软件需求说明: 基于 TEX 的题库整理与试卷构建系统 (0.1 版本)

Mathlang 组织

2019 年 11 月 19 日

目录

1 修订历史

维护者	时间	变更日志	版本
梁钰栋	2019/11/19	• 初步定稿软件需求说明 (本地版本) • 在 GitHub 新建仓库托管代码	0.1

2 简介

2.1 项目目的

基于 TEX 的题库整理与试卷构建系统 (下称本项目) 旨在将试卷的格式与数据分离 (即分离为前端, 后端及数据库): 使用 Json 数据格式或数据库 (如 SQL) 来存储题目数据; 使用 TEX 系统对题目进行排版, 数据与版式之间通过 Python 编程语言操作数据库以生成 TEX 源代码.

本项目的思想来自于使用特定工具解决特定的问题, 同时从工程的角度审视问题, 寻求易于拓展的解决方案: (1) TEX 擅长排版, 但用作运算 (如计数器的数学运算等) 的用户接口较弱, 而支持嵌入式脚本的原生引擎 LuaLATEX 则需要更复杂的逻辑实现, 不利于以后对程序进行改进与维护; (2) Python 为“胶水语言”, 功能库众多, 可以根据特定需求做特定事情 (比如从数据库筛选题目, 制作网页应用等), 但是排版所用库 (例如 PyLaTeX, Python-docx) 提供的功能不如 TEX 丰富且学习成本较大; (3) 数据库仅用来存储数据与筛选数据, 所以在本项目中不能单独使用. 而 Python 存在 ORM (Object Relational Mapping, 例如 SQLAlchemy[1]), 可用来操作数据库. 但是本项目一开始是作为本地版本开发, 数据应追求可读性, 所以使用纯文本记录数据 (例如数据格式 Json) 可以更方便地修改数据.

2.2 文档约定

“基于 TEX 的题库整理与试卷构建系统” 的文档 (下称本文档) 参考 Wikipedia[2] 与 GitHub[3] 的提示进行编写, 其中文档采用 TEX 系统进行排版, 除数据外的代码等部分均托管在 GitHub 上. 需要注意的是, 本文档的源代码中包含大量的交叉引用, 如无必要, 请勿对文件进行复制粘贴, 直接转载到其他网站上.

名词	解释
本地版本	使用 TEX, Python, Json 的版本, 旨在在本地进行题库的管理, 后期可能会在在线版本中提供题库上传与导出的功能, 用以管理个人的题库.
在线版本	使用 TEX, Python, SQL 的版本, 旨在在网页进行题库的管理, 后期可能会在在线版本中提供题库上传与导出的功能, 用以管理团体的题库, 同时在线合作完善题库.
版本控制	对软件开发过程中各种程序代码, 配置文件及说明文档等文件变更的管理.
TEX	本项目使用的 TEX 发行版为 TEXLive 2019, 除此之外的所有发行版 (例如 CTEX) 均不保证模板的正常运行, 更多资料请参考 Read the Docs.
SQLAlchemy	托管在 PyPi 上的 Python 库, 具体版本需要见程序中的依赖声明 (requirements.txt, Pipfile 等), 此外在本地版本中不会使用.

2.3 目标读者与阅读建议

本文档的目标读者为本项目的开发人员及在初期使用本项目的用户. 以下将按照不同的读者类型分别列举其阅读建议. 此外, 如果您不属于此处列举的目标读者, 但是仍想获取阅读建议, 请及时提出 Issue, 我们会在注意到后及时补充至本文档中.

读者类型	阅读建议
开发人员	建议着重阅读第 4 节, 第 5 节及第 6 节, 如果对本文档存在疑问请及时提出 Issue.
初期用户	建议浏览第 2 节及第 3 节, 如果对本文档存在疑问请及时提出 Issue.

2.4 项目范围

本项目重点为题库整理与试卷构建. 在前期项目 Exam 中, 我们完全使用 TEX 编写题库, 通过 TEX 的 `\input` 机制将不同时间的题目分离, 与此同时产生一系列问题: (1) 题目看似分离, 但是在根据已有题目来构建新的试卷时, 还需要将 TEX 的冗余部分重新书写一遍; (2) 为了追求注释部分 `\input` 后仍可以运行, 导言区需要将所有的依赖记录下来, 从而造成性能上的损失; (3) 前期项目提供众多的 `API` (尤其解题框), 构建题库的时间跨度较大导致使用 `API` 不统一.

而本项目在设计时改进了上述的缺点, 即将格式与数据完全分离. 题库存储的时候不需要考虑 TEX 的格式, 试卷构造的时候会针对不同的题目进行导言区的构造 (耗费时间可以忽略不计, TEX 宏包加载的时间远大于程序运行的时间), 在提高题库自定义化的同时, 更好地管理题库的数据.

如果后期添加了在线版本的支持, 则可以将用户们的题库进行整合, 在标签明确的前提下, 可以快速构建试卷. 同时, 可以邀请一部分与试卷无利害关系的用户, 对试卷的内容解析等进行更好地优化, 同时记录修改的历史记录, 用以量化用户的贡献.

2.5 参考资料

[1] BAYER M. SQLAlchemy[G/OL]//BROWN A, WILSON G. The Architecture of Open Source Applications Volume II: Structure, Scale, and a Few More Fearless Hacks. [S.l.]: aosabook.org, 2012. http://aosabook.org/en/sqlalchemy.html.

[2] Wikipedia contributors. Software requirements specification — Wikipedia, The Free Encyclopedia[Z]. https://en.wikipedia.org/w/index.php?title=Software_requirements_specification&oldid=925231559. [Online; accessed 19-November-2019]. 2019.

[3] Jean-Philippe Eisenbarth. A latex template for a Software Requirements Specification that respects the IEEE standards.[Z]. https://github.com/jpeisenbarth/SRS-Tex. [Online; accessed 6-April-2016]. 2016.

3 总体描述

3.1 产品总览

本项目首先开发本地版本, 如果有需求, 则继续开发在线版本 (即套用离线版本, 但使用数据库存储数据, 同时添加类似 `GitHub` 的网页应用).

Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.

3.2 产品功能

版本	功能
本地版本	• 使用 `Json` 数据格式存储题目以构成题库, 纯文本的格式方便用户查看与修改. 但是格式需与在线版本的数据库结构一致, 方便数据库迁移; • 使用 TEX 系统构建试卷, 提供默认样式, 同时支持用户自定义样式; • 使用 `Python` 连接数据与格式, 抽象题库的筛选算法; • 最初提供命令行编译版本, 后期可以提供不同平台的 `GUI` 程序 (使用程序均可跨平台).
在线版本	• 注册新用户, 上传与下载 `Json` 格式的题库, 并且在线对题库进行编辑; • 根据标签对题库进行筛选, 构造试卷, 同时控制分数及难度; • 用户管理类似 GitHub. 公有题库可以通过用户页进行查看, 私有题库则需要通过分享链接进行修改 (或添加为合作人员), 同时进行版本控制; • 在线版本的数据库存储需进行加密, 防止后台出现问题.

3.3 用户类别及其特征

Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.

3.4 操作环境

Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.

3.5 产品设计及其实现中的约束

Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel

operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).

3.6 用户文档

List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.

3.7 假设与依赖

List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).

4 外部接口的要求

4.1 用户接口

Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.

4.2 硬件接口

Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.

4.3 软件接口

Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial compo-

nents. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.

4.4 通信接口

Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

5 系统功能

This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.

5.1 系统特性一

Don't really say "System Feature 1." State the feature name in just a few words.

5.1.1 说明及其优先级

Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).

5.1.2 灵感来源

List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.

5.1.3 功能性需求

Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or

invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

- REQ-1:
- REQ-2:

6 系统其他非功能需求

6.1 系统性能需求

If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

6.2 系统安全需求

Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.

6.3 系统保密需求

Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.

6.4 系统质量指标

Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

6.5 商业授权规则

List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.

7 附录

Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.

7.1 附录 A: 词汇表

Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.