Java题库系统期末大作业报告

谭楷蓁 2153874

**1. 工作需求分析**

经过多年的摸爬滚打，我发现学生在日常的学习工作中，常因不能轻易得到答案而感到绝望，当打开利用搜索引擎而得来的网上题库时，往往均需扫码关注或是付费才能查看答案，令人苦不堪言。上传题目、检索题目和收藏题目的繁琐操作，也是老师和学生的一大烦恼。为解决以上问题，TKZ题库系统应运而生。简洁的UI和实用的功能，是其优势所在。

根据需求分析，我在软件中提供了用户登陆注册、用户基础设置、检索题目、查看题目详情、上传题目、导出题目、收藏题目等功能。

我在软件中设计了四个主页面，分别是首页、检索浏览页面、上传题目页面、用户设置页面。在每一个页面的左侧，设置了侧边栏，以此提供页面切换的目录。

用户登陆注册：每个用户有独立账户，便于用户进行账号管理，与上传和收藏挂钩。

用户基础设置：用户可以在页面内进行个人基本资料的修改，设置用户头像、邮箱、基本信息，查看我的收藏和我的发布。

检索题目：页面顶部设置检索框，可以直接输入关键词查找，也可以选择分科目（语文、数学、英语）、分题型（简答题、选择题）、分难度（简单、中等、困难）查找。

浏览题目：所有题目只显示简略信息，每个页面显示题目数量固定，可以翻页。

查看题目详情：在浏览页面点击题目卡片可显示题目详细信息，包括题干、选项、图片、音频等。进入页面时，答案自动隐藏，点击按钮可查看答案，便于用户进行思考。

导出题目功能：在题目详情页面，可点击导出题目，导出格式为pdf文件，若有音频文件则单独导出。

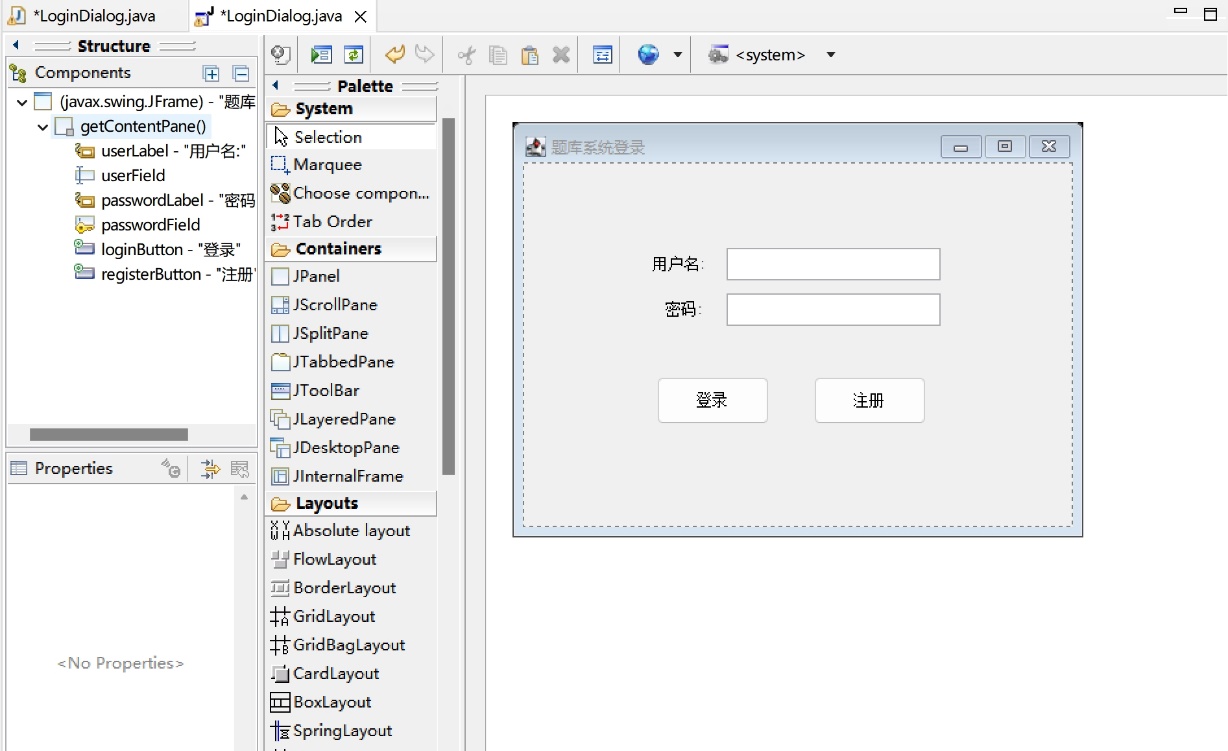
收藏题目功能：在题目详情页面，可点击收藏题目，便于用户进行反复查看和记忆。

上传题目功能：分为直接输入题目、直接导入题目和根据模板导入三种形式。直接输入题目，是手动输入与题目有关的所有信息；直接导入题目，是导入包含题目信息的文件，可能需要用户手动调整数据库映射关系；根据模板导入，是采用规定固定格式的csv文件，可自动识别，且支持批量导入。

**2.实现解析**

2.1 基础框架

在大项目的初始阶段，我曾尝试使用Jcreator+jdk1.6完成开发，随即发现由于编译器的许多限制和版本老旧的问题，限制了许多页面布局和功能设置。一次偶然的机会，我接触到了Eclipse+Windowbuilder插件，发现了它可视化设计的优势和控件可拖动的功能，能够让设计页面布局变得简单且自由。于是，为了实现上述需求分析中提到的功能，我毅然放弃了Jcreator。



在数据库方面，我采用了MySQL。使用了java用于连接MySQL数据库的jar文件，添加到了项目类路径。因为程序中连接的是本地数据库，所以程序在复现时需要事先运行sql文件以导入数据库设计，并且在DatabaseConnection.java中更改本地密码。

为了简化项目结构、简化开发和学习基础原理，我放弃了使用java项目管理工具，如Maven等；还有能提升界面现代化和美观度的工具，如FXML+CSS等。

| 内容 | 对应选择 |
| --- | --- |
| 开发环境 | Eclipse |
| GUI框架 | Java Swing、AWT |
| GUI设计 | Windowbuilder |
| 数据库 | MySQL |

2.2 项目文件架构和运行逻辑

1、**src/com/java**：存放所有的源代码。

* + **database**：存放与数据库连接相关的类DatabaseConnection.java。
  + **models**：存放数据模型类，例如Question.java, QuestionMedia.java，用于表示数据库表的结构。
  + **gui**：存放所有与图形用户界面相关和与后端数据库交互逻辑的类。
    - **main**：存放主窗口的类，MainFrame.java。
    - **panels**：存放所有可切换的面板，如BrowsePanel.java，SetupPanel.java等。
    - **dialogs**：存放弹出对话框的类，如登录对话框LoginDialog.java。

2、**src/font**：存放中文字体资源。

**src/images**：存放图像文件。

**src/media**：存放媒体文件。

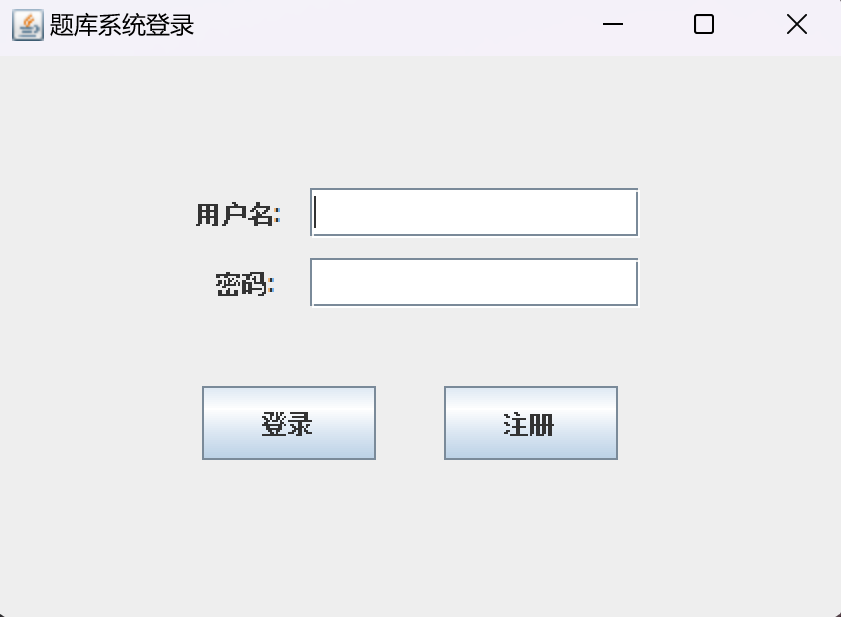
3、**jdbc**：存放项目依赖的外部库，如MySQL JDBC驱动，用于导出pdf文件的iText库，用于svg矢量图操作的batik库等。

项目运行的基本逻辑如下：

运行src/com/java/MainProgram.java，调用LoginDialog。登录成功后，指向控制器MainFrame.java，这是整个应用程序的核心框架，它管理程序的主页面切换，其它所有的面板 (JPanel) 都会在这个框架内展示。它包含了一些基本的布局设置（如侧边栏）和初始化代码。当点击侧边栏按钮时，可切换到对应的具体面板，以实现各种功能。

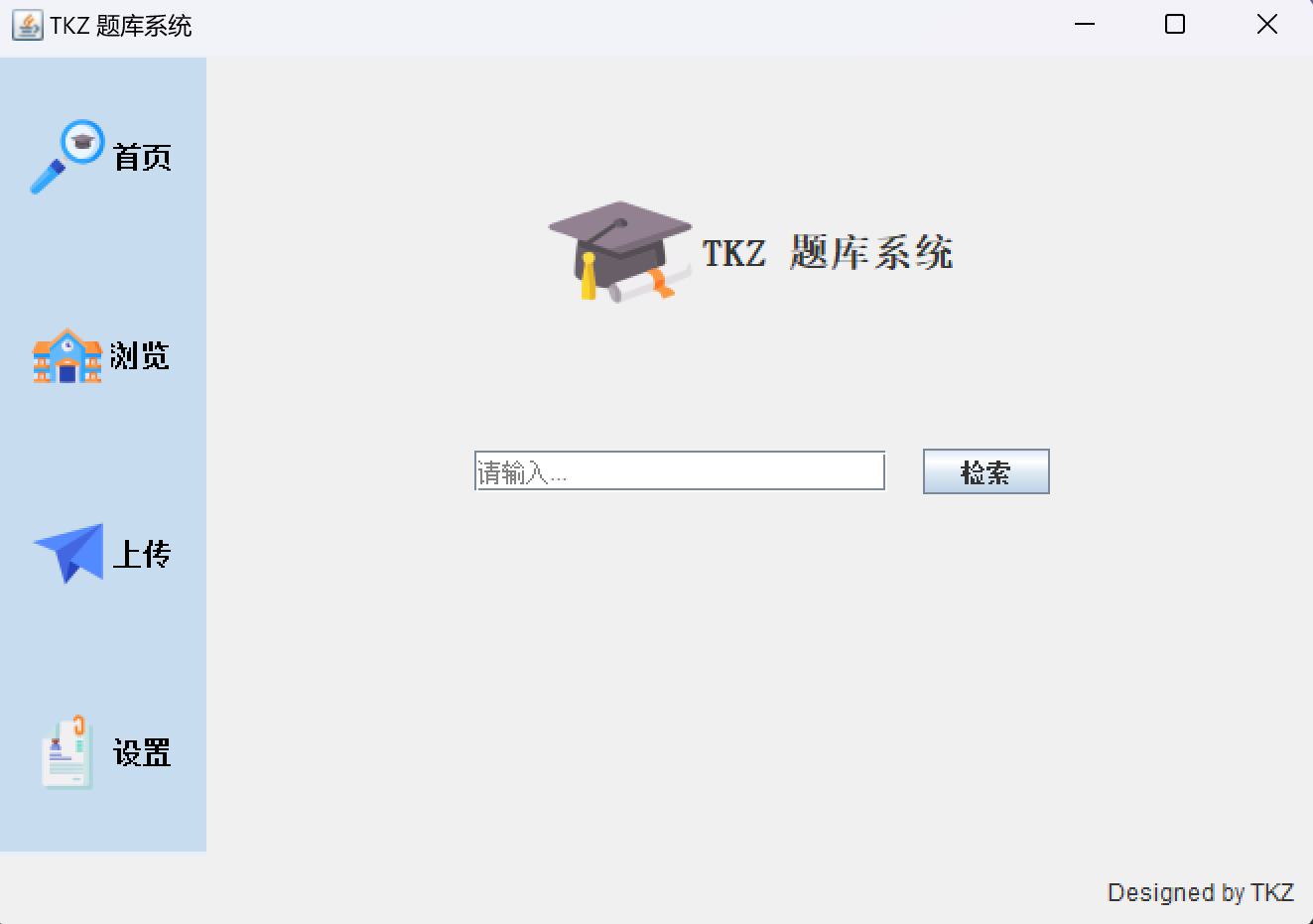
2.3 页面设计与功能

2.3.1 登录、注册页面

用户输入用户名和密码，点击登录按钮后，程序会验证用户信息。如果验证成功，则关闭登录窗口并通知主程序用户已登录。使用PreparedStatement 连接数据库并查询用户信息，确保用户名和密码的正确性。如果用户选择注册，将显示注册页面。

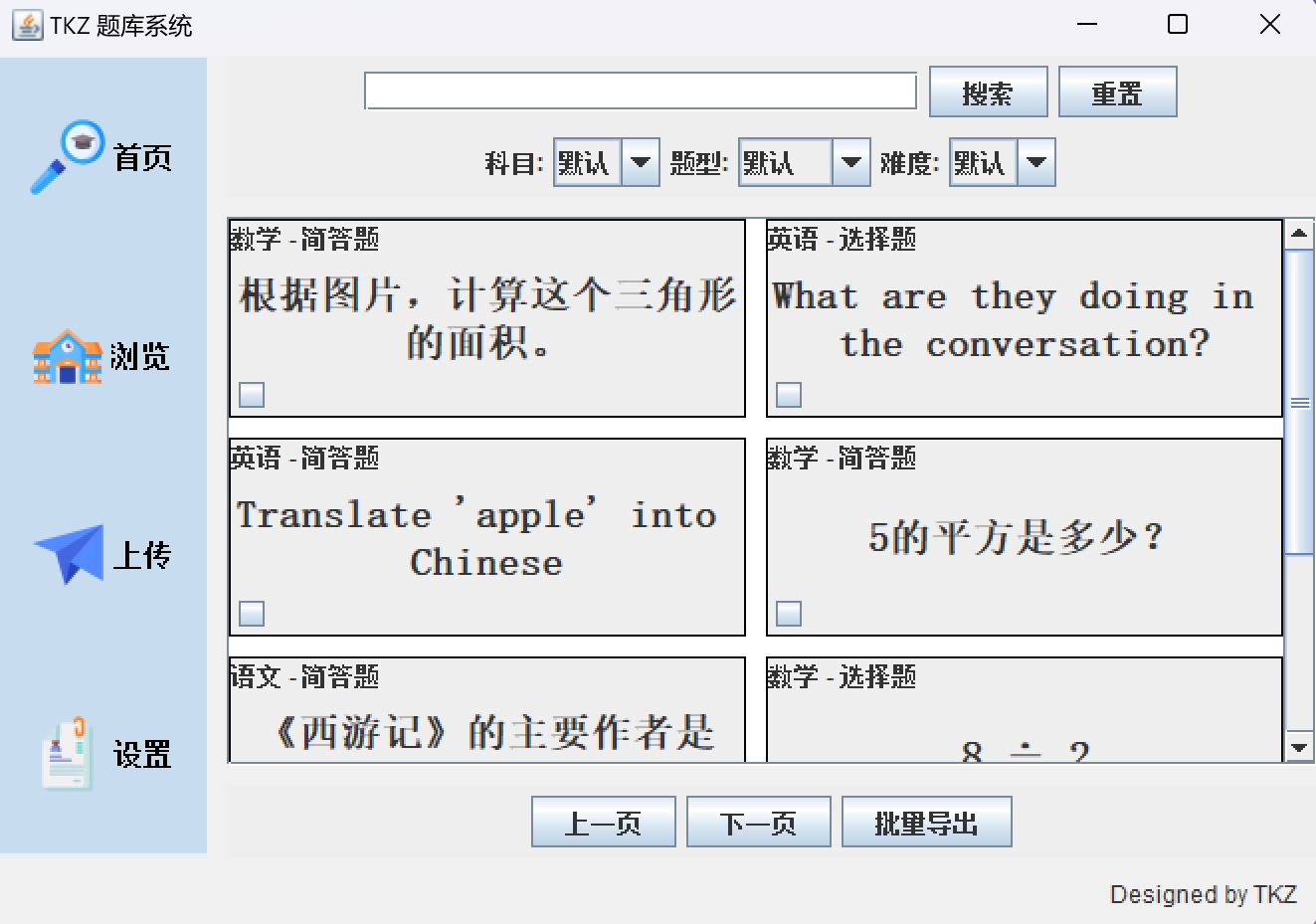
2.3.2 首页



登陆后进入首页, MainFrame.java中设置侧边栏，按钮绑定点击事件，在事件监听器中切换卡片布局面板，从而实现页面的切换。

首页面板名为HomePanel，页面类似Google，中间是一个检索框，图标为TKZ题库系统。检索框的功能与检索浏览页面类似，下文会提到。

2.3.3 检索与浏览题目页面



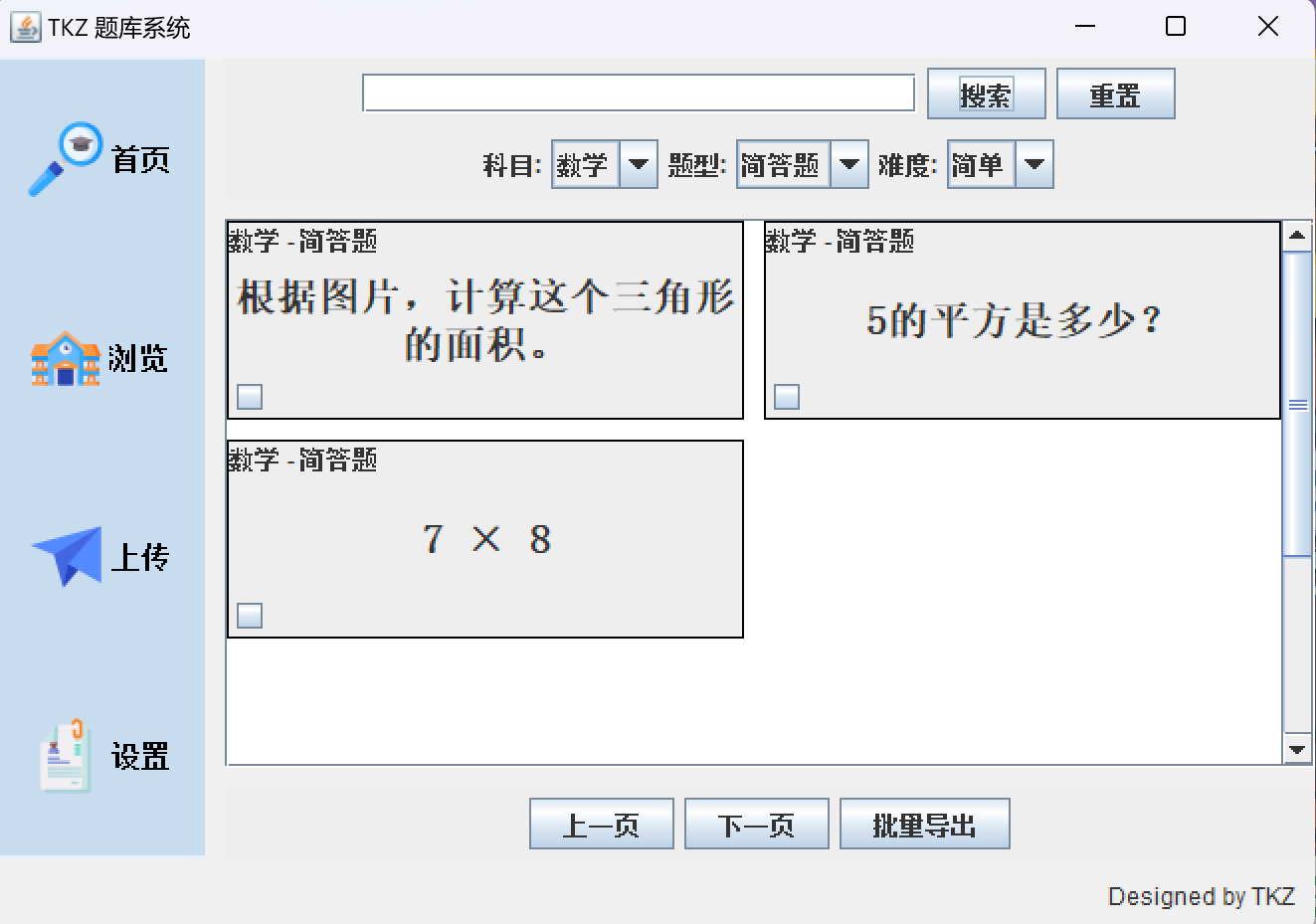
用户可以通过搜索框输入关键词进行搜索，也可以通过科目、题型和难度下拉框进行筛选，点击重置即清空条件。搜索结果以卡片形式展示在结果区域，每个卡片显示题目的摘要信息，用户可以浏览搜索结果，并通过翻页按钮在结果之间导航。点击卡片会显示题目的详细信息。点击每个题目卡片左下角的复选框，可选择题目，再点击下方按钮可进行批量导出。

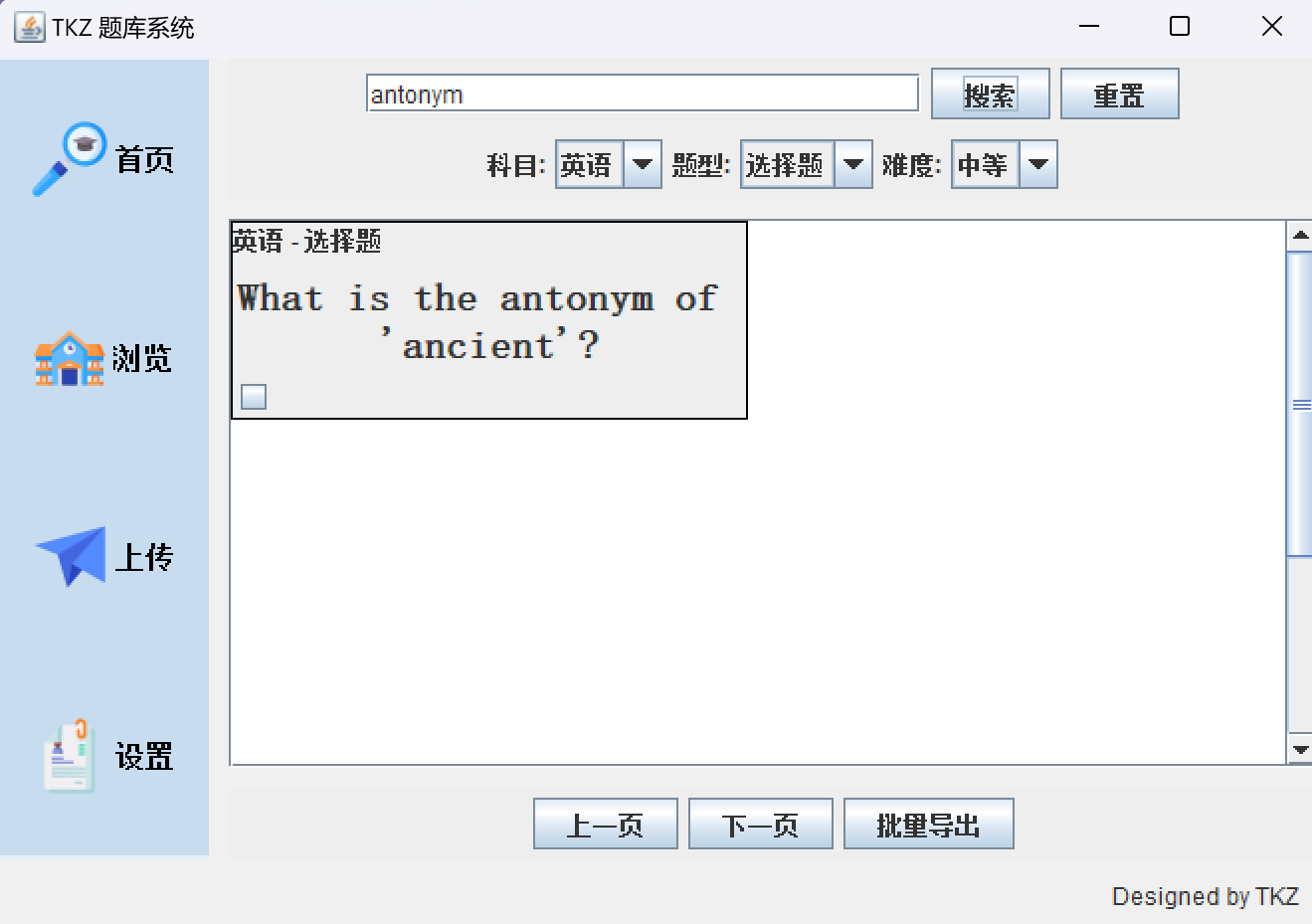
搜索和过滤逻辑：利用 searchQuestions 方法，根据用户输入和下拉框选择的值，使用 PreparedStatement 从数据库查询匹配的题目，并且用ResultSet获取题目信息。

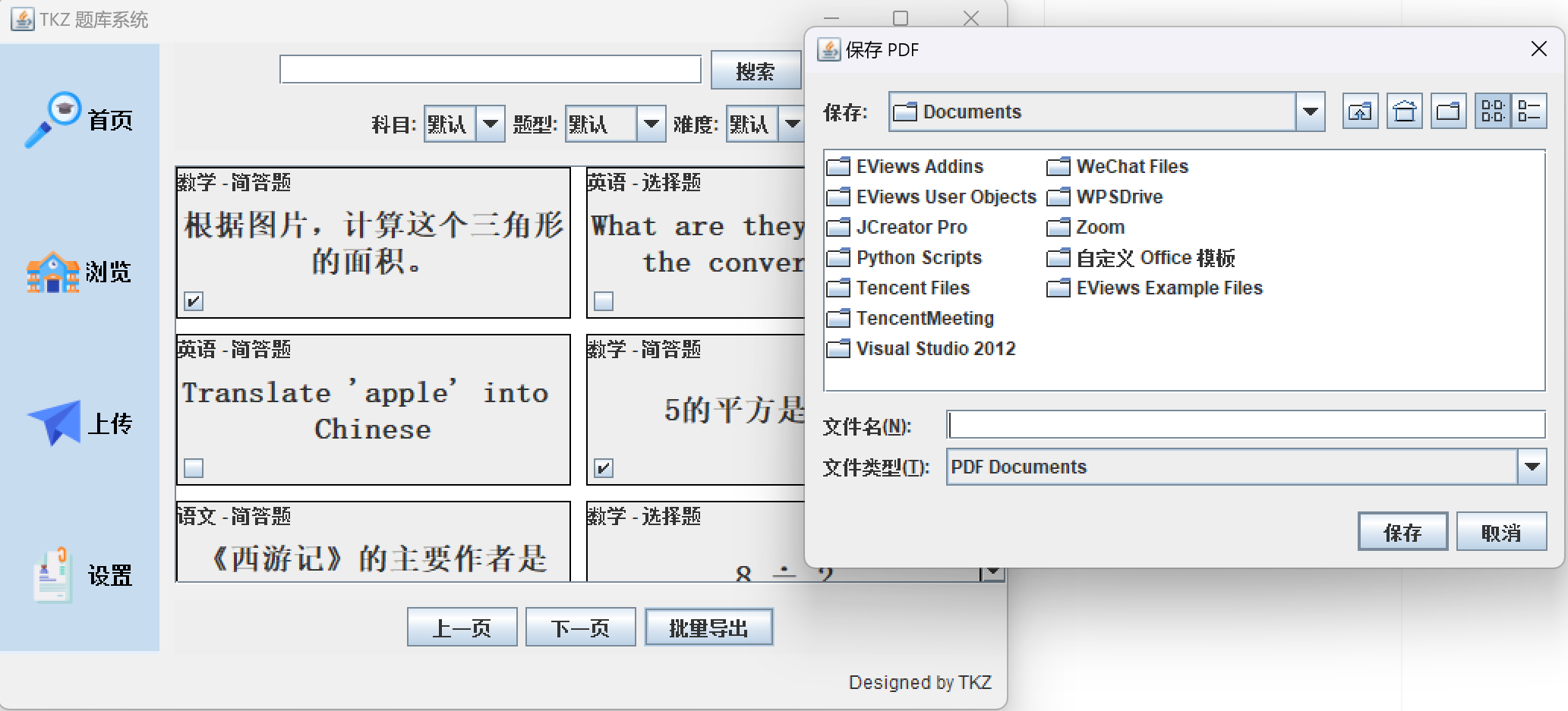
结果展示：updateResultPanel 方法用于更新结果区域，显示搜索结果的卡片视图。

卡片设计：每个题目卡片由 createQuestionCard 方法生成，包括题目的摘要信息。

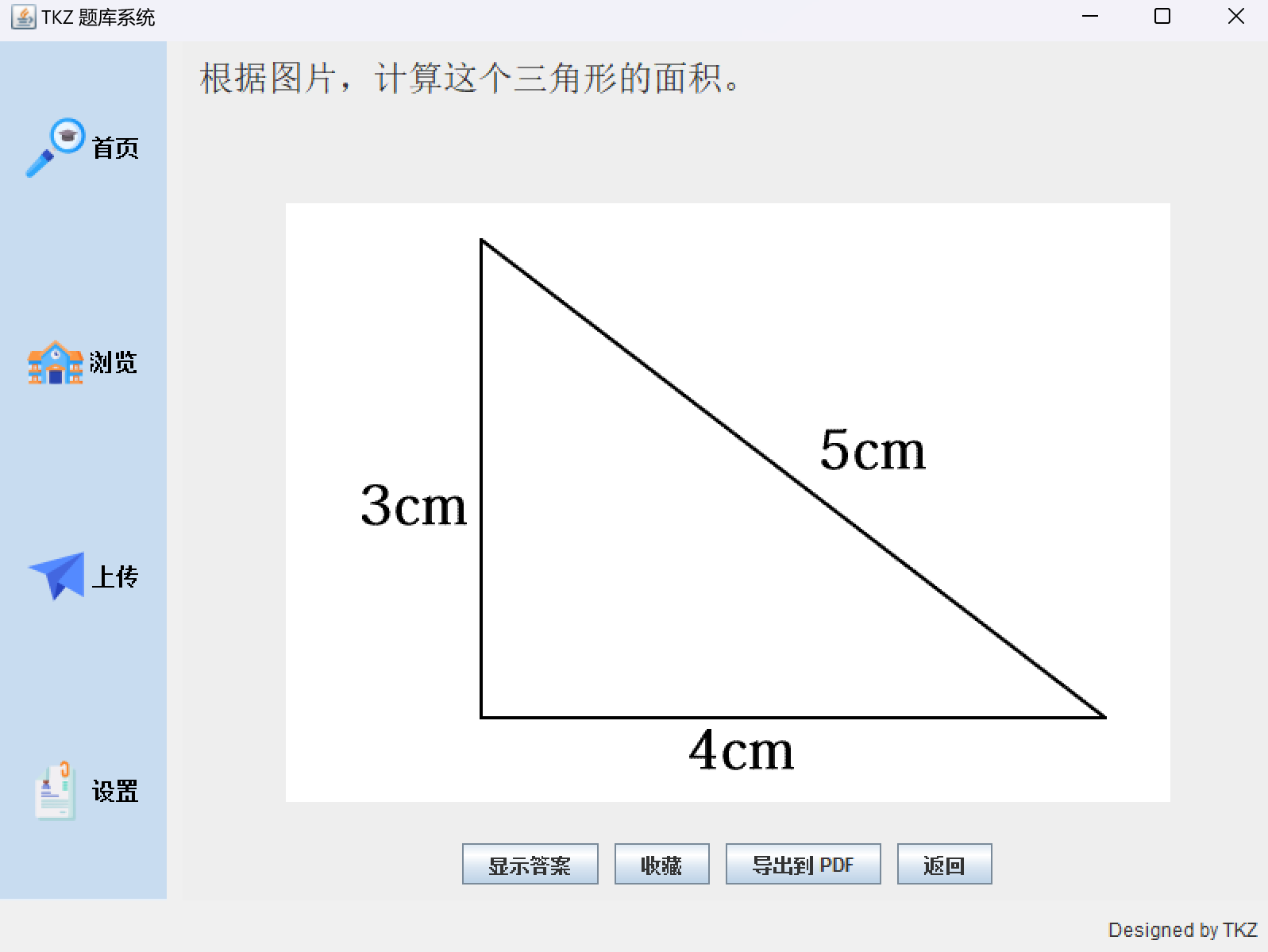
翻页功能：使用 changePage 方法实现翻页功能，通过改变 currentPage 的值来更新展示的题目。

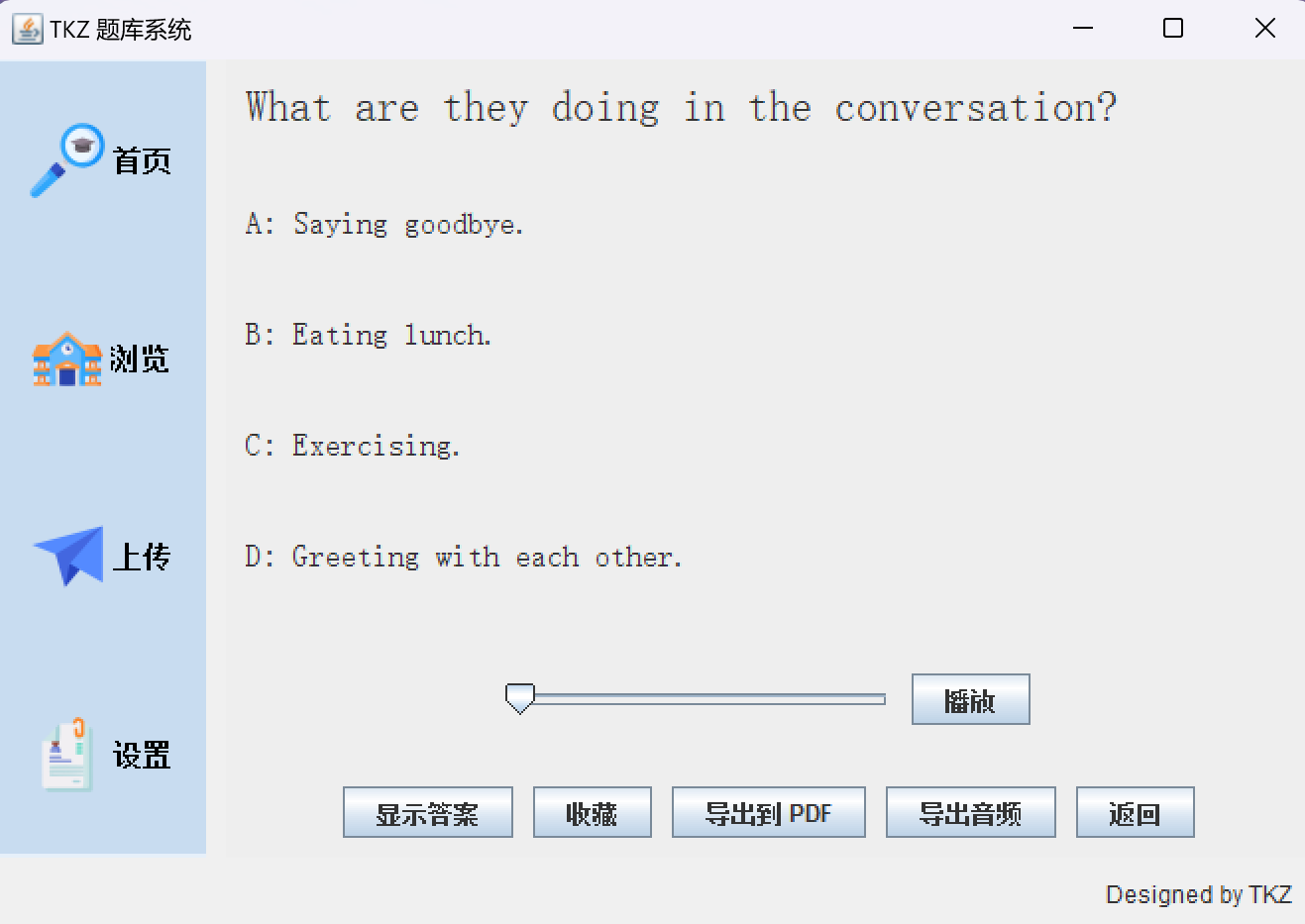






2.3.4 题目详情页面



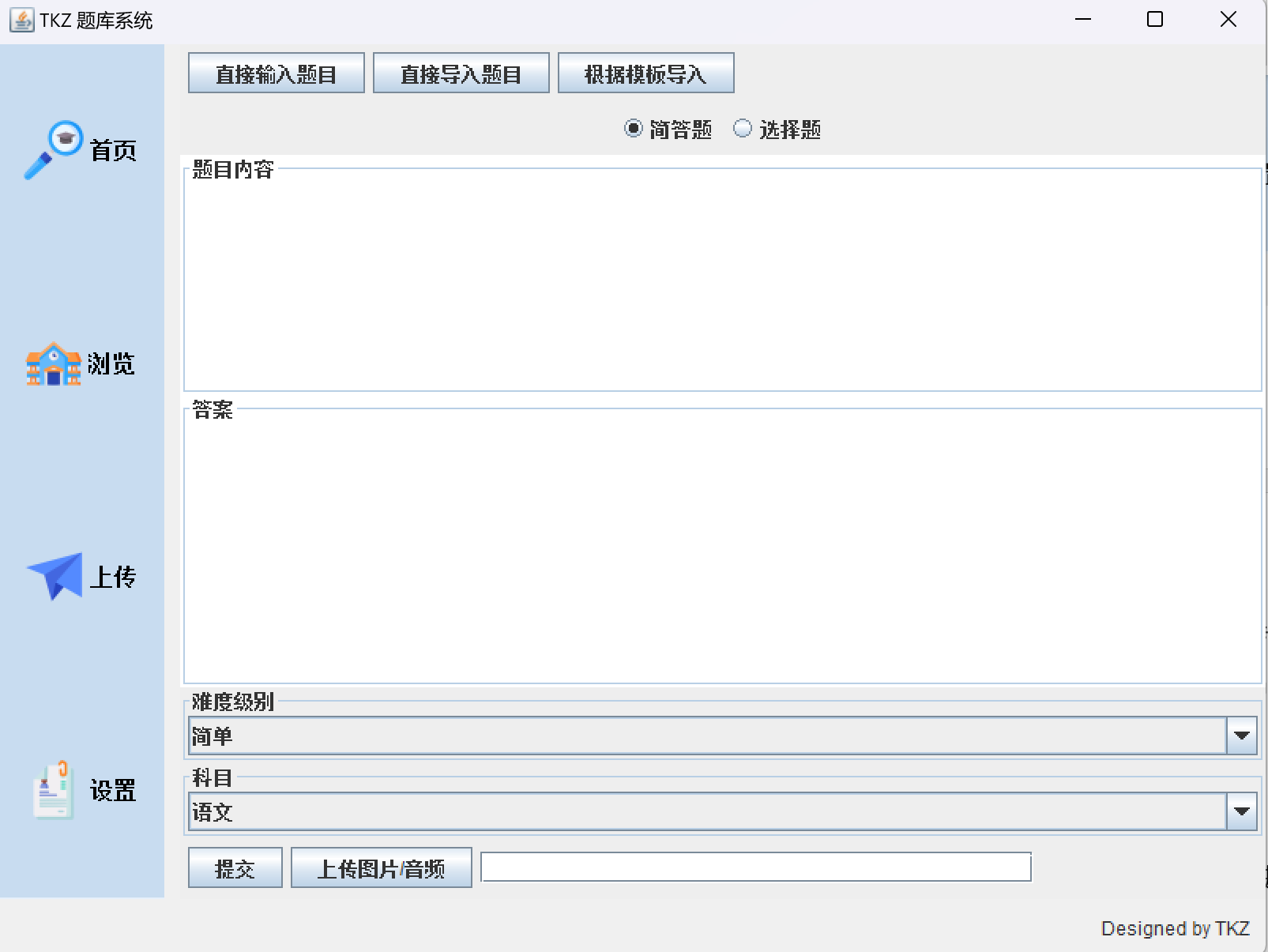


这是题目详情页面。它的功能是显示题目的详细内容，包括图形、图像和音频。使用JtextArea显示不可编辑的题目文本；如果是选择题，则会通过getOptionsFromDatabase 方法从数据库获取选项并显示。图形（矢量图）、图像（位图）和音频的格式分别是svg、png和wav，其地址储存在数据库中。通过 displayMediaContent 方法显示题目关联的媒体内容。对于音频播放控制，通过 toggleAudioPlayback 和 updateAudioProgress 方法，使用Clip对象播放声音片段，并且控制暂停和显示播放进度。

下方一排按钮分别提供显示答案、收藏题目的功能。collectQuestion方法会先验证该题目是否被收藏，然后添加进数据库。对于题目导出，通过iText Java类库，使用exportQuestionToPdf 方法，可将题目的文本和图片的相关信息导出为PDF格式。音频内容可单独导出。返回按钮允许用户回到浏览页面。

2.3.5 上传题目页面

点击三个 Button 可切换CardLayout，切换不同类型题目的答案输入界面。





这是TKZ题库中上传题目的第一种方式，直接输入题目（字面意思）。用户需选择题目类型（简答题或选择题），手动输入题目内容、答案、难度级别、科目。利用JfileChooser，可以在电脑文件中选择上传相关的媒体文件（图片或音频）。提交表单时会检查内容不为空值。



这是上传题目的第二种方式，直接导入题目。用户可以选择文件进行导入，支持文本或CSV文件格式。使用 JTable 和 DefaultTableModel ，实现了表格界面的显示和数据填充。用户仍需选择题目类型（简答题或选择题），根据选择动态调整表格结构（表格列数不同）。表格中将直接展示自动识别出的文件内容和与数据库列名的映射关系，用户可自行编辑，或是拖动数据块进行映射关系的调整。使用哈希表储存旧的映射关系，然后根据重组后的数据更新映射关系

Map<String, Integer> columnToIndexMapping;

columnToIndexMapping = **new** HashMap<>();

调整前：

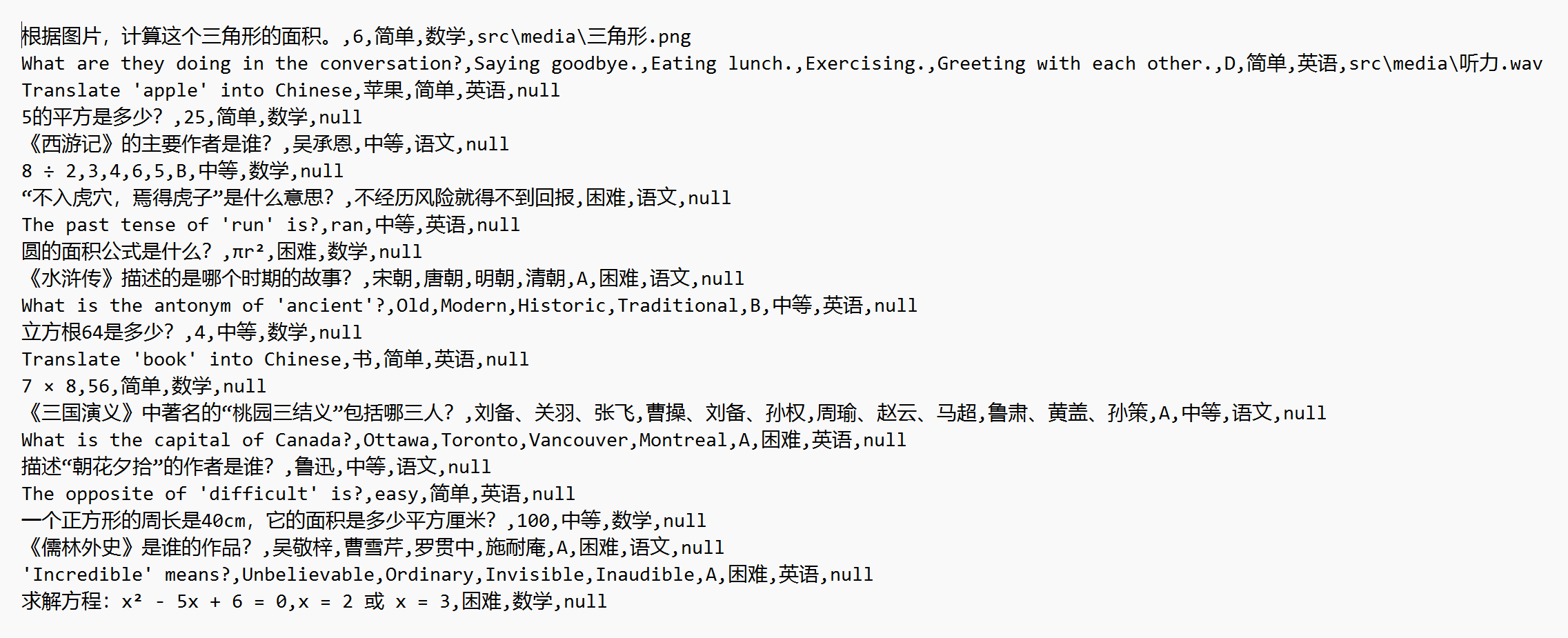


调换“选项D”和“正确答案”，调整后：



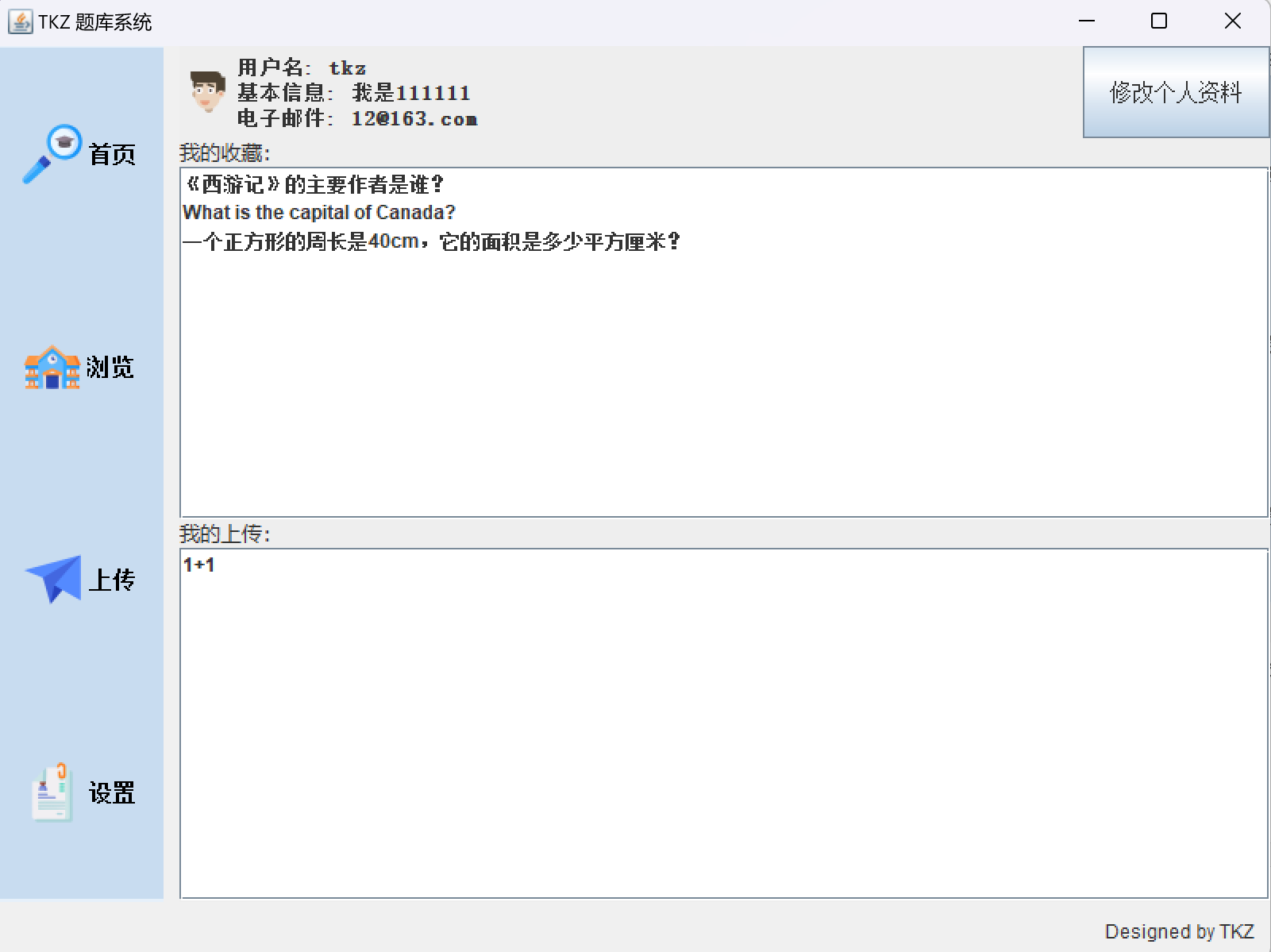


这是上传题目的第三种方式，根据模板导入题目。可读取固定格式的CSV文件，解析数据并自动保存到数据库。文件中数据可以有多行，支持批量导入功能。模板格式在页面中已给出提示，若没有相关数据则输入null。

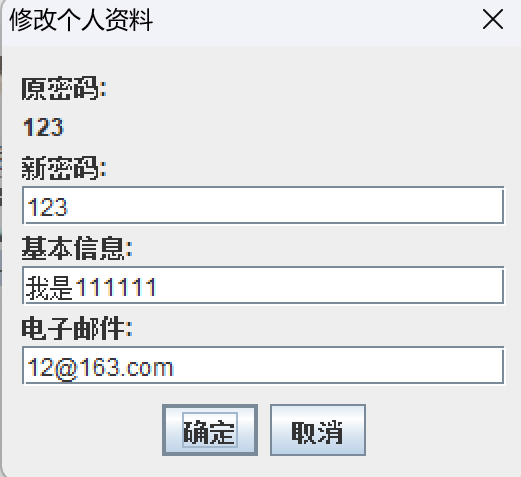


该面板兼容多种题目类型和格式，提供了灵活的数据输入和处理方式；有直接输入、直接导入和模板导入三种形式，满足不同用户需求。

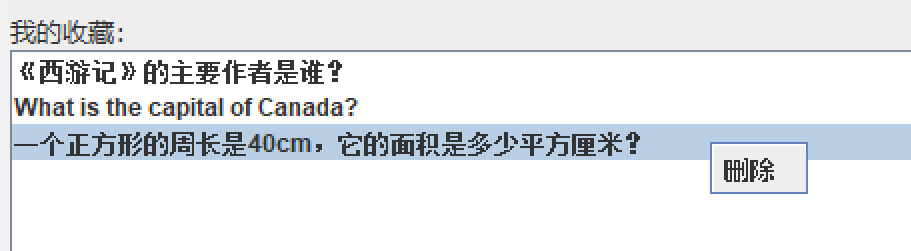
2.3.6 用户设置页面



该界面显示当前登录用户的基本信息（加载时会刷新）。允许用户更改密码、基本信息和电子邮件。当点击“修改个人资料”按钮，弹出一个对话框，用户可输入新的密码、信息和电子邮件。允许用户更改头像，上传头像时自动调整图片大小以适应头像区域。



用户还可以查看他们收藏的题目列表和上传的题目列表。支持右键点击进行更多操作（删除功能），或是双击查看详细信息。在删除操作后，列表会自动更新以反映变化。

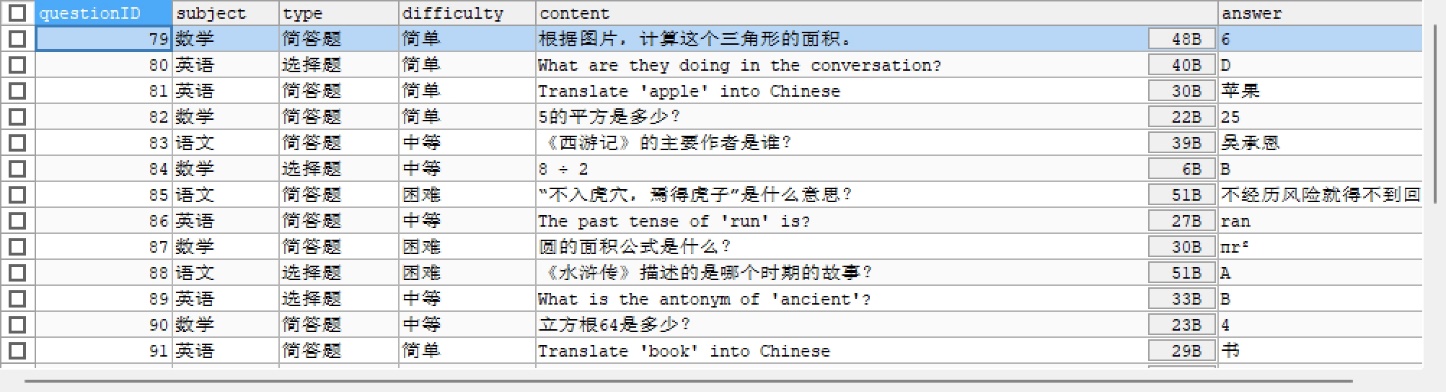


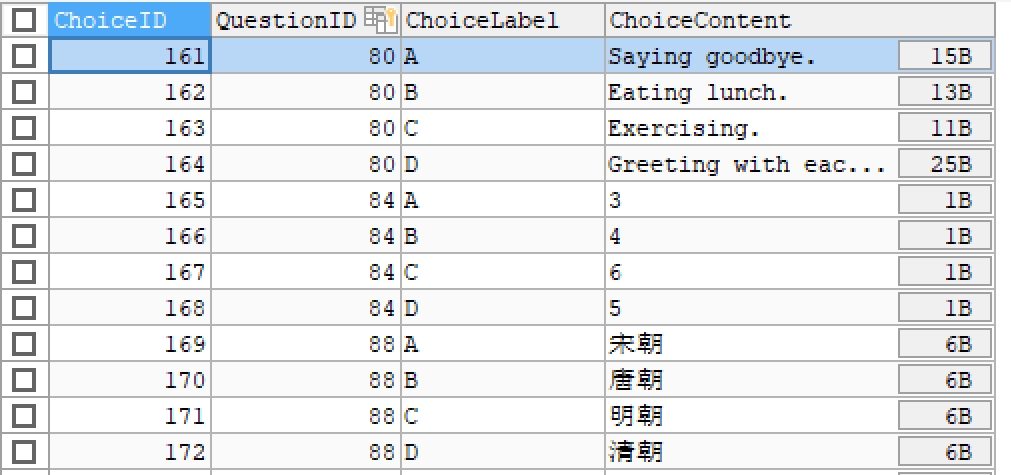
2.4 数据库设计

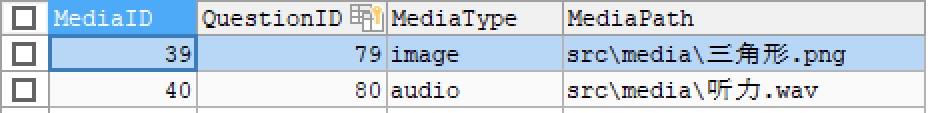
数据库采用MySQL本地数据库，复现该项目时需要运行项目文件夹中的sql文件，且在代码中更换密码。

数据库共6个table，分别为users、questions、favorites、user\_questions、question\_choice、questionmedia。下表是对应关系。

| 内容 | 主要存储内容 |
| --- | --- |
| users | 存储用户的基本信息 |
| questions | 存储题库中的问题 |
| favorites | 存储用户的收藏信息 |
| user\_questions | 存储用户和问题之间的关系 |
| question\_choice | 存储选择题的选项 |
| questionmedia | 存储与问题相关的媒体文件 |

以下是一些表中的内容展示：





MediaType: 媒体类型（如 "image", "audio"）。

MediaPath: 媒体文件的存储路径。

2.5 其他实现

中文字体设置，引入simsun.ttf，解决了中文字符无法导出的问题。

设置icon图标，美化界面。

设置models实体库，作为和数据库交互过程中的临时存储库，以维护安全性。

**3．小结与回顾**

在此次的java项目开发中，我基本实现了所需要的全部功能，保证了一定的美观性。

在尝试完成项目时，我经常遇到一些未知的bug，尝试良久都不知错误在哪里。Java项目中类、函数的相互调用，还有定义变量的作用域总是能把我轻易绕晕。而当我迷茫无助的时候，我时常会浏览一些CSDN文章；亦或是询问ChatGPT，而它总能够如雪中送碳般施以援手。我确信，在当代社会，能够灵活运用AI工具，会写提示词，这样的能力是非常重要的，有助于我们解放生产力。

在此次开发中，我进一步了解了Java项目管理与依赖建立、Java的各种数据类型、Java操作数据库、Java 项目中类的相互调用、Java GUI开发，受益匪浅。当然，我的项目仍然存在着问题。例如未使用前端工具，导致在页面美化上不足；在某些功能上用户体验不够完美；框架运用的缺失，让我的项目减少了许多拓展的可能性。

总之，我将继续对编程语言的学习，为我未来的发展打下坚实基础。