


Exam Management System - PROPOSAL

Roi Tiefenbrunn, Ofek Nov, Mor Abu, Idan Aharoni

Software Engineering
Faculty of Engineering
Ben Gurion University of the Negev

Exam Management System

- Exam Management System
- מערכת ניהול מבחנים

Student Name	ID	Email	Phone	Signature
Roi Tiefenbrunn	209829340	roitief@post.bgu.ac.il	054-7168459	
Ofek Nov	206618175	Ofeknov@gmail.com	052-738-0233	
Mor Abo	209239797	Morabou1@gmail.com	050-334-5338	
Idan Aharoni	206430050	idan.aharoni2@gmail.com	054-973-7462	

Customer

- Customer Name – Dr. Mayer Goldberg
- Customer Representative Person – Dr. Mayer Goldberg
 - Role – Lecturer, Head of Teaching Committee
 - Email – gmayer@little-lisper.org
 - Phone Number - 055-660-3392

Project's Motivation and Purpose

- General Problem:
 - 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, which is faster to grade, and easier to analyze.
 - The creation of challenging multiple-choice tests in the sciences is a non-trivial task.
 - The creation of one-time-use questions is costly and unsustainable.
- Specific Problem:
 - Maintenance of a large and growing pool of questions is proving to be difficult.
 - Current implemented system is local, inaccessible and hard to learn.
 - Division of labor between TAs and test examiners is done by hand and is hard to track.
 - The possibility of tracking quality of questions & answers is crucially needed.
- What is the solution?
 - A web-based System for managing work, subjects, questions.
 - Intended to replace current offline system.

What already exists?

- The client, Mayer Goldberg, has developed a theory of meta-questions and their translation into multiple-choice questions.
- A scheme-based system was created to synthesize & generate multiple-choice and implement said theory.
- The existing system has now been in use for 6-7 years in a large number of courses.
- Current system's low accessibility is too hard for people without deep knowledge and understanding to use comfortably.

Customer's 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 over work velocity and output (= 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.

Project Description

- Frontend+Backend - developed in JavaScript as it is widely user, stable, with frameworks that allow easy implementation of features.
- Simple, intuitive and smooth interface - better UX is key for adoption by new users.
- PostgreSQL based DB – stable and strong relational DB.
- Features – Detailed in org file.

User Interface

- Role based workflow management interface:
 - View and filter content by subjects/keywords/classes.
 - Creation and management of stems, meta-questions, questions, appendices.
 - Creation and management of possible solutions/distractors per question.
- Create new Exam

Project's Risk Assessment

Challenge	How to overcome? (Bonus)
We never Developed in JS	Learn JS

Project's Environments and Languages

- The Project UI and BE will be developed in JavaScript

Evaluation and Testing Plan

- Accessibility Testing
- Unit Testing
- Sanity Testing
- Acceptance Testing
- End to End Testing
- Security Testing

Project's Steps and Timeline

User Interface:

- Web-based
- Login page
- Role dependent dashboard + actions

Workflow Management:

- Ask for work
- Watch pending tasks
- Perform specific task
- Maintain history (continue where left off)
- Manual urgency control
- Automatic/Algorithmic urgency control

Exam Management:

- Meta-question management
- Appendices management (appendices + their meta questions)
- Stem management (stems + their appendices + their meta questions)
- Selection of meta-questions by keywords/substring
- Question production
- Exam production
- Catalog production
- Version Control

