**Exam Management System**

Application Requirements Document (ARD)

Students: Roi Tiefenbrunn, Ofek Nov, Idan Aharoni, Mor Abo

Customer: Dr. Mayer Goldberg

Table of Contents

[Chapter 1 – Introduction 3](#_Toc157719032)

[The Problem Domain 3](#_Toc157719033)

[Context 3](#_Toc157719034)

[Vision 4](#_Toc157719035)

[Stakeholders 4](#_Toc157719036)

[Software Context 4](#_Toc157719037)

[Access 4](#_Toc157719038)

[Regular Use (Course downtime) 4](#_Toc157719039)

[Pre-Exam 4](#_Toc157719040)

[Post-Exam 4](#_Toc157719041)

[Chapter 2– Usage Scenarios 5](#_Toc157719042)

[User Profiles — The Actors 5](#_Toc157719043)

[Use cases : 6](#_Toc157719044)

[Special Usage considerations: 6](#_Toc157719045)

[Chapter 3– Functional Requirements 7](#_Toc157719046)

[Workflow Management System 7](#_Toc157719047)

[Content Management 7](#_Toc157719048)

[Chapter 4 – Non-functional requirements 8](#_Toc157719049)

[Implementation constraints 8](#_Toc157719050)

[Platform constraints 8](#_Toc157719051)

[Special restrictions & limitations 8](#_Toc157719052)

[Chapter 5 – Risk assessment & Plan for the proof of concept 9](#_Toc157719053)

[Risk assessment 9](#_Toc157719054)

[Plan for the proof of concept 9](#_Toc157719055)

[Research: 9](#_Toc157719056)

[Preliminary: 9](#_Toc157719057)

[Proof of Concept: 9](#_Toc157719058)

[Appendices 10](#_Toc157719059)

# Chapter 1 – Introduction

## The Problem Domain

With the increase in the size of the student body, it has become difficult to manage grading within the given time constraints. This has led many courses to adopt a multiple-choice format in their exams, which is faster to grade, and easier to analyze. And still, the creation of challenging multiple-choice tests in the sciences is a non-trivial task and creation of one-time-use questions is costly and unsustainable.

Over the years, Dr Goldberg has developed a unique approach to writing multiple-choice questions in the science, based on the idea of meta-questions:   
A meta-question represents an abstraction over a multiple-choice question, and is a structure from which it is possible to derive several, often a great many, multiple-choice questions. To support the creation of exam questions and the synthesis of exams, Dr. Goldberg has defined a DSL for describing various kinds of meta-questions, and a software system that synthesizes exams from a given selection of meta-questions.   
The system is written in Scheme, and uses text-files to store meta-questions.

As the number of meta-questions grew, the weaknesses of the current system have become apparent, and a re-design is in order. The most notable change in the new design is that it is designed as a workflow-management system that enables different course-staff members to collaborate in creating and developing new questions.

Our proposed solution is to develop a system that will be an upgrade to the existing one, while still relying on the same philosophy and approach of meta-questions maintenance, offering better solutions where the current system fails - with focus on scalability, accessibility, labor management all while being easy to learn and use.

## Context

The system is designed to manage work inside of an academic course between the lecturer(s) and the TAs.   
Through the system will the course administration develop, test and create multiple-choice questions and generate exams.   
The system will be installed remotely on university’s servers and be accessed through web applications, while the data will be stored on databases.

## Vision

* Workflow Management System:
  + Support various roles (for TAs, graders, instructors, etc).
  + Manual+Algorithmical generation of tasks based on priority and needs.
  + Manual+Algorithmical spread of tasks throughout configured workforce.
  + Tracking productivity (= blame feature).
* UI:
  + Web-based system.
  + Dashboard based on Role.
* Content creation:
  + Manage content by subjects/keywords/classes.
  + Creation and management of stems, meta-questions, questions, appendices.
  + Creation and management of possible solutions/distractors per question.
* Output creation:
  + Flexible LaTeX-based creation of exams/keys/solutions.
* System:
  + Have both WMS and content handled in DB.
  + Version control of questions, handled in DB.
  + Easy install, migration, backup and cloning of the system.

## Stakeholders

**Client**: Dr. Mayer Goldberg – Supervisor Lecturer for “Compiler Construction” course.   
**Users**: “Compiler Construction” course administration – lecturers, TAs, assignment checkers.   
**Additional people of interest**: Other courses’ supervisors are interested in the system when it comes to existence.

## Software Context

### Access

Access to the system will be through browsers and only while connected to the university’s network.

### Regular Use (Course downtime)

Users will access the system to maintain question stock based through various tasks completion based on their respective roles.   
High roles will be able to track work velocity and quality of lower-level roles.

### Pre-Exam

High roles will be able to produce exams/keys/solutions through the system and by choosing subjects distribution.

### Post-Exam

Inputting students’ answers will be used to further analyze questions/answers/distractors quality.

### Main Expected functionality:

The system should support the following functionalities:

* Add meta question: Add meta question (questions, solutions distractors) to a specific course.
* Ask for task: A user can get a task to work on, task could be to validate correctness of meta-question solution and distractors, or to add a new meta question. The task will be pulled from a task pool which will prioritize the tasks
* Complete a task: complete a task received from the task pool
* Add/Remove course staff: Course administrator can add and remove staff from his course.
* Generate exam: Course administrators can automatically generate exams which will be based on the meta-questions

# Chapter 2– Usage Scenarios

## User Profiles — The Actors

Certainly! Let's describe the main user categories or profiles (actors) for the Exam Management System:

1. System Administrator:
   * Characteristics:
     + Manages the overall WMS at the department/university level.
     + Handles tasks like migration, installation, updates, and system management.
     + Manages courses within the WMS.
   * Data Exchange:
     + Migrates, installs, and updates the WMS system.
     + Manages courses, including creating, backing up, and removing course entries.
2. Course Administrator:
   * Characteristics:
     + Responsible for managing courses, defining course staff, and overseeing exams.
     + Assigns roles to course staff and defines exam parameters.
   * Data Exchange:
     + Defines course staff and assigns roles.
     + Defines exam parameters, including type, length, date, direction and stylistic elements.
     + Generate exam documents.
3. Course Staff (Instructors, TAs, Graders):
   * Characteristics:
     + Involved in the creation and validation of questions, stems, keys, distractors, and solutions.
     + Performs various activities based on assigned roles.
   * Data Exchange:
     + Works on existing questions, validates, edits, and adds new content.

## Use cases :

1. Manage Courses:
   * Actor: System Administrator and Course Administrator
   * Description: Managing course, this includes tasks such as creating new courses, updating course information, and removing courses when necessary.
2. Manage Exams:
   * Actor: Course Administrator
   * Description: Define and manage exams. This involves specifying exam type, length, direction, date etc.
3. Manage Course Staff:
   * Actor: Course Administrator
   * Description: Define the course staff roles, including instructors, TAs, and graders. They assign roles to individuals, track their activities, and manage any changes in the course staff.
4. Edit and Validate Questions:
   * Actor: Course Staff (Course administrator, Instructors, TAs, Graders)
   * Description: Course staff members edit, validate, and manage questions within the system. They can add, delete, or modify question elements, such as stems, keys, distractors, and solutions.
5. Generate Exam Documents:
   * Actor: Course Administrator
   * Description: Generate exam documents based on the defined parameters. This includes creating different versions of exams.
6. Inspect Changes by Course Staff:
   * Actor: Course Administrator
   * Description: Course administrators review and inspect changes made by course staff members. They can accept, reject, modify, or conditionally validate these changes based on the quality and relevance of the modifications.
7. Search and Edit Questions:
   * Actor: Course Staff (Course Administrator, Instructors, TAs, Graders)
   * Description: Course staff members can search for specific meta-questions and edit them. The system allows users to locate questions based on keywords and strings, facilitating efficient editing and validation processes.
8. View Dashboard:
   * Actor: All Users
   * Description: Users view a personalized dashboard based on their roles in the current/given course. The dashboard provides an overview of relevant information and tasks for the user.
9. Generate Solution for Exams:
   * Actor: Course Administrator
   * Description: Course administrators generate solved exams for distribution to students after an exam is completed. This document includes keys and explanations.

## Special Usage considerations:

The system should be tangible and attractive, due to the high need of user cooperation for the system to be beneficial.

# Chapter 3– Functional Requirements

## Workflow Management System

MH = Must Have

NTH = Nice To Have

| **No** | **Description** | **Priority** | **Risk** |
| --- | --- | --- | --- |
| 1 | **System Administrator** | - | - |
| 1.1 | The system shall support the System Administrator in performing migration, installation, updating, and overall management of the WMS system. | MTH | Medium |
| 1.2 | The system shall provide the System Administrator with functionality to install the WMS on departmental/university systems. | MTH | Medium |
| 1.3 | The system shall implement the ability for the System Administrator to clone the WMS for duplication or testing purposes. | MTH | Low |
| 1.4 | The system shall enable the System Administrator to update the WMS with new features or patches. | MTH | Medium |
| 1.5 | The system shall support the System Administrator in the creation of new course entries within the WMS. | MTH | Low |
| 1.6 | The system shall provide functionality for the System Administrator to back up and remove course entries as needed. | MTH | Low |
| 1.7 | The system shall allow the System Administrator to assign initial roles, particularly the administrator role, for each course. | MTH | Low |
| 1.8 | The system shall provide functionality for the System Administrator to remove and change roles as needed in course staff. | MTH | Low |
| 2 | **Course Administrator** | - | - |
| 2.1 | The system shall enable the Course Administrator to define course staff. | MTH | Low |
| 2.2 | The system shall allow the Course Administrator to assign roles to course staff. | MTH | Low |
| 2.3 | The system shall enable the Course Administrator to define exams. | MTH | Low |
| 2.4 | The system shall allow the Course Administrator to define the exam type (test, quiz). | MTH | Low |
| 2.5 | The system shall enable the Course Administrator to define the exam direction (RTL, LTR). | MTH | Low |
| 2.6 | The system shall allow the Course Administrator to define the exam length. | MTH | Low |
| 2.7 | The system shall enable the Course Administrator to define the exam date. | MTH | Low |
| 2.8 | The system shall allow the Course Administrator to define stylistic elements (fonts & sizes). | MTH | Low |
| 2.9 | The system shall enable the Course Administrator to define frontal matter for the exam, test. | MTH | Low |
| 2.10 | The system shall allow the Course Administrator to define basic headers. | MTH | Low |
| 2.11 | The system shall enable the Course Administrator to define instructions for an exam. | MTH | Low |
| 2.12 | The system shall allow the Course Administrator to define basic layout (number of columns, number of items (4 or 5)). | MTH | Low |
| 2.13 | The system shall enable the Course Administrator to define the number of versions for the exam. | MTH | Low |
| 2.14 | The system shall allow the Course Administrator to select questions. | MTH | Low |
| 2.15 | The system shall enable the Course Administrator to select appendices. | MTH | Low |
| 2.16 | The system shall allow the Course Administrator to generate exam documents. | MTH | Medium |
| 2.17 | The system shall enable the Course Administrator to generate exam versions. | MTH | Low |
| 2.18 | The system shall enable the Course Administrator to generate special versions for reading-impaired (A4, but landscape, and 41% larger). | MTH | Low |
| 2.19 | The system shall enable the Course Administrator to generate exam keys (As PDF, As CSV). | MTH | Low |
| 2.20 | The system shall enable the Course Administrator to generate exam catalog documents. | MTH | Low |
| 2.21 | The system shall enable the Course Administrator to generate solved exams (for handing out to students post-exam). | MTH | Low |
| 2.22 | The system shall enable the Course Administrator to inspect changes made by course staff and take actions such as reject, accept, modify, or condition upon further validation. | MTH | Low |
| 3 | **Course Staff** | - | - |
| 3.1 | The system shall enable Course Staff (Instructors, TAs, Graders) to add, delete, edit, and validate questions. | MTH | Low |
| 3.2 | The system shall enable Course Staff to add, delete, edit, and validate stems. | MTH | Low |
| 3.3 | The system shall enable Course Staff to add, delete, edit, and validate keys. | MTH | Low |
| 3.4 | The system shall enable Course Staff to add, delete, edit, and validate distractors. | MTH | Low |
| 3.5 | The system shall enable Course Staff to add, delete, edit, and validate solutions. | MTH | Low |
| 4 | **WMS Activities** | - | - |
| 4.1 | The system shall allow users to log into the system using an ID and password. | MTH | Low |
| 4.2 | Users should be able to search for a specific meta-question and editing it. | MTH | Medium |
| 4.3 | The system shall allow users to ask to edit the current exam, in which case, the current exam settings give focus. | MTH | Low |
| 4.4 | Users can still write new questions, but those are now suggested to the Course Administrator rather than added into the exam. | MTH | Medium |
| 4.5 | Users can work on existing questions based on their role. | MTH | Low |
| 4.6 | The system shall allow users to ask for a task, the system should offer tasks based on categories and sorted by urgency. | MTH | High |
| 4.8 | If a user active in more than one course, users should be able to select the course. | MTH | Low |

# Chapter 4 – Non-functional requirements

## Implementation constraints

1. Performance (Speed, Capacity, Throughput, etc.)
   * 1. The system should be able to support multiple users (of the same or different types) using the system at the same time.
2. Reliability & Stability
   * 1. the distractors and answers as well as past exams will be persisted in a database and will support data recovery.
     2. In case of errors in internet connection, crash or hardware failure, the system will roll back all related updates until reaching the last stable version.
3. Safety & Security
   * 1. The system will save any sensitive data such as password only encrypted.
     2. The system will not allow any access to its sensitive data to users without permission.
4. Portability
   * 1. The system is web based and can be accessed only while connected to the university’s network.
     2. The system should be accessible from different browsers.
     3. The current requirement is to support Hebrew language only.
5. Usability
   * 1. The system’s users do not have any special expertise in computers or programming; therefore, the system’s interface should be as simple and clear as possible.
6. Availability
   * 1. Unless the system is undergoing updates, the system should be available 24/7

## Platform constraints

SE Project constraints

1. The system will be an interactive system, the inputs come naturally from the end users.
2. The system will need access to student exam answers to analyze questions/answers/distractors quality.
3. The system shall use samples of actual data in case of simulation.

## Special restrictions & limitations

1. The system will be installed remotely on the university’s servers.
2. The system will rely on the cooperation of end users, without it no data will flow in the system.

# Chapter 5 – Risk assessment & Plan for the proof of concept

# Risk 1: Resistance from users to adapt to the new system.

## Users may be resistant to change, especially if they are accustomed to existing systems. This resistance could result in unuseful system, that’s due to the need of users to complete tasks.

## Mitigation:

## Conduct training sessions to familiarize users with the new system.

## Communicate the benefits of the new system, emphasizing efficiency gains and improved functionalities.

## Risk 2: Vulnerability to unauthorized access and leaking of sensitive information:

## Given the nature of the system dealing with exams and sensitive data, there is a risk of unauthorized access leading to data leakage of sensitive information.

## Mitigation:

## Implement security measures, including encryption and secure authentication protocols.

Allow connection to the system only via the University network.

## Plan for the proof of concept

### Research:

* **Examination of Web-Based Workflow Tools:**
  + Investigate popular web-based workflow management tools.
  + Assess their capabilities for role-based dashboards and task distribution.
* **Algorithmic Task Distribution:**
  + Research and develop a preliminary algorithm for task distribution.

### Preliminary:

* **Selection of Web-Based Framework:**
  + Based on research, choose a web-based framework suitable for developing the workflow management system.
* **Basic UI Development:**
  + Implement basic UI features, such as role-based dashboards and content organization.
  + Integrate a simple manual task assignment mechanism.

### Proof of Concept:

* **System Implementation:**
  + Develop a basic version of the system with the following components:
    1. **Task Distribution:**
       - Implement a simplified version of the algorithm for task distribution.
    2. **LaTeX-Based Exam Creation:**
       - Develop a module for flexible LaTeX-based exam, key, and solution creation.
    3. **Database Integration:**
       - Establish basic database integration for handling workflow and content data.

# Appendices