streaming платформи като Spotify и Apple Music отдавна направиха ненужно свалянето на музика в хранилището на телефона или записването на дискове за да може да слушаме музика докато се разхождаме или пътуваме с кола.

Примерите са безкрайни и се увеличават неимоверно всяка секунда.

Технологиите фундаментално са навлезли и в политическата система на много държави и макар, че все още не е силно разпространено е напълно сигурно, че след години ще можем да гласуваме на следващите избори/референдум от комфорта на собствения си дом, забравяйки за дългите опашни в кварталните училища, безкрайното лутане докато намерим стаята, която отговаря на нашия жилищен регион и тъмните стаички, в които се крием за да дадем своят вот. Вместо това ще можем с един клик да дадем своя глас и да продължим спокойно деня си, без да трябва да правим планове, кога да отидем да гласуваме, така че да приключим най-бързо и лесно. Комфорта обаче няма да е единствения плюс на подобна технология. По-голямата сигурност предоставена от софтуерното бъдеще ще надвишава всяка такава на настоящето. Неща като фалшифициране на вотове и човешки грешки ще бъдат премахнати или поне сведени до минимум. Тоновете хартия, които се използват и изхвърлят при всички избори вече няма да са нужни, по-този начин добавяйки и един екологичен плюс в полза на модерното.

Целта на този проект е не да създаде това бъдеще, а да покаже колко лесно и удобно би било то

**Първа глава**

**Анализ на съществуващи разработки**

Идеята за подобно приложение не е нова и можем лесно да намерим други подобни разработки в интернет. Ето няколко примера, изтъквайки техните положителни и отрицателни страни:

**1.1** **myvote.io**

Web базирано приложение за гласуване, което позволява създаването на гласувания и анкети, и наблюдаването на им в реално време. Има версии за iOS и Android, поддържа се както на мобилни устройства, така и на таблети и персонални компютри.

**1.2 voxvote.com**

Мобилно приложение за гласуване. Използва се най-често за задаване на въпроси по време на реални събития като конференции и срещи. Водещият на срещата създава гласуването, а участващите в публиката, чрез приложението, отговарят или гласуват на зададения от него въпрос.

Минус е факта, че можеш да имаш само по десет въпроса на създаден event и имаш право само на 5 event-а. За наличието на повече се заплаща

**1.3 doodle.com**

Doodle е уеб-сайт за създаване на анкети. Негов голям плюс е, че е можеш да продължаваш да ползваш безплатните му функции, колкото често искаш, без да се налага заплащане. Дизайна му е доста минималистичен и “user friendly”. Не е нужно регистрирането на акаунт и предоставянето на мобилен телефон или и-мейл адрес за неговото ползване, което го прави лесно за употреба

Втора глава

Проектиране

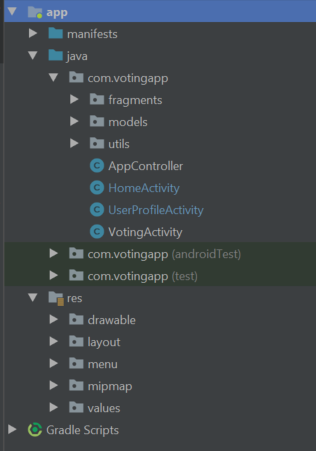
В следната глава ще

**Трета глава**

**Реализация**

В тази глава ще разгледаме как е реализиран проекта. Ще покажем обектите, които изграждат приложението, activity-тата, от който се състои VotingApp и фрагментите, който се използват. Ще дадем примери и за потребителския интерфейс, валидации и програмен код.

**3.1 Структура на проекта**



***Фиг 3.1.*** *Структура на проекта в Android Studio*

Основните папки и файлове в проекта са:

* java - com.votingapp – В нея се намират всички java файлове използвани в приложението.
  + В нея се съдържат acitivity-тата, които изграждат приложението. Това са HomeActivity.java, VotingActivity.java, UserProfile.java. В тази папка се съдържа и AppController.java, който е отговорен за създаване на самото приложение.
  + fragments – В нея се намира файла ListFragment.java, който се използва за показване на списъка от гласувания/анкети/референдуми в VotingActivity.java. Файловете TakeVotingFragment.java, TakePollFragment.java и TakeReferendumFragment.java, които се използват за даване на вот в гласуване, анкети и референдум респективно. Файловете VotingResultsFragment.java, PollResultsFragment.java и ReferendumResultsFragment.java които се използват за показване на резултати от даденото гласуване, анкета или референдум.
  + models – В тази папка се съдържат класовете-модели, от които се изграждат обектите изграждащи приложението. Това са Option.java, Poll.java, Question.java, Referendum.java, UserProfile.java, Vote.java, Voting.java.
  + utils – В utils се намира Key.java, който съдържа всички ключове използвани за комуникиране и изпращане на информация между activity-та и фрагменти.
* res – Папка с ресурсите на приложението
  + drawable – Съдържа всички xml икони, които се използват предимно за NavigationDrawer-а.
  + layout – В тази папка се намират всички layout файлове, които изграждат потребителския интерфейс на всяко activity и фрагмент.
  + menu – В нея се намира съдържанието на основното меню, което се визуализира под формата на NavigationDrawer.
  + mipmap – Съдържа иконата за стартиране на приложението.
  + values – Папката съдържа всички цветове, стилове, размери и низове нужни на приложението.

**3.2 Описание на моделите(основните класове) на приложението**

**3.2.1 Vote.java**

Най-основния модел в приложението е абстрактният клас Vote.java. Неговата идея е да бъде изграждащата частица на всеки вид вот(гласуване/анкета/референдум. Класът е абстрактен и не могат да бъдат създавани обекти директно от него. Той служи единствено като основа! Vote.java съдържа поле “title” от тип String, което държи заглавието на вота, get-ър и set-ър за полети, default конструктор, както и конструктор за инициализиране на title.

**3.2.2 Question.java**

Двата основни елемента, които изграждат един вот са – въпрос и възможни отговори. Класът Question представлява точно този въпрос. В сегашния си вариант класа съдържа само едно поле от тип String, което държи текста на въпроса(questionText), get и set методи за инициализира и достъпване на член-променливата и два конструктора: един празен и един инициализиращ questionText.

Моля да се има предвид, че questionText полето на Question.java не се припокрива с полето title на Vote.java, тъй като заглавието на вота и текста на въпроса могат да се различават. Също така, когато имаме анкета имаме едно генерално заглавие на анкетата, което се пази в title на Vote.java, а отделните въпроси в анкета се пазят в член-променливите questionText на обектите от тип Question(За повече информация виж ???)

В бъдещи разработки могат да се добавят и допълните полета към класа Question, като например:

* important – Поле, което да означава, че въпроса е от висока важност.
* hot – Променлива, която да показва, че много хора отговарят на дадения въпрос
* category – Член, който да разделя въпросите в различни видове категории

**3.2.3 Option.java**

Другия основен елемент на всеки вот – възможните отговори е представен чрез класа Option.java. Класа съдържа текста на дадената опция(optionText), броя пъти, които е избран този отговор(int променливата timesSelected) selectedByCurrentUser ???

Класът съдържа метода *public void increaseTimesSelected(),* който се вика, когато някой избере дадената опция във вот. Той инкрементира променливата timesSelected, по този начин показвайки, че опцията е избрана още веднъж от даден потребител.

**3.2.3 Referendum.java**

Класът Referendum е репрезентацията на референдум, на който се гласува с „Да“ и „Не“. Класа наследява абстрактния(основния) клас Vote.java и съдържа обект от тип Question, в който се пази информацията за въпроса на референдума и два член-променливи обекта от тип Option:

* optionYes – Опцията „Да“. Съгласяване с въпроса зададен на референдума
* optionNo - Опцията „Не“. Несъгласие с референдума.

**3.2.4 Voting.java**

Voting класа на приложението представлява гласуване и наследява основния клас Vote.java. Надгражда го като добавя член-променлива от тип Question и списък от възможни отговори(ArrayList от Option). Това се налага, тъй като дадено гласуване не е ограничено от брои възможни отговори.

**3.2.5 Poll.java**

Моделът Poll.java репрезентира анкета. Както и предишните два вида вот, той също наследява абстрактния клас Vote.java. Класът има една член-променлива “pollContent” тип HashMap<Question, ArrayList<Option>>. Това е Map, който като ключ ползва обект от тип Question(въпрос), а като стойност има свързан списък от възможни отговори(Option)

Юзър профил ???

**3.3 Описание activity-тата в приложението**

**3.3.1 HomeActivity.java**

**3.3.2 VotingActivity.java**

**3.3.3 UserProfileActivity.java ???**