Project Delivery Methodology (PDM)

Software Requirement Specifications (SRS)

Rick Waldron

MarkR

VERISION: 1 REVISION DATE: 27/7/2022

Approval of the Software Requirement Specifications indicates an understanding of the purpose and content described in this deliverable. By signing this deliverable, each individual agrees with the content contained in this deliverable.

Approver Name	Title	Signature	Date

Table of Contents

Section 1 Purpose

The purpose of the Software Requirement Specification is to describe the business requirements in detail. This document will establish the application business requirements, business processes, functional and data requirements. Upon completion of this document, this document will be used for designing the application.

Section 2 Business Requirements

2.1 Define Business Requirements

2.1.1 Business Area – Exam marking

Priority 1:⇒**Application should be user-friendly and appealing to all teachers**

Priority 2:⇒**Application should save time over current processes**

2.1.2 Business Area – Exam feedback

Priority 1:⇒**Application should provide automated feedback for students**

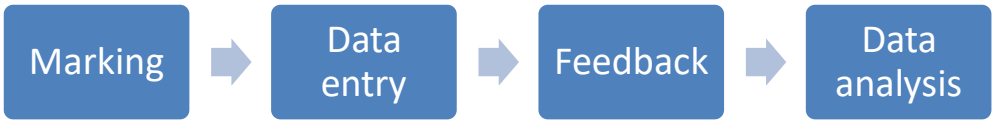
Priority 2:⇒**Application should provide an exam summary for teachers**

2.2 Business Process Model

2.2.1 Business Process Definitions

- **Business Area – Exam marking**
  - Marking: Evaluating student responses on an exam
  - Data entry: Entering data into a program, question-by-question
- **Business Area – Exam feedback**
  - Feedback: Reporting to students their outcomes on an assessment
  - Data analysis: The process of using data to support decision-making

2.2.2 Business Process Flow



2.3 Functional Requirements

2.3.1.1 Function 1: Recording exam results in a database

2.3.1.1.1 Function 1 Purpose

Enter marks into the application for processing.

2.3.1.1.2 Function 1 Inputs

Exam information (class name, students names, emails, number of questions, marks for each question, tags for each question) and each student's mark for each question.

2.3.1.1.3 Function 1 Operations

Setting up the exam in the application, putting exam papers in order and entering class marks one by one.

2.3.1.1.4 Function 1 Outputs

A database with all student marks for each question.

2.3.1.2 Use Case 1: Final exam for a unit

Most commonly, this application will be used for unit exams. Teaching teams will set this up together and mark individually, entering results into a linked database.

2.3.2.1 Function 2: Sending exam reports to students

2.3.2.1.1 Function 2 Purpose

Give feedback to students.

2.3.2.1.2 Function 2 Inputs

The data entered previously is used to create a report. The type of report is selected by the teaching team and calculated based on percentages for certain tags.

2.3.2.1.3 Function 2 Operations

A report button is pressed and options selected.

2.3.2.1.4 Function 2 Outputs

A personalized email to students (and optionally parents) that gives targeted feedback on their exam.

2.3.2.2 Use Case 2: Unit exam feedback

Question tags such as topic, difficulty and type of question are used to create detailed reports. This is presented as an email to students to inform their future study.

2.3.3.1 Function 3: Data analysis for teachers

2.3.3.1.1 Function 3 Purpose

Help teachers reflect on the validity of the exam and the effectiveness of teaching practices.

2.3.3.1.2 Function 3 Inputs

The exam data is used to generate a report, customizable by the teaching team.

2.3.3.1.3 Function 3 Operations

An analysis button is pressed and options selected. The application uses data to generate various statistics.

2.3.3.1.4 Function 3 Outputs

A summary of students results, both overall and question by question. Potentially, grade boundary recommendations could be generated to help accurately translate results in the performance scales.

2.3.3.2 Use Case 3: Exam summary informing exam writing

For subsequent years, teachers could use exam summary to help write better exams. For example, if the first question on the exam (intended to be the easiest) was done poorly by a cohort, it might need to be reworded, modified, moved or deleted from the next years exam. Over time, exams are calibrated to meet the requirements of the syllabus.

2.4 Non-functional requirements

2.4.1 User privacy

2.4.1.1 Feature 1: Privacy policy

The application will make it clear to the user what the intention of all personal information is and make many fields optional. For example, it may be useful to enter student emails in so that generated reports can be sent to individual students. However, this is optional, and the policy will specify that these emails will not be shared or used for any other purpose.

2.4.1.2 Feature 2: User data

User data will not be stored for any other purpose other than generating reports. There will no extraneous data collected, such as location data, and all data can be easily deleted when requested.

2.4.2 Security

2.4.2.1 HTTPS

Hypertext Transfer Protocol Secure (HTTPS) will be used to communicate over the network between the client and the server. This protocol uses encryption to protect data while in transit, making it far less likely for data to be breached in a man-in-the-middle attack.

2.4.2.2 Password requirements

Passwords will need to have a certain number of characters, and require a capital letter, a number and a special character. This will make it much less likely for data to be breached in a brute force attack.

2.4.3 Accessibility

2.4.3.1 Large text

The size of the text will be customizable, so that some people with minor visual impairments may be able to access the software by increasing the size of the text.

2.4.3.2 Colour contrast

The design of the web application will be simple, with contrasting colours such as black and white making the site easier to view for people with colour vision deficiency.

Section 3 Data Management Requirements

3.1 Archive/Purge Requirements

Data would need to be stored as the long as school is using the application. If the school leaves the service, data could be stored for historical reasons if the server space allows, and potentially archived for a specified time (e.g. a year).

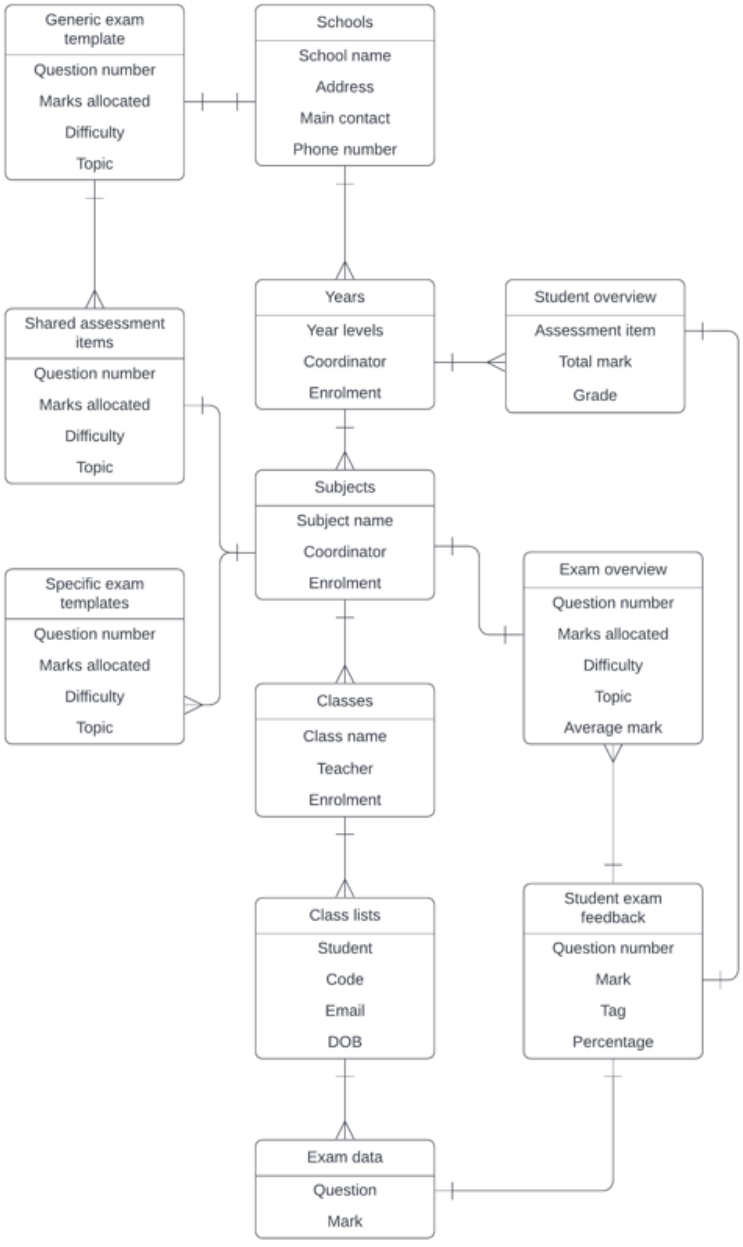
3.2 Audit Requirements

A yearly data audit is recommended to ensure schools are using the application effectively, and to optimise storage space. Schools should be doing their own data audits, and it is not necessarily the responsibility of the application to protect this data. Schools would be encouraged to back the data up on their own systems.

Section 4 Conceptual Data Model

4.1 Table Names and Descriptions

Below is the diagram of the databases for the MarkR project, based on the entity-relationship model.



4.2 Integrity Constraints

One of the major integrity constraints will be ensuring that students in classes are member of the year level and school. There will also be constraints on the type of data that can be entered in most fields. For example, mark data will need to be integers (or half-marks if schools require) and emails will need to be validated as a correct school email address.

Section 5 Reporting Requirements

Monthly reports are required for this application. The purpose of these reports is mainly to fix bugs and respond to client feedback. All members of the software team will work on this report, and it will be finalised by the project coordinator. Annually there will be a data audit performed by the data team at the software company. The reports will be shared through email.

Section 6 References

Document No.	Document Title	Date	Author

Section 7 Glossary

Entity-relationship model: A way of representing the relationship between "entities" such as objects or concepts. Entities can contain attributes, and the links between them can show the type of relationship e.g. one to many.

Section 8 Document Revision History

Version	Date	Name	Description
1	27/7/2022	Version 1	Initial draft

Section 9 Appendices