Software Engineering

Design Phase

**Trello link:** [**https://trello.com/b/Z30sotw5/design-phase**](https://trello.com/b/Z30sotw5/design-phase)

Group members:

* Michail Krugliakov - Student № 100136484
* Kiril Chomaniuk - Student № 100111543
* Ioakim Ioakim - Student № 100134028
* Moaz Hafez - Student № 100111519

Table of Contents

[Glossary 4](#_Toc483919602)

[Similar System Analysis 5](#_Toc483919603)

[GoConqr 5](#_Toc483919604)

[Get Revising 7](#_Toc483919605)

[Use cases 10](#_Toc483919606)

[Relationships between use cases 11](#_Toc483919607)

[Use cases specifications 12](#_Toc483919608)

[Create Study Profile 12](#_Toc483919609)

[Define Tasks 13](#_Toc483919610)

[Define Activities 14](#_Toc483919611)

[Define Milestones 15](#_Toc483919612)

[Edit Tasks 17](#_Toc483919613)

[Edit Activities 18](#_Toc483919614)

[Edit Milestones 19](#_Toc483919615)

[Save Study Profile 20](#_Toc483919616)

[Edit Study Profile 21](#_Toc483919617)

[Capture Activity/Task Time 22](#_Toc483919618)

[Inspect Gantt Chart 23](#_Toc483919619)

[Open Dashboard 23](#_Toc483919620)

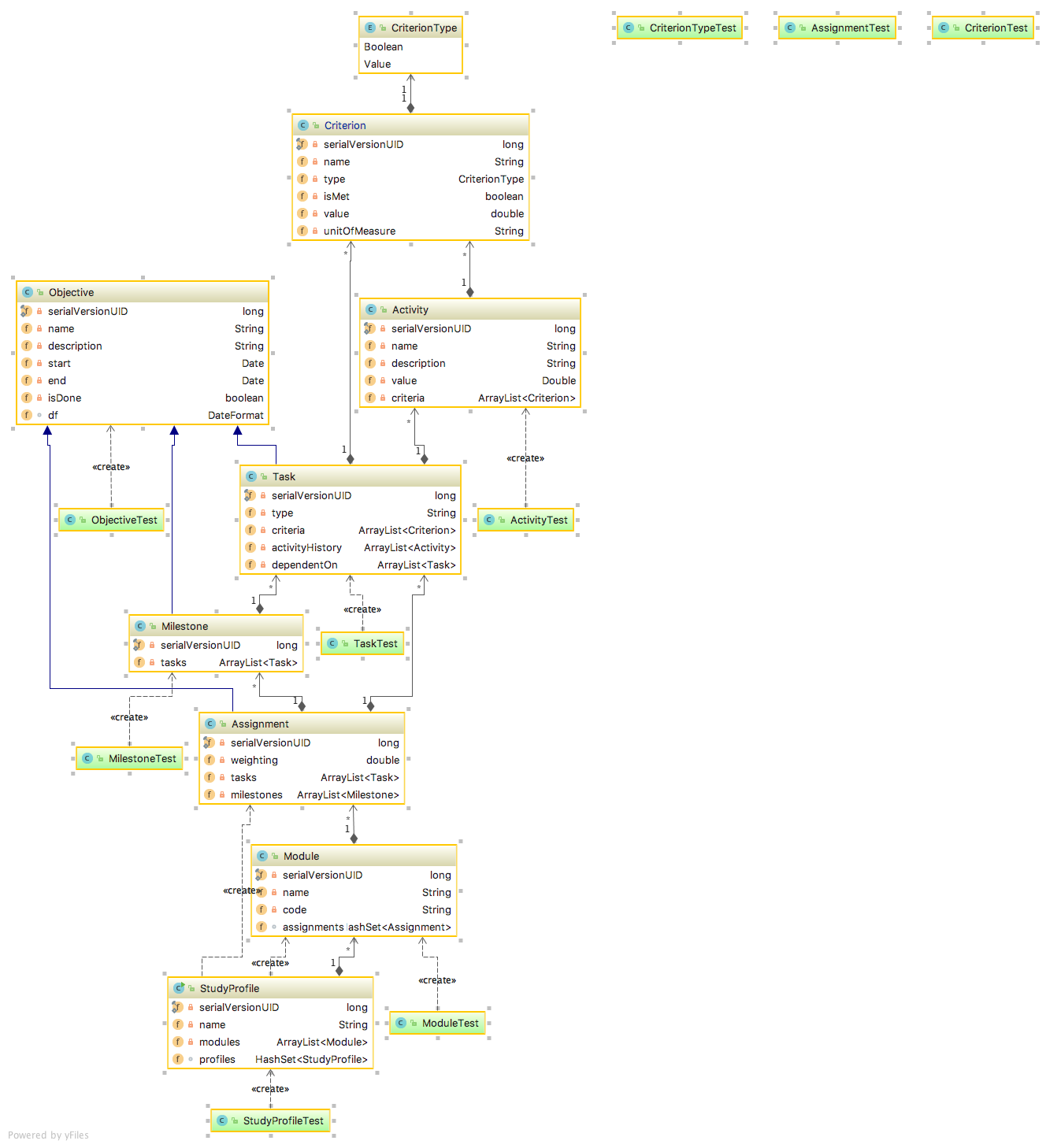
[Requirements 25](#_Toc483919621)

[MoSCoW Analysis 25](#_Toc483919622)

[Potential Classes 26](#_Toc483919623)

[Class Diagrams 27](#_Toc483919624)

[Initial Class Diagram 27](#_Toc483919625)

[The Model 28](#_Toc483919626)

[The Architecture 31](#_Toc483919627)

[State-Transition Diagrams 35](#_Toc483919628)

[Create Semester Profile 35](#_Toc483919629)

[Sequence Diagrams 38](#_Toc483919630)

[Create Activity 38](#_Toc483919631)

[Save Study Profile 39](#_Toc483919632)

# Glossary

**Deadline** - deadline for a certain task, milestone, and assignment. It may change based on extensions or module organiser decisions.

**Define Tasks** - a user should be able to create their own tasks, but they must be based on existing coursework and/or exams and must also belong to an existing task type (e.g. programming, reading, research etc.).

**Task Types** - Categories a task may belong to. These include: Programming, Research, Writing, Revision, etc. Users may also add new tasks types.

**Dependencies (of tasks)** - Tasks may have dependencies on each other e.g. one task cannot be started before another is completed. Other tasks may be done in parallel.

**Task** - a user defined task (thing to be done), that must be based on the user’s module schedule assignments and exams. The completion of a task is based on criteria (defined by the user), that must be met before the task is automatically considered complete.

**Criteria** - a user defined criteria, which can be met by updating it with (adding) activities. It is either *BOOLEAN* (just needs a new activity defined and added to it) or *VALUE* (needs the value of added activities equal or exceed the value of the criteria). If it is *VALUE*, then the user may also define *UNITS* the criteria is measured in (e.g. minutes or horses hugged).

**Study Activities** - A user defined activity, which must be attached to a criteria in a task that contributes to meeting of said criteria.

For example, suppose we create a task “Write Card Class”, for which we define the following criteria: “Finish Commenting” and “Write 3 Unit tests”, where the user defines *UNITS* as “unit tests written”. Now, the user may create a new activity “Done commenting” and add it to the first criteria. Since it was a Boolean criteria, it is now considered met. Next, the user adds activity “Wrote 2 Unit tests” to the second criteria, which now displays that the user has “2 out of 3 unit tests written”.

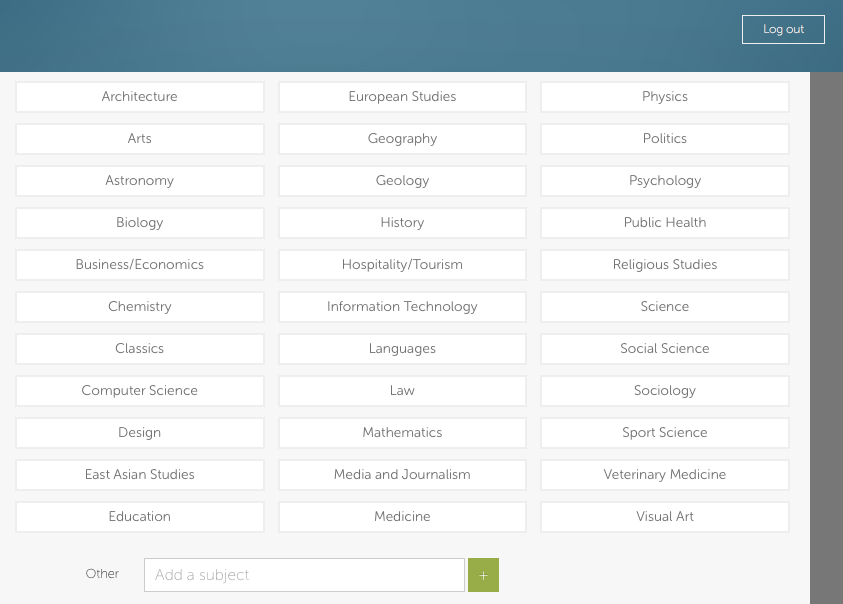
**Milestone** - A user-set goal, that must be achieved by a certain user-set deadline and must have at least one task attached to it that relates to the completion of said goal. Goal can’t be completed until all tasks attached to it are completed.

**Study Dashboard** - Visualisation and highlighting of met, missed or upcoming deadlines. Also lets the user view a Gantt Chart.

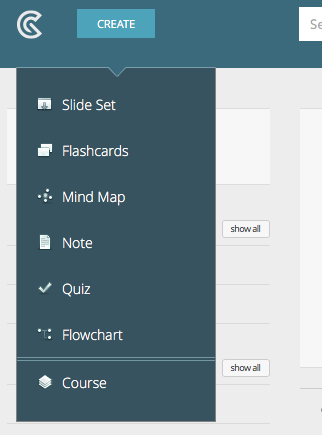
# Similar System Analysis

## GoConqr

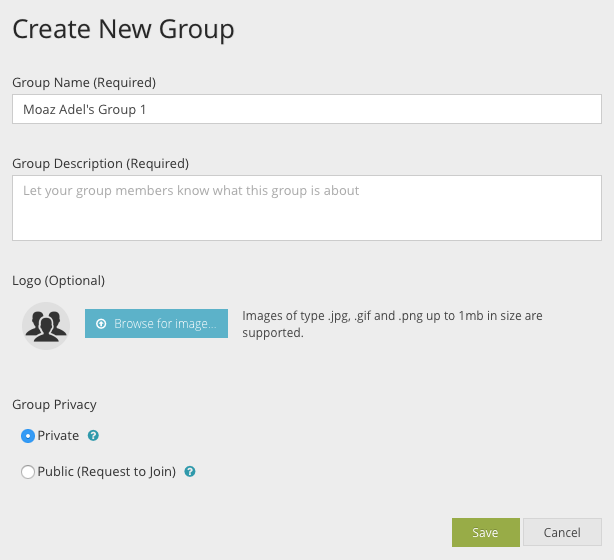
GoConqr is an online database that helps users to create a study environment or plan. By adding first the modules you desire which you chose from a list or you can create your own module if not found in the list.



The system allows the user of adding any learning resources which is great simply you choose the module you would like to add resource to it and then chose the type of resources you would like to add and is great to keep all your study resources in one place.



Using GoConqr you can create groups of study and add more people to these groups, and you can create your own quizzes and notes, which is always great to interact with other users.



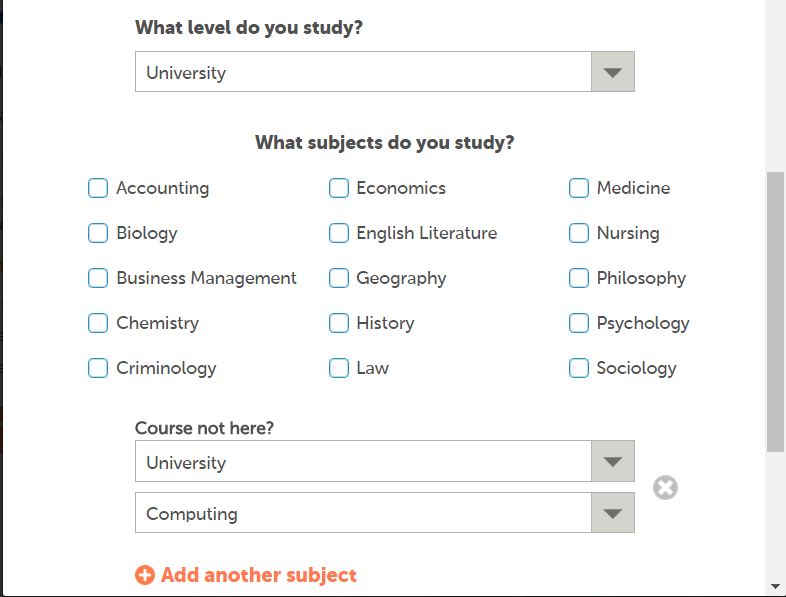
It also has a great feature that allows the user of exporting the calendar he made on GoConqr into his personal calendar.

It lacks:

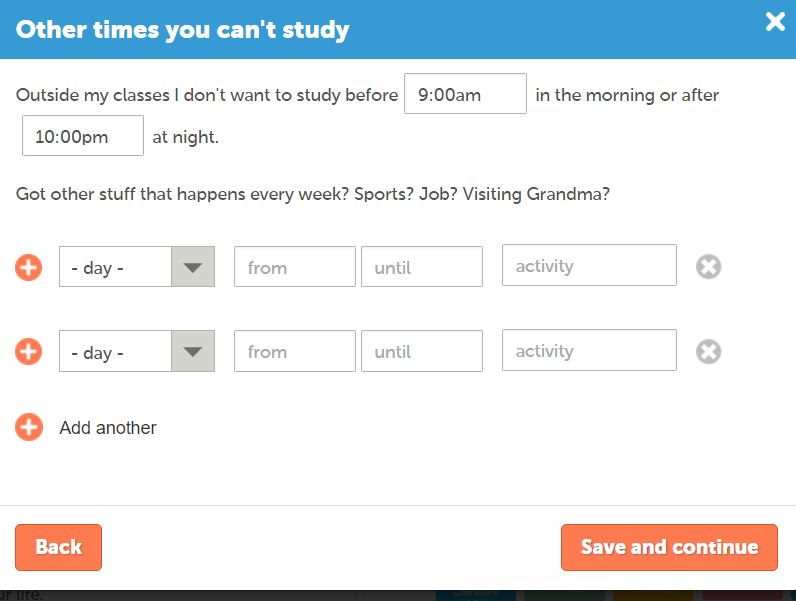
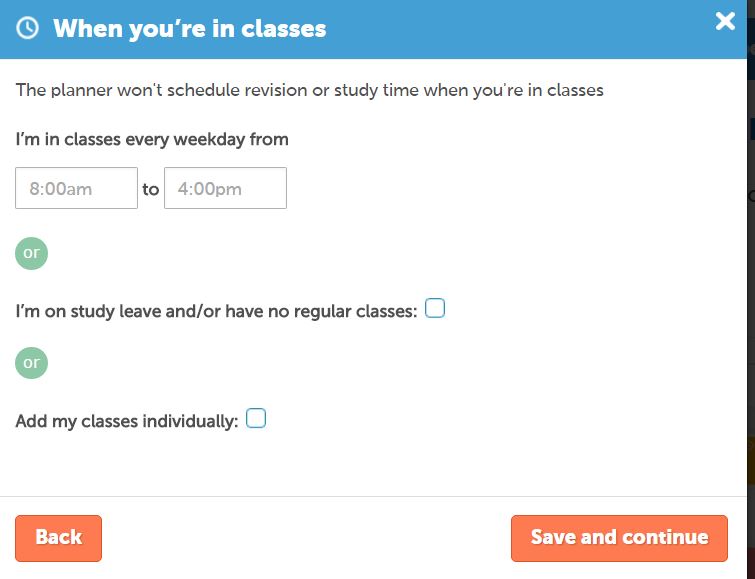
* The ability to add formatted file with all the modules and events (exams, coursework, seminars, etc.) - they all have to be added manually by choosing (Create new Event)
* The feature to Capture Time is quite vague and complicated to edit.
* Option of showing upcoming events or near deadlines on the main dashboard.

## Get Revising

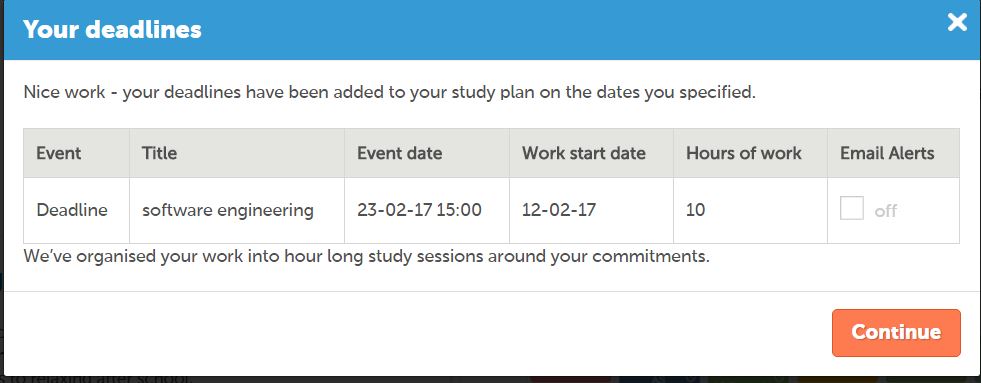
Get Revising is an online study planner that helps you create a study plan according to your needs. First you create your profile and select the course of your study.



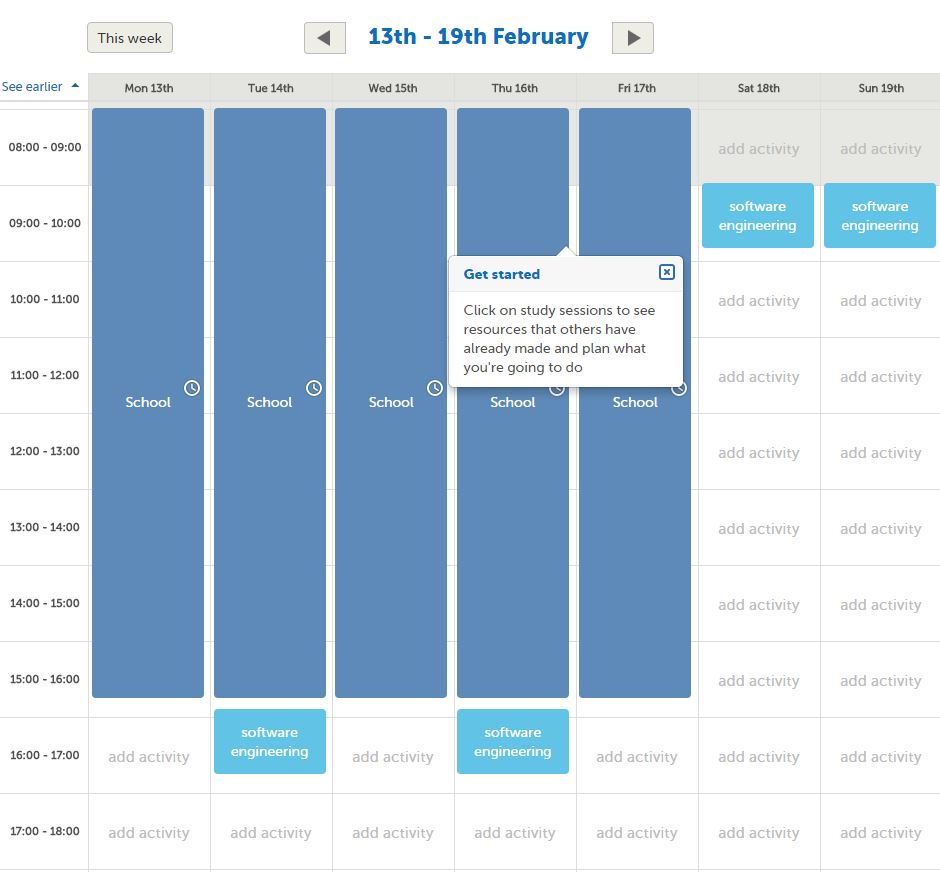
Then you select the time you are in classes and the times you can’t study so it doesn’t allocate any revision slots at that time.



Then you create your upcoming deadlines and allocate the date due and the hours you want to spend studying for that.



And finally it automatically creates a personalised schedule on the needs you provided.



Its main features are:

* Prioritises the modules you find more difficult
* Gives you email reminders to keep you to plan
* Breaks revision in manageable chunks
* Easily updated
* It takes other commitments you have into consideration and builds around them
* Drag and drop functionality to edit the plan
* Its printable
* Suggests resources to help you study

Some things it could have:

* Export feature to personal calendar
* Add times for general study not just for deadlines

# Use cases

**Actors:** Student

**Priorities:** High (3), Medium (2), Low (1)

**Frequency:** High (3) - very frequent, Medium (2) - moderately frequent, Low (1) - infrequent

**Performance target:** Rapid (3) - has to work very fast as these are core features of the program, Responsive (2) - has to work fast and be responsive to user, Unimportant (1) - are additional features that will affect the whole experience much

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Name** | **Primary actors** | **Performance target** | **Frequency** | **Priority** |
| 1 | Create Study Profile | Student | Rapid | Low | High |
| 2 | Define Tasks | Student | Responsive | High | High |
| 3 | Define Activities | Student | Responsive | High | High |
| 4 | Define Milestones | Student | Responsive | Medium | Medium |
| 5 | Edit Tasks | Student | Responsive | High | High |
| 6 | Edit Activities | Student | Responsive | High | High |
| 7 | Edit Milestones | Student | Responsive | Medium | Medium |
| 8 | Inspect Study Profile | Student | Unimportant | Medium | High |
| 9 | Edit Study Profile | Student | Rapid | Low | High |
| 10 | Capture Activity/Task Time | Student | Responsive | High | Medium |
| 11 | Inspect Gantt Chart | Student | Unimportant | Low | Low |
| 12 | Open Dashboard | Student | Rapid | High | High |

## Relationships between use cases

* Define tasks can only happen if study profile is created
* Define activities can only happen if a task is defined as an activity is part of a task
* Define milestones can only happen if task and activities are defined
* Edit tasks can only happen if there are defined tasks
* Edit activities can only happen if there are defined activities
* Edit milestones can only happen if there are defined milestones
* Inspect study profile can only happen if there is a created profile
* Edit study profile can only happen if there is a created profile
* Capture Activity/Task Time can only happen if there are defined tasks or activities

The system accepts a single data file from the hub (which contains the assignment info for all modules).

The student clicks on a button to create a new study profile, he then names it and uploads respective data file to the system. After creating the profile, the user sees a dashboard with all the modules and their assignments, exams etc. with deadlines. From now on, the word assignment means anything from coursework, to exams and module events.

The user may then double-click on any assignment and open a new window with a more detailed description of it. He may then create new tasks for this assignment. After creating at least one task, the user is then allowed to create a milestone, which must have at least one task attached to it.

Double clicking on a task opens a more detailed window of it. There, the user may write a description, add notes and activities. They may also mark the task as done, as long as they don’t have any activities associated with it.

Updating the data file keeps the existing tasks, etc. in the study profile.

## Use cases specifications

### Create Study Profile

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Create Study Profile | |
| Goal in Context | To initialise a study profile for the Student | |
| Scope & Level | Study Planner | |
| Preconditions | User is in the main screen and has downloaded file from the Hub | |
| Success End Condition | A new study profile is created | |
| Failed End Condition | Unable to initialise a new study profile | |
| Primary Actor | Student | |
| Trigger | “New Study Profile” button selected | |
| SUCCESS SCENARIO | Step | Action |
|  | 1 | Student gives name to new Study Profile |
|  | 2 | Application asks the Student for a data file |
|  | 3 | Student loads HUB file |
|  | 4 | Application asks to confirm the creation of the profile |
|  | 5 | The profile is now initialised and system confirms the creation |
| ALTERNATIVE SCENARIO | Step | Branching Action |
|  | 2a | The uploaded file is corrupted/unloadable/incorrect |
|  | 2b | System informs about the problem and returns to step 2 |
| RELATED INFORMATION | | |
| Priority | High | |
| Performance Target | Rapid | |
| Frequency | Low | |
| Subordinate Use Cases | N/A | |
| Channel to Primary Actor | User Interface | |
| OPEN ISSUES | N/A | |
| SCHEDULE | N/A | |
| AUTHOR | Kiril Chomaniuk | |

### Define Tasks

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Define Tasks | |
| Goal in Context | To define/create Tasks in a study profile | |
| Scope & Level | Study Planner | |
| Preconditions | Study Profile created and assignments associated to modules are available | |
| Success End Condition | A new task is added to an assignment in the study profile | |
| Failed End Condition | Unable to create a new Task | |
| Primary Actor | Student | |
| Trigger | “Add Task” button selected | |
| SUCCESS SCENARIO | Step | Action |
|  | 1 | Student is prompted to select the module, assignment |
|  | 2 | Student is prompted to enter task details (name, optional description, due date) |
|  | 3 | Student prompted to define criteria |
|  | 4 | System asks the Student to confirm a new task |
|  | 5 | Task is created and Student is displayed the study profile menu |
| ALTERNATIVE SCENARIO | Step | Branching Action |
|  | 2a | Task with same name already exists in assignment |
|  | 2b | System informs user that task with name like that already exists and returns to step 2 |
| RELATED INFORMATION | | |
| Priority | High | |
| Performance Target | Responsive | |
| Frequency | High | |
| Subordinate Use Cases | N/A | |
| Channel to Primary Actor | User Interface | |
| OPEN ISSUES | Possibly a dropdown menu of existing courseworks, modules? Disallows the Student to add invalid information | |
| AUTHOR | Kiril Chomaniuk | |

### Define Activities

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Define Activities | |
| Goal in Context | To define/create Activities in a task criteria | |
| Scope & Level | Study Planner | |
| Preconditions | Study Profile and tasks for assignments are created | |
| Success End Condition | A new Activity is added to a criteria | |
| Failed End Condition | Unable to create a new Activity | |
| Primary Actor | Student | |
| Trigger | “Add Activity” button selected | |
| SUCCESS SCENARIO | Step | Action |
|  | 1 | Student is prompted to select the module, assignment, task and criteria |
|  | 2 | Student is prompted to enter Activity details (name, etc) |
|  | 3 | Application asks the Student to confirm new Activity |
|  | 4 | Activity is created and criteria is updated |
| ALTERNATIVE SCENARIO | Step | Branching Action |
|  | 2a | If selected criteria is of type Value, then user is also prompted to input a double |
|  | 2b | GOTO step 3 |
| RELATED INFORMATION | | |
| Priority | High | |
| Performance Target | Responsive | |
| Frequency | High | |
| Subordinate Use Cases | N/A | |
| Channel to Primary Actor | Study profile menu | |
| OPEN ISSUES | Possibly a dropdown menu of existing courseworks, modules? Disallows the Student to add invalid information | |
| AUTHOR | Kiril Chomaniuk | |

### Define Milestones

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Define Milestones | |
| Goal in Context | To define/create Milestones in a module assignment | |
| Scope & Level | Study Planner | |
| Preconditions | Study profile created, at least a single task is defined | |
| Success End Condition | Milestone successfully added to an assignment | |
| Failed End Condition | Unable to create a new Milestone | |
| Primary Actor | Student | |
| Trigger | “Add Milestone” button selected | |
| SUCCESS SCENARIO | Step | Action |
|  | 1 | Student is prompted to select module, assignment |
|  | 2 | Student is prompted to enter Milestone details (name, description, date) |
|  | 3 | Student is prompted to add Tasks to the Milestone |
|  | 4 | Application asks the Student to confirm new Milestone |
|  | 5 | Milestone is created, Student returned to Study Profile menu |
| ALTERNATIVE SCENARIO | Step | Branching Action |
|  | 2a | Milestone with same name already exists in Assignment |
|  | 2b | System informs user that milestone with name like that already exists and returns to step 2 |
| RELATED INFORMATION | | |
| Priority | Medium | |
| Performance Target | Responsive | |
| Frequency | Medium | |
| Subordinate Use Cases | N/A | |
| Channel to Primary Actor | User interface | |
| OPEN ISSUES | N/A | |
| AUTHOR | Ioakim Ioakim | |

### Edit Tasks

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Edit Tasks | |
| Goal in Context | To edit and change Task details | |
| Scope & Level | Study Planner | |
| Preconditions | Study Profile created, at least one Task defined | |
| Success End Condition | Desired Task successfully edited | |
| Failed End Condition | Changes to Task are not saved/accepted | |
| Primary Actor | Student | |
| Trigger | “Edit Task” button selected | |
| SUCCESS SCENARIO | Step | Action |
|  | 1 | Student navigates to a Task and clicks on the button |
|  | 2 | Student edits Task details (name, description, date, dependencies, date, criteria) |
|  | 3 | System saves changes confirms and returns to main window |
| ALTERNATIVE SCENARIO | Step | Branching Action |
|  | 3a | Changes create conflict |
|  | 3b | System reports error and returns to step 2 |
| RELATED INFORMATION | | |
| Priority | High | |
| Performance Target | Responsive | |
| Frequency | High | |
| Subordinate Use Cases | N/A | |
| Channel to Primary Actor | User interface | |
| OPEN ISSUES | N/A | |
| SCHEDULE | Due date is version 1.0 release | |
| AUTHOR | Ioakim Ioakim 10/2/2017 | |

### Edit Activities

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Edit Activities | |
| Goal in Context | To edit and change Activity details | |
| Scope & Level | Study Planner | |
| Preconditions | Study Profile created, Tasks are defined, at least one Activity is defined | |
| Success End Condition | Desired Activity successfully edited | |
| Failed End Condition | Changes to Activity are not saved/accepted | |
| Primary Actor | Student | |
| Trigger | “Edit Activity” button selected | |
| SUCCESS SCENARIO | Step | Action |
|  | 1 | Student navigates to an Activity and clicks on the button |
|  | 2 | Student edits Activity details (name, description, value) |
|  | 3 | System saves changes, confirms and returns to main window |
| ALTERNATIVE SCENARIO | Step | Branching Action |
|  | 3a | Changes create conflict |
|  | 3b | System reports error and returns to step 2 |
| RELATED INFORMATION | | |
| Priority | High | |
| Performance Target | Responsive | |
| Frequency | High | |
| Subordinate Use Cases | N/A | |
| Channel to Primary Actor | User interface | |
| OPEN ISSUES | N/A | |
| SCHEDULE | Due date is version 1.0 release | |
| AUTHOR | Ioakim Ioakim 10/2/2017 | |

### Edit Milestones

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Edit Milestones | |
| Goal in Context | To edit and change Milestone details | |
| Scope & Level | Study Planner | |
| Preconditions | At least one Task is defined, at least one Milestone defined | |
| Success End Condition | Desired Milestone successfully edited | |
| Failed End Condition | Changes to Milestone are not saved/accepted | |
| Primary Actor | Student | |
| Trigger | “Edit Milestone” button selected | |
| SUCCESS  SCENARIO | Step | Action |
|  | 1 | Student navigates to a Milestone and clicks on the button |
|  | 2 | Student edits Milestone details (name, description, date) |
|  | 3 | System saves changes, confirms and returns to main window |
| ALTERNATIVE SCENARIO | Step | Branching Action |
|  | 3a | Changes create conflict |
|  | 3b | System reports error and returns to step 2 |
| RELATED INFORMATION | | |
| Priority | Medium | |
| Performance Target | Responsive | |
| Frequency | Medium | |
| Subordinate Use Cases | N/A | |
| Channel to Primary Actor | User interface | |
| OPEN ISSUES | N/A | |
| SCHEDULE | Due date is version 1.0 release | |
| AUTHOR | Moaz Hafez | |

### Save Study Profile

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Save Study Profile | |
| Goal in Context | To save all the data defined by the user | |
| Scope & Level | Study Planner | |
| Preconditions | Study Profile created, tasks and activities defined | |
| Success End Condition | All user data are successfully serialised and saved | |
| Failed End Condition | Data failed to save or corrupted | |
| Primary Actor | Student | |
| Trigger | “Save Profile” button selected | |
| SUCCESS SCENARIO | Step | Action |
|  | 1 | Student creates a new study profile |
|  | 2 | Student plans his study |
|  | 3 | Student selects “Save Profile” button |
|  | 4 | All information is now saved and serialised |
|  |  |  |
| ALTERNATIVE SCENARIO | Step | Branching Action |
|  | 3a | Profile failed to save |
|  | 3b | Inform Student that changes were not saved |
| RELATED INFORMATION | | |
| Priority | High | |
| Performance Target | Responsive | |
| Frequency | Medium | |
| Subordinate Use Cases | Create Study Profile | |
| Channel to Primary Actor | User Interface | |
| OPEN ISSUES | How to save only the things that are different? | |
| SCHEDULE | To be added as soon as user can create a study profile | |
| AUTHOR | Moaz Hafez | |

### Edit Study Profile

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Edit Study Profile | |
| Goal in Context | Edit any changes in the study plan. (Change a deadline, edit a task, create new task, edit dependencies) | |
| Scope & Level | Study Planner | |
| Preconditions | Study Profile created, tasks and activities defined | |
| Success End Condition | Study Profile successfully edited | |
| Failed End Condition | Changes to Study Profile are not saved/accepted | |
| Primary Actor | Student | |
| Trigger | “Edit Study Profile” button selected | |
| SUCCESS SCENARIO | Step | Action |
|  | 1 | Student navigates to a Study Profile and clicks on the button |
|  | 2 | Student edits Study Profile details (name, data file) |
|  | 3 | System saves changes, confirms and returns to main window |
| ALTERNATIVE SCENARIO | Step | Action |
|  | 3a | Changes create conflict |
|  | 3b | System reports error and returns to step 2 |
| RELATED INFORMATION | | |
| Priority | High priority | |
| Performance Target | Rapid | |
| Frequency | Low | |
| Subordinate Use Cases | Create Study Profile | |
| Channel to Primary Actor | N/A | |
| OPEN ISSUES | N/A | |
| SCHEDULE | Due date is version 1.0 release | |
| AUTHOR | Moaz Hafez | |

### Capture Activity/Task Time

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Capture Activity/Task Time | |
| Goal in Context | To capture the amount of time taken to finish a task/activity | |
| Scope & Level | Study Planner | |
| Preconditions | Student defined task/activity, Student marked task/activity as finished | |
| Success End Condition | System successfully tracked and displayed the time taken on a task/activity | |
| Failed End Condition | System failed to track time taken to finish task/activity | |
| Primary Actor | Student | |
| Trigger | Student marks a task or activity as finished | |
| SUCCESS SCENARIO | Step | Action |
|  | 1 | The System now displays the elapsed time from start to finish next to the task/activity |
| ALTERNATIVE SCENARIO | Step | Branching Action |
|  | 1a | System unable to display time consumed |
| RELATED INFORMATION | | |
| Priority | Medium | |
| Performance Target | Rapid(less than 0.5 a second) | |
| Frequency | High(As often as task/activities are defined and finished) | |
| Subordinate Use Cases | Define Task, Define Activity, Edit Task, Edit Activity | |
| Channel to Primary Actor | User Interface | |
| OPEN ISSUES | If user’s start time is at a later date than his finish time (due to editing PC’s date), then time displayed may be negative or undefined | |
| SCHEDULE | To be released as soon as a person is able to define and finish tasks/activities | |
| AUTHOR | Michail Krugliakov | |

### Inspect Gantt Chart

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Inspect Gantt Chart | |
| Goal in Context | To help Student visualise how defined Activities, Tasks and Milestones contribute to module progress | |
| Scope & Level | Study Planner | |
| Preconditions | User defined Tasks and Activities and is viewing the dashboard | |
| Success End Condition | Student informed about his progress in a module | |
| Failed End Condition | Gantt Chart didn’t have data to work with and failed to visualize progress | |
| Primary Actor | Student | |
| Trigger | Dashboard is opened and Gantt Chart button is selected | |
| SUCCESS SCENARIO | Step | Action |
|  | 1 | Student view the Gantt Chart |
| ALTERNATIVE SCENARIO | Step | Branching Action |
|  | 1a | Gantt Chart fails to display properly |
|  | 1b | System informs the user |
| RELATED INFORMATION | | |
| Priority | Low | |
| Performance Target | Unimportant | |
| Frequency | Medium | |
| Subordinate Use Cases | Define Task, Define Activity | |
| Channel to Primary Actor | User interface | |
| OPEN ISSUES | How to display data in a helpful and visually pleasant way | |
| SCHEDULE | To be released by version 1.0 of Study Planner | |
| AUTHOR | Michail Krugliakov | |

### Open Dashboard

|  |  |  |
| --- | --- | --- |
| USE CASE NAME | Open Dashboard | |
| Goal in Context | Lets Student examine how semester study is progressing | |
| Scope & Level | Study Planner | |
| Preconditions | Student defined tasks and activities | |
| Success End Condition | Student opened dashboard and received relevant information | |
| Failed End Condition | Dashboard didn’t open | |
| Primary Actor | Student | |
| Trigger | “Open Dashboard” button selected | |
| SUCCESS SCENARIO | Step | Action |
|  | 1 | Student views Dashboard |
| ALTERNATIVE SCENARIO | Step | Branching Action |
|  | 1a | System fails to display the Dashboard or wrong data displayed |
|  | 1b | System informs the user |
| RELATED INFORMATION | | |
| Priority | Medium | |
| Performance Target | Rapid | |
| Frequency | Medium | |
| Subordinate Use Cases | Define Task, Define Activity | |
| Channel to Primary Actor | User Interface | |
| OPEN ISSUES | Should approaching deadlines notification display once per session or each time Dashboard is opened? | |
| SCHEDULE | To be integrated in version 1.0 | |
| AUTHOR | Michail Krugliakov | |

# **Requirements**

## MoSCoW Analysis

Class

Property

Must have

* Load module, coursework and deadline information from a defined file format
* Ability to define and edit study tasks, criteria, details, milestones and deadlines
* Ability to record study activities that contribute towards completing task criteria
* Dependencies can exist between tasks (such as one task can only be started after another has been completed)
* Support a Study Dashboard that highlights upcoming, missed and finished deadlines, progress towards milestones.
* Assignments must have % of module weightings displayed as well
* The system must be able to change deadlines

Should have

* Serialisation of all tasks/activities/milestones of the study profile
* Visualise activities, dependencies, intermediate milestones and deadlines in a Gantt chart representation as well as a study progress dashboard that highlights upcoming, missed and finished deadlines, progress towards completing milestones and time spent for each module
* User should be able to define new types of tasks

Could have

* Student can request help from the adviser to make a plan
* Online functionality
* Receive notifications for deadline changes from the Hub
* Login support
* Achievements system, rewards, offers
* Dashboard deadlines can be clicked on and navigate the user to respective detailed page

Won’t have

* Online registration, because user would need to be using the UEA login system
* Activity being attachable to multiple tasks and contributing to all of them

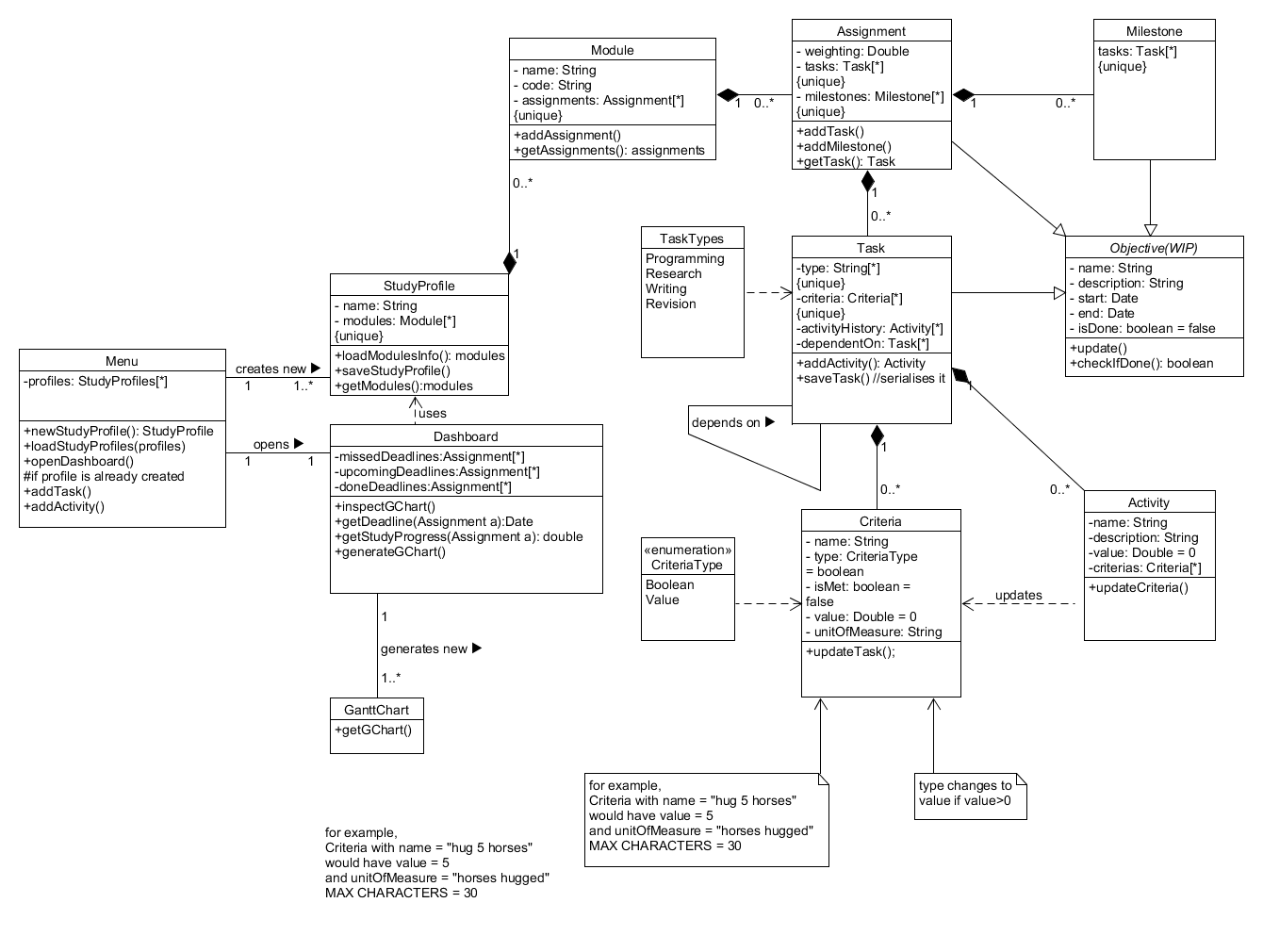
The priority for system components are determined by the analysis of the requirements specification. It is assumed that online functionality would be beneficial to have in future, yet it is not crucial for overall workflow of the system and allows to shift most of the development time to implementation and testing of the critical parts of the application.

## Potential Classes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property** |  | **Potential Classes** |  | **Relation** |
| Deadline |  | Module |  | Coursework is part of module |
| Details |  | Coursework |  | Criteria is part of task |
| Dependencies |  | Task |  | Milestone is part of task |
| Progress towards |  | Criteria |  | Activity is part of task |
| 10% of module weightings |  | Milestone |  | StudyDashboard is part of study profile |
| Serialisation |  | Activities |  | Assignment is part of module |
| Upcoming |  | StudyDashboard |  | Gantt chart is part of study profile |
| Missed and finished deadlines |  | Assignment |  | StudyProgress is part of study profile |
| Progress |  | StudyProfile |  |  |
| Time spent |  | GanttChart |  |  |
| Online |  | StudyProgressDashboard |  |  |
| Notification |  |  |  |  |
| Log in |  |  |  |  |

# **Class Diagrams**

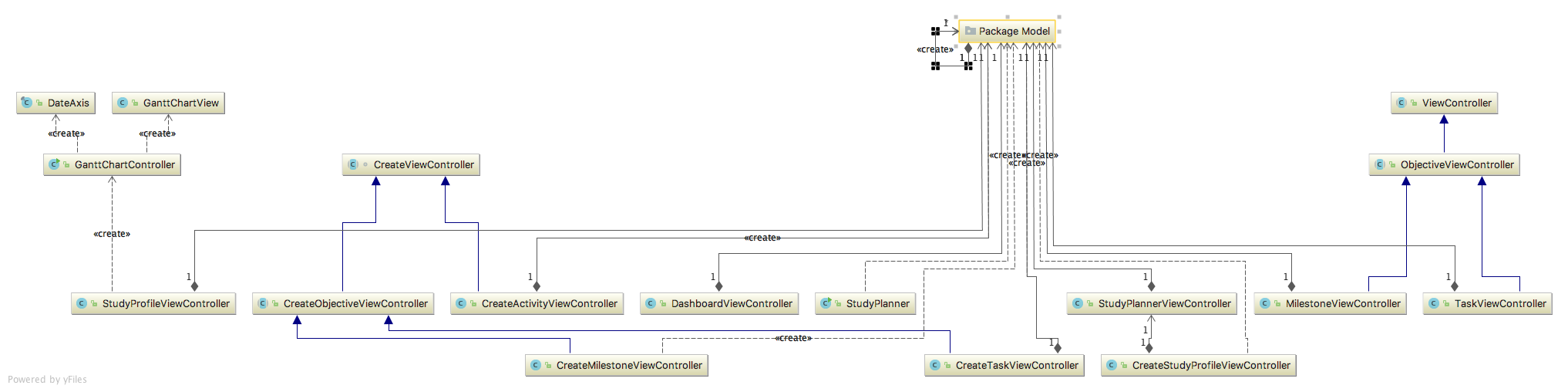
## Initial Class Diagram

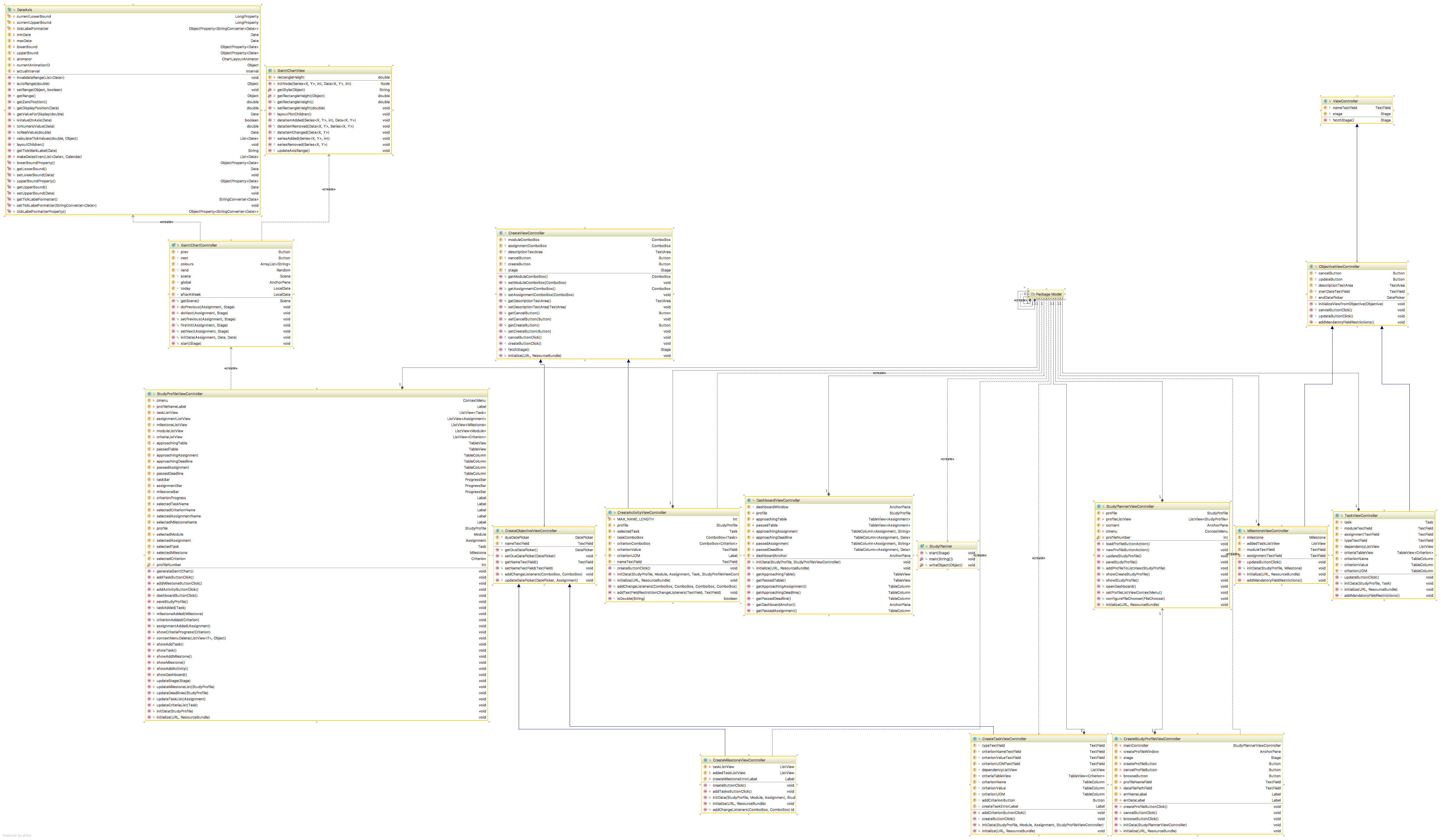


## The ModelMacintosh HD:Users:Moaz:BitBucket:diagramModel.png

A choice was made to make a Study Profile that contains all the module information. Assignments exist for each module and for each assignment a task is created that is comprised of Criteria and Activities. For each assignment to be complete, the assigned Criteria need to be met and all the activities to be completed. The Objective is an abstract class, from which Milestones, Assignments and Tasks are inherited. This diagram is modelling the behaviour of the important components of the application.

## The Architecture





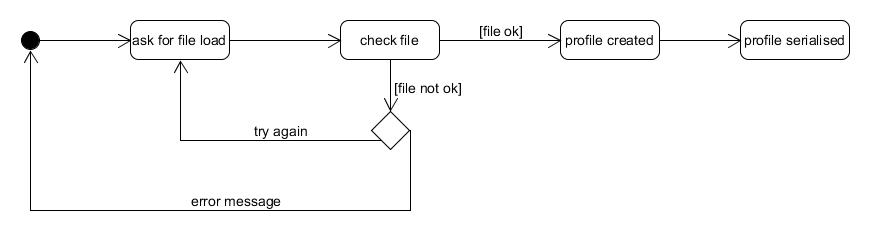
MVC with Fields and Methods

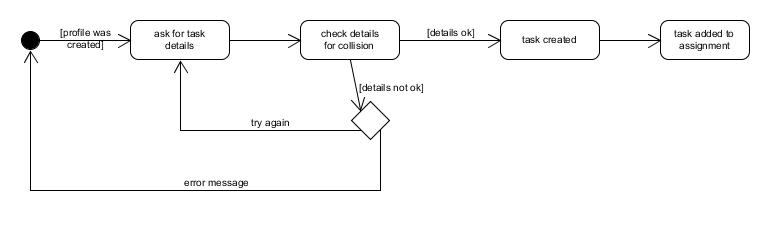
Application’s architecture is the Model-View-Controller. The above-mentioned model classes are being used in conjunction with the View (MenuView, StudyProfileView, DashboardView, GanttChartView) classes and Controller (MenuControl, StudyProfileControl, DashboardControl) classes.

## **State-Transition Diagrams**

These diagrams show the states that the application is going through to complete a certain action of the application.

### Create Semester Profile

 Define Tasks

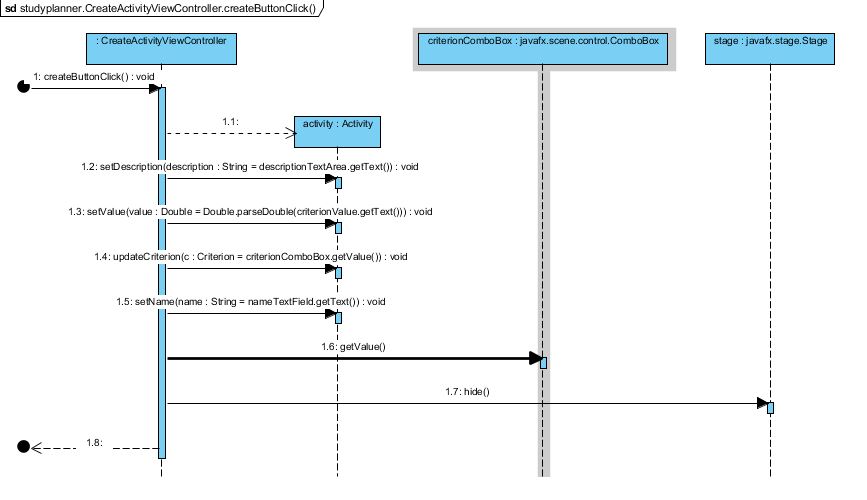


## 

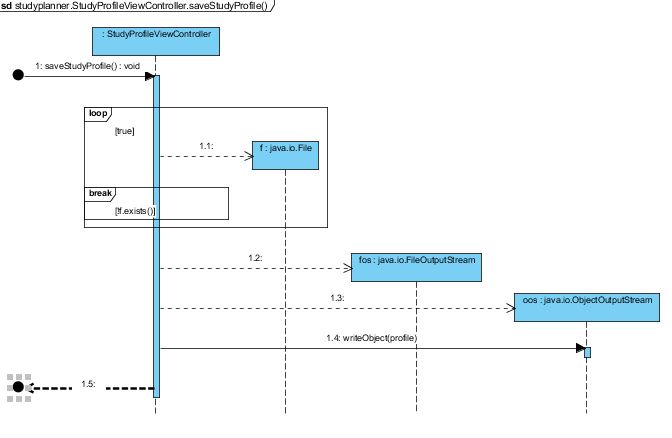
## **Sequence Diagrams**

These diagrams show the interaction between the system objects (Model-View-Controller parts) and the user in order to complete a certain action of the application. These interactions are ordered by time.

### Create Activity



### Save Study Profile



Development Plan

As the assignment is a group project we developed our system as a git project which enabled us to share our code and each team member to work on different parts of the code. Online meetings happened at the end of each week to check on progress and resolve any issues. We initially have set a weekly plan that later changed as development decisions were made that changed the flow of work. We initially used a Java project and Swing for view development but we later switched to JavaFx as we found it to be a much easier to develop way. The development IDE was NetBeans.

Overall the approach followed was that each team member created a part of the model and tested it and then all the parts were merged and tested as a whole. Then decisions were made for how the view was going to look like and it was assigned to a team member to create the views and then the controllers were created. At the same time other team members worked on reading the hub files and in serializing the objects.

Testing Plan

When developing a computer system it is critical to test the functionality and behaviour of the system to make sure that is acting in the desirable way and providing the correct output as a code with no errors is not enough to show that the system is working properly. So following a test-driven development is the best way to achieve that.

Whenever the individual code of team members was to be merged, the team made sure that that before merging the individual code was tested and when merged the cross-class methods were tested to see that the system works as a whole. Also while writing the code we had simple in-line tests and we used debugging console to see that the system was behaving correctly.

We used Junit tests to test that all the methods of the system behave correctly. We made sure that every object is created correctly and every change that a method makes happen correctly and the system has the desirable behaviour.

Also while developing the system they have been numerous merge issues because of the changes of the code that lead us to clean the test classes and create them again.

In order to assert passing of failing the criteria, after creating the test we had two variables, expResult and result for expected result and actual result respectively and we checked that they were equal.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case** | **Activity** | **Assignment** | **Criterion** | **Criterion type** | **Milestone** | **Module** | **Objective** | **Study profile** | **task** |
| ActivityTest.testUpdateCriterion | X |  | X | X |  |  |  |  |  |
| ActivityTest.testGetName | X |  |  |  |  |  |  |  |  |
| ActivityTest.testSetName | X |  |  |  |  |  |  |  |  |
| ActivityTest.testGetDecription | X |  |  |  |  |  |  |  |  |
| ActivityTest.testSetDescription | X |  |  |  |  |  |  |  |  |
| ActivityTest.testGetValue | X |  |  |  |  |  |  |  |  |
| ActivityTest.testSetValue | X |  |  |  |  |  |  |  |  |
| ActivityTest.testGetCriteria | X |  | X | X |  |  |  |  |  |
| ActivityTest.testAdd\_Criterion | X |  | X | X |  |  |  |  |  |
| ActivityTest.testAdd\_ArrayList | X |  | X | X |  |  |  |  |  |
| AssignmentTest.testAddTask | X | X | X |  |  |  |  |  | X |
| AssignmentTest.testAddMilestone |  | X |  |  | X |  |  |  | X |
| AssignmentTest.testGetWeighting |  | X |  |  |  |  |  |  |  |
| AssignmentTest.testSetWeighting |  | X |  |  |  |  |  |  |  |
| AssignmentTest.testGetTasks | X | X | X |  |  |  |  |  | X |
| AssignmentTest.testSetTasks | X | X | X |  |  |  |  |  | X |
| AssignmentTest.testGetMilestones |  | X |  |  | X |  |  |  | X |
| AssignmentTest.testSetMilestones |  | X |  |  | X |  |  |  | X |
| AssignmentTest.testToString |  | X |  |  |  |  |  |  |  |
| CriterionTest.testUpdateTask | X |  | X | X |  |  |  |  | X |
| CriterionTest.testGetName |  |  | X | X |  |  |  |  |  |
| CriterionTest.testGetType |  |  | X | X |  |  |  |  |  |
| CriterionTest.testSetType |  |  | X | X |  |  |  |  |  |
| CriterionTest.testIsMet |  |  | X | X |  |  |  |  |  |
| CriterionTest.testSetMet |  |  | X | X |  |  |  |  |  |
| CriterionTest.testGetValue |  |  | X | X |  |  |  |  |  |
| CriterionTest.testSetValue |  |  | X | X |  |  |  |  |  |
| CriterionTest.testGetUnitOfMeasure |  |  | X | X |  |  |  |  |  |
| CriterionTest.testSetUnitOfMeasure |  |  | X | X |  |  |  |  |  |
| CriterionTest.testToString |  |  | X | X |  |  |  |  |  |
| MilestoneTest.testGetTasks |  |  |  |  | X |  |  |  | X |
| MilestoneTest.testSetTasks |  |  |  |  | X |  |  |  | X |
| MilestoneTest.testAddTask | X |  | X |  | X |  |  |  | X |
| MilestoneTest.testRemoveTask | X |  | X |  | X |  |  |  | X |
|  | **Activity** | **Assignment** | **Criterion** | **Criterion type** | **Milestone** | **Module** | **Objective** | **Study profile** | **task** |
| MilestoneTest.testUpdate | X |  | X |  | X |  |  |  | X |
| ModuleTest.testGetName |  |  |  |  |  | X |  |  |  |
| ModuleTest.testGetCode |  |  |  |  |  | X |  |  |  |
| ModuleTest.testGetAssignments |  | X |  |  |  | X |  |  |  |
| ModuleTest.testSetName |  |  |  |  |  | X |  |  |  |
| ModuleTest.testSetCode |  |  |  |  |  | X |  |  |  |
| ModuleTest.testAdd |  | X |  |  |  | X |  |  |  |
| ModuleTest.testToString |  |  |  |  |  | X |  |  |  |
| StudyProfileTest.testGetName |  |  |  |  |  |  |  | X |  |
| StudyProfileTest.testGetModules |  |  |  |  |  | X |  | X |  |
| StudyProfileTest.testSetName |  |  |  |  |  |  |  | X |  |
| StudyProfileTest.testToString |  |  |  |  |  |  |  | X |  |
| TaskTest.testGetType | X |  | X |  |  |  |  |  | X |
| TaskTest.testSetType | X |  | X |  |  |  |  |  | X |
| TaskTest.testGetCriteria | X |  | X |  |  |  |  |  | X |
| TaskTest.testSetCriteria | X |  | X |  |  |  |  |  | X |
| TaskTest.testGetActivityHistory | X |  | X |  |  |  |  |  | X |
| TaskTest.testSetActivityHistory | X |  | X |  |  |  |  |  | X |
| TaskTest.testGetDependencies | X |  | X |  |  |  |  |  | X |
| TaskTest.testSetDependencies | X |  | X |  |  |  |  |  | X |
| TaskTest.testToString | X |  | X |  |  |  |  |  | X |

University of East Anglia

School of Computing Sciences

2017