Versions:

Date	Version	Author	Comments

Analysis&Design for Physics Experiment Simulator

Continut:

Avizat:	**Aprobat:**
Company/University Name: Address: Universitatea Tenhica Cluj-Napoca Romania	

1. General Presentation

The project will provide an easy to use interface for simulating physics experiments. Each experiment will be stored in its specific file, such that it will be easier to add/remove/edit experiments.

2. Theoretical Fundamentals

Main theoretical fundamentals will be physics. The project should be able to compute any experiment, from any branch of phisics. So a basic knowladge of physics and what you want to achive through this program is required. It will provide some theoretical background on each experiment, but this might not always be enough.

3. IT Technology

This program will be developed using xtend/java programming language, and using the eclipse RCP framework to export the applicastion as exe, for the sake of portablity. Also Google Injection is used in some cases.

4. List of functionalities

Provides a list of local experiments, Search action for the list, Easy to add experiments and a lot of info about them.

Experiments will be stored locally and a preference page will be available to allow the user to set the location of the experiment folder.

The initial functionalities are the following:

- View a list of local experiments
- Select and view details of experiments from the list
- Run a selected experiment
- Debug the experiment using a console
- Create or modify existing experiments

5. General Presentation of Actors and Access Rights

Simple application that can be used by anyone. You do not need special access rights to add/edit/remove experiments, as all experiments are local. A public repository containing some basic experiments will be available, where you will need access from the developer to post new experiments.

6. Use-Case Diagrams

7. System Architecture

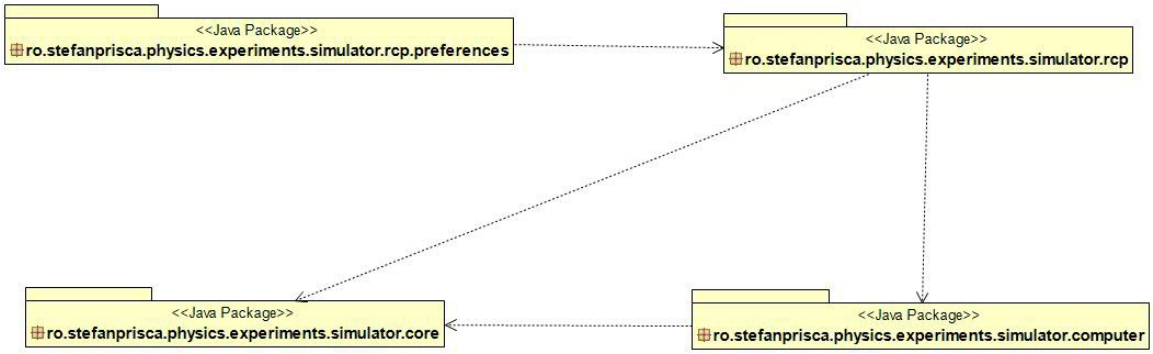
The project will be split in the UI part and the "worker" part. As it is developed using the eclipse RCP framework, the UI part will be made using eclipse extension points (Views, Perspectives, etc).

Therfore there will be 2 main packages :

ro.stefanprisca.physics.experiments.simulator containing the working classes

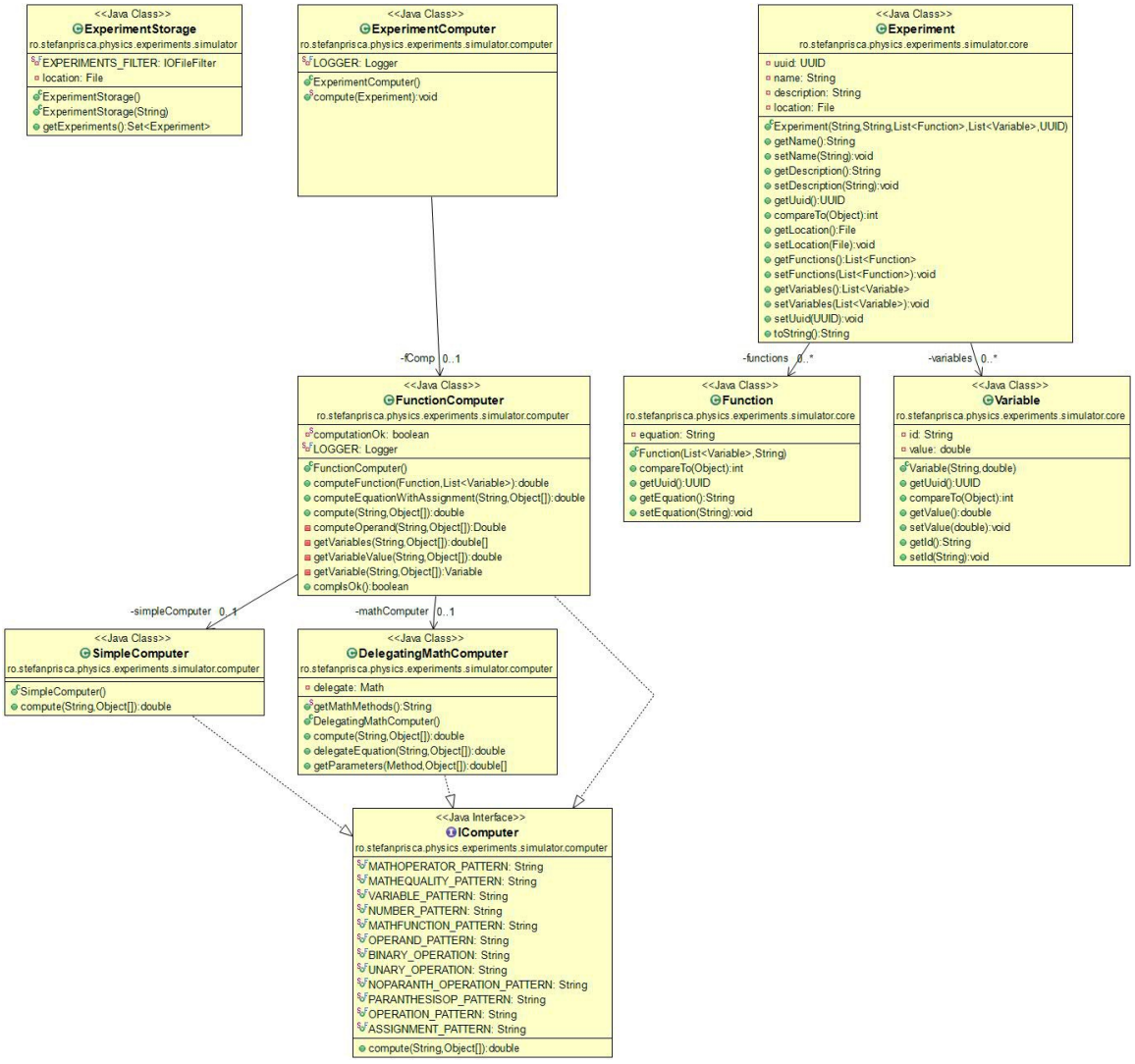
ro.stefanprisca.physics.experiments.simulator.rcp containing the UI parts

The tests will go in r.s.p.e.s.tests

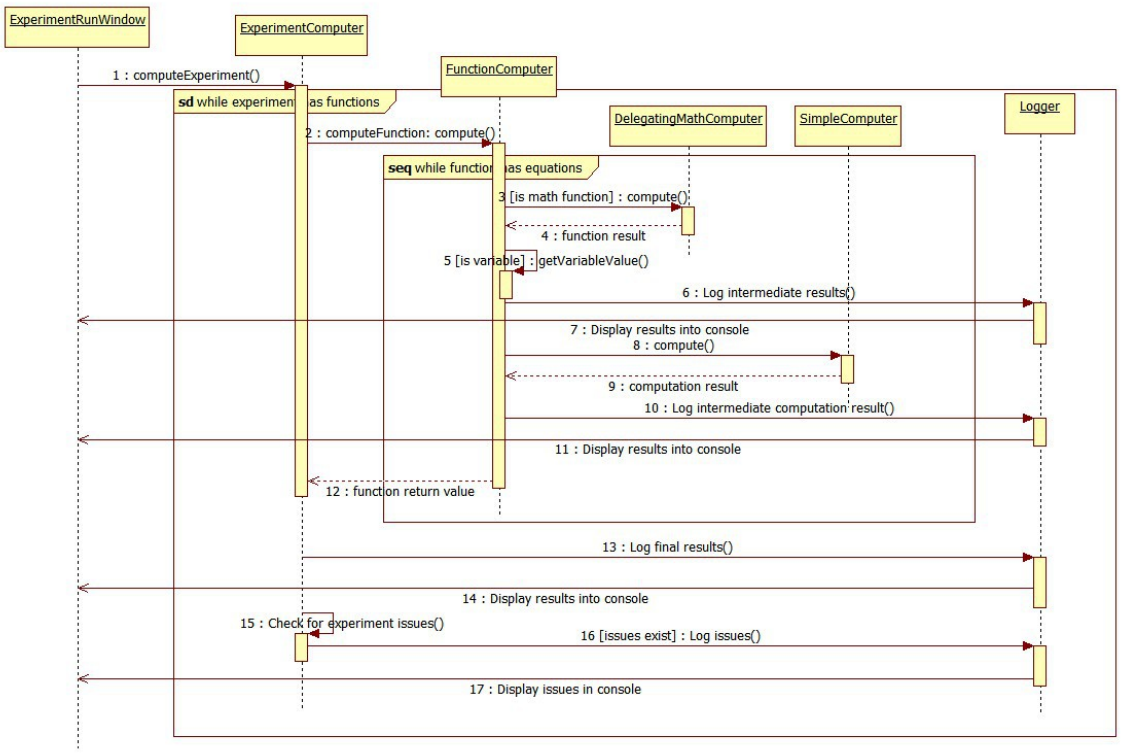


8. **Design** (detailed presentation of UML diagrams)

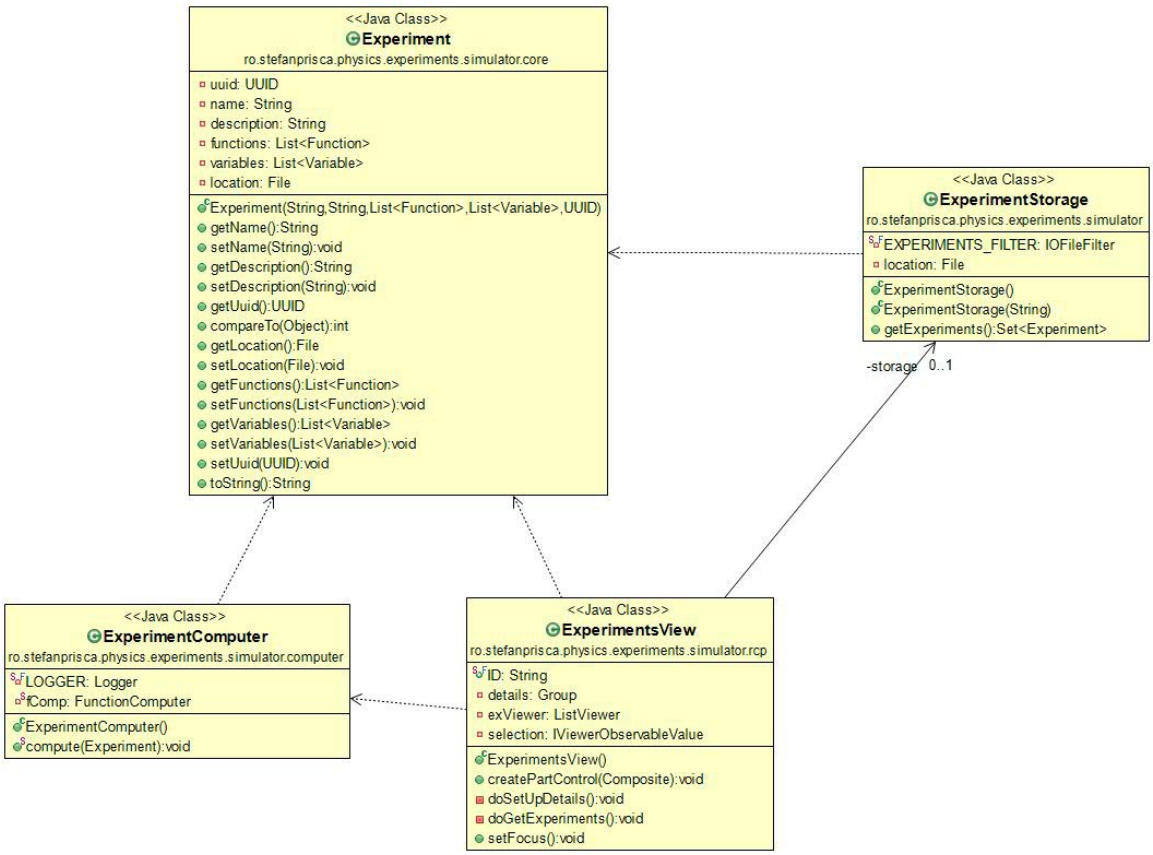
Here is an UML class diagram that shows the relations between the experiment computers :



The following sequence diagram is showing how experiments get computed:



This next diagram will present the core UI structure, and what classes it uses to show the experiments:



9. Operating Mode/ Interactivity Features

First, the user will select an experiment from the list. This brings up the following:

1. Some details about the experiment will pop up in the main frame
2. the user will be able to run that experiment. This means that:
    1. The user will be required to input the necessary parameters for the selected experiment
    2. The experiment will do the computations required and post the results in a console. This will be done as interactively as possible using a logger to display step-by-step what happens in the experiment.
    3. If a computation went wrong (e.g. division by zero) an message will be displayed in the console suggesting to look for issues in the log.
    4. The user will be able to analyze intermediate results and debug the experiment if something went wrong.

Besides running experiments, a wizard that will help you create your own will be available. So not only you can test existing experiments, but you can add your own contribution to science! Cool, eh?

10. Portability

Given the fact that it is written in java and the project is exported as an exe file using the eclipse RCP development framework, I'd say portability is strong with this one! :D

11. Competing Software

Phet: http://phet.colorado.edu/en/simulations/category/physics .

12. Glossary &Bibliography

- Eclipse RCP development: http://wiki.eclipse.org/index.php/Rich_Client_Platform
- Xtend developing language: http://www.eclipse.org/xtend/