CS102	**Spring 2018/19**	Project Group	3B
Instructor:	**David Davenport**		
Assistant:	**Mehmet Başaran**		

~ Nito ~

Internationals

Ziya Mukhtarov, Javid Baghirov, Abdul Mannn, Alper Sari, Mokhlaroyim Raupova, Adeem Adil Khatri

Criteria	TA/Grader	Instructor
Presentation		
Overall		

Detailed Design Report

(version 1.0)

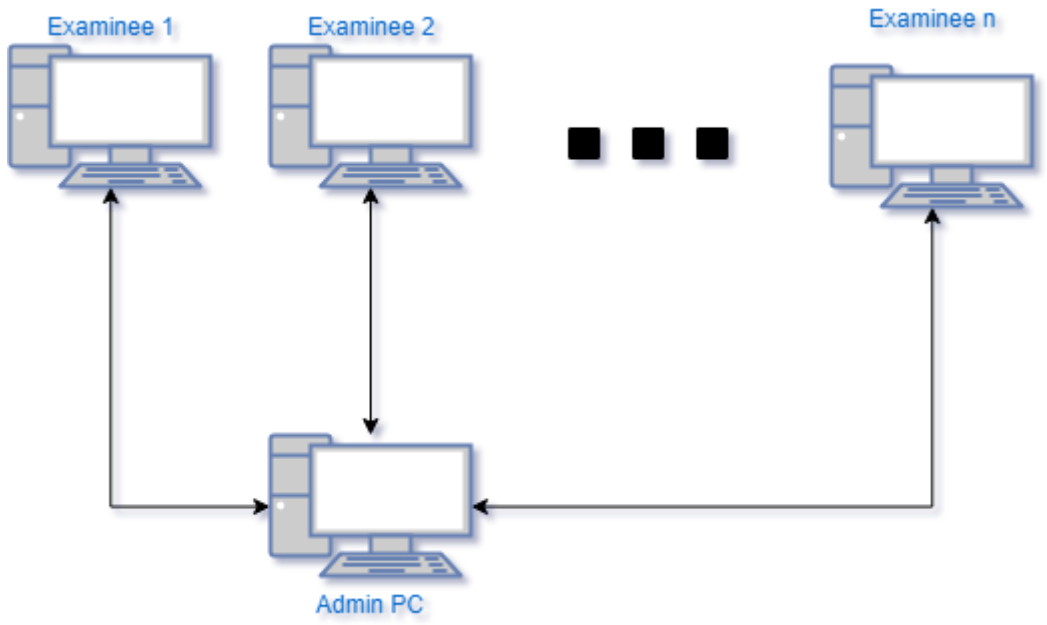
16 April 2019

1. Introduction

Nito – the exam system aims to provide its users with a modern and elegant solution to digitalize the old-fashioned tools (e.g. paper, pencil) used during programming exams. Using almost all features that technology and the internet provide us with, Nito strives to fill this gap as unfortunately there is almost no application which gathers such anti-cheating services into one software.

2. System Overview

2.1 Network



This is the basic network diagram showing the connections between Nito interfaces during an exam. The first version of the program will only support one administrator for each exam. That admin interface will be connected to all examinee interfaces via both UDP and TCP protocols. UDP protocol will be used for live screen view feature, while all other requests will be done using TCP as it is more reliable than UDP.

The first version will only be available for the Windows operating system. The admin PC will function as a server, while examinees will be connected as clients.

All relevant classes will probably implement Java's Serializable interface in order to be sent over the network.

2.2 Storage

Nito will use file-based storage. All of the exam data will be stored in the Admin PC. The relevant data will flow from admin to examinees and kept at a temporary folder until the exam ends. The progress of examinees will be backed up to their memory continuously. Once the exam ends or submit action occurs, all of the solutions will be transferred to admin pc and stored in folders.

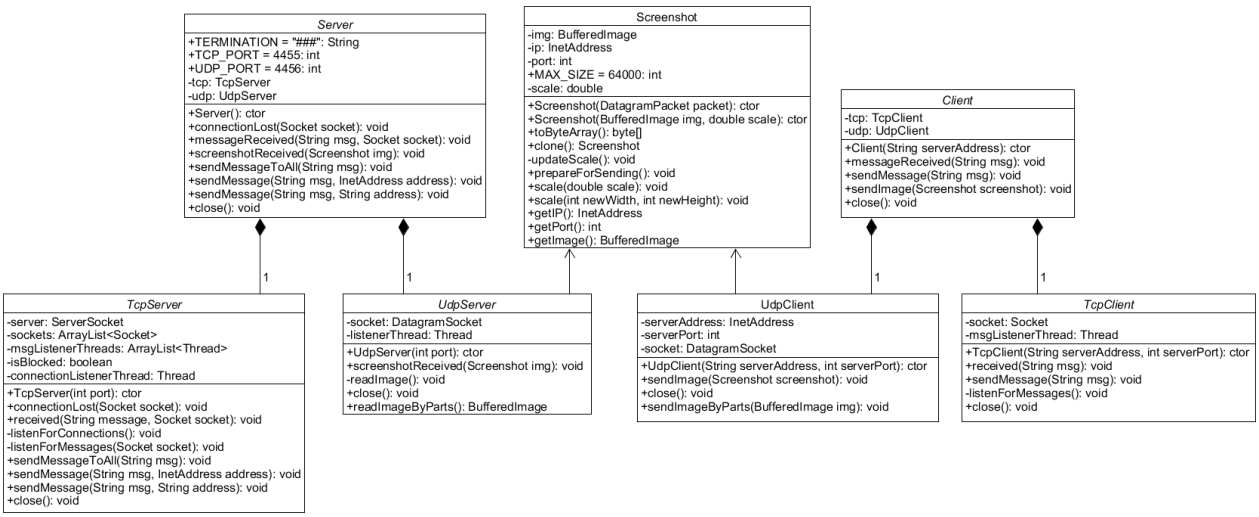
2.3 Graphical User Interface (GUI)

For designing GUI's, JavaFX library will be used. As a part of the process, FXML and CSS languages may also be used. Nito interface will use third-party layout managers for some complex layouts such as MigLayout. For some features, Swing components might be integrated into JavaFX as it lacks some rare components (e.g. JDesktopPane).

3. Core Design Details

Below you can find the first drafts of UML Class diagrams. Please note that these are very incomplete and inaccurate diagrams. They should be seen as outlines of the real UML diagram.

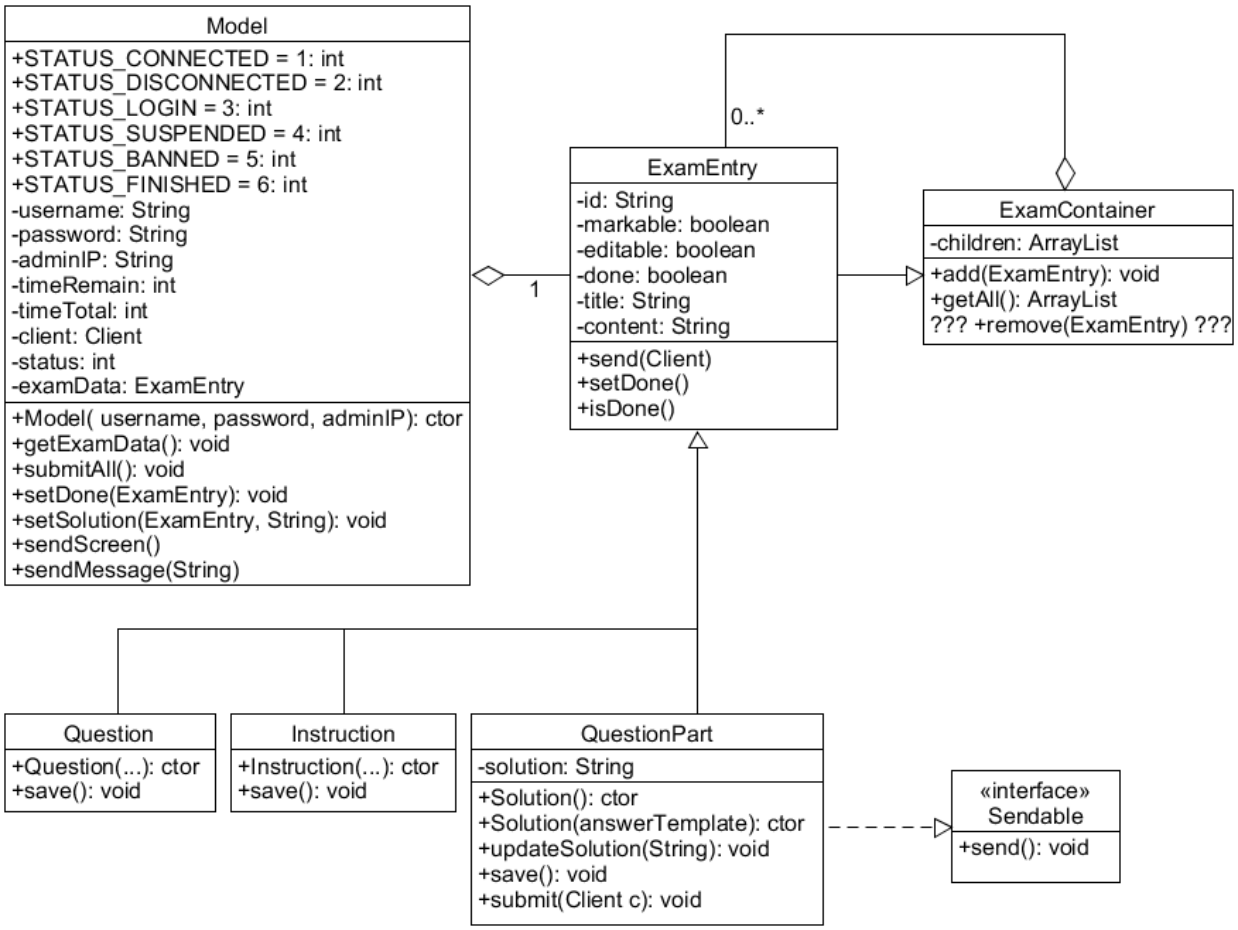
3.1 `network` package



This package provides basic network features such as sending messages or `Screenshot`s. Both `Server` and `Client` classes contain TCP and UDP versions. Admin interface will use `Server` class, while examinee interfaces will use the `Client` class to communicate over the network.

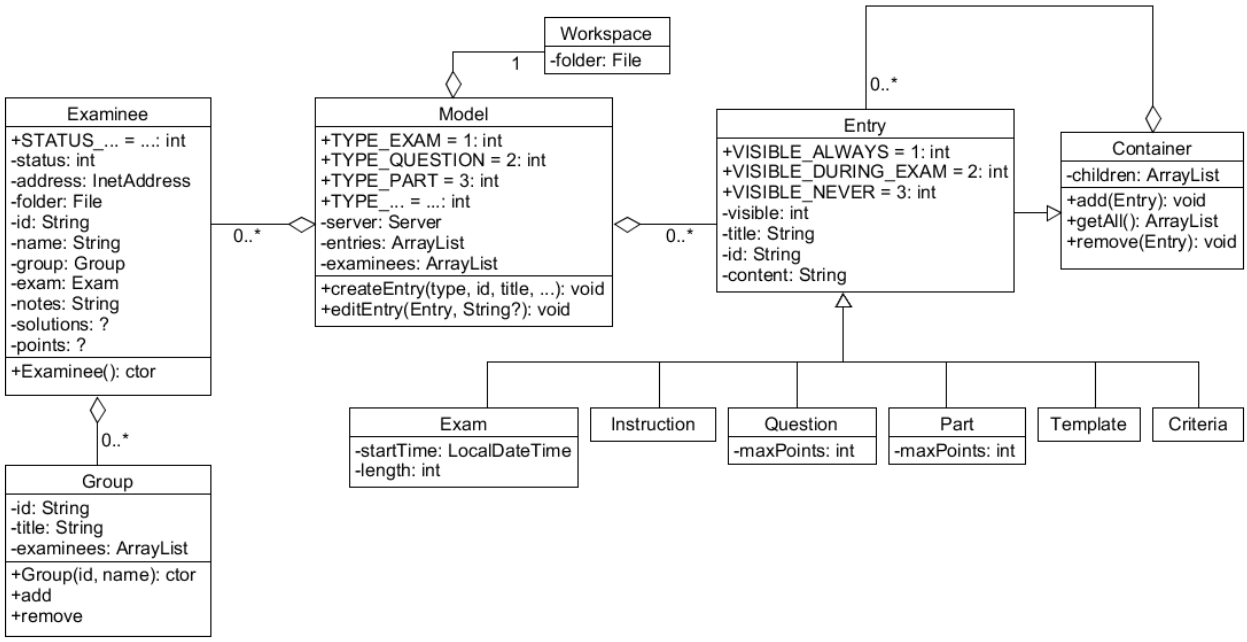
`Screenshot` class is a wrapper class for `BufferedImage` and other relevant data. It also provides functionalities needed for Nito application.

3.2 `examinee` package



This is the classes for examinee interface. `ExamEntry` and `ExamContainer` classes are similar to Swing classes in the sense that it is possible to create complex entries because each entry is a container. Other than that, everything should be self-explanatory, as there is not much information in these diagrams.

3.3 `admin` package



This UML diagram shows the classes for the admin interface. `Entry` and `Container` classes are nearly the same as the `examinee` package. `Examinee` class will provide functionalities for one examinee and `Group` class will be used for grouping them according to some data, e.g. section number.

3.4 External API's

Our application will use a lot of external libraries in order to provide its functionalities. Below you can see the list of libraries we have in mind for now. In the revised version of this document, the list will be complete.

JNativeHook – This library provides global keyboard and mouse listeners. We will use it for blocking keyboard shortcuts in examinee interface for anti-cheating.

ControlsFX – JavaFX library that contains a lot of additional components for modern GUI's. We will use it for breadcrumb, but other interesting components might also be included in the project.

4. Task Assignment

Below are the assigned tasks for each person in our group. After a long discussion, we decided to implement only the core Nito features as we have very little time. Ziya Mukhtarov plans to improve this project during Summer and make it useable for CS101/102 courses.

Javid Baghirov	Examinee interface – GUI
Abdul Mannan	Admin interface – Exam preparation GUI

Mokhlaroyim Raupova	Admin interface – Exam Monitoring and Grading GUI
Alper Sari	Examinee interface – model classes
Adeem Adil Khatri	Admin interface – Entry and Container classes and subclasses
Ziya Mukhtarov	Admin interface – other model parts Examinee interface – anti-cheating services Network package Everything else that is not mentioned in this table and is required to run Nito

5. Summary & Conclusions

This report lists the details of Nito software. It is just the drafted version and it will be revised/improved in the future.