TRABALHO FINAL DE CURSO

Engenharia Informática 2012-2013

"Discipulum Organisen"

Realizado Por:

João Pinto nº 20081875

Professor Orientador:

Professor Doutor Sérgio Guerreiro

Acknowledgements

To my father, for his patience, company, help and support during my struggles with the math courses of the degree. Would not be the same without you.

"Doubt that the stars are fire
Doubt that the sun doth move
Doubt truth to be a liar
But never doubt I love." --Shakespear, Hamlet

Abstract

Marc Aurelius once said that the secret of all victory lies in the organization of the non-obvious [12]. During my undergraduate computer science degree I missed a tool that could aggregate all my course needs and information in a single place allowing me to focus on the essentials instead of wasting time thinking how to organize my student life.

When the time came to do a final project as the last required step to finish my undergraduate degree, I recalled my difficulties in finding a piece of software that could fulfil my needs during my studies.

I decided then that it was an ideal opportunity to build one from scratch, doing all the necessary steps for creating a software that begun from a single idea on a piece of paper to its completion as a functional software.

The idea is to build a software that any student can use to organize his degree, from assignments to exams, from grades to teachers.

My main goal is to build a fairly simple and usable piece of software, that the users (students) can use during their degree without too much hassle or trouble, not a complete solution to every need. As the time allotted to this project is short and would not be enough for such an endeavour. It is also the aim that the project has the necessary foundations to allow it to be picked in the future by me or anyone interested in continuing the work.

Keywords: Windows Application, C# , student, Organizer

Resumo

Marco Aurélio escreveu uma vez que o grande sucesso de toda e qualquer vitória reside na organização do não-óbvio [12], durante a minha licenciatura em engenharia informática sempre senti a falta de uma ferramenta que agregasse num só sítio toda a informação e necessidades relevantes para o curso de modo que eu pudesse-me focar no essencial em vez de perder tempo a pensar como devia organizar a minha vida académica.

Ao me ser pedido para realizar um trabalho final como último passo necessário para completar a licenciatura, lembrei-me das minhas dificuldades e necessidades durante a licenciatura e decidi ser a oportunidade ideal para construir uma ferramenta do género de raiz percorrendo todos os passos necessários para construir um *software* desde a sua idealização num pedaço de papel até à sua conclusão como um software funcional.

A ideia é construir um *software* que permita aos estudantes organizarem a sua vida académica, desde trabalhos e exames a notas e professores.

O meu objectivo principal ao realizar este projecto é que o resultado seja um software relativamente simples e fácil de utilizar onde os utilizadores (estudantes) possam usar durante o seu percurso académico sem problemas ou complicações, não uma solução completa adequada a cada necessidade especifica pois o período de duração do projecto é curto e insuficiente para tal solução. É também um objectivo do projecto construir as bases necessárias que permita que o projecto seja extensível no futuro por alguém interessado em continuar o seu desenvolvimento.

Palavras-chave: Windows Application, C#, Vida académica, organização

Table of Contents

List of Figures

1. Introduction

Today we see computer science in almost every area of our society, from banking systems to libraries, from shopping to sports, it is Omni-present and because of that it is easy to forget how young the field is.

Even if we take into account the short years of human lifetime, its origins and its definition as a science in its own right emerge only from the Second World War making it less than 70 years old.

University Life on the other hand had its origins in the medieval era with the first known universities in Europe being the University of Bologna (1088) and the University of Paris (1150).

We might be allured to think that the university, being ordinarily one the first ones in society to adopt not only new technologies but also new ways of thinking about life and the challenges associated with it, would also be one of the first to change radically with the advances in computer science.

To be fair this is true in most areas of university life, it has indeed changed profoundly with the arrival of computers. But if we focus on how students do study, how they organize themselves, how they gather information and how they store it, we can easily conclude that it have not change that much in the last 65 years.

If we allow ourselves some time to scrutinize how students work nowadays, we can observe them still printing thousands and thousands of pages that teachers give them in electronic format, we observe them buying and consulting technical books that they truly only need for a semester but are unavailable in digital format, paper book organizers like the "The Palgrave Student Planner" are still one of the most popular and bought books by students and they still

organize their notes and course information in written notebooks, not taking advantage of the amazing advances that computer science have made to our life.

The aim with this project is to be able to construct a simple but practical tool that allows students to organize their student life in a more modern and hassle freeway if they want to. Allowing for a more efficient and lasting method to, not only, make and preserve their notes and comments about the degree they are engaged in but also to organize and have a centralized view of their assignments, exams, grades and calendars.

The project will be constructed in the Framework .NET using the language "C#" and it will be made with the intention of making a iOS or Android version on it that can easily synchronize with the desktop version, although the main focus will be the desktop one and only if time allows will the mobile version be made.

1.1. Structure of the report

In the first chapter is made a brief introduction to the project. Chapter two makes a short reflection on how students work. In chapter three a revision is made of the state of the art regarding solutions for the problem arisen in introduction. Chapter Four addresses the technology used in this project. Chapter five explains the main decisions made during the project. Chapter Six is the most important one as it addresses the System Modelling and Design of the project. Chapter seven exhibits the results from the software development and chapter eight is the last chapter and include the conclusion and suggestions for future work.

2. Theoretical Framework

Since the aim of this project to build a software tool to help students study, we need to understand what components are needed for success in this area and how can we can build a tool than can help students achieve it.

According with Stella Cottrell award-winning author of several books and papers about study skills [5][6] there are five main study-skills components

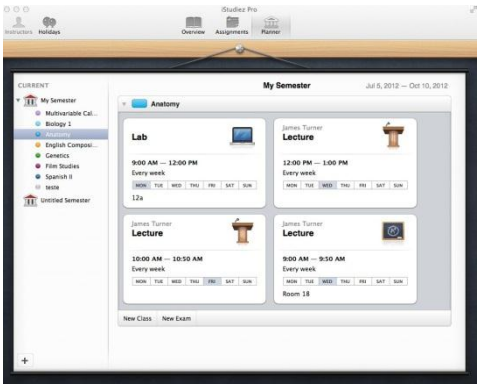
1. Self-awareness and self-evaluation
2. Awareness of what is required
3. Methods, organization, strategies
4. Confidence and permission
5. Familiarity: practice and habit

I will, obviously, focus on the methods, organization and strategies component since the others, although very interesting, are beyond the scope of this project.

3. State of Art

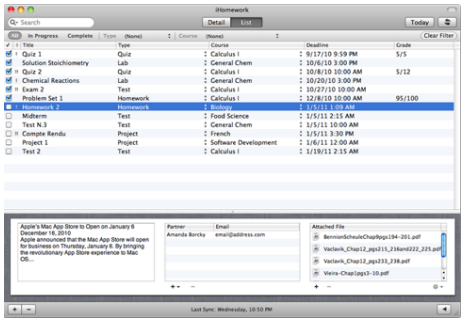
With the spectacular rise of the popularity of the tablet market and the apps store associated with them, some applications appeared to try to tackle the need described in the introduction, some becoming so popular that a desktop version was made. The more popular are:

iStudiez Pro™: First Created in 2009 for the iPhone™, it has won several awards since including Best College Student App (2011) and Best Young Adult app (2010). Its main advantage is its presence in most apple hardware combined with good sync technology that allows the users to always have the information with them. It lacks however class note taking capability. (Fig 3.1) [7]



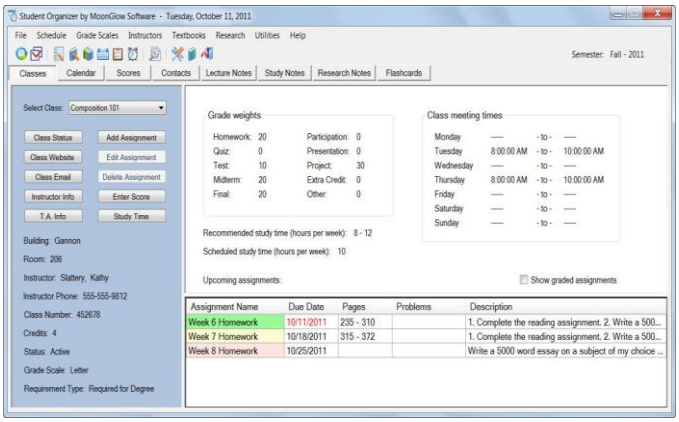
3.1 Screenshot istudiez Pro

iHomework™: As iStudiez Pro™, it started as a popular iphone/ipad application, afterward launched in Mac app Store in January 2011 with a huge success. Its main strength is it accessible user interface, as downside it lacks many features; For example it doesn't show classes on the assignment calendar,, and doesn't offer much more functionality than the built-in calendar and contacts. [21] (Fig 3.2) [8]



3.2 Screenshot iHomework

Student Organizer™ by MoonGlow®: This software is a popular School planner, homework and note organizer for windows. Its main strength is it ample series of features like classes recording and flashcards technology. It lacks however a mobile version and sync capability. (Fig 3.3) [9].



3.3 Screenshot Student Organizer

4. Technology Used

One of the main decisions that a developer needs to make when building a piece of software is what language and platform he should use.

Today there is a huge array of choice, from Windows and its .Net platform or the apple osx™ and its Cocoa platform, from desktop or Web, Traditional Operative system software or tablet with the ever popular iOs and Android operative systems.

My choice was to build the application in .net platform using the language c#, the main reasoning being that it was the platform that was most used during my undergraduate degree, with several courses using it. It makes sense for the final project of a degree to use a technology taught during it length and not a brand new technology.

Visual Studio: Visual studio is a suite of applications created by Microsoft® Studios to give developers a compelling development environment for windows and .Net platforms. Visual Studio can be used to write both Windows applications and Web Applications.

There have been numerous version of visual studio since its creation, the first version was launched in 1997 and combined development tools like visual basic and visual c++. Visual studio 6 coincided with the release of visual basic 6 and visual studio.net 2002 was released along the first version of the .NET framework.

The last version is Visual Studio 2012, but the one used in this project is visual studio 2008 due to some compatibility errors in the last version.

.NET Framework: The .NET Framework is a software framework developed by Microsoft® that runs primarily on Microsoft® Windows™. It includes a large library and provides language interoperability across several programming languages.

Programs written for the .NET Framework execute in a software environment known as the Common Language Runtime (CLR), an application virtual machine that provides services such as security, memory management, and exception handling. The class library and the CLR together constitute the .NET Framework.

C# Language: C# is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented and component-oriented programming disciplines. It was developed by Microsoft® within its .NET framework. The main advantage of C# that is a modernized version of C++ having its power but simplifying its use, making coding less time consuming and with a better memory management.

5. Decisions

When building a software project, there are a certain amount of decisions that have to be made. One of the main ones is without a doubt the motivation and reasoning to build the software.

One can argue that there are a huge amount of generalist applications that can be used by students to organize themselves. From well known applications like Microsoft® word [15], Scrivener [16], OneNote [17] or a most recent Evernote [18] can all do the job fairly well, so why build a specific application for that ?

I think that the reason behind developing a specific software was given by the users in the last few years.

If we look back to the last five years, with the release of mobile operative systems like iOs [19] and Android [20] we have seen an explosion of relative simple applications that fulfil very specific functions.

The average application users responded so well to this that Apple® released with huge success its own application store for its main operative system Mac Osx [13].

This step was then followed by Microsoft® for its own operative system (Windows store) [14]. Since then thousands of applications by independent developers have been created and have been very successfully.

This is the reason why I think it is safe to think that users are sending a clear message, stating that they want simple and intuitive software to do specific and common tasks, and not huge and overcomplicated do-it-all applications.

6. System Modelling and Design

6.1. UML

The Unified Modelling Language (UML) is a specification language, backed by single meta-models, that helps in describing and designing software systems, particularly software systems built using the object-oriented (OO) style[1].

Some developers consider UML simply as a specification language to be used for brainstorming and high-level documentation. Others consider UML to be a pictorial programming language, generating code from it or synthesizing it from existing code [1].

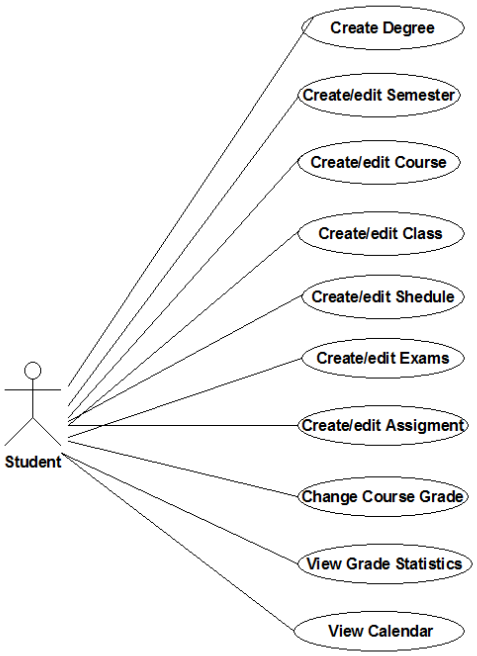
The UML is controlled by the object Management Group (OMG), an open consortium of companies. The OMG was formed to build standards that supported interoperability, specifically the interoperability of object-oriented systems.

The UML was born out of the unification of the many object-oriented graphical modelling languages that thrived in the late 1980's and early 1990's. Since it appeared in late 1997, it has relegated the other languages to history books.

6.1.2 User Cases

A user case is a static description of some way in which a system or a business is used, by its customers, its users of by other systems. [1]

In the case of this project, the software only has one kind of users, the student, so we only have one kind of user case (Fig 6.1)



6.1 Student user case

6.2. Requirements analysis

The main idea behind this project is to create a software tool that allows students to organize all aspects of their academic life; therefore students should be able among other things to:

1. Be able to load a previous created degree or create a new one (in the folder of his choosing).

2. Be able to create a degree with the specifications detailed below:

   - Degree Certification
   - Area of Study
   - University
   - Country
   - Language

- Number of Years
- Number of semesters per year
- Start of first Semester
- End of second Semester
- University Website

3. After the degree is created he should be able to:

- Create a course
- Delete a course
- Create an exam in the calendar
- Delete an exam in the calendar
- Add a teacher
- Delete a teacher
- Add an assignment
- Delete an assignment

4. Be able to create a course with the specifications detailed below:

- Name
- Year
- Semester
- Credits
- Grade
- Classes
- Exams
- Teacher
- Biography

5. After the course is created he should be able to:

- Add/delete classes
- Add a teacher and/or teacher Assistant
- Add/delete exam
- Add/delete Assignment
- Add/delete Class
- Add/delete Biography
- Add/delete grade

6. Create an exam in the calendar to create a exam with the specifications detailed below:

   - Date
   - Course
   - Name
   - Start time
   - Duration
   - Location

7. Create a Class of a specific course

A student should be a able to create a class with the specifications detailed below:

- Name
- Start Date
- End Date
- Start time
- Duration
- Repetition (which days of the week)
- Location
- Teacher/Instructor

After the class is created he should be able to:

- Add/delete contents given in the class
- Add/delete notes from the class in a full editable text

8. Add a teacher

A student should be a able to add a teacher to the database with the specifications detailed below:

- Name
- Department
- Tittle
- E-mail
- Contact Number
- Office Hours
- Office Location
- Website

9. Create assignments

A student should be a able to create a assignment with the specifications detailed below:

- Name
- Course
- Due Date
- Priority
- Description

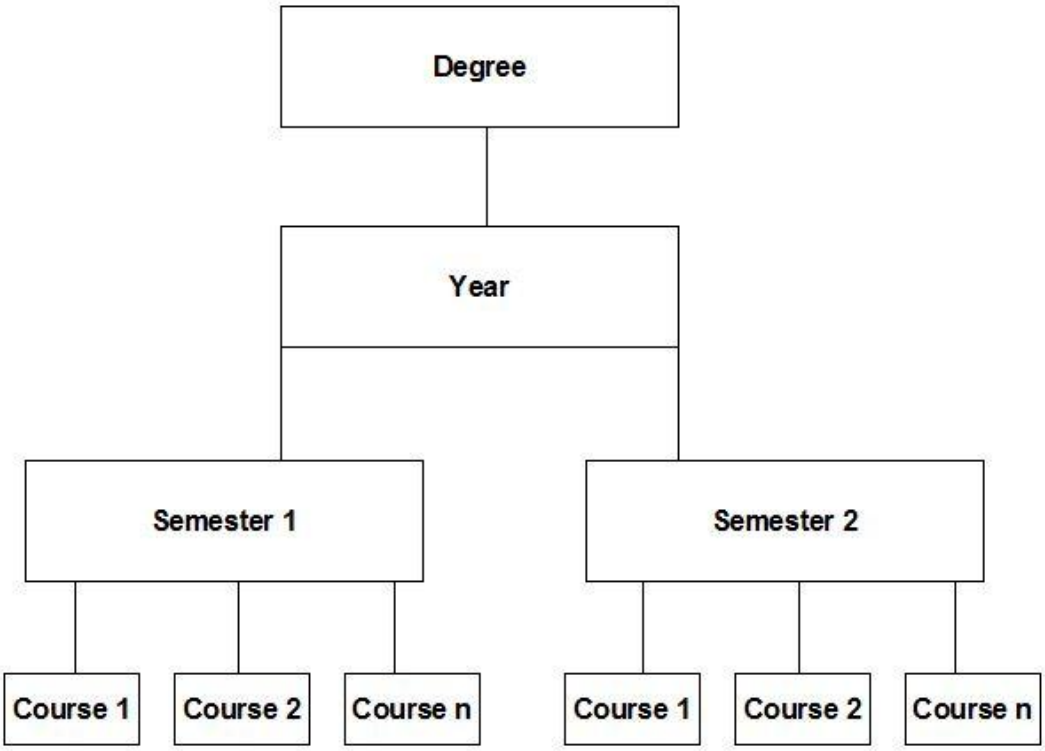
10. Give a grade

The software should be able to give information about the grades of the degree, including:

- Final grade
- Grade by year
- Grade by Semester
- Best Grade
- Worst Grade
- Other useful statistics

6.2.2 Structure

If we think about a regular degree (undergraduate or graduate at MSc level) we can easily reach a simple pattern, we have the degree which is divided by the number of years of the degree and each year is itself divided by semesters that have a finite number of courses. Putting it in simple figure we have what it can be seen in fig 6.2.



6.2 Degree Structure

Having established that, we can think about each item more specifically:

- The degree is as the name suggests the degree that the student is using the software to organize for

- The Semester represents a fixed amount of time where certain curses occur through a specific year of the degree.

- The Course represents, as also the name suggests, a specific course of the degree that occurs in a specific year and a specific semester.

- The assignments are specific works or to-do's that the student has to do in a specific course, it has a due date, a priority (low, medium and high) and when the student wants to be reminded of it (if at all)

- A schedule is a specific event (class, work, other) that repeats at a specific day of the week for a specific time frame (usually a semester) its associated to a course.

- A teacher is, as the name suggest, the lecturer of a certain course(s)

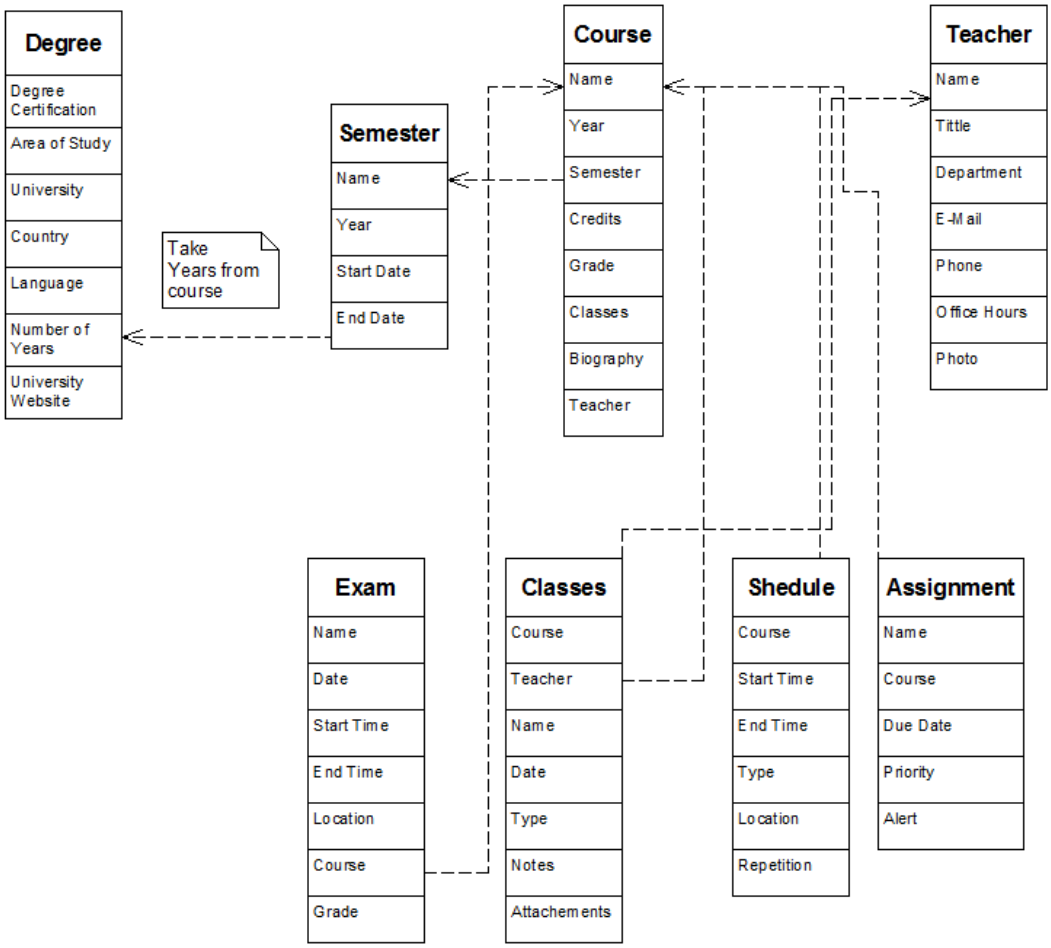
- The classes are the annotations the student wants to register of a specific course that occurred at a specific date. The student may also may save any kind of attachment (pdf, example, other) that is associated with that specific class.

- The Exams are evaluations that occur at a specific date; they are associated to a certain course and have the possibility of having a grade associated to them.

6.2.3. Class Diagram

A class diagram describes the types of classes in the system and the various kinds of static relationships that exist among them. Class diagrams also show the properties and operations of a class and the constraints that apply to the way objects are connected.[1]

Having determined, in the previous pages, the main objects needed for the applications we can now build a UML Class Diagram for the application (fig 6.3)

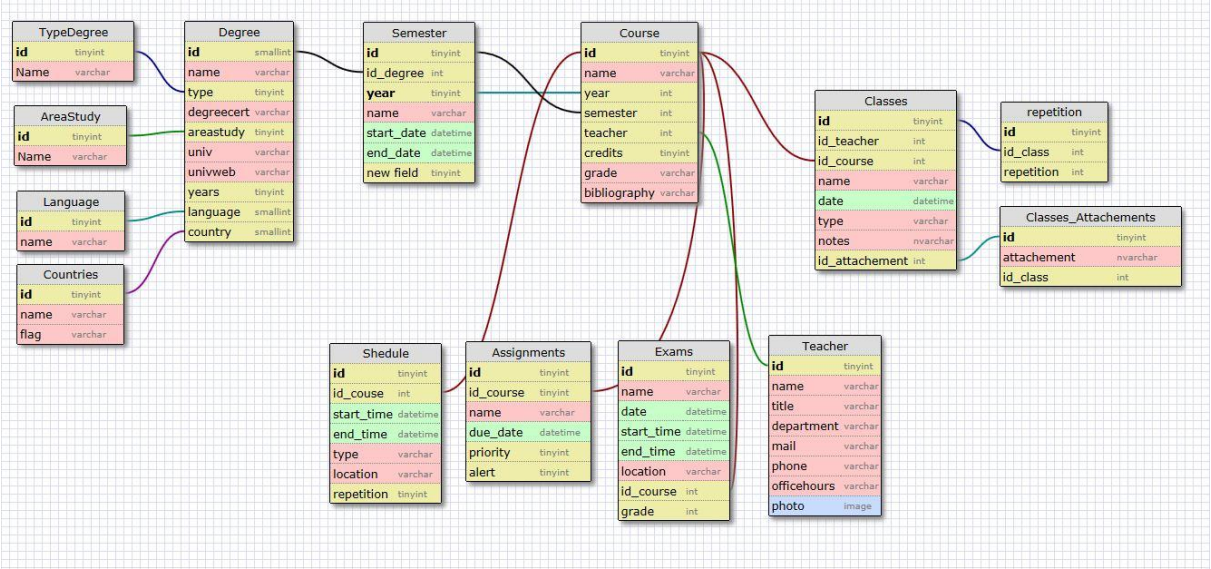


6.3 Class Diagram

6.3 Database Structure (fig 6.4)

A database is nothing more than a set of related information. A telephone book, for Example, is a database of the names, phone numbers, and addresses of all people living In a particular region. [11]

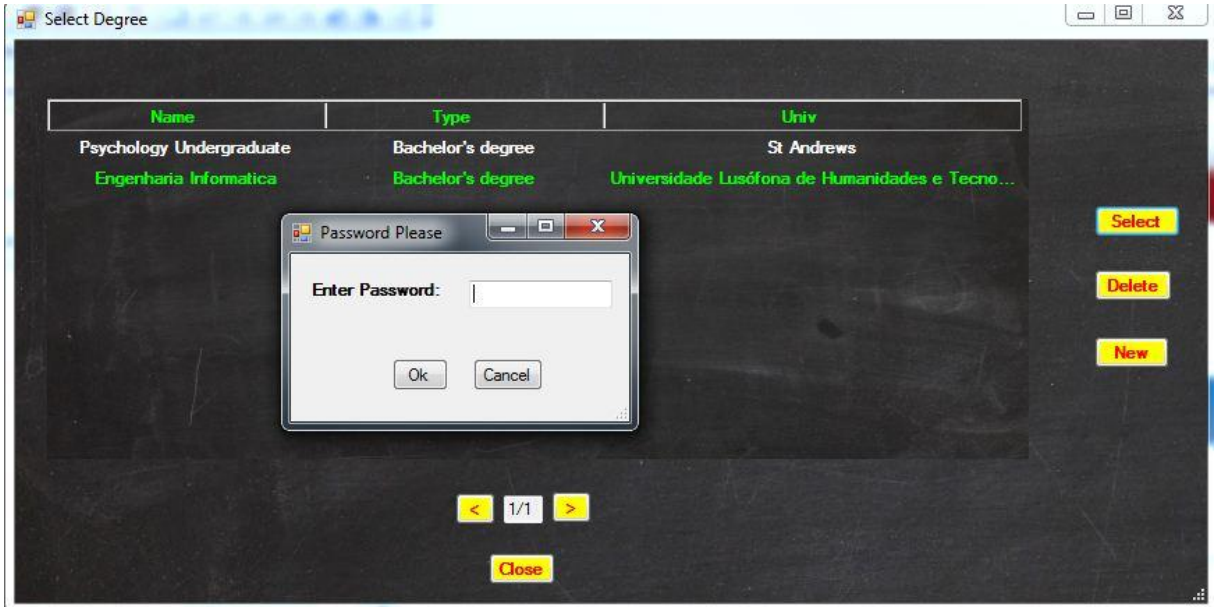
SQL stands for Structured Query Language. SQL is used to communicate with a database. It is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database.



6.4 database structure

7. Results

7.1 Application start-up, loading a degree

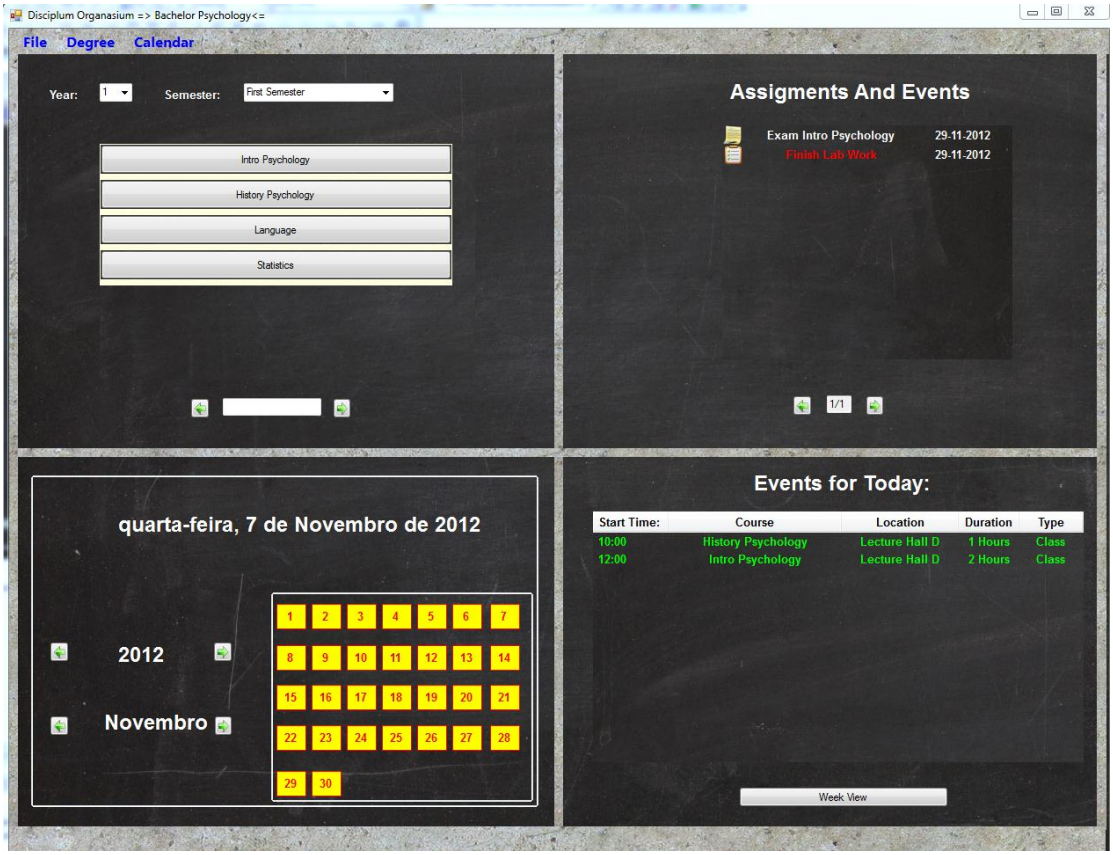


7.1 Degree Select

When the application starts, the user is presented with the degrees previously created (fig 7.1). He can select one by pressing the button "Select" and input the degree password.

He also has the possibility to delete a degree and to create a new one.

7.2 Main Screen (first screen that appear after loading a degree)



7.2 Main application screen

As it can be seen in figure 7.1, the main window of the application is divided in four Squares. The first square (top left corner) has the courses associated to the active degree, by default it shows the courses that are in the semester which start date and end date are within the current date of the system; but users can also filter by year, semester and name. Clicking on any course will bring the main window of that specific course, which will be discussed further along this report.

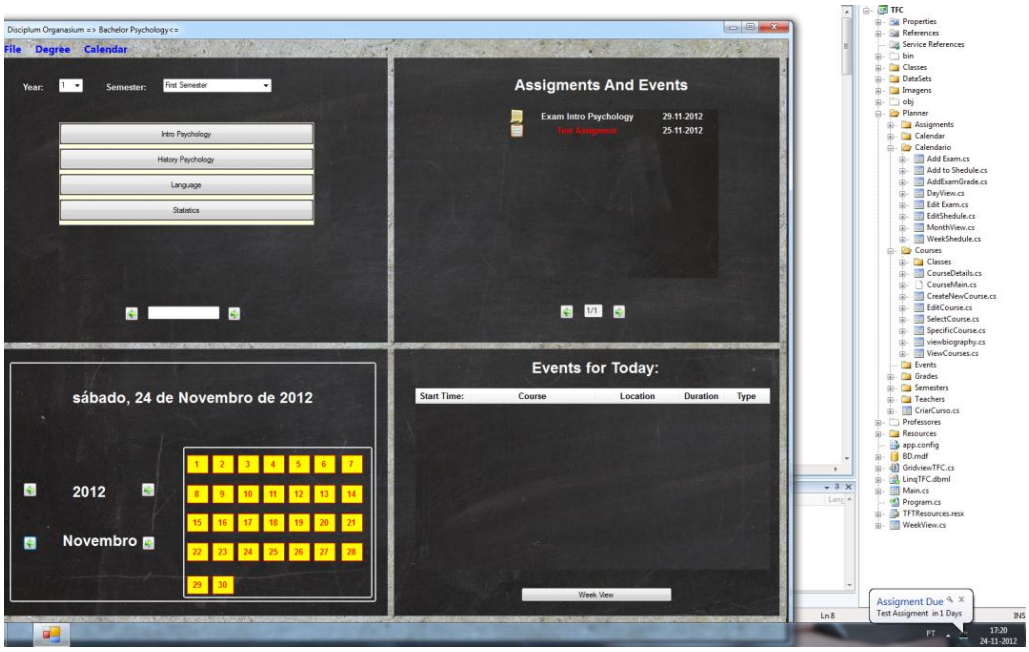
The second square (top right corner) shows any events (Exams or assignments) which date is further that the selected date, i.e. future events that could interest the student.

The third square (low left corner) allows the users to change the date, so they can check past and future events.

The last square (low right corner) shows events (Classes, assignments, exams and so on) for the selected day in chronologic order.

The Week view button will show the student the events for the next seven days (from the selected date) which will be discussed in more detail in the next pages.

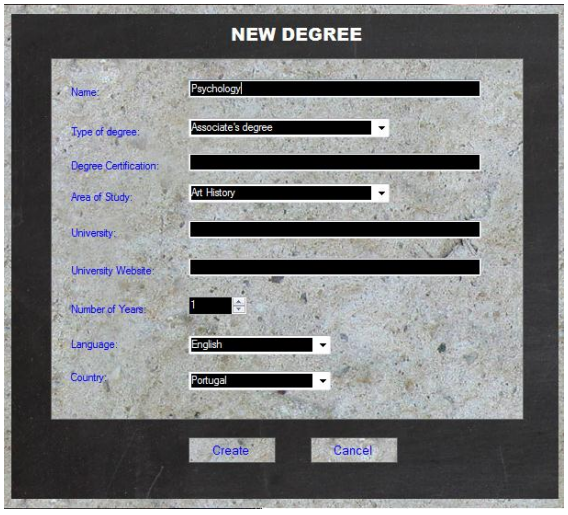
Last but not least, this area of the software is also responsible for alerting the student of an assignment. When creating an assignment a student has the option of choosing when he wish to be alerted (in due date, day before, two days before and so on) and when the time comes for the alert, it will be shown in the icon tray like any normal windows programs (fig 7.2).



7.3 Assignment alert

7.3 Create New Degree

The first thing a student needs to do to start using the software is to create a new course, this is done by going to the toolbar File -> New.

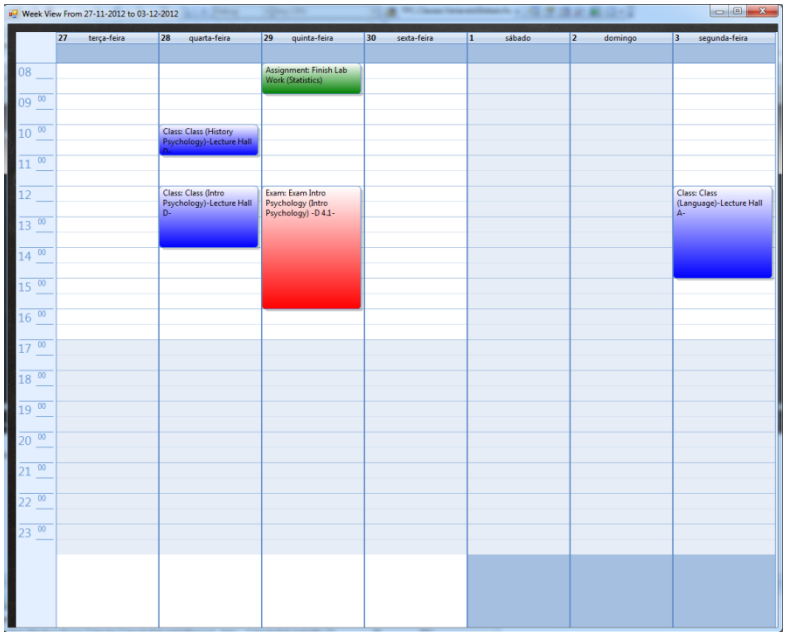


7.4 Create new degree

Here (fig 7.3) the student can type the name of the degree (cannot be blank), its type, certification, area of study, university, university website, number of years the degree consists, language the degree is lectured and the country.

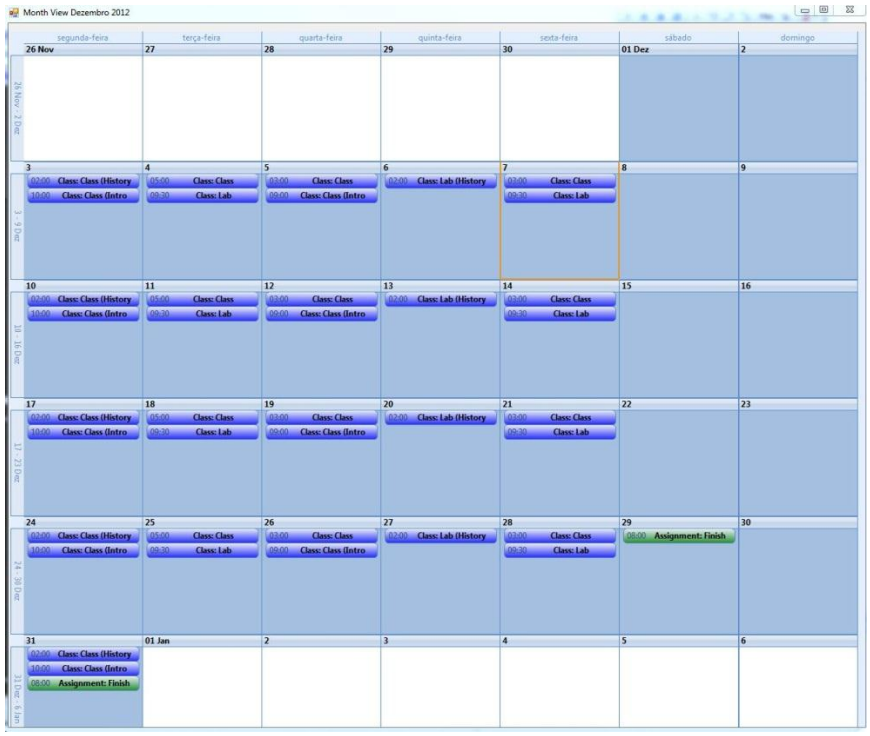
7.4 Calendar (week view and month view)

As it was mentioned earlier in this report, the student has the option to see his event in a week view mode (7 days from the selected day) (fig 7.4).



7.5 Calender week view

He also can see the full month view selecting the Calendar -> Month View in the respective toolbar (fig 7.5)



7.6 Calendar month view

7.5 Main Course Screen

When a specific course is selected in the main screen, the main screen of that specific course is presented (fig 7.6)

7.7 Main course screen

As the main application window, this form is divided in four main squares. The first one (top left) has the course information (Year, Semester, Biography credits and grade) and allows the user to set or update the Biography or grade by pressing the respective button).

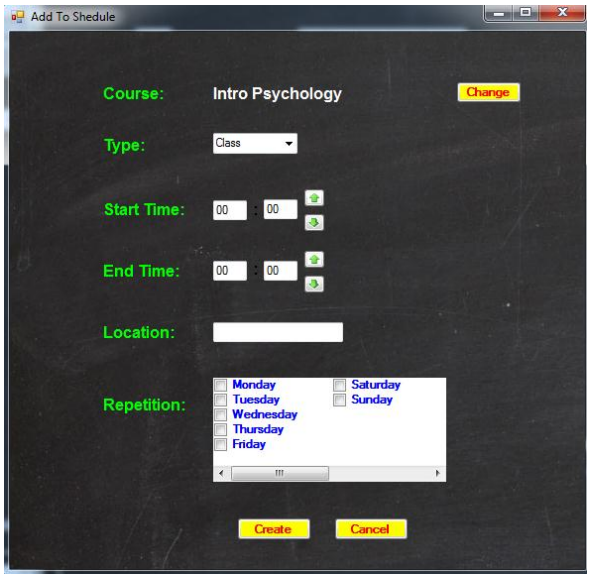
The second square (Top Right corner) has the information of the schedules associated with that specific course showing its information as it is shown in the picture. It allows the user to edit, delete and create a new schedule.

The third square (Low left corner) has the information of the Classes associated with the course. It allows the student to view, edit, delete and create a new class.

The last square (Low right corner) has the exams associated to the course, showing its date, location and grade. It allows the user to change the grade, edit, delete and create a new exam.

7.6 Create a new Schedule

By selecting the option to create a new schedule (via the course button or the toolbar) the student is presented with the following screen (figure 7.7)

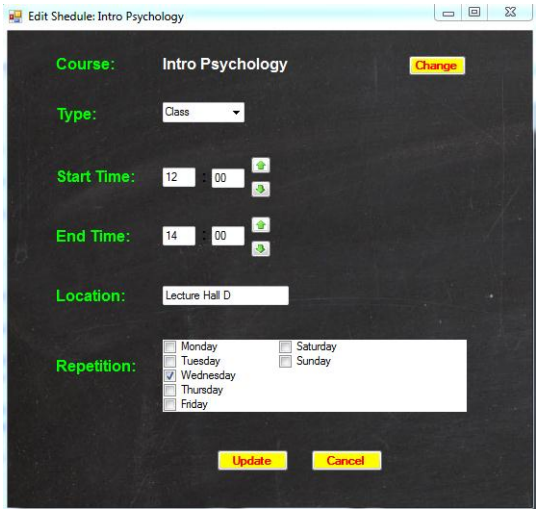


7.8 Ceate new schedule

Here the student has the option to select the course which the schedule is referring to: the type (class, lab and so on), the start time, end time, the location and which day(s) of the week it repeats in the corresponding semester in which the course is allocated.

It has three validations: a course has to be selected (cannot be null), the end time must be after the start time and a day must be selected.

7.7 Edit Schedule

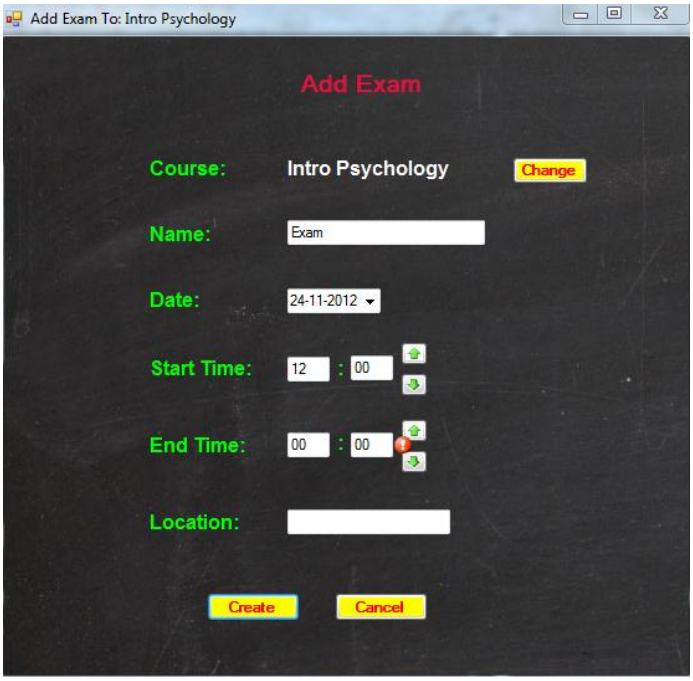


7.9 Edit schedule

The student also has also the ability to edit a schedule, changing the entire fields that were discussed previously. (fig 7.8)

7.8 Create a new exam

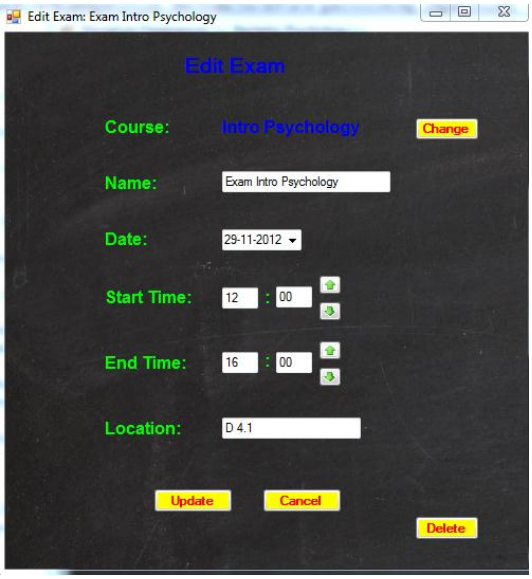
By selection the option to create a new exam (by the course button or the toolbar) the student is presented with the following screen (fig 7.9)



7.10 Create exam

Here the student has the option to select which course the exam refers to, the name of the exam (default is "Exam"), the date, the start time, end time and location. The form validations are three: the exam must have a course select (cannot be null), the end date must be higher than the start time and the name cannot be null.

7.9 Edit Exam

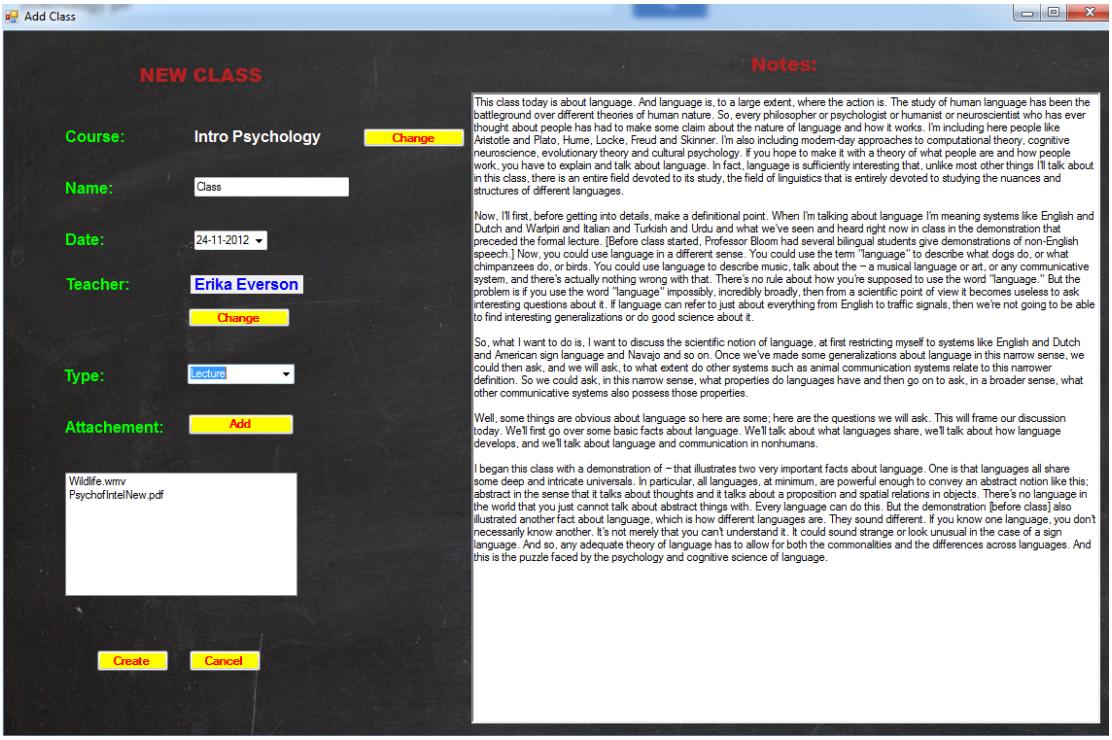


7.11 Edit exam

As with the Schedule, the student can also edit each exam changing the fields that were previously discussed. (Fig 7.10)

7.10 Create a New Class

By selection the option to create a new Class (by the course button or the toolbar) the student is presented with the following screen (fig 7.11)



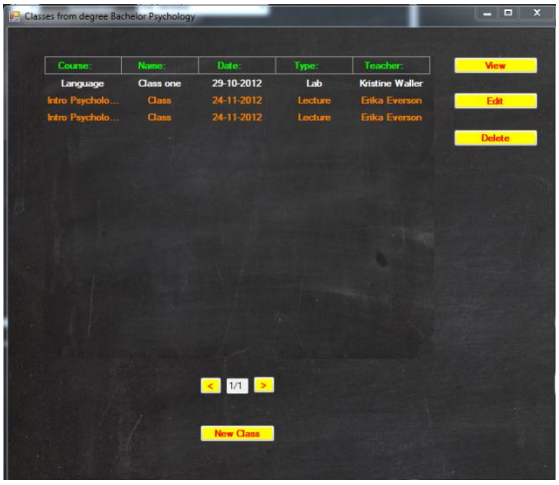
7.12 Create new class

Here the student has the option to create a Class from a specific course in a specific date. He can change course (cannot be null), change teacher (also cannot be null), specify a name (default is "Class"), select a date and a type (Lecture, Lab and so on) and he can add any type off attachment he feels is relevant for that specific class. He can also add the annotations he wishes for that specific class.

Besides the validations already mentioned the name cannot be blank.

7.11 View Class

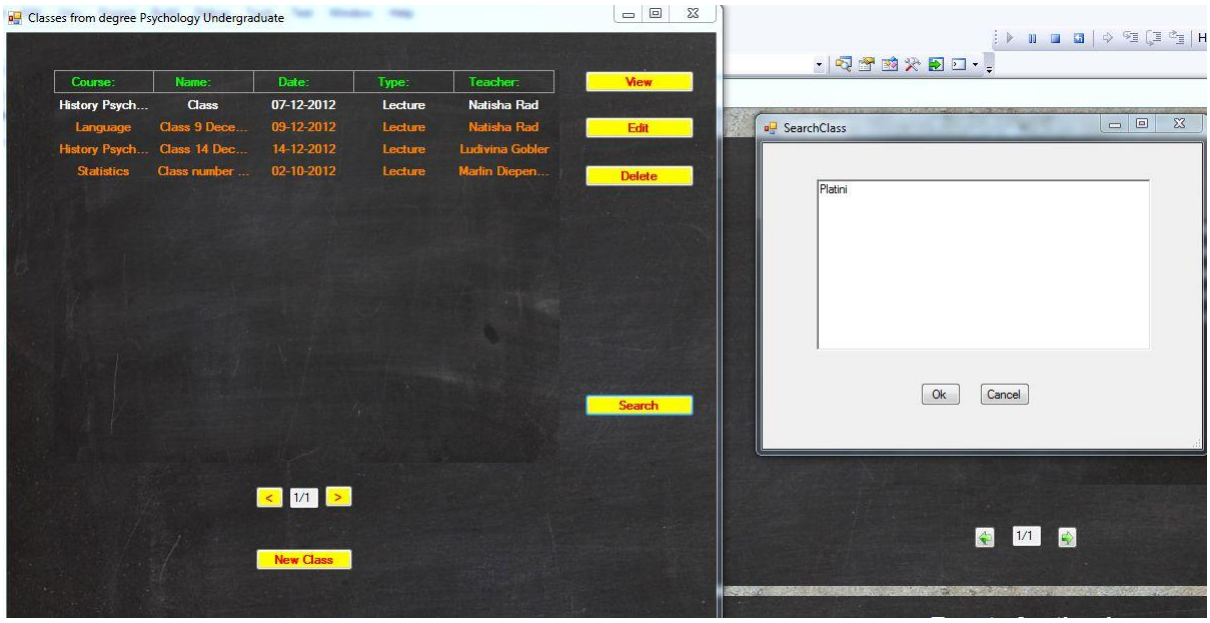
By selecting in the toolbar Degree -> Classes -> View Classes the student can view all classes created (fig 7.12)



7.13 View classes

Here (fig 7.12) the student can view, edit, delete and create a new class

7.12 Search Class



7.14 Search Class

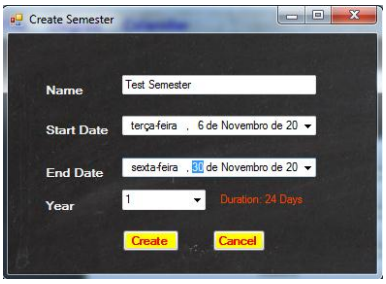
Users can also search a class by a key-phrase (fig 7.14), pressing the search button and inputting a search string.

7.13 Edit Class

Like all the events shown previously, the student can edit a specific class the fields that were previously discussed.

7.14 New Semester

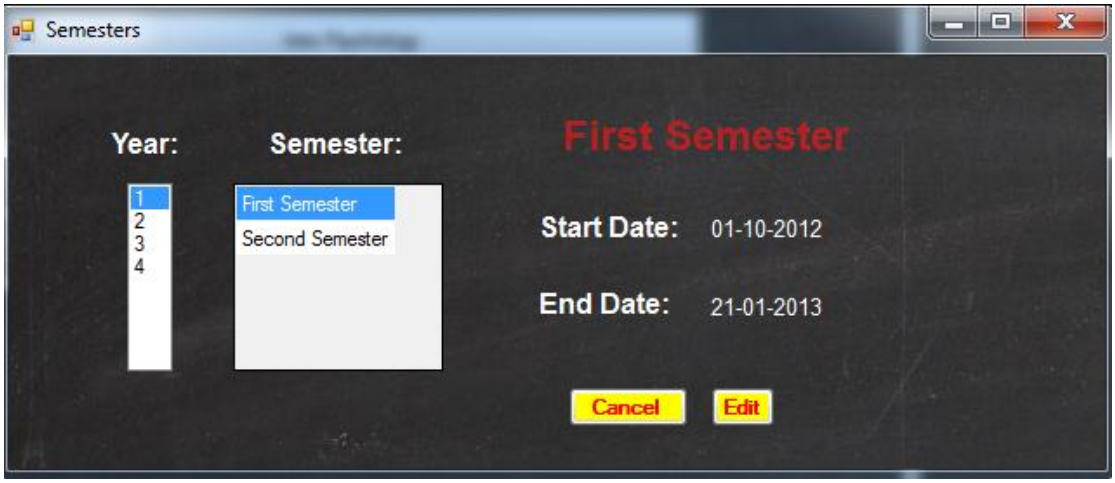
Every Degree has a semester, so before the user can create courses he must first create the semester which a course will be associated:



7.15 Create Semester

Here (fig 7.13) the student has the option to create a semester: he can choose a name (cannot be blank) a start date and an end date (end date must be higher than the start date). He can also select which year of the degree created the semester belongs to.

7.15 View Semester

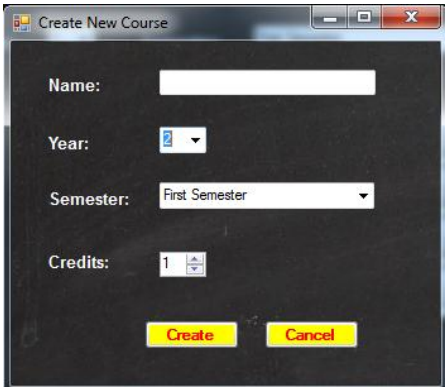


7.16 View semester

By going to the toolbar Degree -> Semester -> View Semester (fig 7.14) the students have the ability to see the semesters previously created (filtered by year). Here they can choose to edit the semester to change the fields that was previously discussed.

7.16 New Course

The heart of a degree is obviously the courses, so before the application can start to be useful, a course must be created:



7.17 New course

As illustrated in figure 7.15 the student have the option to create a course, he can select a name (cannot be blank), a year and a semester (the selected year must have a semester associated) and the number of credits the course is worth.

7.17 View Course

By going to the toolbar Degree -> Courses-> View Courses the students have the ability to see all the courses associated to the Degree:

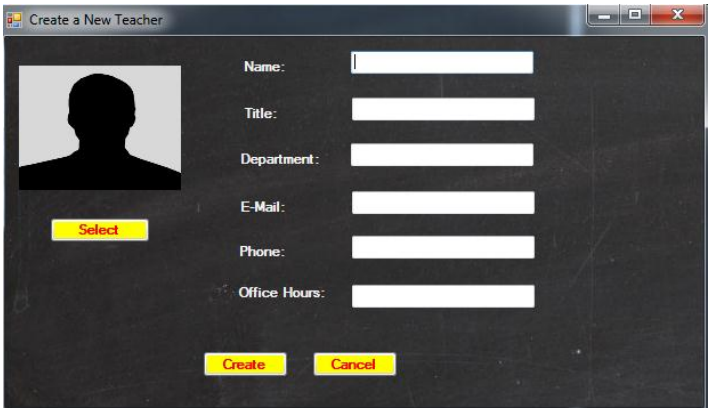
7.18 View current courses

Here (fig 7.16) they can filter a course by its year and semester, when a course is selected. Besides the course information they can:

1. Associate / change course teacher
2. View / create / edit a schedule
3. View / create / change Grade of exams
4. Open classes associated with the course (detailed further in the report)
5. Open assignments associated with the course (detailed further in the report)
6. Edit the course

7.18 Create New Teacher

Every Course has at least one Teacher, so the student has the ability to create a new teacher and associate him or her to one or more courses:



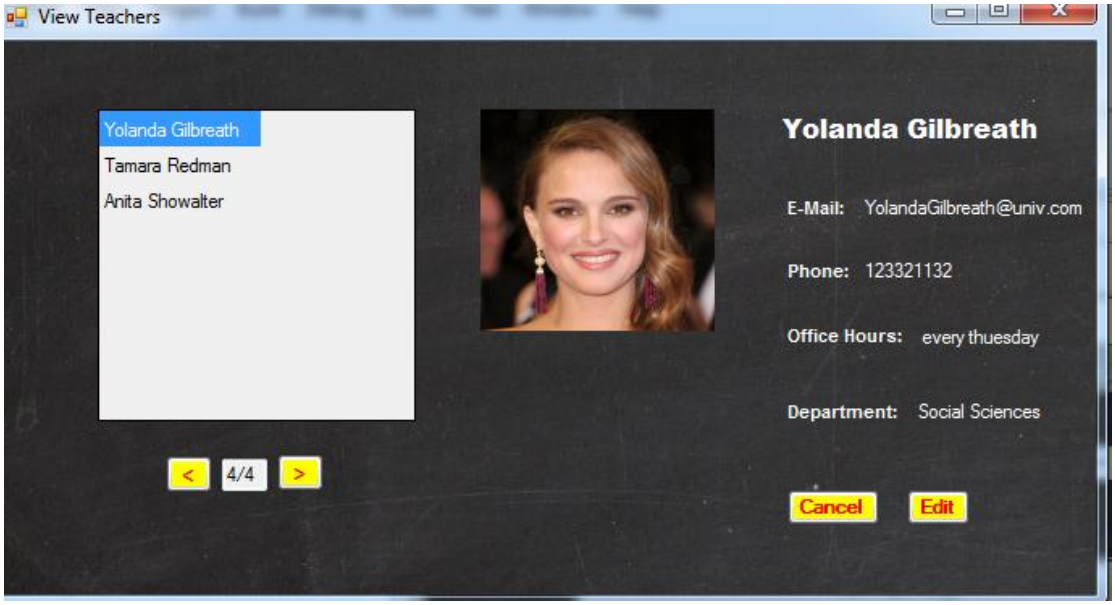
7.19 Create new teacher

Here (fig 7.17) the student can select a name, tittle, department, E-mail, phone, office hours and a photo to the teacher.

The name cannot be blank, the e-mail must have a valid format and the photo must be from a valid picture format (jpeg, bmp and so on)

7.19 View Teacher's

By going to the toolbar degree -> teachers-> view teachers the students have the ability to see all the Teachers previously created.

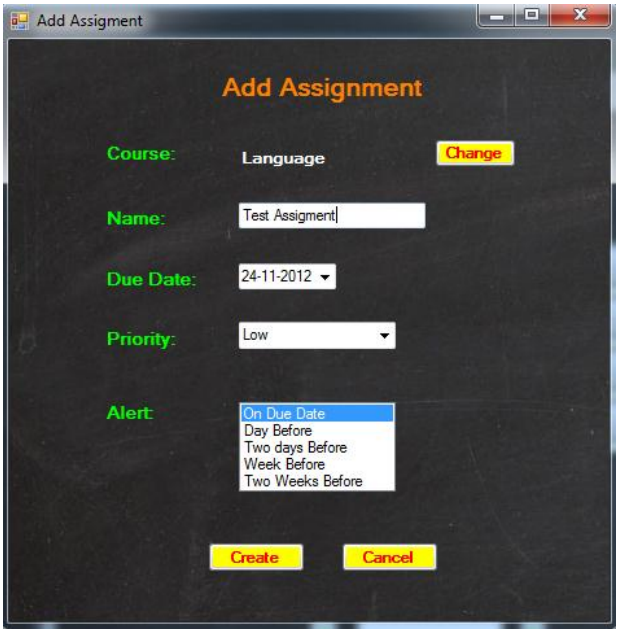


7.20 View Teachers

Here (fig 7.18) they can not only see all the teachers previously created along with their information but also edit their information.

7.20 Create New Assignment

By selection the option to create a new Assignment (via the course button or the toolbar) the student is presented with the following screen (fig 7.19)



7.21 Add assigment

Here the students have the ability so select the course of the assignment (cannot be null), the name (cannot be null) a due date, a priority (low, medium, High) and the time he wishes to be alerted by the application for that particular assignment.

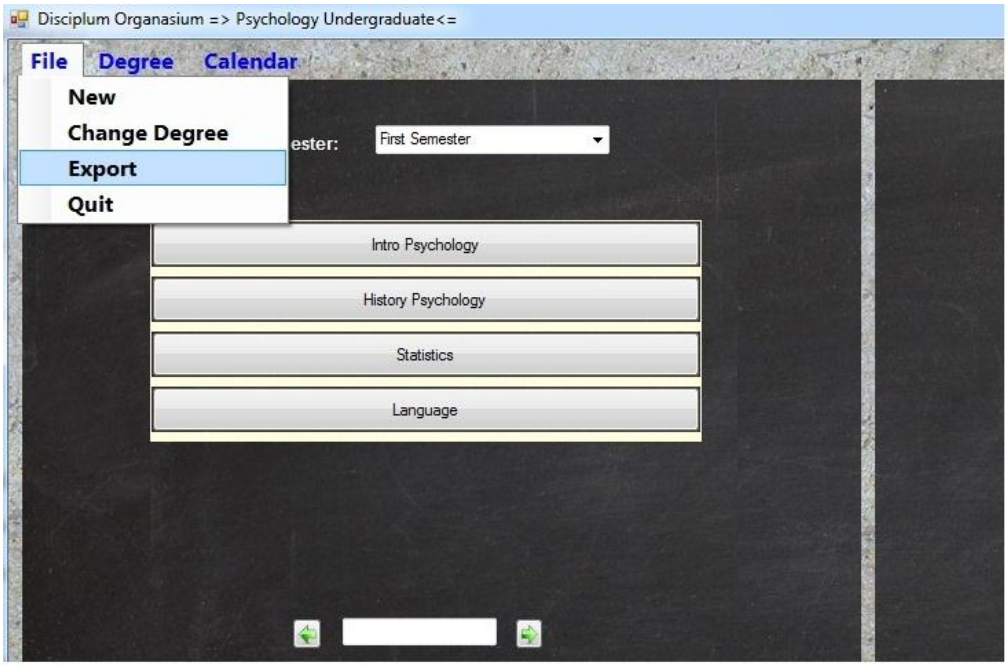
7.21 View Assignment's

By selecting in the toolbar degree -> assignments -> view assignments the student can view all assignments created (fig 7.20)

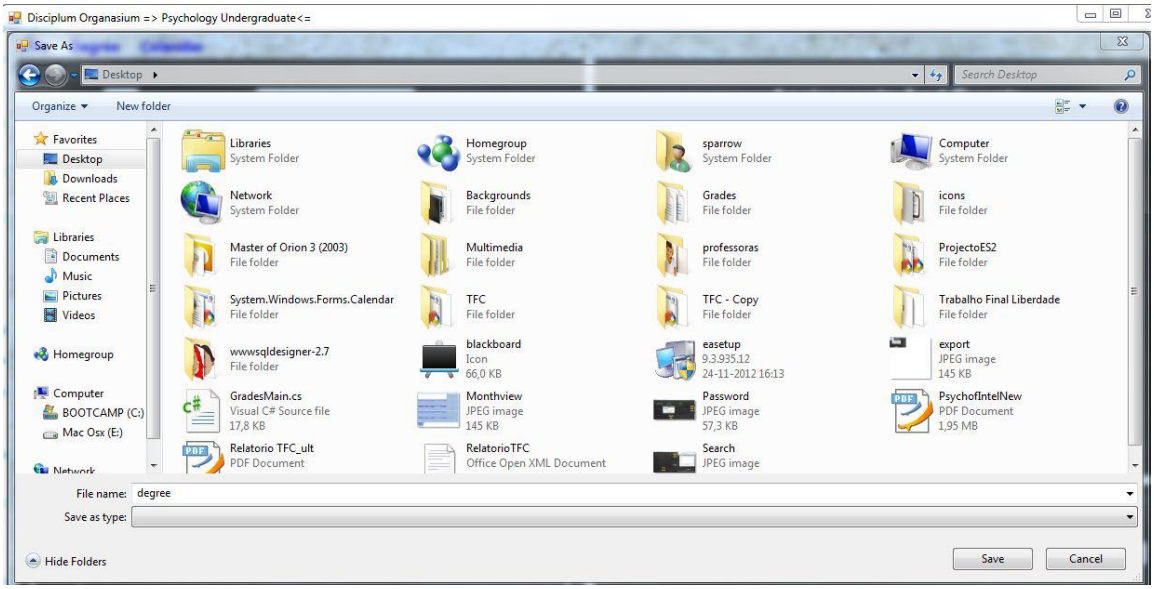
7.22 View assignments

Here (fig 7.20) the student can see all assignments created and he can edit, delete and create a new assignment.

7.22 Export Degree



7.23 Export Degree I



7.24 Export Degree II

Users can also export the course to a text file, by going the toolbar File-> export and selecting a folder and a name to export the degree.

7.23 Grades

The students also have the ability to see his or hers grades as overall being going the toolbar -> grades (fig 7.21)

7.25Grades form

Here (fig 7.21) the student can filter his grades; he can filter by year or semester or see them all together, the program then give him some statistics about the selected courses grades, the overall Grade, the median, the spread, the best and worse grade.

He also has some graphs to help him analyse his grade.

8. Conclusion and future work

At the inception of this project the aim was to build a software that could help students organize their degrees.

The project begun by doing a requirement analysis of the needs of students. Analysing what kind of tool could satisfy them and how such a tool should be built. UML was used to model it.

Afterwards the technology question arose: which programming language should be used and why ? in the end it was decided to use C#.net™ and SQL express™ in Microsoft Windows™.

Finally the time was allotted to build the solution, correcting bugs and testing the application.

It was never the aim of the project to build a complete solution, as the time an manpower allotted to the project was too short, but to build a software easy to use and appealing, that a student could use to organize his degree, without wasting too much time trying to understand how it works, and that, we believe, was achieved.

For future work, there are a myriad of ideas that can be implement in software of this kind, some of the more important are:

1. Grade Internationalization

The software is prepared for Portuguese grade system, with grades going from 1 to 20 but this is not a universal system, the English system for example is from A to F, and more very different systems exist through the world. One important improvement would then be to change the grading system to match the country where the degree is taken.

2. Mobile Version

In the last five years the computer Industry has changed its shift dramatically, with the mobile phone and tablets market predicted to outsell the traditional computer market as soon as next year [10]

For this reason one important step is to make a mobile version of the software (iOS, Android or both) that can sync automatically with its Desktop version so that the student can always have the latest information available.

3. Flash Cards

Using flashcards develops and improves retention of information and improves the connection of synapses. A flashcard's obverse side would usually contain a singular image of an object, organism, letter, word, or number depending on its use. Such minimalism in design allows a person to focus on the card's image when it is flashed. [22]

Therefore developing a Flash Card feature would help students memorize information faster, making their study time more efficient.

4. Integrated Recorder

Developing a feature that can record a class while he is taking notes would ensure that the student don't miss any important information. It would also allow the student to pay more attention to the classes not worrying about the necessity of taking notes.

References

[1] O' Docherly ,Mike – Object-Oriented analysis & Design, John Wiley & Sins, Ltd

[2] B.Dunaway, Robert – The Book of Visual Studio .Net, No Starch Press, 2002

[3] Skeet , Jon – c# in depth, Manning Publications 2008

[4] Powers, Lars; Snell, Mike – Microsoft Visual Studio 2008 Unleashed, Sams, 2008

[5] Cottrel, Stella – The study Skills Handbook, palgrave macmillan, 2008

[6] Cottrel, Stella – The exam Skills Handbook, palgrave macmillan, 2008

[7][Online].Available: http://istudentpro.com/.[Acessed November 2012]

[8][Online].Available: http://ihomework.element84.com/Mac.html/.[Acessed November 2012]

[9][Online].Available:http://www.moonglowsoftware.com/studentorganizer.html/.[Acessed November 2012]

[10][Online].Available:http://www.techspot.com/news/49169-microsoft-says-tablets-will-outsell-desktop-pcs-next-year.html/.[Acessed November 2012]

[11] Bealieu , Alan – Learning SQL, O'Reilly, 2009

[12]Aurelius, Marcs – Meditations, Penguin Classics, 2006

[13][Online].Available: http://www.apple.com/osx/apps/app-store.html/.[Acessed November 2012]

[14][Online].Available: http://www.windowsstore.com/.[Acessed November 2012]

[15][Online].Available: http://office.microsoft.com/en-us/word/.[Acessed November 2012]

[16][Online].Available: http://www.literatureandlatte.com/scrivener.php/.[Acessed November 2012]

[17][Online].Available: http://office.microsoft.com/pt-pt/onenote/.[Acessed November 2012]

[18][Online].Available: http://evernote.com/.[Acessed November 2012]

[19][Online].Available: http://www.apple.com/ios/.[Acessed November 2012]

[20][Online].Available: http://www.android.com/.[Acessed November 2012]

[21][Online].Available:http://www.tabletpcreview.com/default.asp?newsID=1543&review=Best+Student+Organizer+and+Notetaking+Apps+for+the+iPad/.[Accessed November 2012]

[22][Online].Available: http://EzineArticles.com/6288984/.[Accessed November 2012]