

Uggleträsket - uppbyggnad

Här följer en beskrivning av Uggleträsket med avseende på hur applikationen är uppbyggd samt redogörelse av de designval som tagits om det anses nödvändigt.

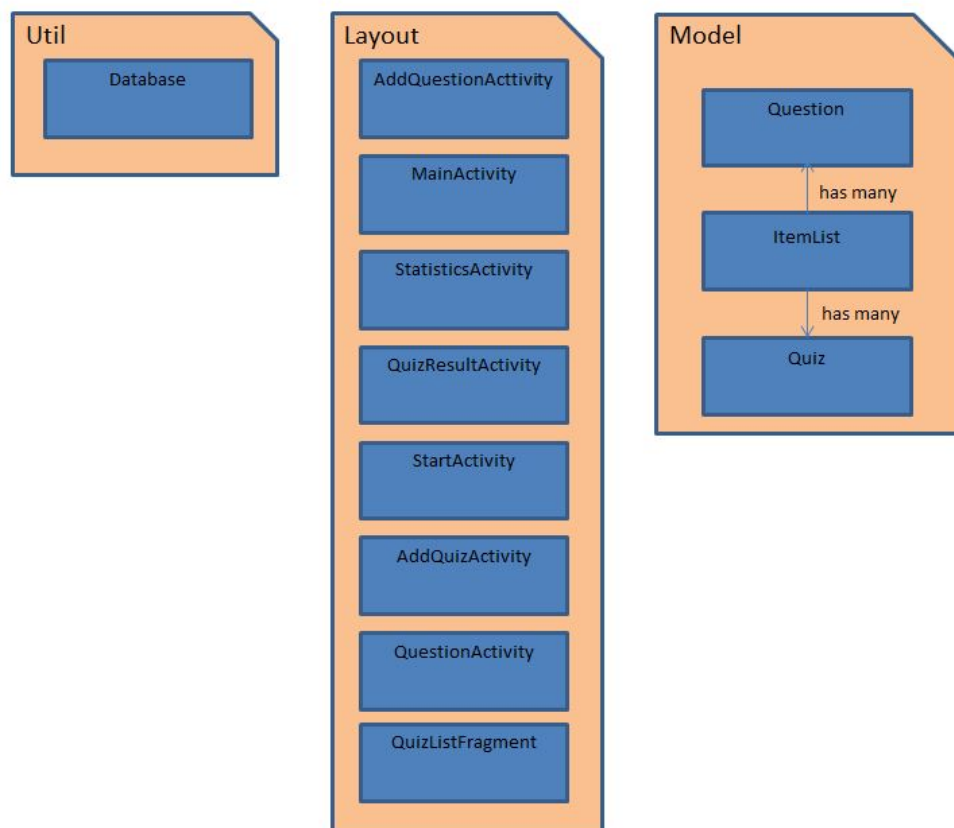
API

Minimum SDK-nivå: 11

Maximum SDK-nivå: 21

Minimumnivån för API är satt till 11 för att se till att DialogFragment, som används i QuizListFragment, stöds av samtliga användare. I och med detta blir applikationen kompatibel med 90,4 % av alla Android-telefoner vid tiden då detta dokument senast uppdaterades. Maxnivån bör hela tiden vara satt enligt den senaste utgivna API-versionen, vilket för tillfället vad är 21.

Klassschema



Arkitektur

Här redogörs för hur applikationens filer är strukturerade.

Paket - layout

- AddQuestionActivity
- AddQuizActivity
- MainActivity
- QuestionActivity
- QuizListFragment
- QuizResultActivity
- StartActivity
- StatisticsActivity

I paketet återfinns de Java-klasser som är direkt knutna till specifika XML-layouter.

Paket - model

- ItemList
- Question
- Quiz

I paketet återfinns de klasser som utgör applikationens funktionsmässiga byggstenar. Dessa är ett direkt resultat av den objekt-orienterade struktur som ligger till grund för applikationen.

Paket - util

- Database

I paketet återfinns applikationens hjälpklasser.

Designval

Här redogörs för vissa designval för att underlätta förståelse och fortsatt utveckling av applikationen.

QuizListFragment

Listan på alla quiz som finns i den utnyttjade databasen används i flera olika klasser. Listan gjordes därför till ett fragment för att kunna återanvändas på ett enkelt sätt. Klassen tar en sträng som bestämmer vilken data som skickas vidare till vilken activity när ett quiz väljs. Om fragmentet skall användas i ett nytt sammanhang kan det därför vara nödvändigt att i klassen utöka if-satsen som jämför den strängar och väljer nästa aktivitet.

ItemList

ItemList är en generisk klass som används för att lista olika objekt. Innan klassen kodades fanns två klasser, QuestionList och QuizList, som hade samma metoder. Klassen ökar kodåteranvändandet och gör det även enkelt i fortsättningen att nyttja listor med andra objekt. Exempelvis kan listor med lärare eller studenter vara aktuellt i framtiden.

Database

I denna klass återfinns all hantering av datatransferering till och från databasen. Eftersom att detta används av flera klasser och att flera anrop använder samma kod har det valts att läggas i en separat publik klass, vilket förebygger duplicering av kod.