Technical University of Moldova

Chair Computer Science

Report

On Designing Informational Systems

Laboratory Work Nr. 1

Done by: st. gr. FAF-091	Melniciuc Constanin Bibilici Victor
Checked by:	Zarea Ivan

Chişinău – 2012

Contents

1) Creating the system vision

a) Business Context

Quizz tool Kit is a program which will generate quizzes based on a given subject. It will be used by teachers to test they're students, also will be used by students to prepare for their exams. It must be set on a Windows Platform System, this of course will be something new for teachers and students, and they will have to acknowledge with the interface and the possibilities of the program.

b) Bussines Opportunities

This program have a great future, it can be used by anybody, for creating tests on any subject you have. All you need is the material to study, some plain text files which contains necessary information. This is a great idea for making the work easier for teachers and educating a necessary work flow for teachers, this means that any paragraph will have to be specially tagged so that the program would understand that based on this it should a create a question. This is however may seem complicated but this is really important for modern world, because any report and material have to be well organized. This will organize as well the people and they will find easily what they need, taking into consideration that they organized really well everything else.

c) Problem Description

Problem

a) Test creation consumes lot of time, and working power.
b) Test may not correspond to the material students studied, and it may be hard to find necessary material in a large file.
c) Students don't know what to study for the test.
d) Security issues may appear when checking the reliability of a concrete test.

Consequences

a) Teachers may not have enough time to do his own needs
b) Teachers may be not prepared in time to the test, so test may be not ready in time.

c) Students may prepare really well for their future exams, knowing with a high rate what questions they will have at their tests.
d) Taking into consideration that program will generate a unique Quick Response Code (QR Code) tests cannot be replaced, this way cheating is excluded.

Successful solutions

a) Program automatically generates and creates tests, this way minimizing the time consumed by teachers on preparations.
b) Students know what they should prepare for the future tests.
c) Tests are always ready in time.
d) Tests are really secured and can’t be prepared for being switched.

Task of the system

a) Create tests based on material it has as an input.
b) Generate QR code for secure the digital assignment.
c) Recognize the QR code.
d) Recognizing correct answers.
e) Create a web-based interface

2) Identifying the stakeholders

Table 1 **–** Stakeholders and their role in the project.

Stakeholder	**Role**	**Description**
Teachers	Creates the tests	Introduces necessary material as in input.
Students	Study the material	
Developers (Melniciuc Constantin / Bibilici Victor)	Develop and sustain the program	- Monitoring the workflow of the program - Solving the problems appeared. - Creating the system
QR Code generator	Secure every single test	Generates unique QR Code
QR Code Database	Contains all QR codes	It’s a database where all generated QR codes are kept.
Question generator	Generates Tests based on inputted material	Taking the input material and based on the specially tagged themes generates questions.
Input file	Material database	Contains the necessary text based on which tests are created.
User Interface	Making the work easier	Has an friendly user interface
PDF Generator	-	Generates .pdf files with questions.

3) Documenting the functional requirements of the system

Functional requirements:

Quiz tool Kit Software

- The operating system required for use is MS Windows 2000, MS Windows XP, or MS Vista.
- Require any browser to be installed.
- The quiz toolkit is properly loaded in browser.
- Supporting the pdf format for scan.
- Use of QR code for later recognition.

Quiz tool kit Hardware

- The application can use local or networked printers.
- Will not open from outside without using browser.
- Opens correctly from inside browser.

System interfaces

- The application provides a general introduction to the Quiz tool kit. It also provides links to the validation documents.
- Is properly formatted for printing.

Regulatory requirements

- Quiz tool kit have the ability to generate accurate and complete copies of records in both readable and electronic form.
- Users are able to select search queries.
- Users are able to customize the criteria used to view data from the search.
- Users are able to export data to pdf format.

Program time-outs

- The application provides the Quiz tool kit with automatically time-out after 30 minutes of non-activity.
- Limiting access of authorized individuals.

Other functionalites

- The oriented name of the signer.
- The date and time when signature was executed.
- Application of an electronic signature requires use of user id and password.
- Data secured with an electronic signature cannot be edited or deleted unless the electronic signature is removed.
- Include the printed name of the user applying the electronic signature.
- Include the Date/Tune when electronic signature was applied.
- Electronic signature is human readable.
- There are no specific requirements on system performance.

It also provides backup/recovery. Quiz tool kit will be backed up daily on the LAN.

4) Non-functional Requirements

Product family-oriented attributes

- The program should work on several platforms.
- The program can be modify, it should be possible the addition of new functionalities.
- It also can be reusable. We can reuse components, code, designs, and even requirements in other systems.

Process-oriented attributes

- Maintainability: changes to functionalities, repairs.
- Readability: of code, documents.
- Testability: ease of testing and error reporting.
- Understandability: of design, architecture, code.
- Integrability: ability to integrate components
- Complexity: degree of dependency and interaction between components

Reliabilty measures

- The precision of calculations shall be at least 1/106
- The system defect rate shall be less than 1 failure per 1000 hours of operation.
- No more than 1 per 1000000 transactions shall result in a failure requiring a system restart.

Availability measures

- The system shall meet or exceed 99.99% uptime.
- The system shall not be unavailable more than 1 hour per 1000 hours of operation.
- Less than 20 seconds shall be needed to restart the system after a failure 95% of the time.

Security measures

- Examples of requirements
- The application shall identify all of its client applications before allowing them to use its capabilities.
- The application shall ensure that the name of the employee in the official human resource and payroll databases exactly matches the name printed on the employee's social security card.
- At least 99% of intrusions shall be detected within 10 seconds.

Testabilty measures

- The delivered system shall include unit tests that ensure 100% branch coverage.
- Development must use regression tests allowing for full retesting in 12 hours.

Portabilty measures

- No more than 5% of the system implementation shall be specific to the operating system.
- The meantime needed to replace the current Relational Database
- System with another Relational Database System shall not exceed 2 hours. No data loss should ensue.

5) Architectural sketch:

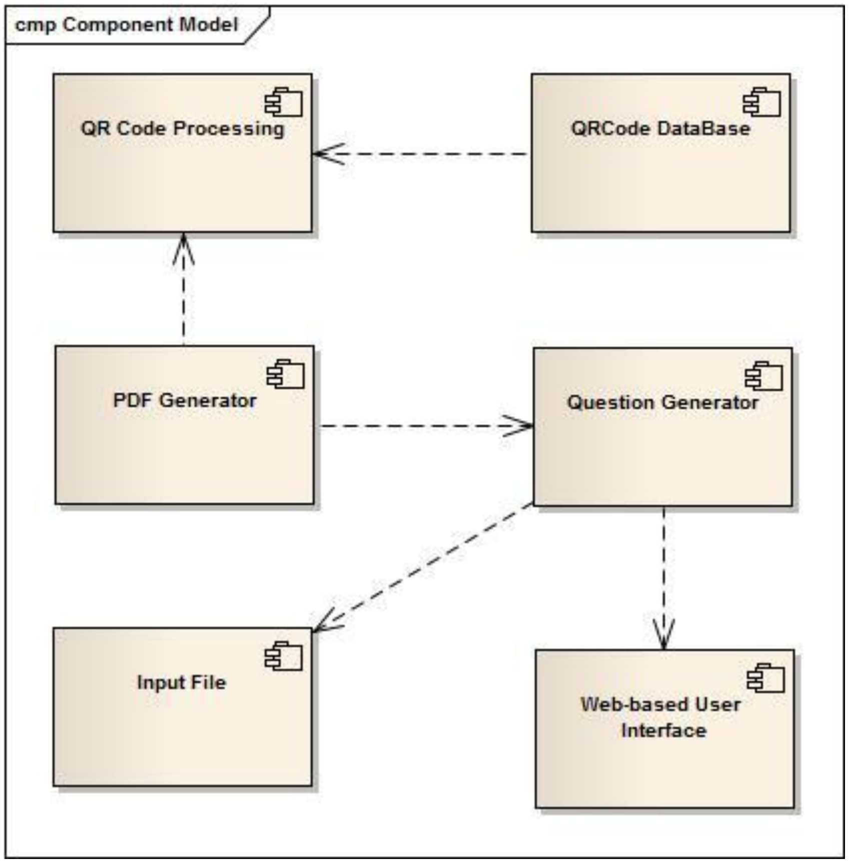

Figure 2 – Structural Model of the application;

To understand the workflow of the program, I will explain everything in the above shown diagram.

We need the generated .PDF file which contains the question for a quiz. As a consequence we need a “Question Generator”. On its order Question Generator need the “Input File” in which the information is well-organized.

Also we need the QR code processing part which generates and then recognize the QR code, used instead of a signature. This I a great security issue. QRCode is kept in the DataBase for later recognition if necessary.

The User Interface – is an independent module which specifies where to find the input file for Question Generator.

6) Functional Blocks

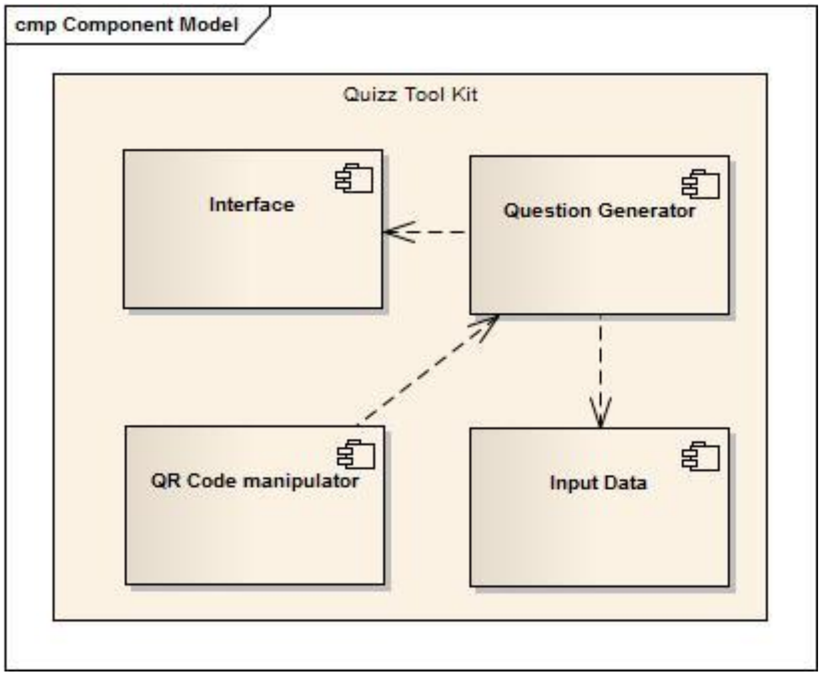

Figure -1 Functional Blocks of the application

Question Generator depends on the Commands introduced by user from the interface. Also it depends on the Input Data.

Next the QR Code manipulator depends on Question Generator, so if there will be some Quizzes Generated there will be the necessity in generating QR Code, and introduce them into the Data Base.

I’ve grouped up Generator of Questions and Generator of PDF files because it’s really connected to each other. If there are no question to generate no PDF file will be generated.

QR Code DataBase and QR Code Processing are grouped as well, because if there are no generated QR Codes, there is nothing to introduce in the DataBase.

7) Selecting architecture

I see that as a Service Oriented Architecture. Because every building block can play one or both roles:

a) *Service provider* – This application has a Web-Based interface, it contains all the needed information. All independent blocks in this application can be used as provider either as consumers.

b) *Service consumer* – The application's consumer is the teachers and students. Basing on the interface it will be used either to recheck the answers either to create quizzes for students. This is a powerful tool for any teacher, as well as for students.

Main reason why I've chosen this architecture type is because it reflects mostly the way the system works, and it will be the easiest way (in my opinion to be shown).

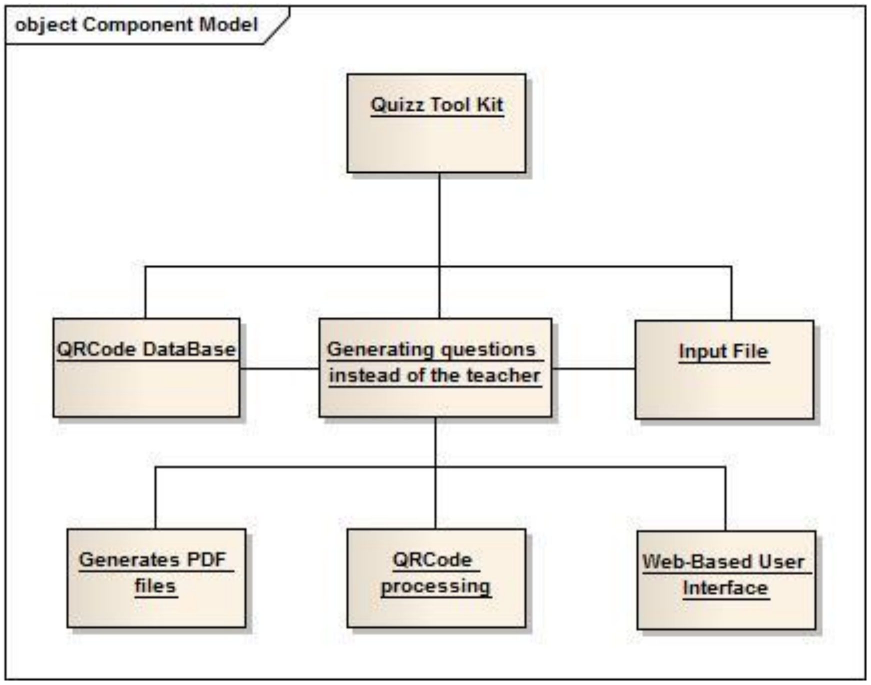

Figure 3 – Service Oriented Architecture represented graphicaly.

8) Architectural principles

Loose coupling – In my application there are 2 independent parts QR Code processing and Question Generator (it uses input file, the Web-Based User Interface and PDF Generator). Paradigm achieved.

High cohesion – Our application contains 4 independent modules:

- QR Code processing & QR Code Database
- PDF Generator
- Input file & Question Generator
- Web-Based User Interface

I can say that all these modules are following the high cohesion principle, because they are pretty much independent and are easy to test.

Design for change – In this application can be easily added new modules for developing the idea.

You can easily redesign the interface, because it's Web-Based, and add desired buttons, after which you can create a new module that will do some other functionality. This is really reliable.

Separation of concerns – Every single module in my application has its own responsibilities. All the subclasses included have also some responsibilities. Therefore I can assume that this principle is achieved.

Information hiding – In my application the processing modules like quiz generator and QR code processing are compiled, therefore it will be hard to reverse engineer without the source code. This principle is also achieved.

Liskov Substitution – This principle is used by giving different input files, with different information.

Modularity – The application contains blocks which are having a common communication interface, and they will interact well between them.

Convention over configuration – The application uses an input file, once declared that file may contain anything and still it will work. Once the decisions are predefined the application runs anyway. This way this paradigm is also satisfied.

9) An architectural sketch (The Final Version)

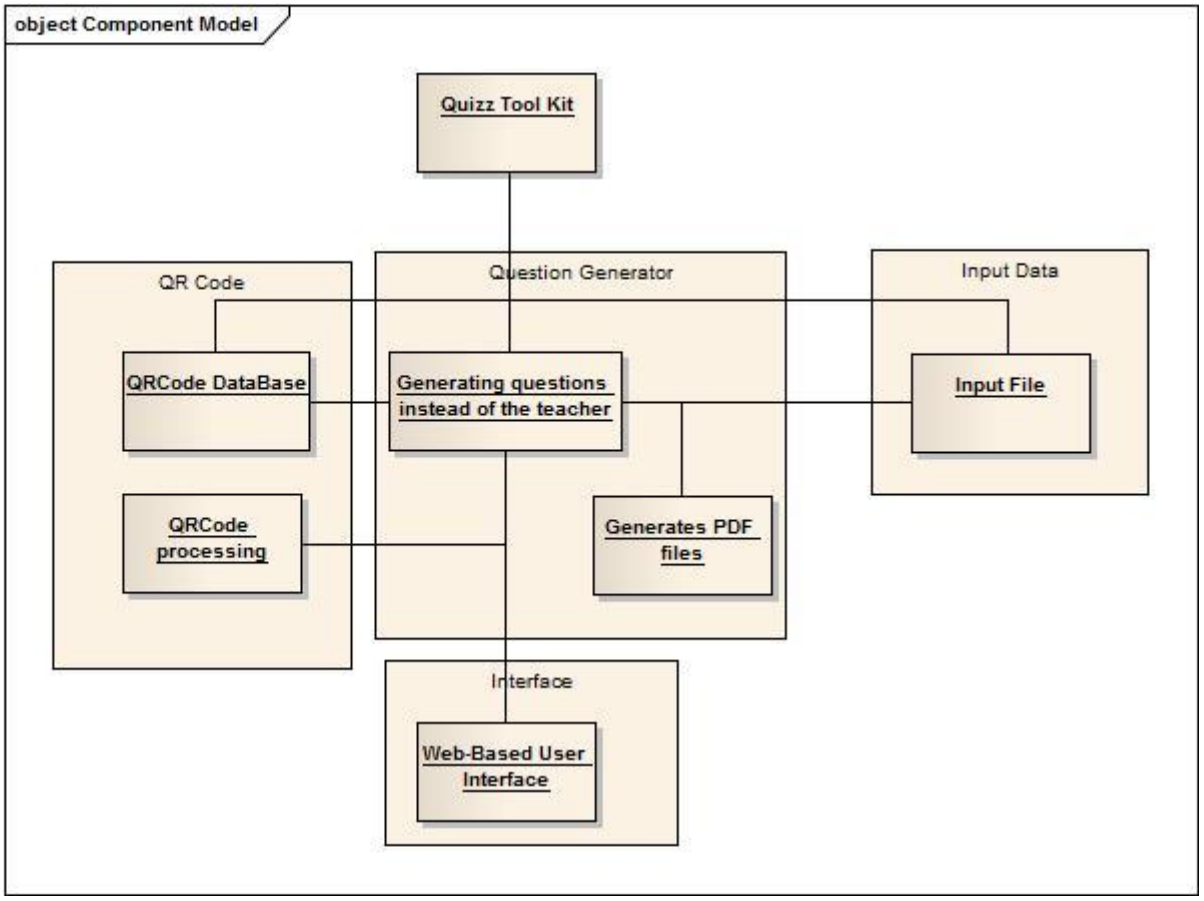

Figure 4 – Classified Version of Service Oriented Architecture represented in graphical manner.