UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

VIRTUAL TUTOR: INFORMATION RETRIEVAL IN MOODLE AND PARAMETRIZATION VIA A BACKOFFICE APPLICATION

Ricardo Jorge Correia Costa

Mestrado em Engenharia Informática
Especialização em Arquitetura, Sistemas e Redes de Computadores

Trabalho de projeto orientado por:
Prof.ª Doutora Ana Paula Boler Cláudio
Prof.ª Doutora Maria Beatriz Duarte Pereira do Carmo

2018

Acknowledgments

I would first like to thank my thesis advisors doctor professor Ana Paula Boler Cláudio and doctor professor Maria Beatriz Duarte Pereira do Carmo for allowing me to be a part of this project, for motivating me to achieve a better outcome and the integral part they had in the development of the project.

To the professors at Universidade Aberta for the help provided during the project, with an emphasis on the help provided by doctor professor Vítor Rocio.

I would like to thank the volunteers of the performed tests, the data received was invaluable, thank you for the time taken to perform the tests.

To my colleagues from Faculdade de Ciências da Universidade de Lisboa and my friends that accompanied me for the duration of the Bachelor and Masters degree, most notably Paulo Antunes, Inês Gouveia, Joaquim Afonso, Henrique Mendes, Ion Cornei, Tomás Peixinho, Telmo Santos, Tiago Santos and Miguel Falé for everything they shared or gave me, every helping hand and every listening ear meant the world to me. I would like to thank especially my project colleague Catarina Alves for being an amazing colleague and providing amazing moments that made the experience of working in the project unforgettable.

I would like to thank my family and my girlfriend for the endless support and encouragement provided every day throughout the years that gave me the ability and motivation to achieve my goals and finish writing this thesis. Without their support, this feat would not have been possible. Thank you.

This work is supported by UID/MULTI/04046/2013 centre grant from FCT, Portugal (to BioISI - Biosystems & Integrative Sciences Institute) and by PTDC/IVC-PEC/3963/2014 project grant (VIRTUAL TUTORING – the virtual tutor as learning mediating artefact in online university education) also from FCT. I would like to thank everyone that works at these institutions for enabling me to work on this project.

Resumo

As plataformas de e-learning como o Moodle são sistemas *online* que permitem a criação de unidades curriculares para armazenar e mostrar uma vasta quantidade de material que auxilia no período de aprendizagem e criar uma ponte de interação entre os alunos e o professor. Alunos que previamente não tinham possibilidade de aprenderem, com estes recursos têm a possibilidade de entrar em cursos e aprender conteúdos novos sem a presença física em aulas. Estes benefícios levam a que uma quantidade sempre crescente de estudantes adquira interesse em cursos *online*. Contudo, estes modelos de aprendizagem dependem de uma quantidade significativa de informação que tem de estar na página da unidade curricular. A quantidade de informação combinada com a possível baixa frequência de acesso à página da unidade curricular pode sobrecarregar um aluno.

Com o número de alunos inscritos a aumentar e tendência de substituir actividades presenciais com versões presentes *online*, torna a capacidade de um professor supervisionar e aconselhar todos os alunos cada vez mais difícil, o que leva a que alunos se possam sentir sozinhos e sem acompanhamento, criando situações em que o seu desempenho é inferior ao desejado ou que, no limite, desistem.

Este trabalho insere-se no projeto de investigação "TUTORIA VIRTUAL – o tutor virtual artefacto mediador de aprendizagem no ensino superior online", financiado pela FCT, com referência PTDC/IVC-PEC/3963/2014 que envolve investigadores de 3 instituições: a Universidade Aberta de Lisboa, o Instituto Superior Técnico de Lisboa e a Faculdade de Ciências da Universidade de Lisboa.

Neste trabalho desenvolveu-se uma solução tecnológica na forma de uma aplicação WebGL com um tutor virtual antropomórfico implementada na plataforma Moodle que fornece suporte síncrono ao aluno como complemento a experiência de aprendizagem. A aplicação designada Virtual tutor application, tem como objectivo criar uma relação de empatia com cada aluno que a utiliza durante o período de duração do curso usando texto e expressões faciais do avatar, motivando-o a ter um melhor desempenho e a não desistir. A aplicação foi desenvolvida usando a plataforma Unity e envolve dois componentes, o componente de interface com o nome InterfaceManager (desenvolvido no contexto de outra tese) e o componente de aquisição e organização de dados, WebManager, que é descrito nesta tese.

Também foi desenvolvido neste projeto, uma aplicação Android que recorre a avatares cujo objectivo é realizar tarefas de acompanhamento e aconselhamento ao longo da experiência de aprendizagem do aluno em todas as unidades curriculares, o componente descrito nesta tese embora não esteja ainda integrado nesta aplicação, possui a capacidade de distinguir a plataforma (móvel ou WebGl) e operar da forma desejada, no caso da aplicação Android requisitar informação sobre todas as unidades curriculares e na aplicação localizada no Moodle requisitar a unidade curricular com base no identificador fornecido.

O componente WebManager foi desenvolvido na forma de um plugin que pode ser incorporado em qualquer aplicação Unity com relativa facilidade e possui a capacidade de comunicar com o Moodle e uma base de dados externa através de pedidos web. Deste modo é possível adquirir e guardar informação sobre o aluno e a unidade curricular crucial para a interação com o aluno.

A aquisição desta informação permite que o componente InterfaceManager tenha a capacidade de mostrar ao aluno informações sobre a estrutura da unidade curricular e os seus componentes, sobre as novidades na página desde o seu último acesso sem que o aluno seja sujeito a navegar a vasta informação disposta na página da unidade curricular e apresentar conselhos e indicações com base no estado do aluno na unidade curricular, este estado é calculado com base no desempenho acadêmico na unidade curricular, assiduidade na página e as interações na mesma, que, no caso de um curso online funciona como uma turma virtual. A expressão de conselhos e indicações é feito com base em frases criadas com o auxílio da equipa pedagógica da Universidade Aberta baseadas na atividade do aluno com o propósito de melhorar a experiência de aprendizagem na duração do curso.

O plugin é capaz de adquirir informação relativa à unidade curricular e ao aluno através do uso de serviços web nativos à plataforma Moodle e a serviços web desenvolvidos no contexto desta tese localizados no servidor que contém a base de dados externa, os serviços web desenvolvidos extraem informação de uma combinação de tabelas da base de dados externa que contém uma réplica da tabela de logs do servidor Moodle. A informação requisitada pelo plugin envolve, mas não está limitado a: notas do aluno, que materiais acedeu na unidade curricular, quantos dias já acedeu a unidade curricular ou quantos posts já fez sem a necessidade de criação de componentes adicionais no Moodle. O uso de serviços web existentes no Moodle permite o uso do plugin com versões base do Moodle desde que sejam versões iguais ou superiores a presente no projeto devido a existência dos serviços web está dependente da versão.

A comunicação com a base de dados externa é realizada recorrendo ao uso de serviços web contidos em ficheiros php para oferecer um intermediário capaz de interpretar as variáveis fornecidas pela aplicação e fornecer informação com base

nessas variáveis, estes ficheiros removem a necessidade de fornecer informação sensível ao servidor como credenciais para aceder à base de dados e permitem a adição de passos para filtrar e autenticar as variáveis como medidas de segurança.

Complementando o Webmanager, uma aplicação Back Office foi desenvolvida em Unity que permite ao professor da unidade curricular definir e guardar parâmetros na base de dados externa, esses parâmetros são depois adquiridos pelo componente WebManager e usados como base dos cálculos de avaliação do estado do aluno que irão ajudar a parametrizar o empenho posto pelo aluno e informar o InterfaceManager das falas e informação para definir as expressões faciais que o avatar deve usar, levando a uma melhor experiência de aprendizagem dos alunos.

Para garantir o funcionamento correto durante a duração de uma unidade curricular, testes de operação foram realizados pela equipa da Faculdade de Ciências da Universidade de Lisboa em unidades curriculares criadas especificamente para o efeito na duração de dez dias com contas de alunos teste desenvolvidas para este propósito divididas entre todos os participantes. Durante este período de tempo, os participantes seguiram as ações definidas nas tabelas presentes no anexo que envolveram a entrega de componentes de avaliação e interações nos fóruns, permitindo assim verificar as possibilidades consideradas.

Pela realização de testes de carga ao servidor usando uma aplicação desenvolvida em Python para identificar objectos no ecrã e contar o tempo entre ciclos, foi possível encontrar uma desvantagem associada com o uso do plugin: O tempo significativo que demora a aquisição da informação desejada do Moodle e da base de dados externa. A causa principal para a existência deste problema deve-se à quantidade de pedidos que são necessários para obter a informação e a necessidade de ordenar determinados pedidos. Este tempo está também dependente do tempo de resposta da plataforma, combinando este tempo com a organização forçada certos pedidos com a limitação da plataforma WebGL que não tem acesso a sockets para uma comunicação mais rápida impede a implementação de várias alternativas com o objectivo de melhorar o desempenho do componente.

A acompanhar as limitações de desempenho do componente WebManager, a aplicação reside num bloco HTML na página da unidade curricular e a sua inicialização e operações associadas a abertura e carregamento da página. Esta associação causa restrições à aplicação como não estar acessível em páginas que não contém o referido bloco e ser inicializado sempre que a página é carregada, isto é, não é possível manter um estado facilmente adquirido sempre que muda a página forçando a aplicação a requisitar e organizar toda a informação de novo e realizar os cálculos outra vez.

Testes com professores foram realizados para obter uma opinião de alguém experiente no ensino online e confortável com o uso da plataforma Moodle. Nestes testes

foi explicado previamente em que consistia o projeto e foi mostrado um tutorial em vídeo, após a visualização do vídeo os professores interagiram com a aplicação numa unidade curricular de testes usando contas de utilizadores com estados de alunos diferentes. Após realizarem tarefas básicas foram convidados a responder a um questionário. Com base nos questionários realizados foi possível concluir que a maioria destes professores concordaram que as capacidades fornecidas pela aplicação, como o acesso mais rápido ao material de ensino na página da unidade curricular são uma vantajosas para o ensino. Contudo muitos afirmam que é necessário melhorar o desempenho em termos do tempo de carregamento da informação antes que se torne viável a sua utilização em unidades curriculares reais.

Palavras-chave: Tutoria virtual, Moodle, Tutor Virtual , Tutor online, Sistemas de tutoria inteligentes

Abstract

With the ever-increasing number of students taking online courses, combined with the overbearing amount of information present that overwhelms students, a large group of students become demoralised and can, ultimately, decide to give up on the course.

This work is funded by the research project "VIRTUAL TUTORING - the virtual tutor as learning mediating artefact in online university education", reference PTDC/IVC-PEC/3963/2014 which involves researchers from three institutions: Universidade Aberta de Lisboa, the Instituto Superior Técnico de Lisboa and Faculdade de Ciências da Universidade de Lisboa.

In this thesis, we present a complement to the learning experience with the goal of accompanying the student: an application with an anthropomorphic virtual tutor(avatar) implemented in the Moodle environment that can provide synchronous support to the student, support is done in the form of text presented in the application and the facial expressions.

The Virtual Tutor informs the student of changes in the course page without the need to check the vast information in the page, offer feedback to the performance of the student and create an empathy relationship over the course duration to improve the learning experience while motivating the student to continue working towards a good grade in the course. The solution involves two major components: an interface component that displays information alongside the avatar to the student and an information storage and retrieval component, by the name WebManager, that gathers information from Moodle and from an external database and provide it to the interface component. Tests with professors were made to receive the opinion of someone experienced in online teaching.

A Back Office application was developed to enable a professor to define specific parameters to adapt the behaviour of the Virtual Tutor to a specific course, leading to an experience more aligned with the goals of the specific for the students.

Keywords: Virtual tutoring, Moodle, Virtual Tutor, Online tutor, Intelligent Tutoring Systems

Contents

List of Figures

List of Tables

Abreviations

AI Artificial Intelligence. 8

GB GigaByte. 53

HTML Hypertext Markup Language. 68

ICGI International Conference on Graphics and Interaction. 2

ID Identifier. 21

IP Internet Protocol. 16, 20

ITS Intelligent Tutoring System. 9

JSON JavaScript Object Notation. 34

LMS Learning Management System. 5, 12

Moodle Modular object-oriented dynamic learning environment. 1–3, 5–12, 15–26, 34, 39, 44, 48, 50, 53, 56, 57, 65, 67–70

MyST My Science Tutor. 9

PHP PHP: Hypertext Preprocessor. 6, 20, 30, 34, 49

RAM Random-Access Memory. 53

SQL Structured Query Language. 20, 49

VCAT Virtual Cultural Awareness Trainer. 8

VLE Virtual Learning Environment. 11

VT Virtual Tutor. 10

WebGL Web Graphics Library. 16, 18, 20, 21, 46, 65, 67

Chapter 1

Introduction

Learning management systems like Moodle are popular platforms to host pages that contain a vast volume of study material for a specific subject or course. This type of systems allows various kinds of students to complete courses that were previously impossible due to constraints, enabling students to acquire more knowledge.

However, this model relies heavily on a significant amount of information being available on Moodle, that coupled with the possible reduced frequency of login from the student, can be overwhelming, this can demoralise the student and possibly lead to the student ultimately dropping the course.

This thesis fits into the investigation project "TUTORIA VIRTUAL – o tutor virtual artefacto mediador de aprendizagem no ensino superior online" that has the goal of developing a Virtual Tutor application capable of accompanying the student during the course duration. In this thesis, in particular, the primary goal was to create a technological solution capable of retrieving and storing information from Moodle with the purpose of providing a Virtual Tutor application (that provides an avatar playing the role of tutor in the course) information necessary to accompany the student during the course duration, enhancing the learning experience, informing the student of new content and supporting the learning experience.

Additionally, to improve the versatility of the Virtual Tutor application, a back office application was developed to give the professor of the course the ability to define specific parameters of the Virtual Tutor.

With the use of the Virtual Tutor, we can understand the impact that such an application has added support to e-learning.

1.1 Motivation

A course structured learning environment like Moodle relies heavily on displaying a significant amount of information. The large quantity of information requires constant observation for the student to keep up to date with the program which can be overbearing when combined with the limited time that students have to study, this pressure can demoralise the student, leading to worse grades or causing the student to give up.

Depending on the situation, a student can have limited communication with the professor, reducing the feedback received regarding the work performed, leaving the student without a grasp regarding his/her performance in the course.

The problems indicated previously provide a determent for the student that hinders the learning experience.

1.2 Objectives

The primary objective of this thesis is the creation and development of a component that is capable of retrieving and storing relevant information from the Moodle server and an external database server to the Virtual tutor application located in a course page developed by the team at Faculdade de Ciências da Universidade de Lisboa and the Virtual Tutor application for Android developed by the team at Instituto Superior Técnico de Lisboa. The solution must be able to:

- Gain access to the information about the Moodle course and determine if alterations occurred to the course page since the last saved login.
- Evaluate the student according to the parameters defined by the professor and offer positive feedback to create a strong bond with the student.
- Keep a coherent and up-to-date record of the student with the help of an external database.

The proposed solution combined with the back office application enables an application to inform the student of alterations in the course such as the creation of new content or posts in the forum, and the ability to evaluate the student depending on the parameters defined by the professor and the performance of the student.

1.3 Contributions

The main contributions of this thesis are:

(i) A Unity plugin capable of communicating with a Moodle server and an external server enabling a Virtual tutor application to accompany the student.

(ii) A Back office application developed alongside the Unity plugin with the purpose of allowing a professor to configure essential parameters for the correct operation of the Virtual tutor application and store said parameters in the external database.

(iii) Formulas were developed to assess the state of the student using variables defined by the professor of the course and the interactions the student performs in the course page to provide feedback relative to the experience of the student.

(iv) Tests performed with professors from Universidade Aberta to receive the opinion of someone experienced in online teaching.

Additionally, a scientific paper was produced and submitted for publication as a long paper in the International Conference on Graphics and Interaction 2018 (ICGI 2018).

1.4 Work plan

From the 1st of October to the 30th of June:

- Development of the necessary processes for the access and register of information of each user on Moodle.
- Development of the Back Office application.
- Perform operation tests to fine-tune the solution.
- Perform stress tests to analyse the impact the application has on the Moodle platform.
- Test developed work with professors from Universidade Aberta.

From 1st of July to 30th of July:

- Write the final report.
- Contribute in the writing of the paper for publication.

Situations that caused delay:

- Moodle not responding due to insufficient RAM, in this situation the server had 1 GB of RAM for the Moodle platform, currently it contains 8 GB of RAM.
- The creation of the external server that hosts the external database that was necessary as an alternative that did not create additional tables in the Moodle database.
- Fine-tuning the application to the multitude of possible cases.

1.5 Document structure

This document is organized in the following order:

- Chapter 2 – **Concepts and Related Work**

  The chapter presents concepts related to e-learning and Moodle, as well as works related to virtual avatars with a comparison of the final product with the one offered in this work, and works that detail advantages that virtual tutors can bring to a course.

- Chapter 3 – **VirtualTtutor - Plugin and Back Office**

  The chapter describes the environment used for the development of the Unity plugin and the back office application, detailing every component existent in the developed work and explaining the formulas created and the tables in the external database.

- Chapter 4 - **Tests**

  The chapter describes tests made with the purpose of evaluating the application and presents the results and conclusions derived from these tests.

- Chapter 5 - **Conclusions and future work**

  The chapter presents the conclusions made after the development of the project, followed by suggestions for future work.

Chapter 2

Concepts and Related Work

In this chapter concepts related to e-learning and Moodle are presented, as well as works related to virtual avatars with a comparison of the final product with the ones offered in this project, and works that detail advantages that virtual tutors can bring to a course.

2.1 Concepts

In this section, a short description is made regarding a set of concepts that are considered essential for the understanding of the current project and the development environment.

2.1.1 E-learning

Although e-learning lacks a precise definition universally accepted by researches, Arkoful et al.[14] proposes the following description focused on the current project: e-learning refers to a learning system based on formalised teaching combined with the use of electronic technologies such as the internet to improve the learning experience.

Such learning systems rely on access to a vast amount of information that can help a student learn more efficiently and more comfortably and at their own pace. Additionally, the existence of a virtual environment shared by some individuals can motivate the student not only to learn but to interact with professors and peers.

The e-learning platform used in the duration of the project is Moodle and is considered to be internet-based learning, this is due to having the content available on the internet, increasing the availability of the learning material further.

2.1.2 Learning Management System (LMS)

Learning management systems is a software application that provides a virtual environment that enables the management and delivery of e-learning courses[10].

The essential functions provided by an LMS for the administration of a course are the creation, organisation and distribution of content. The LMS used for the duration of the project is Moodle provided by Universidade Aberta.

Moodle (modular object-oriented dynamic learning environment) is an open-source learning management system written in PHP. The flexibility provided by Moodle combined with the ability to add plugins efficiently make it not only capable of implementing additional features efficiently but also the prime candidate to create and develop an application capable of accompanying a student during the learning process.

2.1.3 Web service

A web service is an online service that transforms a request from a user into an operation that can communicate with several applications to obtain a response[2][9].

The web services used on the project are either supplied by Moodle or custom-made. In Moodle web services, it is necessary that a valid token is supplied with the parameters and name of the function when making a request, as it provides authentication of the user to the service. The custom-made web services were used to communicate with the external database.

2.1.4 External service

An external service is a service created on Moodle that contains a set of web services[4].

Users are required to be associated to take advantage of the web services. Users must also have the necessary capabilities and permissions to use these web services.

2.1.5 Token

A token is a key given by Moodle that allows access to an external service, this enables a user to gain access to the available web services[6]. Administrators can create tokens or users with the capability “moodle/webservice:createtoken”. In the spectrum of the project, communication with Moodle is vital to acquire data about the student, and the course. The usage of a token enables a user to continue to use an external service even though the service might have suffered alterations.

2.1.6 Module

A module[3] represents and contains information about one of the activities and resources found on the course page. In Moodle, a structure called topic serves as a container and displays of a selection of modules.

The module is an essential component of the course page, as it allows the representation of various types of learning materials. The plugin requests limited information of each module, expanding the request for assignments and forums.

2.2 Related Work

This section presents the review of papers done during the project duration. On the first part, the analysis of a set of virtual tutors is made and compared with the two applications created in the current project.

2.2.1 Virtual Tutors

A tutor in a virtual environment can be of various types. A human tutor using an avatar in a virtual environment like Second Life or AI can constitute a virtual tutor. Regarding virtual tutors, a series of papers are analysed taking account several types of virtual tutors and compared to the current project. At the end of this subsection, we present a table comparing the approaches.

Development of a Virtual Assistant for Alerts and Notifications in a Learning Environment

Amaral et al.[13] continued the development of a Moodle integrated virtual assistant developed by Maciel et al.[21]. This paper describes the developed Android application that could provide the alerts and notifications of the aforementioned virtual assistant. This version can engage pro-actively with the student, prompting him to do specific tasks or to make a study plan in proximity to an exam or other kinds of evaluations that might be imminent without the need of accessing a browser.

VCAT: Virtual Cultural Awareness Trainer

Johnson et al.[18] developed the VCAT training model used by the military services. It is a web-based course that incorporates interactive games as a way to introduce trainees to information in order to acquire basic operational intercultural competence, by questionnaires and role-playing examples, this helps learns to associate knowledge gained to a situation.

STEVE: An Animated Pedagogical Agent for Procedural Training in Virtual Environments

Johnson et al.[19] created STEVE. It acts as an instructor and assistant that explains what is needed and how to do it without disrupting the user on a virtual world. STEVE is considered a virtual body (hand or full body as examples given) that the user can see and watch it interact with the world increases the sense of presence of the user, keeping the user focused.

Desenvolvimento de um Assistente Virtual Integrado ao Moodle para Suporte a Aprendizagem Online

Maciel et al.[21] describe their implementation of an avatar into Moodle which can assist the student learning by relating pertinent information given by the professor via voiceover, stating that a student can become overwhelmed with the data in Moodle, demoralising him. This paper is of particular relevance since it aligns perfectly well with the context of our Virtual Tutor.

Cognitive Intelligent Tutoring System based on Affective State

Rajkumar et al.[23] propose a new framework for an Intelligent Tutoring System (ITS) that also aims to understand the emotions that lay beyond the spoken words and take this to contextualise the interaction better. The tutor processes the persons emotional state by utilizing Active Appearance Model for facial feature extraction, a Camshift technique is used to track the hand pose and orientation, lastly, speech features were extracted from continuous speech and produced utilizing a Natural Language Processing unit to generate the proper phonetic forms by analysing text and predicting prosodic information. The virtual tutor has a data model that represents its beliefs, intentions, knowledge and desires and plans its behaviour by explicitly handling these models. By utilising this information, the tutor becomes able to assess the needs and purposes of the student.

My Science Tutor: A Conversational Multimedia Virtual Tutor

My Science Tutor(MyST) is an ITS developed by Ward et al.[24] that was designed to assist elementary school students in science-related subjects via an interactive multimedia environment. Students interact with an always present virtual 3D character via natural spoken dialogue. The tutor can ask open questions related to animations, illustrations or interactive simulations shown on the screen. For added realism, the avatar produces visual speech, synchronising both head and face movement with her speech. MyST utilises media to facilitate interaction, by providing multimedia the student is better able to think and reason about the question. The

tutor also analyses the spoken explanations to be able to determine what the student knows to formulate new questions. The avatar is designed to behave like an effective human tutor to promote rapport between the virtual tutor and student.

Comparison

Regarding this project, two applications of the Virtual tutor application were created, they are both interactable applications and use the same avatars however they serve different purposes. The variants will be referred as VT Moodle for the application developed by the team at Faculdade de Ciências na Universidade de Lisboa (FCUL) and VT Android when referring to the variant developed by the team from Instituto Superior Técnico (IST).

VT Moodle is a virtual tutor that resides in the course page, notifying the student of news and giving information regarding the performance on the course, the application retrieves data about the student and the page. The student can, through the use of buttons, navigate the information presented in Moodle.

VT Android app aims to accompany the student during the course duration in a mobile application, providing dialogues for increasing empathy with the student all the while incentivising the student to keep a study plan to improve academically.

Table 2.1 compares the virtual tutors analysed with the two applications created during the project regarding platform that the virtual tutor runs, dimensionality of the environment/avatar, projection of text using text-to-speech, if the virtual tutor can be interacted with by the user, realism of the environment and if the virtual tutor is capable of acting according to player actions (reactive) and or if it can operate without previous interaction of the user (proactive).

	Platform	Model type	Natural language	Interactive	Realism	Action type
VCAT	PC	3D	No	Yes	Medium	Reactive
STEVE	PC	3D	Yes	Yes	Low	Proactive/Reactive
Moodle assistant	PC/Android	2D	Yes	No	Low	Proactive
Cognitive ITS	Undefined	3D	Yes	Yes	High	Proactive/Reactive
MyST	PC/Tablets	3D	Yes	Yes	Medium	Proactive/Reactive
VT Moodle	PC	3D	No	Yes	High	Proactive/Reactive
VT Android	Android	3D	No	Yes	High	Proactive/Reactive

Table 2.1: Comparison table

Comparing the existing virtual tutors with the ones developed in the current project we can verify that a vast majority of examples are in a 3D environment or use 3D objects, this seems to have an association with the interactivity of the application with the user. The only case that is neither 3D nor interactive is the Moodle assistant as it mostly serves as a message relayer from student to professor.

2.2.2 Additional analysis

The following papers although not presenting a virtual tutor are essential for the project nonetheless, referring to the importance of critical factors to consider when developing a virtual tutor in Moodle.

Virtual Learning Environments

The paper of Dillenbourg et al.[16] starts by stating that a virtual learning environment (VLE) is not limited by supporting virtual reality technology nor by the scope of age or level of the target audience. VLEs have multiple features such as being a social space for users to act upon it in a manner acceptable and logical. Users can also be from various geographical origins. The environment must also integrate multiple technologies and pedagogical approaches and must have virtual counterparts for physical objects. The authors state the effectiveness virtual learning environments have in improving education by reducing educational system costs and improving the quality of education, however, virtual media is viewed by professors mostly as a tool to use not a substitute.

The Impact of Avatar Realism and Eye Gaze Control on Perceived Quality of Communication in a Shared Immersive Virtual Environment

Garau et al.[17] present a study to measure how vital are the realism of the avatar and the kind of visual contact it establishes with the users. To test this, an experiment in which an avatar of lower quality (without defined gender) and two avatars of higher realism (with defined gender) along with two different types of eye gaze. The authors state that a certain consistency is required, a lower fidelity model should exhibit a low fidelity behaviour. We would argue that this results in an uncanny valley effect, in which the fact that a less realistic model shows a more "human" behaviour causes a certain feeling of eeriness and disgust with the user.

Intelligent Tutoring Systems and Pedagogical Experience Manipulation in Virtual Learning Environments

In this paper Lane et al.[20] expose some of the nuances and the differences between human and virtual tutoring, some solutions to improve Virtual Learning Environments (VLEs) such as incorporation of a pedagogical agent to serve as an entity for the learner to "bond" with or altering the educational experience to a more subtle and manipulative way.

The paper also states the importance of what kind of tutor and what kind of feedback we want him to give, as well as the realism of the environment that helps the learner immerse and ultimately improve the learning experience.

Evaluating the Impact of Interface Agents in an Intelligent Tutoring System

Moundridou et al.[22] make emphasis on the presence of a virtual agent in an intelligent tutoring system which increases the easiness and enjoyment of users, however it does not increase the learning incomes nor attentiveness significantly, and users feel more comfortable interacting with a virtual agent of similar status, such results might prove useful for our work. This work serves a certain proof-of-concept of what we are trying to achieve. The main conclusion from this study is that the existence of a pedagogical agent does, in fact, improve the learning experience of the student.

2.3 Conclusion

In this chapter, several projects have been analysed that focus on the representation of a virtual tutor with the objective of assisting an interested party in the learning experience, some projects that focus on the impact that such virtual tutors can have in the learning experience and analysed critical concepts in the developed project. As shown in the displayed projects, only one set of them revolve around a virtual tutor assisting in a Moodle platform, however, the analysed projects merely deliver information requested by the professor to the students as opposed to the features in the Virtual Tutor project that aim to assist the student with the content presented in the course page and create empathy.

Chapter 3

Virtual tutor - WebManager and Back Office

3.1 Introduction

The project created by Universidade Aberta (Uaberta) aims to create a mediator, with an anthropomorphic representation, between the student and the professor in a curricular unit, with the primary goal that the student feels more support and becomes motivated towards improving grades. The project is being developed by three institutions: Universidade Aberta de Lisboa, the Instituto Superior Técnico de Lisboa and Faculdade de Ciências da Universidade de Lisboa. The project is divided into two parts:

- The teams from the three institutions worked together on the requirements gathering phase for identification of use cases and subsequent creation of crucial components necessary for the interaction with the student, for example, phrases used by the virtual tutor.
- The creation and development of a mediator for the online learning management system Moodle and a mediator for mobile devices running with the operating system Android. The team from Faculdade de Ciências na Universidade de Lisboa (FCUL) was in charge of implementing the mediator for the Moodle platform, while the team from Instituto Superior Técnico (IST) developed the solution for the Android system.

The mediator developed by FCUL is composed by three modules: A interface module created by Catarina Alves (Master student in computer engineering of FCUL); An artificial intelligence module developed by Ana Marisa Salgueiro(Master student of Faculdade de Letras da Universidade de Lisboa); An information retrieval and storage module that will be described in this chapter.

During the creation and development of the information retrieval module, a Unity[7] plugin was produced. The plugin is capable of communicating with the

Moodle server and an external server both provided by Universidade Aberta in a WebGL or Android environment. Accompanying the mentioned plugin a back office application in Unity was developed as an interface that allows professors to define the parameters that influence the calculation of the state of the student by interacting with the external database, the Back Office application offers a temporary solution while the integrated back office application is in development by the team of UAberta.

This chapter will discuss the developed work regarding the information retrieval and storage module produced during the project duration, the environment of the project, approaches possible to each option and the reasoning for the chosen path, followed by a section dedicated to explaining the operation of each component.

3.2 Project Environment

Unity was the game engine decided to develop the Virtual Tutor application. This cross-platform game engine allows for the creation of an application that can present and animate the avatars that serve as a representation of a human tutor. In the supported platforms that Unity is capable of producing, WebGL (Web Graphics Library) is the platform that can present the application in a Moodle course page. Using WebGL as a solution for the implementation of the application enables the usage of a HTML block to contain the compiled application without the use of plug-ins.

To make the processes of communication with Moodle as adaptable as possible to Unity, C# is the language used to develop the solution and the web services supplied by Moodle 3.4 as the source of information from the server.

The WebGL Virtual tutor application was established to run in the Moodle environment more specifically when the student access the course page, in that scenario, authentication of the student was complete. In contrast, the Android virtual tutor application was expected to run in an environment where access to the internet is limited or non-existent, creating the possibility of the application initialising without the ability to retrieve information from the Moodle server. Authentication is required at the start of the Android application, this can be prevented by storing credentials on the device, removing the necessity of requesting them to the user on each start.

Communication in Unity WebGL is restricted. The use of sockets to maintain a constant connection with the server is disabled due to browsers not having direct access to IP sockets, reducing the available methods to communicate via the functions WWW and UnityWebRequest. For simplicity of usage, the chosen function was WWW.

To use the WWW function in Unity, it is necessary to run the function inside a coroutine. Despite coroutines virtually not having a performance overhead, the only form to guarantee that a coroutine as finished is to yield for the return, this, in turn, stops the function that calls the coroutine until it completes. In the current project, a function calls multiple coroutines that handle different requests, combined with the inherent delay in web communications means that yielding for a coroutine means delaying the entire process of information retrieval. The chosen approach for communication is the same for Android and WebGL.

3.3 State of the Student

The values collected from the back office application contribute to the evaluation of the state of the student. The WebManager plugin retrieves these thresholds and calculates the state of the student, the values are a combination of two values:

1. Evaluation that depends on the grade of the student in each e-folio;

2. Attendance that depends on:

   (i) The frequency of logins made by the student on the page of the discipline;

   (ii) The number of posts made by the student in the forums.

With this, it is possible to position the student in one of nine sectors of performance as presented in the graph contained in Figure 3.1.

Concerning Evaluation, the professor defines a minimum threshold and a maximum threshold. Taking into account these two thresholds, there are three regions: below the minimum, above the maximum and between the two values. In the case of Attendance, the professor defines a minimum and maximum thresholds for the frequency of logins, the number of posts in the forums and also assigns a weight to frequency and posts in the calculation of the attendance. The combination of these parameters also defines a minimum and a maximum threshold for the attendance, also defining three regions. So, combining Evaluation and Attendance, there are nine regions, as illustrated in Figure 3.1.

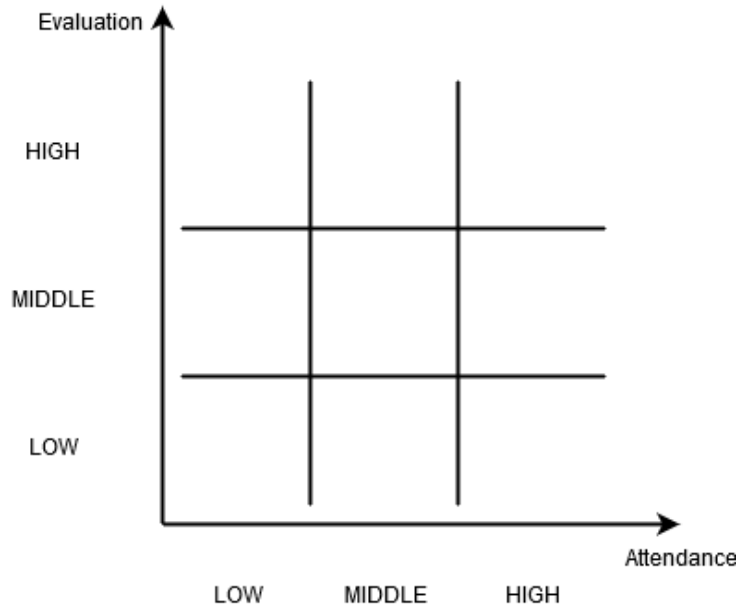

Figure 3.1: Performance chart.

The behaviour of the avatar (expressed by facial expressions and text) is generated according to the region where the state of the student is located in. The phrases requested from the external database take into account the calculated region. The professor defines all the thresholds and weights in the interface of the back office and allows him/her to somehow customise the behaviour of the avatar in the discipline.

3.3.1 Evaluation

Evaluation measures the performance of the student in the course. Equation 3.1 and Equation 3.2 have the purpose of putting the student in one of three regions depending on the grades of the student and the thresholds defined by the professor:

- HIGH for a student that is considered to have great grades;
- MIDDLE for a student that is considered to have mediocre grades;
- LOW for a student that is considered to have low grades.

Only the evaluated assignments of the student count for the calculation.

$$EvaluationValue = \frac{\sum \text{student grades of the e-fólios given}}{\sum \text{maximum grade of the e-fólios graded}} \tag{3.1}$$

$$Evaluation(EvaluationValue) = \begin{cases} HIGH, & \text{for EvaluationValue} \geq \text{Aprov\_High} \\ MIDDLE, & \text{Aprov\_Low} \leq \text{EvaluationValue} < \text{Aprov\_High} \\ LOW, & \text{for for EvaluationValue} < \text{Aprov\_Low} \end{cases} \tag{3.2}$$

Where:

- Aprov_Low - the Minimum threshold for evaluation, defined in the back office application, has a value between 0 and 100 and marks the border between a low and middle Evaluation.
- Aprov_High - the Maximum threshold for evaluation, defined in the back office application, has a value between 0 and 100 and marks the border between the middle and high Evaluation.

3.3.2 Attendance

The resulting value is between 0 and 100 and comparing it with the attendance values defined in the back office allows for the restriction of attendance to three cases, low, middle and high as shown in Equation 3.4.

$$AttendanceVal = LoginImp * ValLogin + PostImp * ValPosts \tag{3.3}$$

Where:

- ValLogin - Value of the Login, calculated in Equation 3.5.
- ValPosts - Value of the Posts, calculated in Equation 3.6.
- LoginImp - Weight of Login, defined in the back office application and has a possible value between 0 and 100, the sum of this value and PostImp is one hundred.
- PostImp - Weight of Posts, defined in the back office application and has a possible value between 0 and 100, the sum of this value and LoginImp is one hundred.

$$Attendance(AttendanceVal) = \begin{cases} \text{HIGH}, & \text{for AttendanceVal} \geq \text{Assid\_high} \\ \text{MIDDLE}, & \text{Assid\_low} \leq \text{AttendanceVal} < \text{Assid\_high} \\ \text{LOW}, & \text{for for AttendanceVal} < \text{Assid\_low} \end{cases} \tag{3.4}$$

Where:

- Assid_low - the Minimum threshold for attendance, defined in the back office application and lower than Assid_high, has a value between 0 and 100 and marks the border between a low and middle Attendance.
- Assid_high - the Maximum threshold for attendance, defined in the back office application and higher than Assid_low, has a value between 0 and 100 and marks the border between a middle and high Attendance.

Logins

Equation 3.5 is used to calculate a number between 0 and 1 to measure the assiduity of the student in terms of the login frequency. Later this value is used in combination with the importance of the logins defined in the back office to make the login portion in the attendance Equation 3.3.

$$ValLogin(AvgLogin) = \begin{cases} 1, & \text{for AvgLogin} \geq \frac{1}{\text{TLMin}} \\ 0, & \text{for AvgLogin} < \frac{1}{\text{TLMax}} \\ Max(0, AvgLogin \times TLMin), & \text{for } \frac{1}{\text{TLMax}} \leq \text{AvgLogin} < \frac{1}{\text{TLMin}} \end{cases} \tag{3.5}$$

Where:

- AvgLogin - Relative frequency of the logins made during the course duration, the number of days the student used the application in the course stored in the external database divided by the days since the start of the course. The number of days since the start is calculated by obtaining the difference between the value saved in the database defined in the back office application and the current day.

- TLMin (ThresholdLoginMinimum)- the Minimum threshold for login, defined in the back office application and lower than TLMax, marks the border between a middle and high frequency of login.

- TLMax (ThresholdLoginMaximum)- the Maximum threshold for login, defined in the back office application and higher than TLMin, marks the border between a low and middle frequency of login.

Posts

Equation 3.6 calculates a value between 0 and 1 to measure the intervention of the student in terms of the number of posts made by the user in the course.

The calculations take into account every post made by the user regardless of utility and content because the content of the post is not evaluated. Later the value is used in combination with the importance of posts defined in the back office to

make the post portion in the attendance Equation 3.3.

$$ValPosts(nPosts) = \begin{cases} 1, & \text{for nPosts} \geq \text{TPostMax} \\ 0, & \text{for nPosts} < \text{TPostMin} \\ \frac{nPosts}{Max(nPosts, TPostMax)}, & \text{for TPostMin} \leq \text{nPosts} < \text{TPostMax} \end{cases} \tag{3.6}$$

Where:

- ValPosts - Value of the Posts
- nPosts - Nr. of posts made by the student since the beginning of the course.
- TPostMin (ThresholdPostMinimum)- the Minimum threshold for posts, defined in the back office application and lower than TPostMax, marks the border between the low and middle number of posts.
- TPostMax (ThresholdPostMaximum)- the Maximum threshold for posts, defined in the back office application and lower than TPostMax, marks the border between the middle and high number of posts.

3.4 Virtual Tutor Application Architecture

There are two major components of the architecture of the virtual tutor application in Moodle:

- InterfaceManager: it is responsible for representing the interface and the avatar, access the information stored in the WebManager component to inform and interact with the student, this component was developed by my colleague Catarina Alves.
- WebManager: Component that requests and stores information about the student and the course for the duration of the interaction. This component is described in this chapter.

The necessary information is provided by the Moodle server that hosts the application and an external server responsible for housing information regarding the student and the virtual tutor. The application inserts this information into the external server during the interaction with the student. The information is stored in dedicated classes (UserData and Course), and can be accessed via the DataManager

class that is available in the WebManager controller class. The interaction of the components of the application and the external servers are depicted in Figure 3.2.

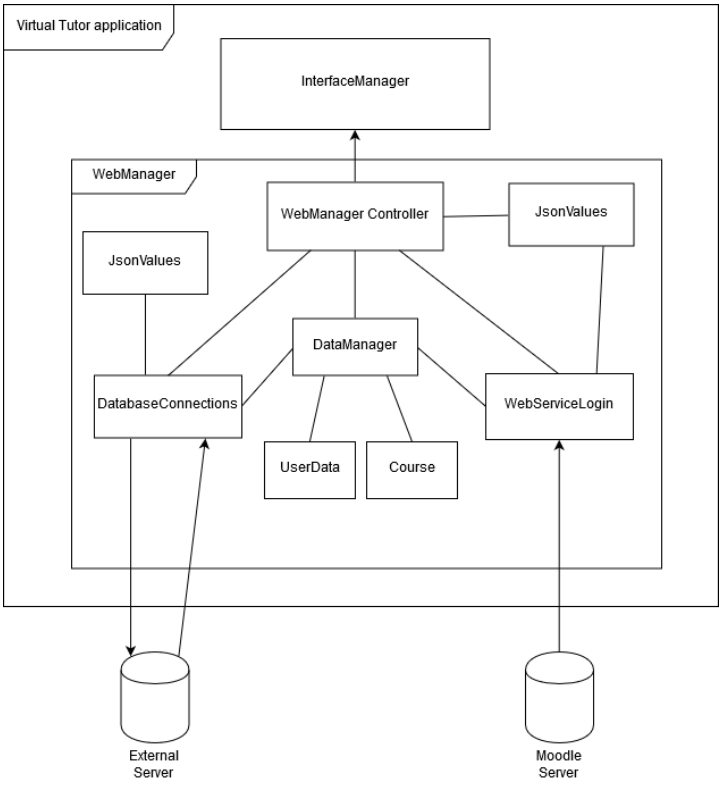

Figure 3.2: Virtual Tutor application components.

WebManager

The WebManager controller class acts as a controller of the other classes in the plugin, receives the credentials and coordinates the operations of communication with the servers.

This class is responsible for interpreting the responses for the external database and calculates the state of the student as well as store captures the information from the external database.

DataManager

The DataManager class holds the UserData and Course classes, manages the information retrieved and removes collisions in the Course components.

The classes WebManager and WebserviceLogin access DataManager to store, deliver and access information about the user and the courses. For each component type that has received all the information by WebserviceLogin, a boolean is set to true to signify the information about the component type is replicated in Course. When all the components are retrieved, DataManager performs a collision removal function and associates the news received with the respective object of the course.

WebserviceLogin

The WebserviceLogin Class receives the credentials from WebManager and retrieves the information of the student and course and sends it to the DataManager class for management.

The class uses coroutines to retrieve the information and interprets it using the nested classes in JsonValues class. As discussed previously, to guarantee the completion of a coroutine it is necessary to yield for the return. In the project, not all the information is required to inquire further information, for example, getting the information about the forums should not make the request for the grades to wait. Yielding only for the return of mandatory requests such as acquiring the names and identifiers of the courses as opposed to yielding for all the requests can reduce the information retrieval process from 9 to 5 seconds.

Previously, when retrieving the information about the forums, the contents were requested, this would imply a request for each discussion and the posts contained inside which in turn would cause a performance impact, an alternative method was reached using the external database to only request the number of posts made by the student.

DatabaseConnections

The DatabaseConnections class is responsible for preparing the requests of the plugin to the external database based on the parameters received.

The DatabaseConnections class contains the URLs of the PHP files that contact with the external database. The methods prepareRequest and prepareRequests receive the parameters necessary to get the information from the database and run coroutines that execute the makeRequest and makeRequests respectively.

The URL created to communicate must send the variables to the corresponding PHP file, the formula for the URL goes as such:

$$URL = location + filename + variables + hash$$

- Location is hardcoded in the class, and the choice of using the normal or secret one is stated as a parameter
- Filename is a variable given to the method and corresponds to the name of the file in the database, the name of the files and a description of their function is detailed in subsection 3.7.8.
- Variables are sent to the external server in the alphabetical order to construct the same hash as the one that is a part of the URL.

- Hash is obtained by the MD5 hashing algorithm by the concatenation of the variables sent and a secret key shared with the external database.

UserData

The UserData class receives and contains the necessary information about the student to allow identification.

Course

The Course class serves as a container for the information about the course after the information retrieval ends. It contains the topics of the course as well as the components that are inside, the e-fólios and forums. Information is stored in nested classes to simulate the structure of the course page.

The class also verifies the information received by the grade report and detects the assignments that exist in the category "e-fólios". If a student has received the grade for all these assignments and is part of the group "Avaliação Continua", the variable checkGradeReport is set to true. When this occurs, the student is incentivised to check the grade report to know if it is possible to take the "p-fólio".

JsonValues

This class serves to contain the possible formats of the responses received. For better readability, the received format from both the Moodle and the external database PHP files is JSON. For Moodle to provide this format, the following text needs to be in the request: "&moodlewsrestformat=json". The PHP files in the external database are programmed to supply only in the JSON format or a string of error. The diagram of the class is decomposed into Figure B.13, Figure B.14 and Figure B.15, present in the appendix.

3.4.1 Moodle communications

To access the web services is necessary to provide a valid token with permissions for the requests, it is required to supply Moodle an account username and password that is allowed to access a service that contains the web services to acquire a token. A web service communication is illustrated in Figure 3.3.

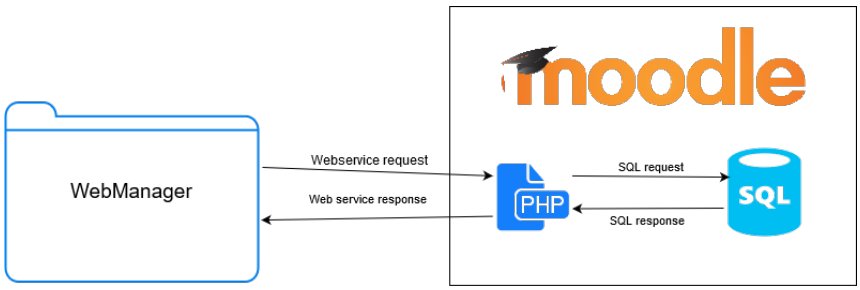

Figure 3.3: Communication flow of the plugin with Moodle.

For an account to be allowed to use a service, it is necessary to add the account manually to the list of permitted accounts.

With this in mind, there were two approaches:

- Associate the student with the service and get a token for the student;
- Associate a non-student account to the service and get the token for that account.

In the WebGL environment, not only would it prove to be cumbersome to gain a valid token without making the student input the account password every time the application started, as it would implicate that one communication with the username and password of the student would go unprotected. Android could circumvent both the request for credentials from the user and the token acquisition for a one-time retrieval and storage of the token.

Considering the scenario in which the students have access to the service, it would make so that each student that uses the Virtual Tutor would also need access to the service, requiring students to possibly have the permissions that are not supposed to have the ability to make the requests. In contrast, using a non-student account allows the management to be more straightforward and removes the worry of managing permissions for each student.

Using the scenario involving the non-student account, it is required to have access to the student information while using the account. However, the clearance level was insufficient with the role of a “student” as information specific for the student would be out of reach for the account, requiring the role in the course to increase to “manager” or “teacher”. This approach would also face the same problem of transmitting the credentials unprotected in addition to them being hard-coded into the program if token requisition makes part of the communication process. For an alternative, the account would not have a traditional login instead opting to make it only operate with web services, removing the chance of hijacking the account for malicious intents although in this case, the token now has to be hard-coded instead of the credentials.

The web services utilized in the plugin can be found in the following tables:

- Table 3.1 describes the web services and the Moodle version they became available;

- Table 3.2 indicates the number of times the web services are used during the information retrieval process and the time they are requested.

Web service	Moodle Version	Description
core_user_get_users	2.5	Used to retrieve information about the user.
core_enrol_get_users_courses	2	Retrieves the information of the courses the student attends.
core_course_get_contents	2.2	Retrieves the course topics.
mod_assign_get_assignments	2.4	Gives the information about the assignments (e-fólios).
mod_assign_get_grades	2.4	Gives the grades of the assignments.
gradereport_user_get_grade_items	3.2	Gives the information about the grade report.
mod_forum_get_forums_by_courses	2.5	Gives the information of the forums.
mod_forum_get_forum_discussions_paginated	2.8	Gives the information about discussions contained in a forum.
mod_forum_get_forum_discussion_posts	2.7	Gives the information about posts contained in a discussion.
core_course_get_updates_since	3.3	Checks the new information of a course regarding a timestamp.
core_group_get_course_user_groups	2.9	Gives the list of groups the student is a part of.

Table 3.1: Web services used in the plugin - short description

Web service	When Used	Number of Times Used
core_user_get_users	When student Id is received	Once
core_enrol_get_users_courses	After core_user_get_users and when course Id is received in WebGl	Once
core_course_get_contents	After received Courses	x Nº of Courses
mod_assign_get_assignments	After received Courses	x Nº of Courses
mod_assign_get_grades	After received Assignments	x Nº of Courses
gradereport_user_get_grade_items	After received Assignments	x Nº of Courses
mod_forum_get_forums_by_courses	After received Courses	Once
mod_forum_get_forum_discussions_paginated	After received Foruns	x Nº of Foruns
mod_forum_get_forum_discussion_posts	After received Discussions	x Nº of Discussions
core_course_get_updates_since	After receiving the list of the user logins	x Nº of Courses
core_group_get_course_user_groups	After received Courses	x Nº of Courses

Table 3.2: Web services - usage in the plugin

3.4.2 External Communication

Unity in a WebGL environment does not have access to IP sockets, prohibiting the direct connection with the SQL database located in the external database. With the direct approach impossible, the alternative chosen to allow communication was the usage of PHP files as intermediates in the communication as depicted in Figure 3.4.

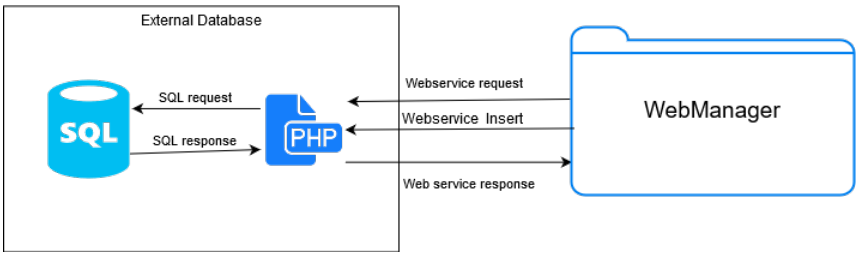

Figure 3.4: Communication flow of the plugin with External database.

This approach came with additional benefits, such as control over the format received and the possibility of using coroutines to communicate with the external database.

Rejected alternative

An alternative approach considered would not require the use of an external database. It would use the Moodle tools named Notes, this alternative would store the information on text files and could be accessed via the existing web services of Moodle. This notes could be divided in personal or course related, the course related would keep information regarding the interaction of the student and the course while the personal notes can be used to store information regarding the interaction between the application and the student.

Figure B.14 shows the structure created in the plugin to support the alternative.

However, the usage of such an approach would incur in a performance penalty, the more extended the data saved, the higher the penalty since reading a more extended text file would imply a higher waiting period.

3.5 WebManager Plugin

The plugin acts depending on the environment. In the case of WebGL, the supplied values are the student identifier and the course identifier since it is expected to run inside a specific course page. In contrast, the Android version runs in an external environment, with that in mind an additional step is required: authenticating the user of the application. This requires the username and password and to prevent subsequent logins, the credentials should be stored in the device. The top-down view of the operation of the plugin is presented in Figure 3.5 and instructions on how to implement the plugin into a Unity application are in Appendix A.

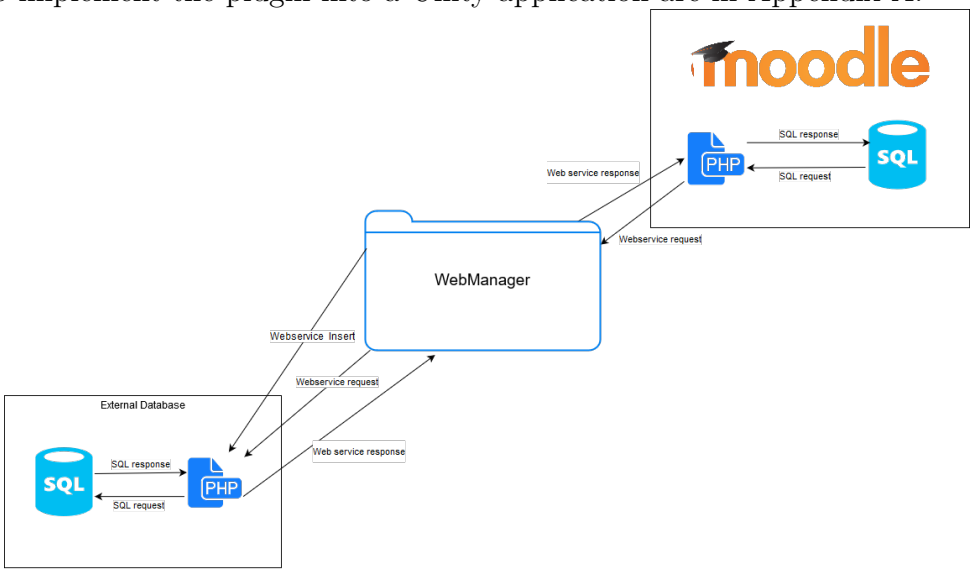

Figure 3.5: Communication flow of the WebManager plugin - Top-down view

3.5.1 Communication

The communication flow with Moodle and the external database needs to follow specific steps to ensure proper functionality, between the ordered steps, exists additional steps that only require that the previous mandatory steps be completed, this is done to retrieve information faster. It is possible to divide the communication flow into two phases, a first phase dedicated to retrieving core information, and a second phase that acts accordingly to the information received in the first part.

First phase: Acquisition of core information

The plugin manages communication with the Moodle server and an external server as depicted in Figure 3.6 for the retrieval of pertinent information for further interactions.

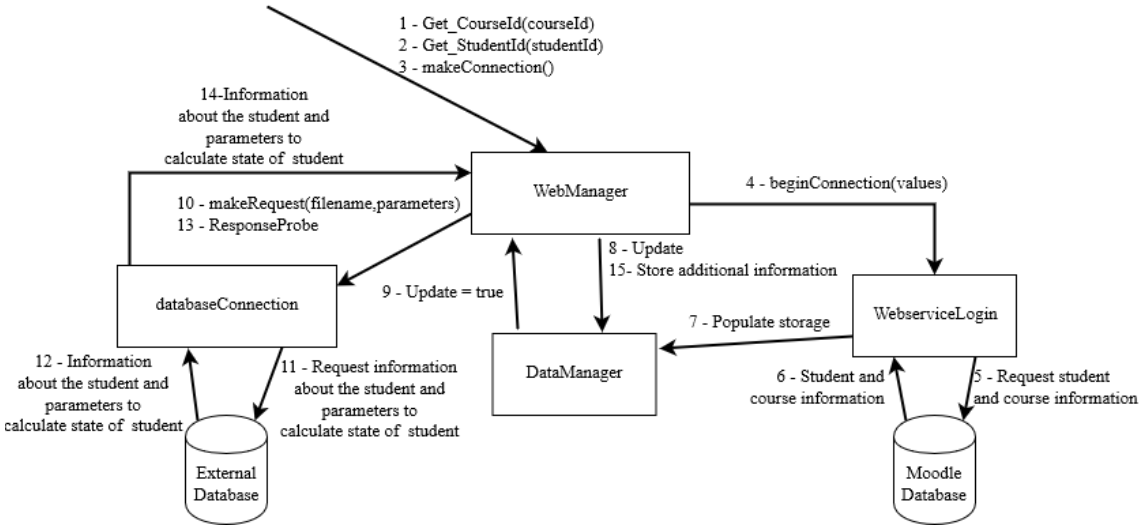

Figure 3.6: Communication flow of the WebManager plugin - phase 1

As depicted in Figure 3.6, for the retrieval and storage procedure to begin, it is required to supply the credentials and request the start to the WebManager class. The plugin after receiving the credentials communicates with Moodle to retrieve the information required to identify the student and the course and the environment of the course page that consists of:

- The structures that constitute the course page, the topics and modules contained in each topic;
- The Assignments and the grade attributed to the student;
- The name of the forums;
- The groups that the student belongs to.

After receiving this information, the plugin communicates with the external database and retrieves the information about:

- The identifier of the tutor the student chose for the course;
- Parameters defined by the professor for the course;
- Logins made by the student;
- State of the student in the course;
- Phrases that the avatar uses to communicate with the student;
- Activity of the student on the replicated logs of Moodle;

The steps 8 and 13 are a part of a probing procedure done by WebManager that runs on the update method with the purpose of verifying if specific information was retrieved and act upon this knowledge to begin communications with the external database. The update[8] method is a built-in method from the MonoBehaviour class and is called in every frame. The MonoBehaviour[5] class is the base class from which every Unity script derives.

Additionally, a design difference between DatabaseConnections and WebserviceLogin is the class that receives the content: in the case of WebserviceLogin, the information is sent to DataManager for storage as opposed to DatabaseConnections that stores the received information in a hash table to be later accessed and retrieved by the WebManager class. The reasoning behind the design difference comes down to the type of information collected. In WebserviceLogin, the information obtained from Moodle regards only the course and student, so storage in the appropriate class is in order. The information retrieved from DatabaseConnections relates to the student, course and avatar. This information can implicate changes in operation such as no chosen tutor for the course, making it more suitable to be first viewed by WebManager.

Second phase: Calculation and specific requests

After the first batch of information is obtained, the plugin acts depending on the captured information to perform additional actions, such as, calculate the state of the student, confirm if the student has previously entered the course and used the application today and verify which updates occurred in the course page that the student did not view yet. The depiction of the process can be viewed in Figure 3.7.

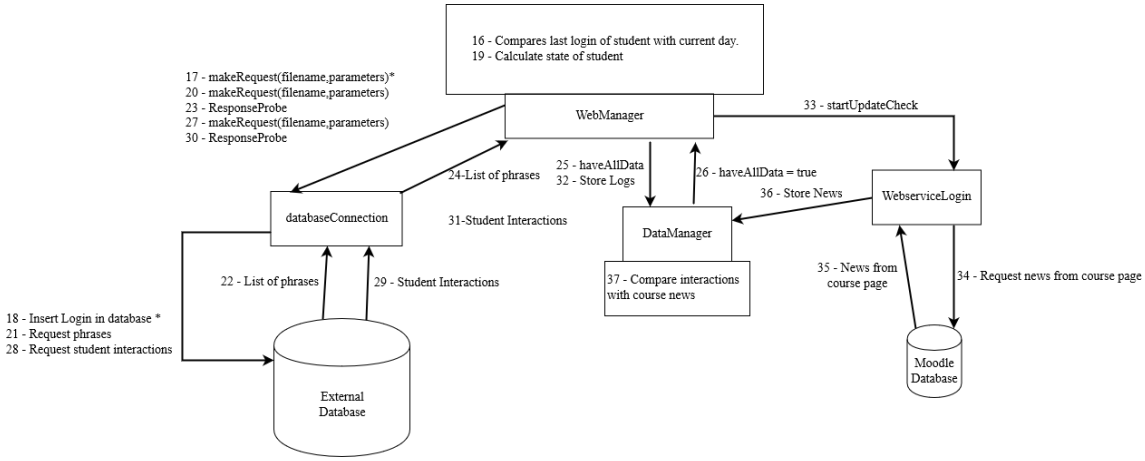

Figure 3.7: Communication flow of the WebManager plugin - phase 2

The second phase of communications revolves around the acquisition of information regarding interactions between the student and Moodle and interactions between the student and the avatar. When the information about the course page and the log of logins is retrieved, the plugin can ascertain if the student has already interacted today and the timestamp to ask Moodle for updates in the course page.

The sequence of steps 17 and 18 are marked with a (*). These steps only occur when the student interacts with the application for the first time in that day and symbolise the storage of the date in the external database and does not require the capture of the response. The set of steps 20 to 24, 25 to 26, 27 to 29 and 31 to 35 can co-occur however for the calculation for the news in the course to execute as desired the step 30 must occur before 36.

News refers to an update received by course_get_updates_since that has a timestamp after the last interaction of the student recorded in the log table. Updates refer to alterations in modules of the course page; these updates can be alterations of settings in the module, adding a post to a forum[12]. The information received regarding an update contains the definition of the update, the module that suffered the update and the time it occurred. The update received is compared to the most recent interaction the student had with the module if the update happened before the interaction then the plugin considers that the student already viewed the update; if the update happened after the interaction recorded in the logs an additional verification is necessary, that verification compares the timestamp of the update to the most recent recorded login (the last day the student used the application not counting the current day), this allows for the separation of old and recent news, for example, an old news is a news already indicated to the student but not yet viewed while a recent news happened between the current and last login and the student needs to be informed.

The calculation to classify news as old or recent requires information from Moodle, as it contains the timestamp of the change in the object on the course page;

The timestamp received is compared to the last interaction made by the student with the object to assess if the news has been consulted by the student.

3.6 Back Office Application

The purpose of this application is to provide a more intuitive and straightforward approach to define the parameters of any given course in the database without recurring into manual insertion to the external server. It serves as a temporary solution while the integrated back office application is in development by the group of Universidade Aberta.

The application only requires the injection of the identifier of the Moodle course to request the desired information from the external database for visualisation and possible alteration.

The classes defined for this program prevent the insertion of incorrect values in the database. The UML diagram of these classes is divided into Figure B.16, Figure B.17 and Figure B.18 in the appendix.

3.6.1 Main Screen

Figure 3.8 shows the main menu of the back office, it identifies the course and allows the definition of the start date.

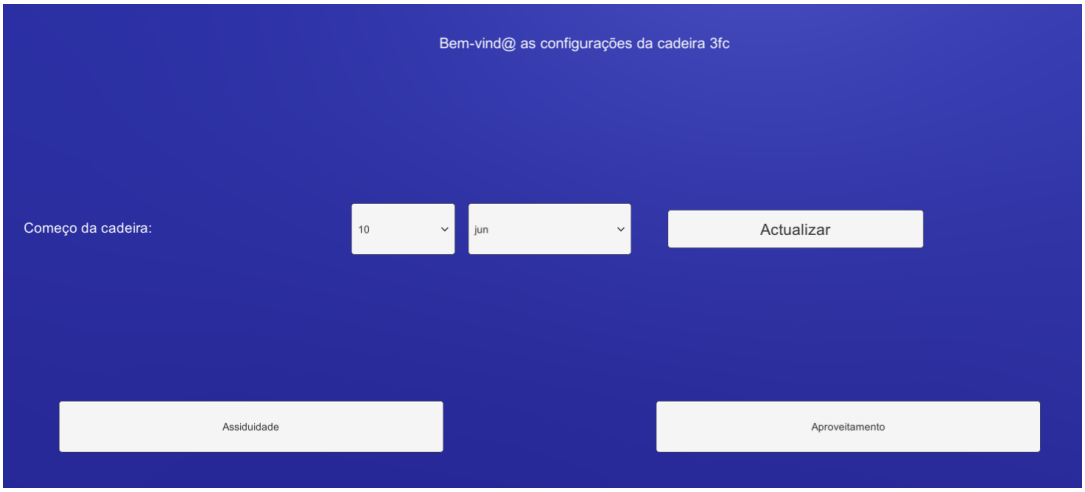

Figure 3.8: Start menu of back office.

The start course date value cannot be accessed in the Moodle web services and provides critical information for the attendance calculation. To define the start course date the day and month are required, the application assumes that the course starts in the current year.

The main screen contains three buttons used to perform actions in the application:

- Actualizar - Submits the value of the start course date to the database;
- Assiduidade - Changes to the attendance screen;
- Aproveitamento - Changes to the evaluation screen.

The class basicInformation provides the conversion of time and display of the course name.

3.6.2 Attendance Screen

This screen presented in Figure 3.9 is responsible for the display and update of the attendance values stored in the external database for the selected course. The parameters are then used in the calculations of the Attendance part of the state of the student.

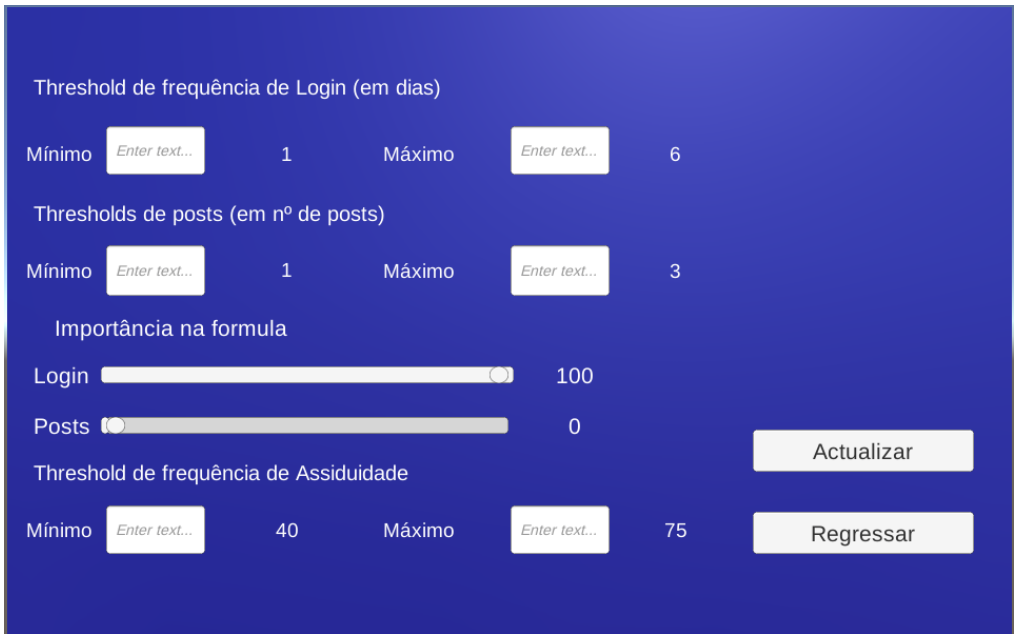

Figure 3.9: Attendance menu of back office.

To capture the parameters to update the external database, the application retrieves the values in the input fields. However, if no value is defined, it will get the already established value that is shown to the right of the input field. The sliders allow the definition of the importance of logins and posts in the attendance formula. The sliders are always prepared to do a combined value of hundred, to prevent invalid importances.

The Minimum value must always be lower or equal to the Maximum value, so if the values are not in those ranges, the application will write in red text in the bottom “Erro no envio” and will not attempt to update the database.

3.6.3 Evaluation Screen

The screen depicted in Figure 3.10 is responsible for the display and update of the evaluation values stored in the external database for the course.

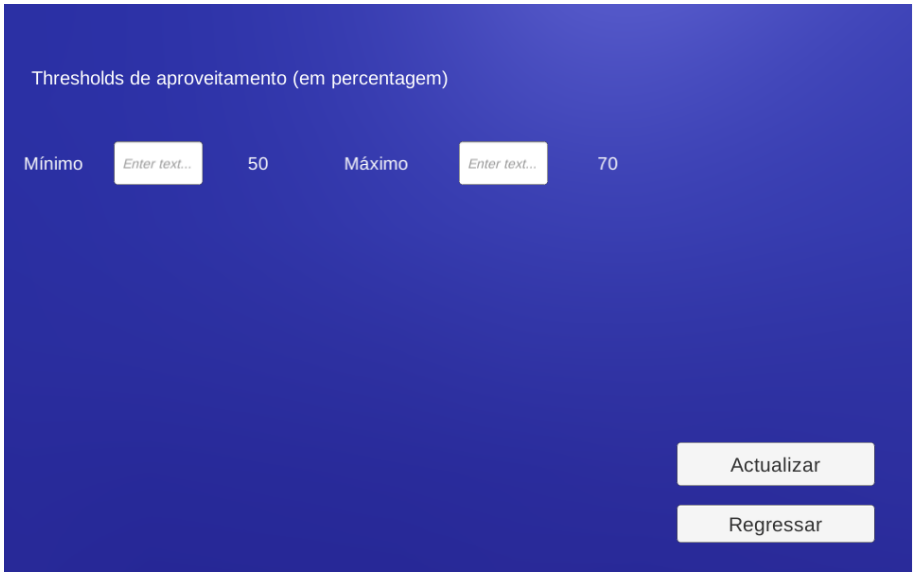

Figure 3.10: Evaluation menu of back office.

The taken approach is equal to the Attendance screen. To capture the parameters to update the program gets the value in the input fields, however, if no value is defined, it will get the already established value. The values inserted are then used in calculations of the Evaluation part of the state of the student, the. The Minimum must always be lower or equal to the Maximum, so if the values are not in those ranges, the application will write in red text in the bottom “Erro no envio” and will not attempt to update the database.

3.7 External Database

The Virtual Tutor requires information that Moodle cannot provide. The parameters to define the behaviour or the storage of logins of the student are some of the pieces of information that Moodle does not have. The existence of the external database allows for information relevant for the Virtual Tutor to be accessible in several devices. The following tables hold information relevant to the desired behaviour Virtual Tutor.

3.7.1 Parameters table

The table presented in Figure 3.11 holds the parameters to define the behaviour of the Virtual Tutor in contrast to the performance of the student. The values saved constitute a maximum and minimum that serve as thresholds.

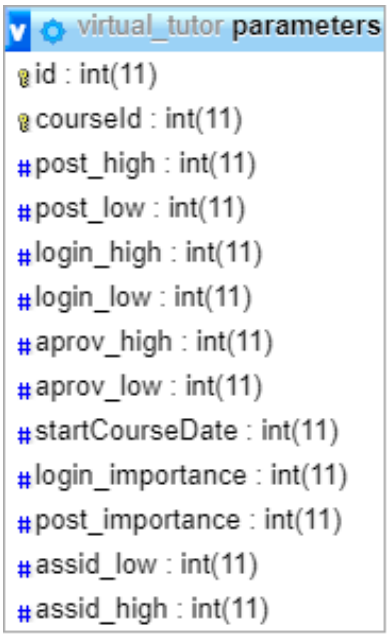

Figure 3.11: Parameters table.

3.7.2 Logins table

Keeps a record of the days the student has used the application as shown in Figure 3.12.

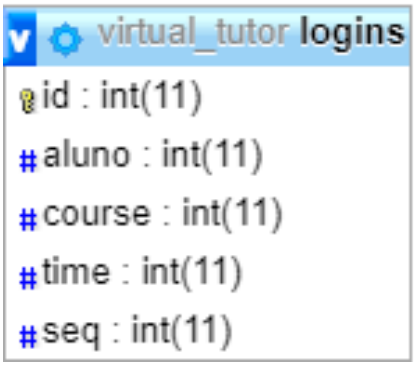

Figure 3.12: Logins table.

To prevent duplicate information that would affect the attendance calculation, the plugin only attempts to insert another login if the current day is different from the most recent saved login. Currently used to help calculate the attendance of the

student. The time value stores UNIX timestamp in an integer. The seq value refers to the order of the login in the sequence, can help in analyses of a pattern of logins.

3.7.3 Student_performance table

Keeps the record of the state of the student done by the plugin and stores the information as in Figure 3.13.

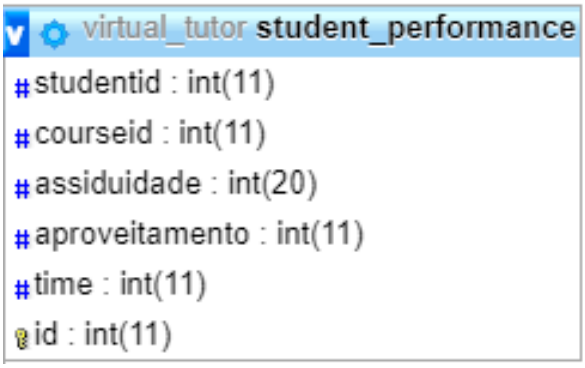

Figure 3.13: Student_performance table.

The plugin sends the information after the calculations are complete and only once per day. No current usage, however, can be used to track student performance along the course duration.

3.7.4 Tutor_selection table

Stores the student selection for the avatar in a given course as in Figure 3.14, the information is given only once for each student and course in WebGL .

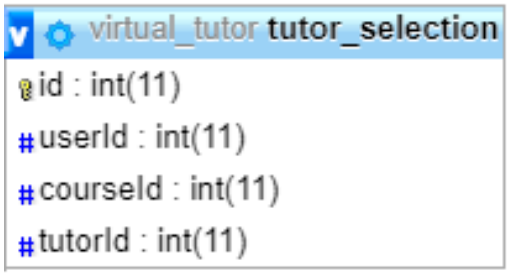

Figure 3.14: Tutor_selection table.

3.7.5 Questions table

The table is used to store the questions made by the student to the Virtual Tutor, the table is described in Figure 3.15.

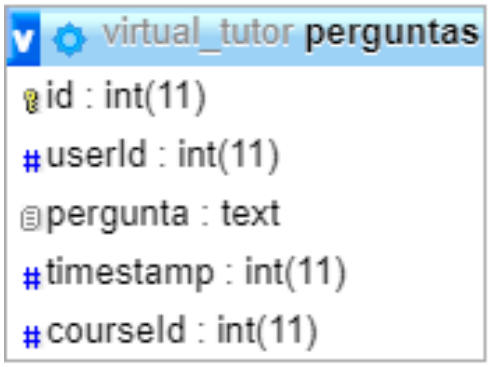

Figure 3.15: Questions table.

This table is a temporary replacement until the artificial intelligence module is prepared to answer questions from the students, can be used to develop a feature that allows the professor to view the questions made by the students.

3.7.6 Falas table

The table presented in Figure 3.16 stores the phrases that the Virtual Tutor displays in situations regarding the performance of the student.

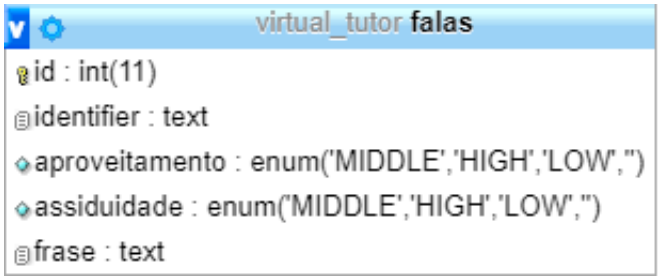

Figure 3.16: Falas table.

Phrases contained in the database are of the following type:

- Feedback phrases regarding evaluation in the greetings phase;
- Feedback phrases regarding attendance in the greetings phase, these combined with the previous phrase constitute the greetings expression;
- Phrases that reflect on the performance in an e-fólio;
- Phrases that direct the student to the grade report when the e-fólios are finished;
- Phrases to appear when accessing the initial window after the first time.

These phrases can have the attributes of assiduidade(attendance) and aproveitamento(evaluation) empty for situations in which the information does not rely on the performance of the student.

The existing developed phrases are a combined effort with the pedagogical team of Universidade Aberta and are in Figure C.1.

3.7.7 Event Logs

The table depicted inFigure 3.17 is a replica of the information on the Moodle database table, this prevents direct access to the Moodle database, protecting Moodle. One important note is that additional features can revolve around this table, enabling more information to be available for the virtual tutor.

Figure 3.17: Event Log table.

3.7.8 PHP Files

To communicate with an external database without the use of sockets, the creation of web services is necessary. These web services create a connection with the database and formulate a SQL query from the parameters given and transmit the response. Additionally, the usage of web services enables the creation of additional steps to add or increase security or versatility of the web services. The connection to the database is made internally, preventing exposure of credentials. The variables sent by the client are sanitised to avoid attacks to the database. As referenced in DatabaseConnections, a hash is created using the parameters and the secret key, and transmitted from the plugin to the web service This gives the two advantages:

- Security: if the function generates an equal hash from the sanitised parameters means the client is not malicious;
- Authenticity: if the function creates an identical hash to the one received means the client has the secret key, in our case, it means it is the plugin or the back office application.

The PHP files can also modify the response from the database and convert it into a desirable format, the web services created to communicate with the external database are contained in the following PHP files:

- Connection.php: Establishes a connection with the database and supplies the secret key to other web services to validate the requests.
- Login.php: Contains the functions to manage the list of logins made by the student supplied and requested by the Tutor Virtual.
- Parameters.php: Used to insert, alter or retrieve the parameters that define the Virtual Tutor.
- Phrase.php: Supplies the phrases to the plugin based on the performance of the student.
- Question.php: Stores and sends the questions made by the students in the Questions & Answering("Q&A") section available in the interface of the Virtual Tutor application.
- Student_performance: Stores the performance of the student in a course for the day, essentially creating a history of each student.
- Tutor.php: Stores the selection of the avatar made by the student in a given course.

The services are called supplying the parameter "function".

3.8 Conclusions

This project brought the development of two programs, a Back Office application and a plugin. They communicate with each other through the use of an external database.

Through the use of web services supplied by Moodle the plugin can retrieve information about the course and the student. The preparation steps for correct operation of both applications consist of creating an account responsible for the communication and insert the token and Moodle URL in the plugin.

Chapter 4

Tests

This chapter describes tests made with the purpose of evaluating the application as describes the conclusion that can be taken from the results. The tests performed had the purpose of assuring the correct operation of the plugin facing several controlled scenarios, the impact that a WebGL Unity application can have in a server hosting a Moodle platform and tests with the participation of professors of Universidade Aberta to acquire opinions regarding the application.

4.1 Test environment

The environment available for the tests was a Moodle platform located on an Amazon server with eight GB of RAM.

The application used in the tests was developed by my colleague Catarina Alves, this application used the plugin to retrieve data from Moodle. The application depicted in Figure 4.1 is located in the course page and provides feedback to the student regarding information on the course page and the performance of the student in the course.

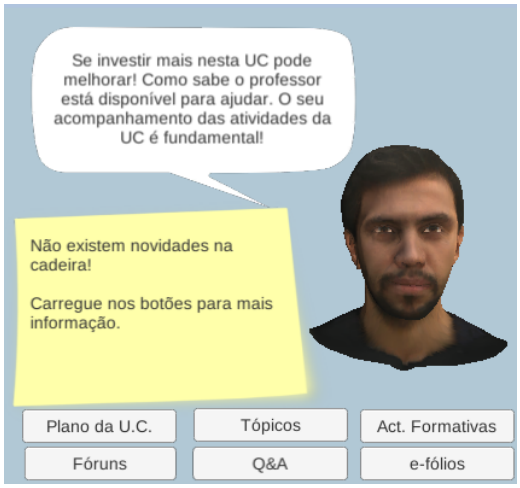

Figure 4.1: Interface of the Virtual Tutor application.

The Virtual Tutor application contains the avatar chosen by the student that gives feedback to the student via text expressed in the speech bubble and through facial expressions, these are defined using the calculations of the state of the student and the phrases received from the external database. To communicate information from the course, the post-it notes inform the student of the existence of news in forums, e-folios and the student can consult components that have news on the buttons below. For example, if a forum has a new post, the post-it will say that there are news in forums, the student then can access the list of forums clicking on the forums button, the Virtual Tutor will then show a list of the forums, on this list:

- Forums with recent news have the button red;
- Forums with some news that are not recent and have not been consulted yet have the button yellow;
- Forums without no news to report have the button the grey colour.

The student can then use the buttons to navigate through the course page and consult the desired information.

4.2 Operation Tests

The operation tests involved two scenarios that simulate a course similar to various ones presented in Universidade Aberta over a ten-day period. The course pages and students were created exclusively for these tests.

The number of test students used for the scenarios involved the possible cases of changes in Evaluation of the student in the course duration. The created test students were distributed between the team at Faculdade de Ciências da Universidade de Lisboa and the interactions with the course page are described in Table D.1, Table D.2 and Table D.3. The participants would perform the tasks in the interaction tables and would take notes of the feedback of the virtual tutor and the news information.

The objective of these tests is to evaluate the performance of the plugin regarding extracting data, the feasibility of features and possible alterations to the equations in an environment similar to an actual course. Additionally, these tests were performed to verify if the application could distinguish news in the course page using only the web service provided by Moodle.

The calculations for the attendance part of the state of the student were incomplete at that period of time, this is due to the attendance formula not being defined. As such, Equation 4.1 helps to determine if the plugin is capable of calculating an

equation for attendance consistently, using the variables that change during the course duration.

$$\begin{aligned} Attendance = LoginImp \times \frac{\text{nº of logins made}}{\text{nº days since start of the course}} \\ + PostImp \times \frac{\text{Posts made}}{max(\text{PostMax, Posts made})} \end{aligned} \tag{4.1}$$

Where:

- LoginImp - Importance of Login = 65
- PostImp - Importance of Posts = 35
- PostMax - Maximum threshold for Posts = 3

For the calculation of the Attendance, the number of days the student logged in and the number of posts made was used because it is important for the student to access and interact with the course page to remain up to date with the course.

4.2.1 Two folio course Tests

The test environment is a custom-made course named "Two folio course", as the name implies, the page contains two e-fólios. These serve as the evaluation for the test students, the e-fólios are part of the category "e-fólios" that make part of the continuous evaluation.

The structure of the course is as such:

- One introduction that contains the announcements forum and the curricular plan;
- One topic for each e-fólio of the course accompanied by a forum to discuss the assignment and forum to discuss the topic.

The test students created are based on the possible evaluation results shown in Table 4.1, and the actions made in Table D.1. These students belong to the group "Avaliação continua" which indicates their desire to complete the course by continuous evaluation, i.e. the course evaluation includes e-fólios.

4.2.2 Three folio course Tests

The test environment is a custom-made course named "Three folio course", as the name implies, the page contains three e-fólios, these serve as the evaluation for the test students, the e-fólios are part of the category "e-fólios" that make part of the continuous evaluation.

The structure of the course is as such:

Starts	E-folio 1	E-folio 2	State	Student
Low	44	46		John1
Low	44	44		John2
Middle	68	75		John3
Middle	68	68		John4
Middle	45	43		John5
High	70	70		John6
High	70	65		John7

Table 4.1: Evaluation for test students of e-fólio 2 course.

- One introduction that contains the announcements forum and the curricular plan;

- One topic for each e-fólio of the course accompanied by a forum to discuss the assignment and forum to discuss the topic.

The test students created are based on the possible evaluation results shown in Table 4.2, and the actions made in Table D.2 and Table D.3. These students belong to the group "Avaliação continua" which indicates their desire to complete the course through continuous evaluation, i.e. the course evaluation is made using e-fólios.

Stats	E-fólio 1	E-fólio 2	E-fólio 3	State	Student
Low	44	80	86		Stan1
Low	44	46	43		Stan2
Low	44	46	46		Stan3
Low	44	44	44		Stan4
Low	44	44	50		Stan5
Middle	65	80	50		Stan6
Middle	65	80	80		Stan7
Middle	50	35	35		Stan8
Middle	50	45	60		Stan9
Middle	65	65	80		Stan10
Middle	50	50	30		Stan11
High	70	30	30		Stan12
High	70	50	50		Stan13
High	70	65	90		Stan14
High	70	70	70		Stan15
High	70	70	40		Stan16

Table 4.2: Evaluation for test students of e-fólio 3 course.

4.2.3 Results

During the tests, the plugin showed the ability to calculate the formulas for evaluation and attendance within the desired expectations.

During the tests, it became apparent that the current formula to calculate Attendance would prove ineffective as the login section would impact the calculated value in a substantial manner, leaving a student with a low initial attendance with hardly any form to recuperate.

With the cases conducted, it was deemed insufficient to rely solely on the web service provided by Moodle to inform the student of news in the course page, this is due to the inability of the application to identify information already access by the student. To solve this problem the usage of the logs containing the interaction of the student with the course page was needed, leading to the final solution of comparing news to recorded interactions of the student replicated in the external database.

Using the cases provided by the tests, allowed for the development of Equation 3.5, Equation 3.6 and Equation 3.3.

4.3 Stress Tests

Since initialising the Unity environment in the course serves as an additional burden to the Moodle server, exists the necessity for tests that verify the ability of the server to support the application running without causing undesired behaviour in the server.

The plugin communicates with the Moodle using the token of a dedicated account, Moodle perceives as that account logging in and requesting data, and this effectively transforms the act of using the plugin in a Moodle course as another user using the course page at the same time.

With this in mind, the server needs to be able to support twice the communication. If the server contains flaws such as memory leaks, it can lead to undesired behaviour from the server such as poor performance.

4.3.1 Sequential Tests

The sequential tests performed aimed to evaluate the performance of the Moodle server facing several logins in sequential order.

A python script was developed to perform the tests. It executes the act of logging in, accesses the course page and logs out when the virtual tutor has finished retrieving data. The script uses image recognition using the method locateOnScreen from the pyautogui module to identify objects on the screen and act accordingly, requiring a previous sequence of manual interactions with Moodle to create the

desired images to be used in the script.

This cycle repeats the desired number of times and the accounted time is stored in a log file.

The course page used for the tests was the previously referred “2 e-fólio Course”. The student used was the test user “John3”.

4.3.2 Parallel Test

The parallel test aimed to evaluate the performance of the server facing several logins made at the same time.

Much like the previous test, this aimed to evaluate the performance of the server facing multiple communications. With the combined effort of ten individuals, the concurrent test involving the login of twenty users was possible.

In this test, the previously created course page “3 e-fólio Course” was used as the environment, and the “Stan” student combined with newly added “Stan”s as agents.

Each participant in the test used the Chrome browser to login with one account, and the incognito mode to log in with another. This is due to Moodle maintaining the login through the use of a cookie and this cookie prevents the use of additional tabs to log as different users.

4.3.3 Results

The sequential tests show on average the application taking around 15 seconds to start as shown in the following Figure 4.2.

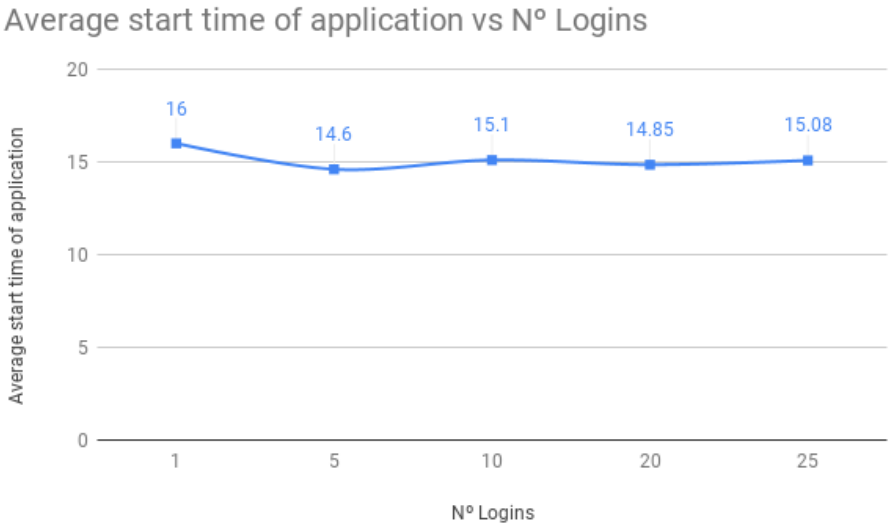

Figure 4.2: Average of start time regarding number of logins.

The primary cause for the extended time is Unity needing to initialise its environment in every course page access. This process takes approximately ten seconds as my colleague Catarina Alves points out, putting the information retrieval process to take about five seconds.

With the previous information in mind, taking into account the restrictions in Moodle is worrisome. The gathering of data of the course combined with the time Unity takes to initialise can be too long for most users.

The parallel test shows the ability for the Moodle server to hold high amounts of data transference over twenty accounts at the same time without causing undesired behaviour in the server, this amount compares to forty since the plugin functions as an account.

4.4 Tests with Experts

The tests with experts that were teachers at Universidade Aberta revolved around experts visualizing the usage of the information gathered by the WebManager plugin through the interaction with the Virtual Tutor application developed by my Colleague Catarina Alves. The expert volunteers acted within a set of activities and gave feedback regarding their experience. In the following section, the results of this experiment are presented and analysed.

4.4.1 Environment

The previously created course page "3 e-fólio Course" is used for the course environment, thirty mock-up students composed of already created students for the three folio course tests and new students were used as the accounts for the tests. The students are divided into groups of two, one of the students will have the parameters of evaluation and attendance in the "High" making it a good student with the tutor already defined, while the other will have the parameters in "Low" making it a lousy student who has not yet chosen the tutor.

4.4.2 Preparation

The expert views an introductory video. After that, the volunteer is accompanied in performing activities already defined in the activity script in the role of two students, one with evaluation and attendance low and other with the parameters high and giving commentary on the appearance and dialogue of the Virtual Tutor as well as answering the question proposed. When the activity script is complete, the volunteer is invited to answer a questionnaire which contained:

- Questions that helped identify the profile of the user;

- A combination of questions that used a Likert scale and questions with multiple answers and open answer to analyse the expectations of the professors regarding the application and the service provided.

4.4.3 Results

Participants

The group of participants was composed of fourteen volunteers. The participants had a majority in the age group between 41 and 44 (72% of the total participants, evenly distributed in gender) and the remaining experts fit in the age group over 55 (28%, 21% female and 7% male) as shown in Figure 4.3.

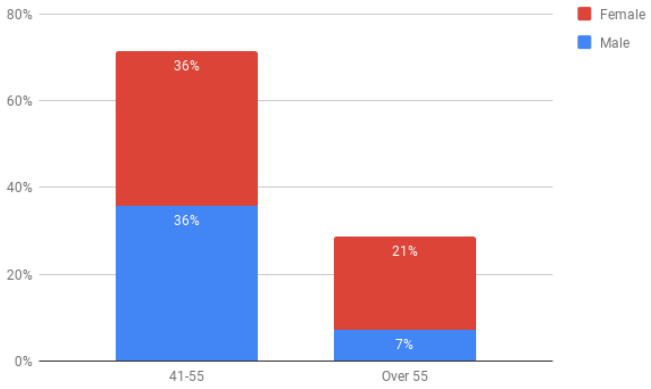

Figure 4.3: Age group of participants.

Figure 4.4 shows that 57% of the experts are female while 43% are male.

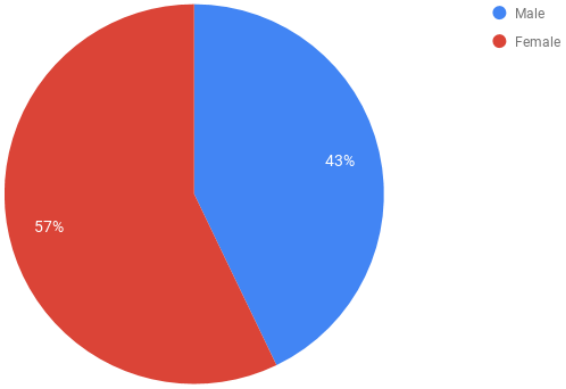

Figure 4.4: Gender division of the experts.

System Usability Scale - SUS

SUS consists of a questionnaire with ten items that have five response options on a Likert scale, from Strongly agree to Strongly Disagree. The even-numbered questions are an opposition to the odd-numbered to retain the expert attention to the context of the questions[15]. The calculations for the SUS score are as follows: - Take one value from the response of the odd-numbered items; - Make the subtraction between the value of 5 and the score attributed in the even items; - Make the sum of all the previously acquired values; - Multiply the value obtained in the previous step by 2,5; - Make the average of all the previous values.

The SUS score is between zero and a hundred, and if the score is above 68 indicates the data is considered reliable. The calculated value for the expert tests is 78.39; therefore, the sample is deemed to be reliable.

Expectations vs reality

The participants responded to a series of statements that can help to get a grasp of the view that the expert has on the application, one of the statements was "The utilisation of the virtual tutor:", this question had five possible answers to be selected and the added option of providing an open answer, to simplify the received feedback, the open answers were not added to the graph however they are taken into account in the conclusions section. A graph of the established responses is presented as Figure 4.5.

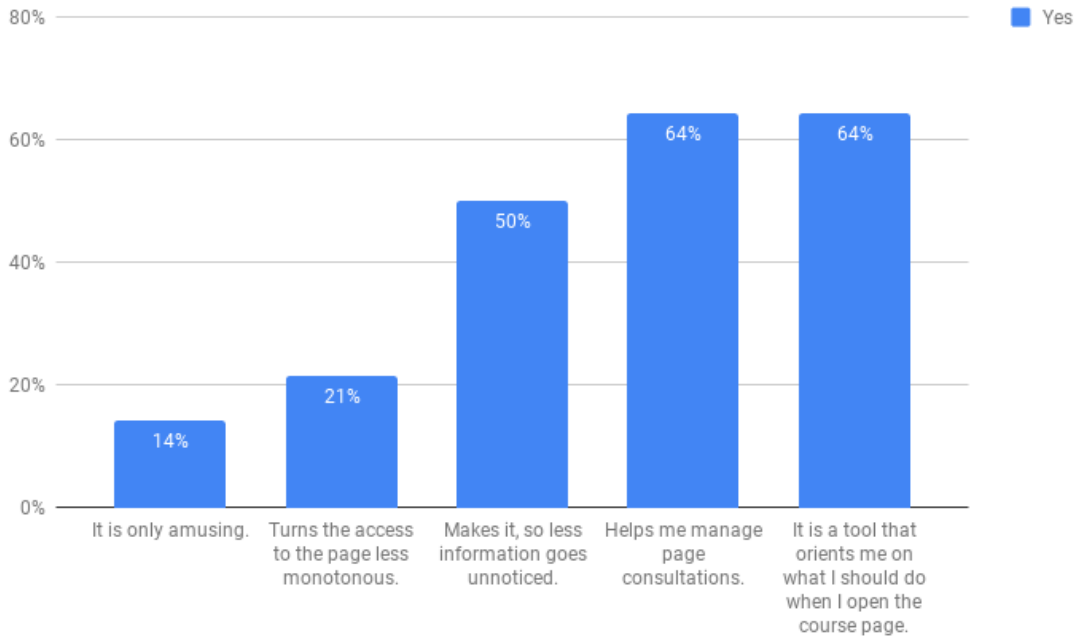

Figure 4.5: Opinions on the utilisation of the virtual tutor by the experts.

As depicted in the graph, a minimal quantity of the experts conclude the appli-

cation is only amusing (14%). A higher, albeit a still small number of the experts, find the app to make the access to the page less monotonous (21%).

In our opinion, a reason for this low number is due to the application taking an extended period to load, and this causes the access to the page to remain as or become even more tiresome than usual. However, the experts also highlight the usefulness of the application in orientation and information relaying, 64% indicate the application helps them manage page consultations, 64% consider the application as a tool that can orient the user when the course page is open, 50% fell the application makes so that less information goes unnoticed . As for what advantages the virtual tutor application can bring to the course page, the experts highlight the news indication, the positive effects of an interactive medium bring to an e-learning course, for example, increasing of focus on information on the course, and supply an interactive medium to improve the interaction of the student with the course removing the unease feeling when presented with vast amounts of information. Experts also desired for the application to contain cognitive capabilities to respond to questions, discern information on forums and indicate links that can help the student.

Facing the statement "The Virtual Tutor, in a general sense, corresponded to my expectations.", a small amount of the experts consider that the application did not live up to expectations (7% strongly disagree and other 7% lean towards it), while 36% stand neutral and 50% lean towards living up to expectations (14% strongly agree while 36% lean towards it), the statistics can be view in Figure 4.6.

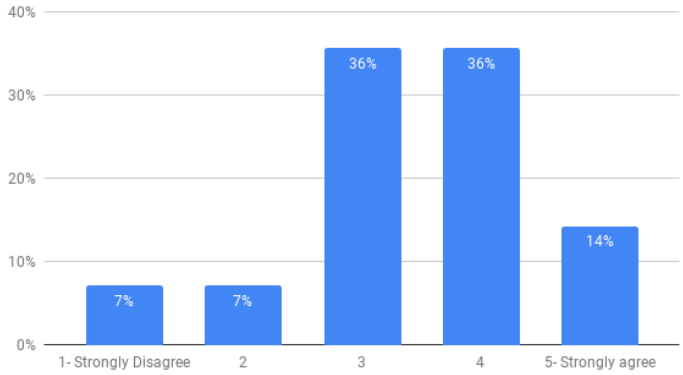

Figure 4.6: Statistics regarding the statement "The Virtual Tutor, in a general sense, corresponded to my expectations".

'A visual component in the main screen used in the interface of the Virtual Tutor application resembles a yellow "post-it" note, on this component the information regarding news is presented. The news is relayed by indicating there is news in a specific type of object in the course page such as assignments or forums, prompting the user to access the appropriate screen to know more about the information relayed, for example, when an e-fólio (assignment) receives changes, the "post-it" will

show “there is news in the e-fólios”.

Figure 4.7 shows the statistics regarding the opinion of the experts about the usefulness of the “post-it” component.

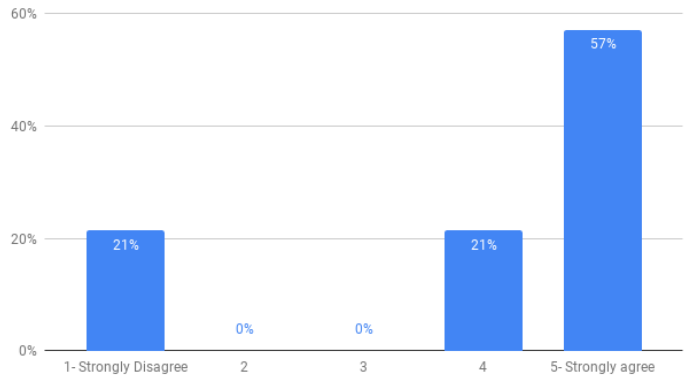

Figure 4.7: Statistics regarding the statement “The indication in the yellow “post-it” helped me identify news in the course”.

A vast majority of the experts consider this component to be helpful in gaining information about the news in the course page (57% strongly agree while 21% lean towards it), only having 21% of the experts to consider the component not to be helpful.

4.5 Conclusions

During the operation tests, additional alterations were made to the plugin to ensure the expected behaviour during the information retrieval and storage process and develop or improving the corresponding formulas to calculate the state of the student.

With the results from the stress tests, we can conclude that the plugin developed in the project requires a noticeable amount of time to retrieve the information due to the limitations in WebGL, that combined with the initialisation time of Unity can create extended waiting times. Adding to the already noticeable issue the necessity of loading the application every time the page starts, can deter the usage in the Moodle page.

With the tests and interactions with experts, it can be concluded that restrictions and setbacks existent on the Moodle application variant have a substantial impact on the evaluation made by the experts. Despite this, the experts also stated that the application allows for more management of the information by the part of the student, alleviating the pressure caused by a large amount of content present in the course page.

Chapter 5

Conclusions and future work

5.1 Conclusions

In the scope of the project, the creation of a plugin was proceded, capable of communicating with a Moodle server in a WebGL or Android environment and retrieve relevant information that can allow an application to acquire student information and identify the current period in the course.

For the plugin to have access to the necessary information, the Moodle web-services were used to communicate with the Moodle server and custom-made web services were created and placed in the external database to store and retrieve information not accessible in Moodle. The creation of a provisional back office application was necessary to accompany the previously mentioned plugin. The back office enables configuration of the thresholds in the external database while the integrated back office by the team at Universidade Aberta finishes development.

The development of the plugin and application was performed using Unity to allow the creation and deployment of the plugin with relative ease in the Unity environment. Additionally, the plugin and back office allow for the addition of complementary features with the objective of enhancing the user experience.

Formulas were developed and implemented in the plugin to assess the state of the student, the variables that allowed for the definition of these formulas involved variables defined by the professor for the course and the interactions made by the student to provide feedback regarding the learning experience of the student.

Following the creation of the plugin, tests to evaluate several components of the work developed were made:

- Operational tests were designed to assess and fine-tune the operation of the plugin facing information interpretation in mock-up course pages created in the test Moodle server;

- Stress tests to evaluate the load generated by the application on Moodle to

determine the viability of the developed work concluded that the application takes an extended period to start, enough time to be noticeable;

- Expert tests with volunteers professors from Universidade Aberta were made to assess the impression made by the application created by Catarina Alves that contains the plugin, and receive feedback regarding the developed work.

5.2 Future Work

With the integrated back office development complete, it is imperative to alter the communication with the database, as the format of the response and the request will change to mimic the web services of Moodle. Additionally, the order of the requests must change to enable a faster data acquisition, discarding the dependence on the update thread to verify the acquisition data in the Webmanager controller class before requesting additional information. The creation of custom-made web services in Moodle is a vital step towards reducing information acquisition time.

Finally, to remove the most highlighted problem with the application in Moodle, a different approach must be made to the start-up of the application, currently being tied with the HTML page of the course makes the start-up of the application more aggravated since by changing pages the application starts again, possibly launching a separate page that hosts the application can solve the issue.

Appendix A

Technical manual

A.1 Application Highlight

The project uses Unity as a platform to display the avatar. Since Unity uses the program language C# as the primary supported language, it was the primary language used in this project.

The project has two components:

- A plugin with the name Webmanager that contains C# scripts that communicate with the web services provided by the Moodle platform and with ones custom-made located in an external server and store information.
- A back office application that lets professors define the parameters that influence the behaviour of the virtual tutor depending on the student performance.

A.2 Webmanager Plugin

The WebManager plugin is composed of a set of C# scripts that store information requested to Moodle and an external database and calculates the student performance based on data retrieved.
web services supplied by the Moodle platform were used to communicate with Moodle. A token associated with an account by the name of "Tutor Virtual" was used to gain access to the web services, said account needs to have the role "manager"("gerente") in the course page, this limits the service to be used only in the desired courses.
The account does not have a traditional method of login(email and password). Instead, the only way to act with it is through web service authentication, prohibiting the use of the account in ways it was not intended.
Additionally, to prevent the usage of the token for unintended usage, it only has permission to use web services that are capable of reading data. The web services used are shown in Table 3.1 and Table 3.2.

A.3 Moodle web services

Moodle has built-in web services that allow other systems to perform operations in Moodle. To enable the web services, the following steps are required:

1. Activate the web services in the advanced features of Moodle;
2. Activate the protocols of the web services;
3. Create a user with the necessary permissions to make the requests. The attribution is done with a role;
4. Create a service and add the desired functions;
5. Associate the user to the service and guarantee that the user has all the permissions;
6. Generate a token of the service for the user.

The presence of the generated token on every request is necessary.

A.4 How to install plugin

To incorporate the plugin in a Unity application the following steps are required:

- Drag the plugin file to the assets environment, alternatively, chose Assets→Import Package→Custom Package (A.1) and chose the desired file.

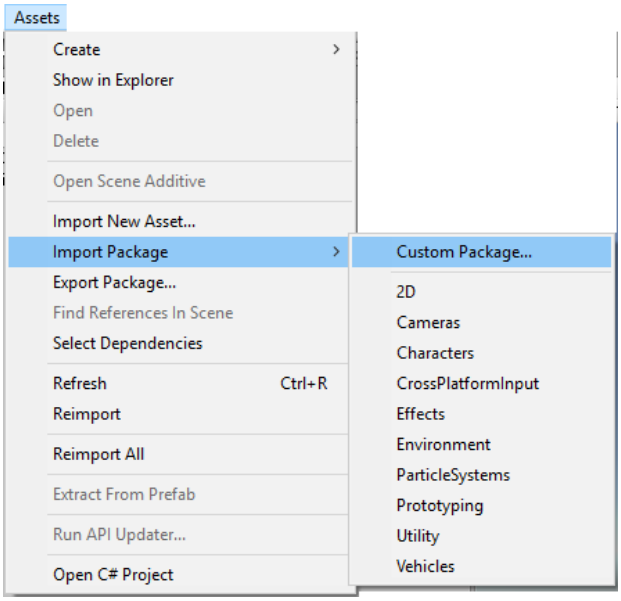

Figure A.1: How to import a plugin

- After selecting all the files, click Import (A.2).

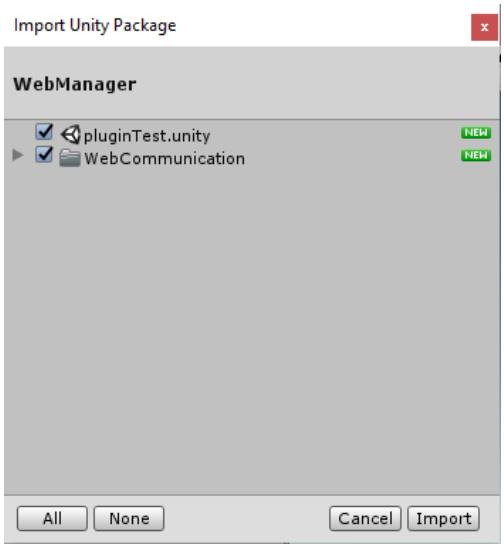

Figure A.2: Import window

- In the assets environment, open Webcommunication→Prefabs and drag the prefab moodleLogin to the Scene.

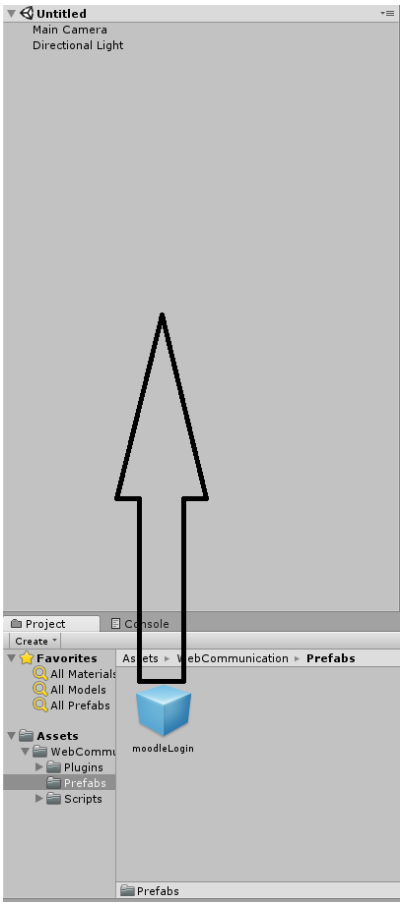

Figure A.3: Drag prefab

A.5 Working with the plugin

The plugin initiates the information retrieval process with the method MakeConnection in the WebManager component of the moodleLogin gameObject. Depending on the Unity platform, the process acts differently and requires different data:

- Android: In this environment, the plugin expects to have received the username and password of the student via the methods getUserName and getPassword respectively.
- Non-Android: In this environment, the plugin expects to have received the id of the course and the student via the methods Get_t and Get_userId respectively.

The moodleLogin gameObject contains a component of the type DataManager. It is responsible for supplying the information about the student and of the courses.

Additionally, the component contains global variables of the type Boolean that help discern when a specific type of information was collected, between the variables exists one with the name EveryThingDone, this variable serves to inform that all the information was received, to prevent conflicting or invalid information, it is advised to only access the data when the boolean is true.

Appendix B

UML class diagram

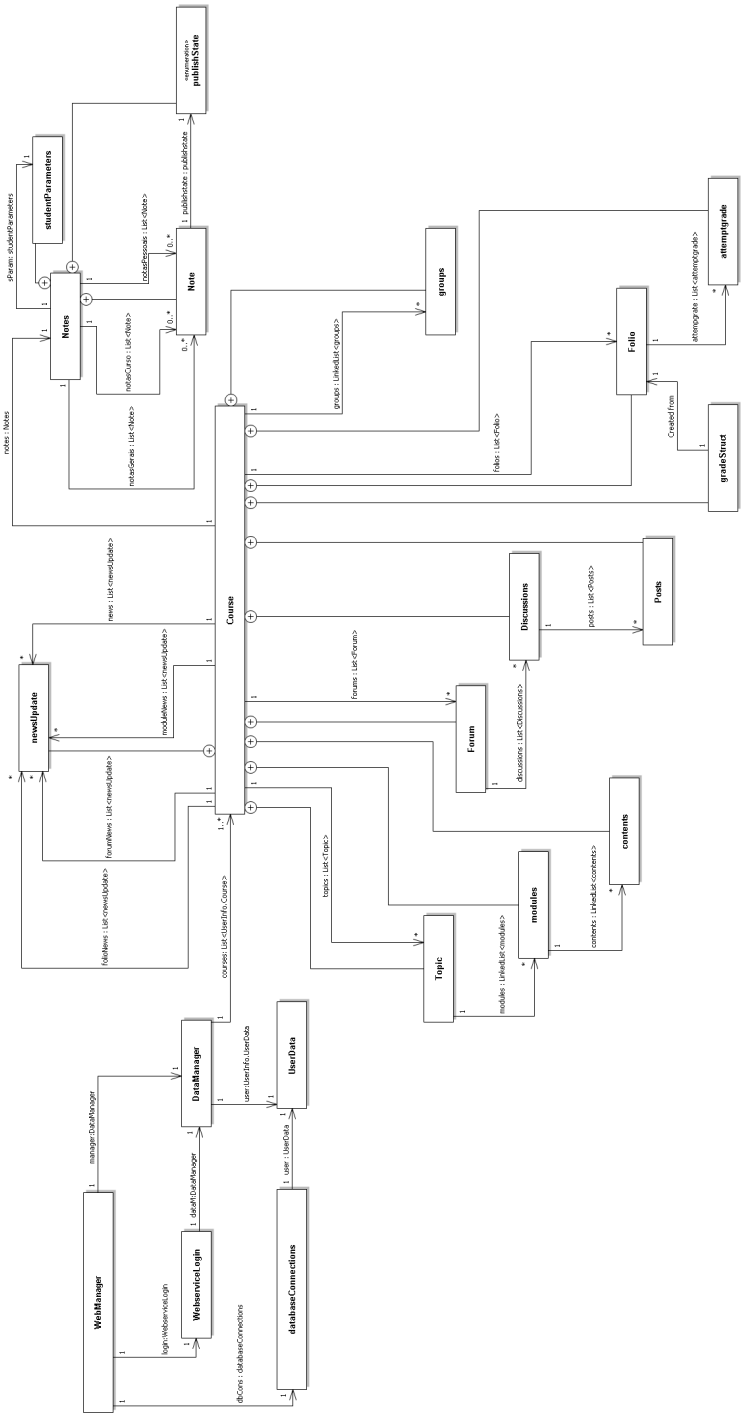

Figure B.1: Simplified structure of plugin classes.

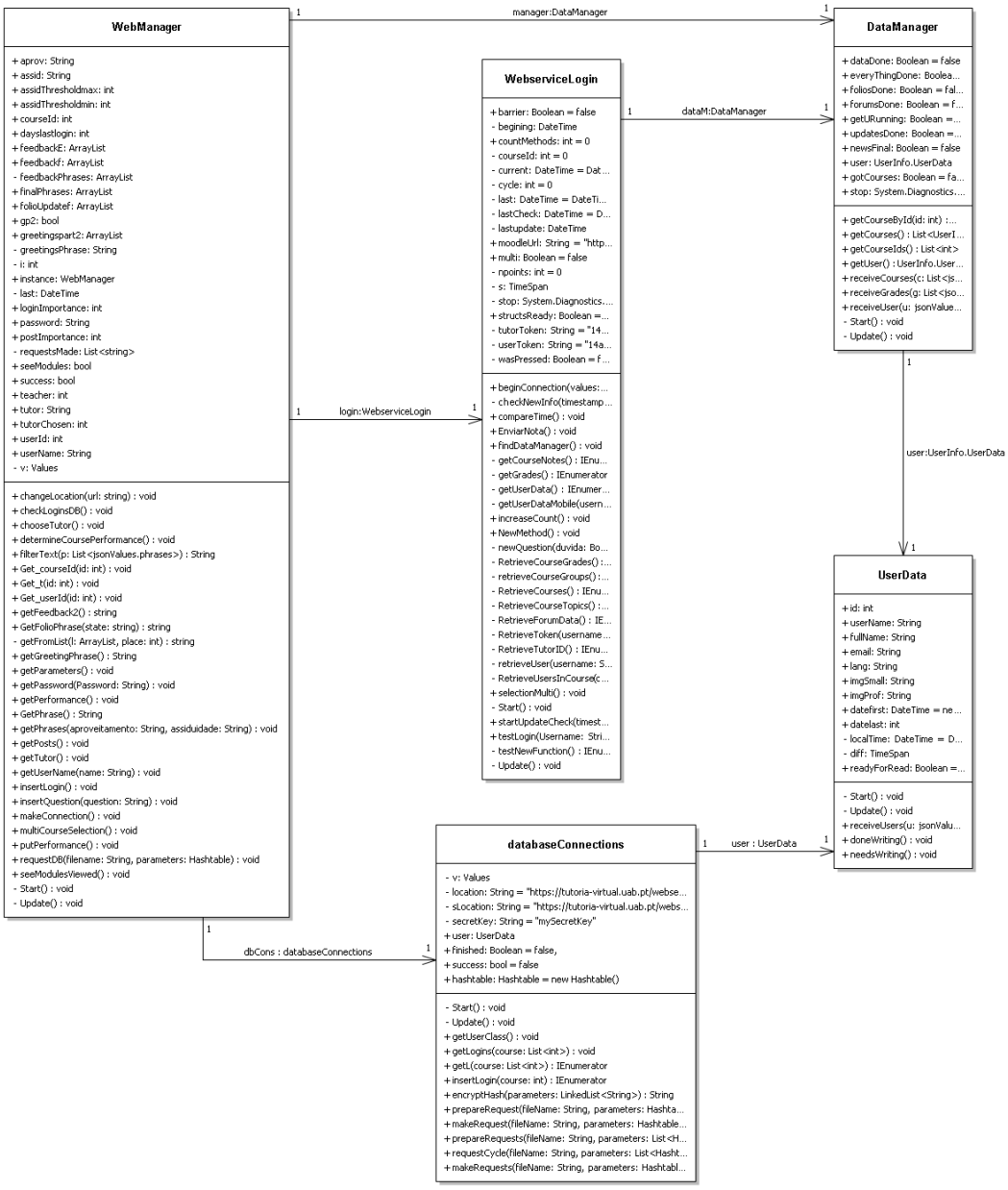

Figure B.2: WebManager plugin Class Diagram - Information Retrieval

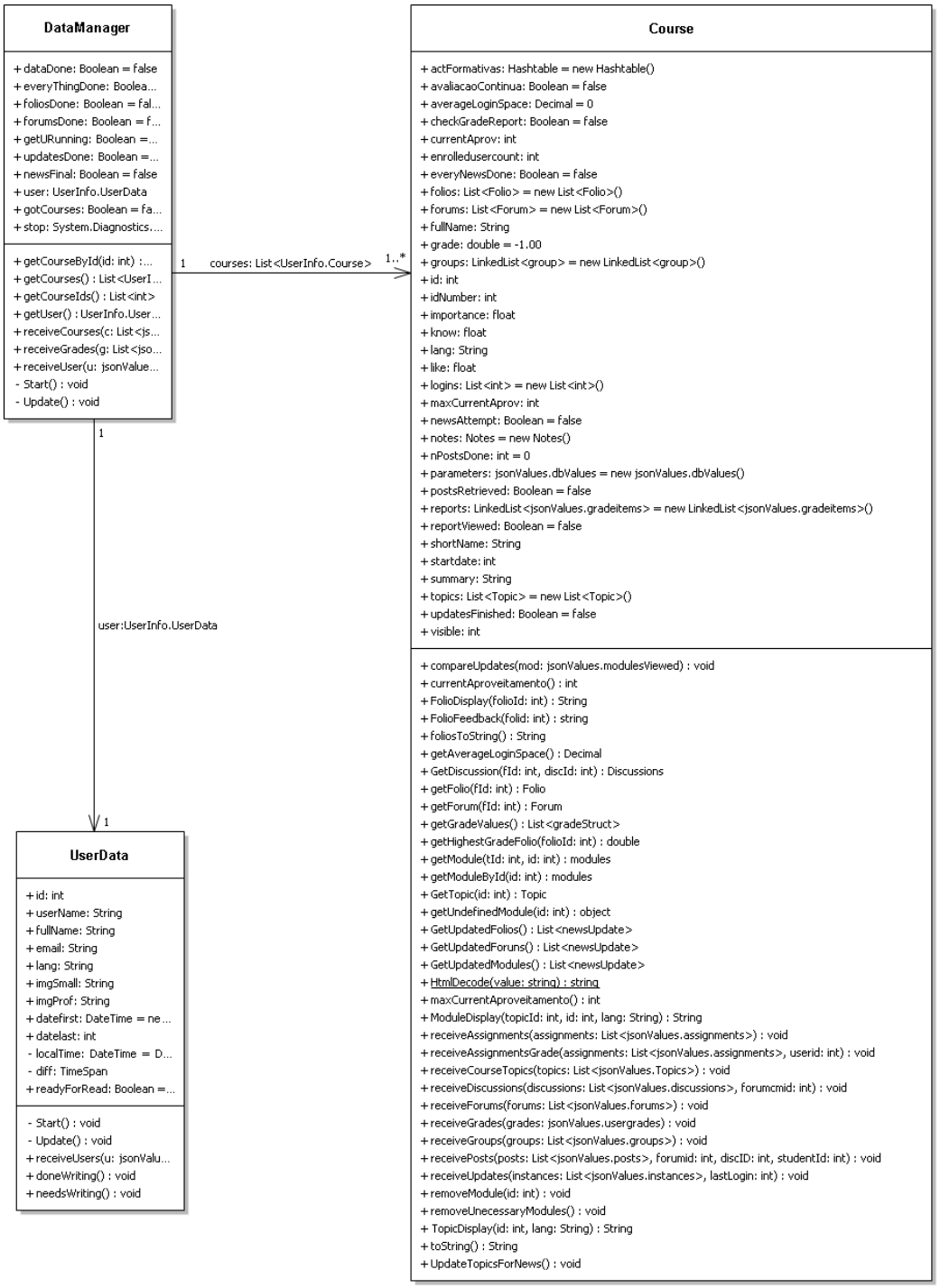

Figure B.3: WebManager plugin Class Diagram - Information Storage

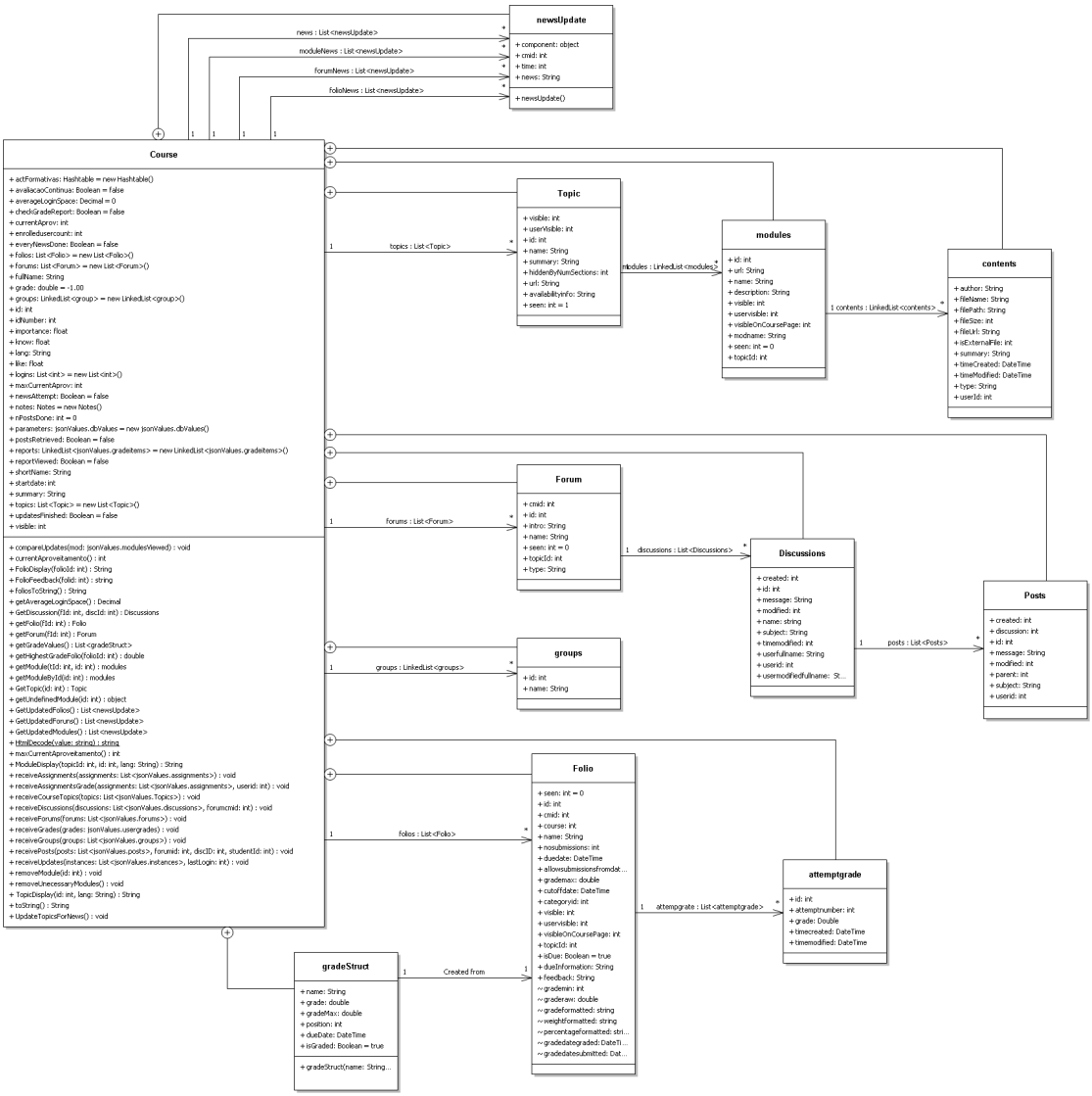

Figure B.4: WebManager plugin Class Diagram - Course Data

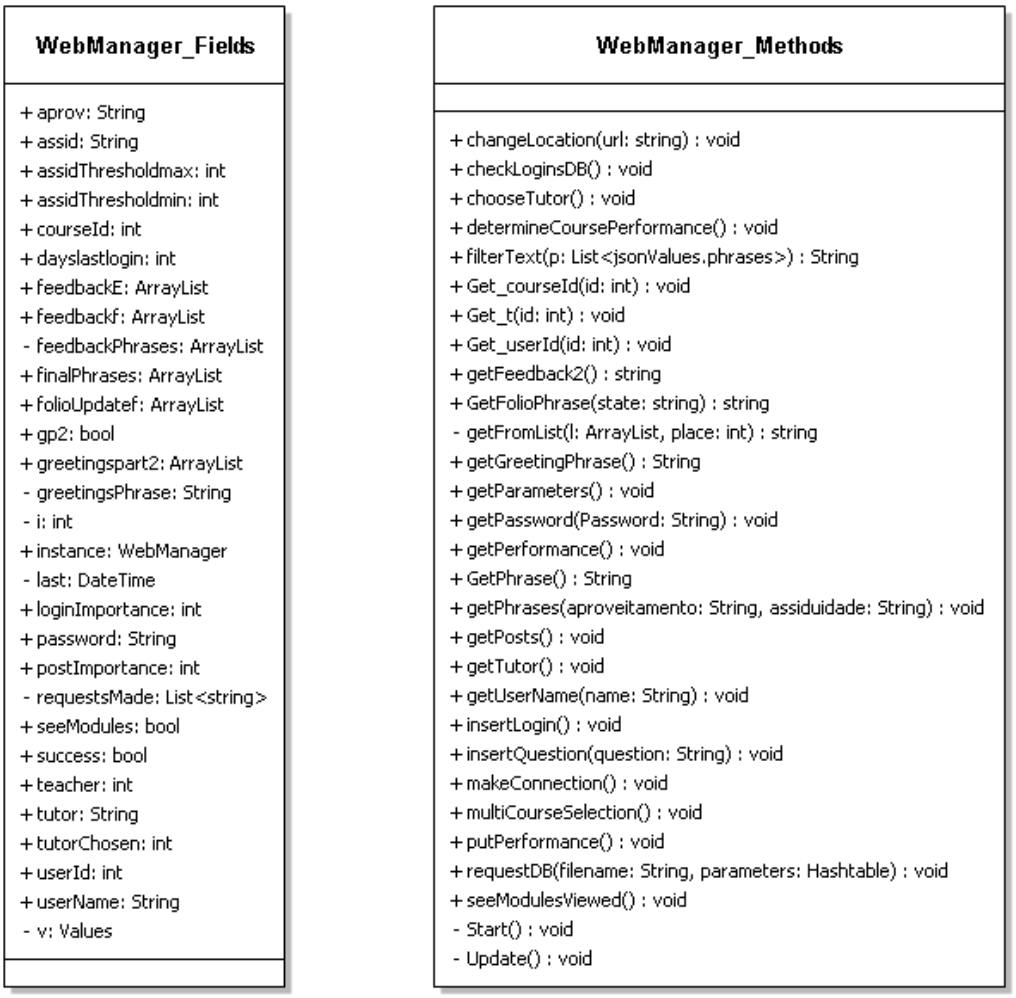

Figure B.5: WebManager class.

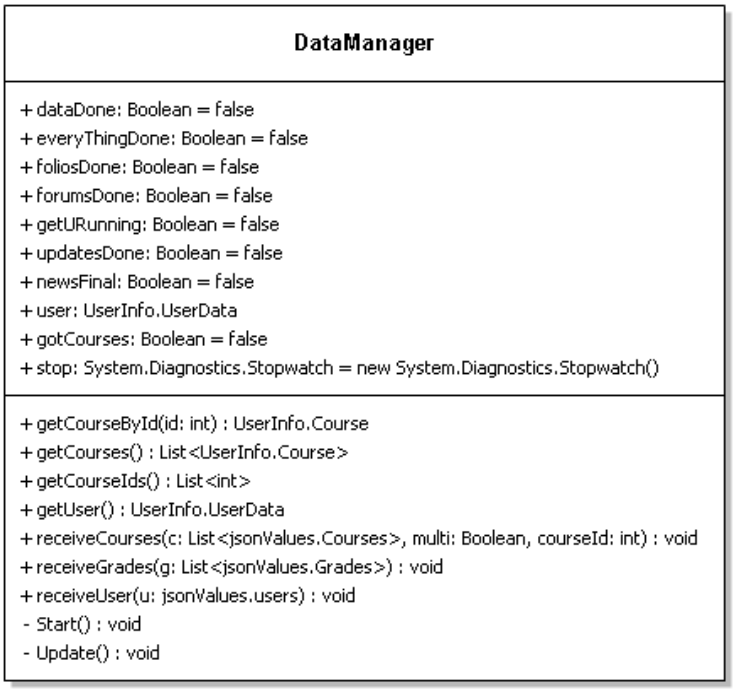

Figure B.6: DataManager class.

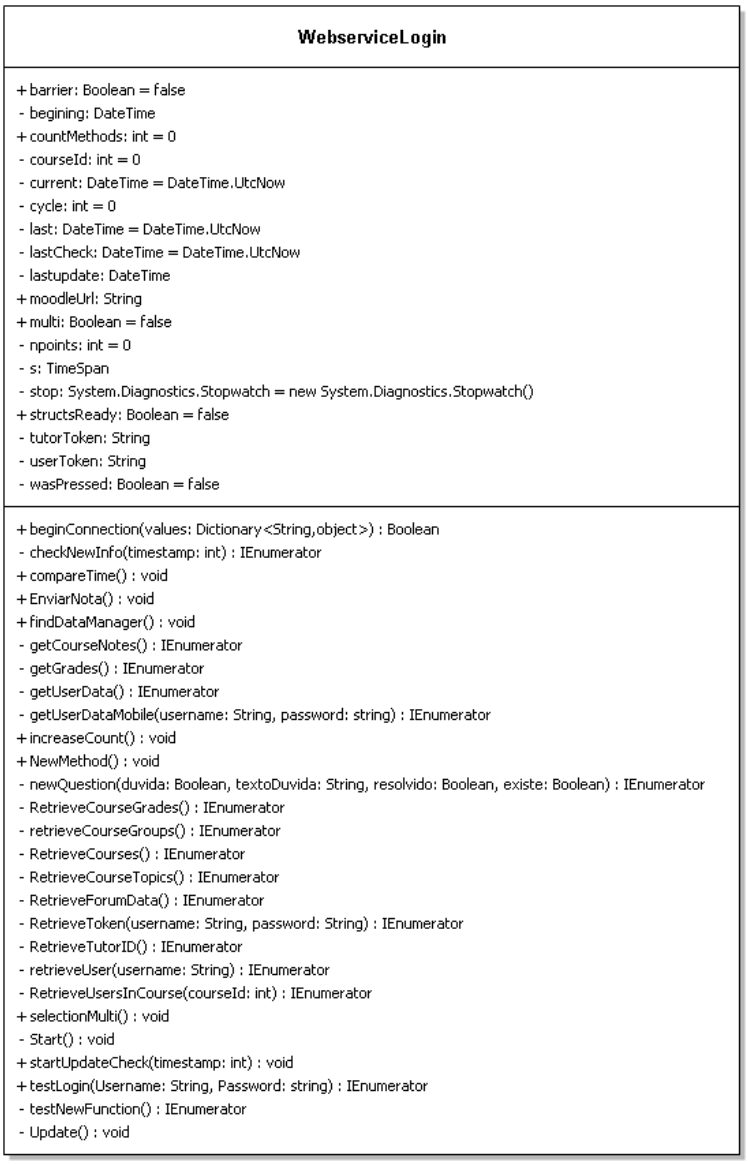

Figure B.7: WebserviceLogin class.

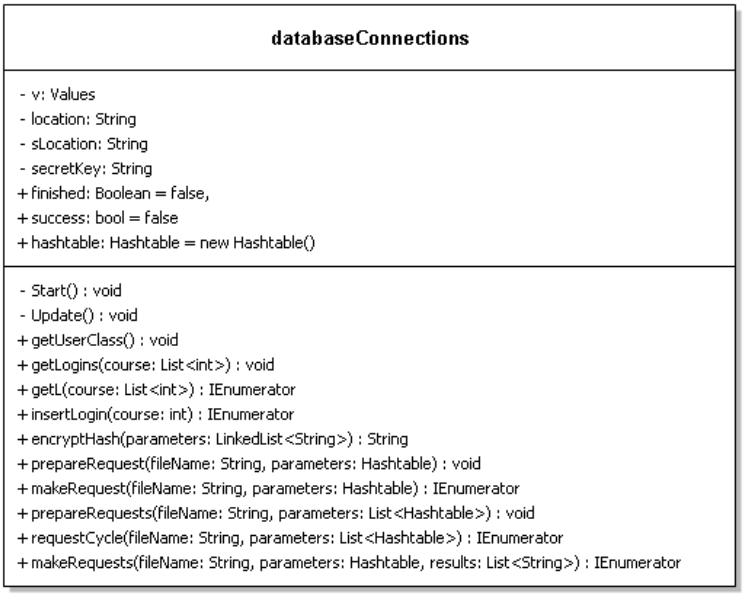

Figure B.8: DatabaseConnections class.

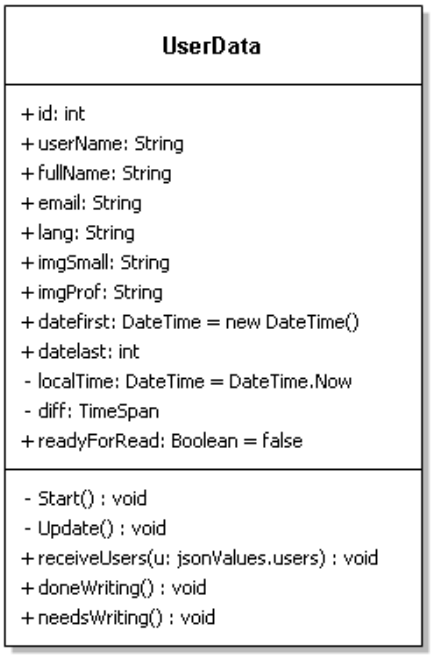

Figure B.9: UserData class.

Figure B.10: Fields of the Course class.

Figure B.11: Methods of the Course class.

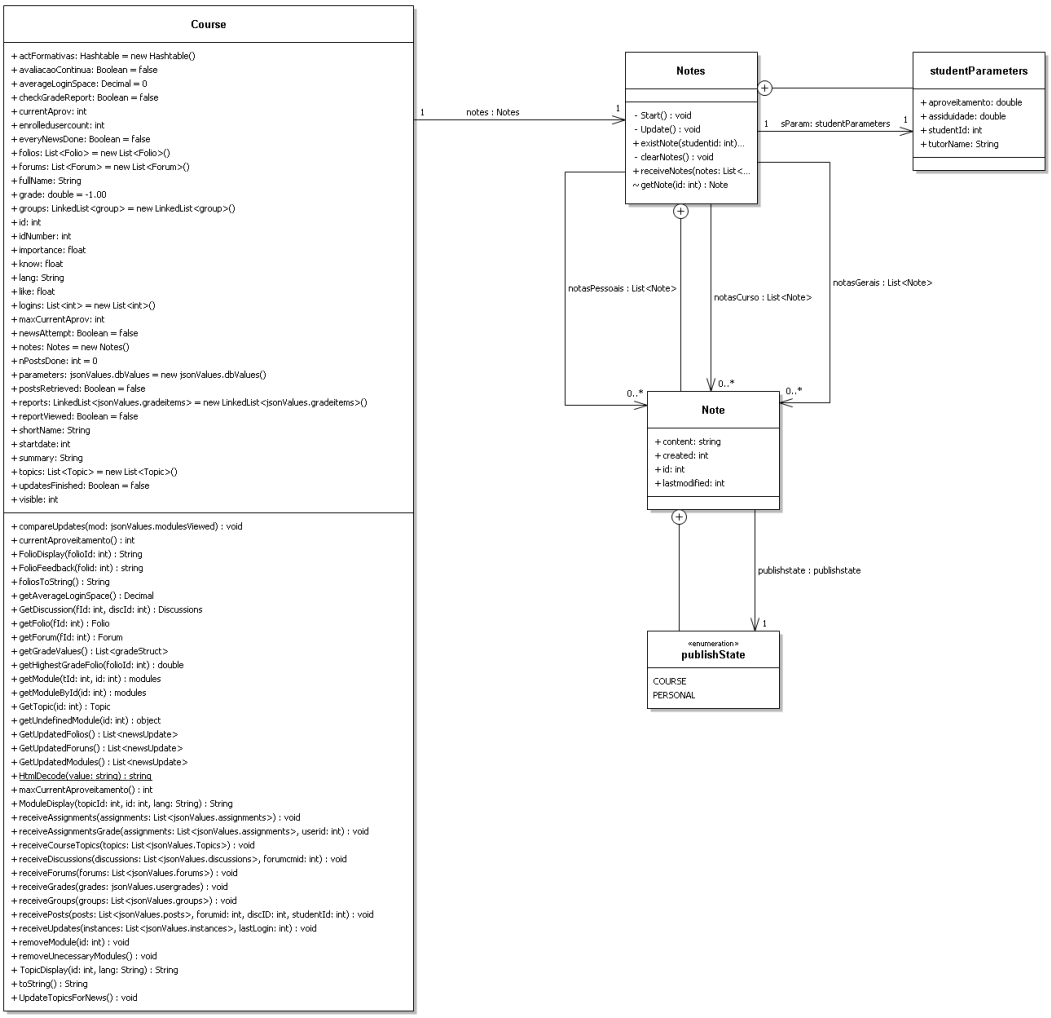

Figure B.12: WebManager plugin Class Diagram - Notes

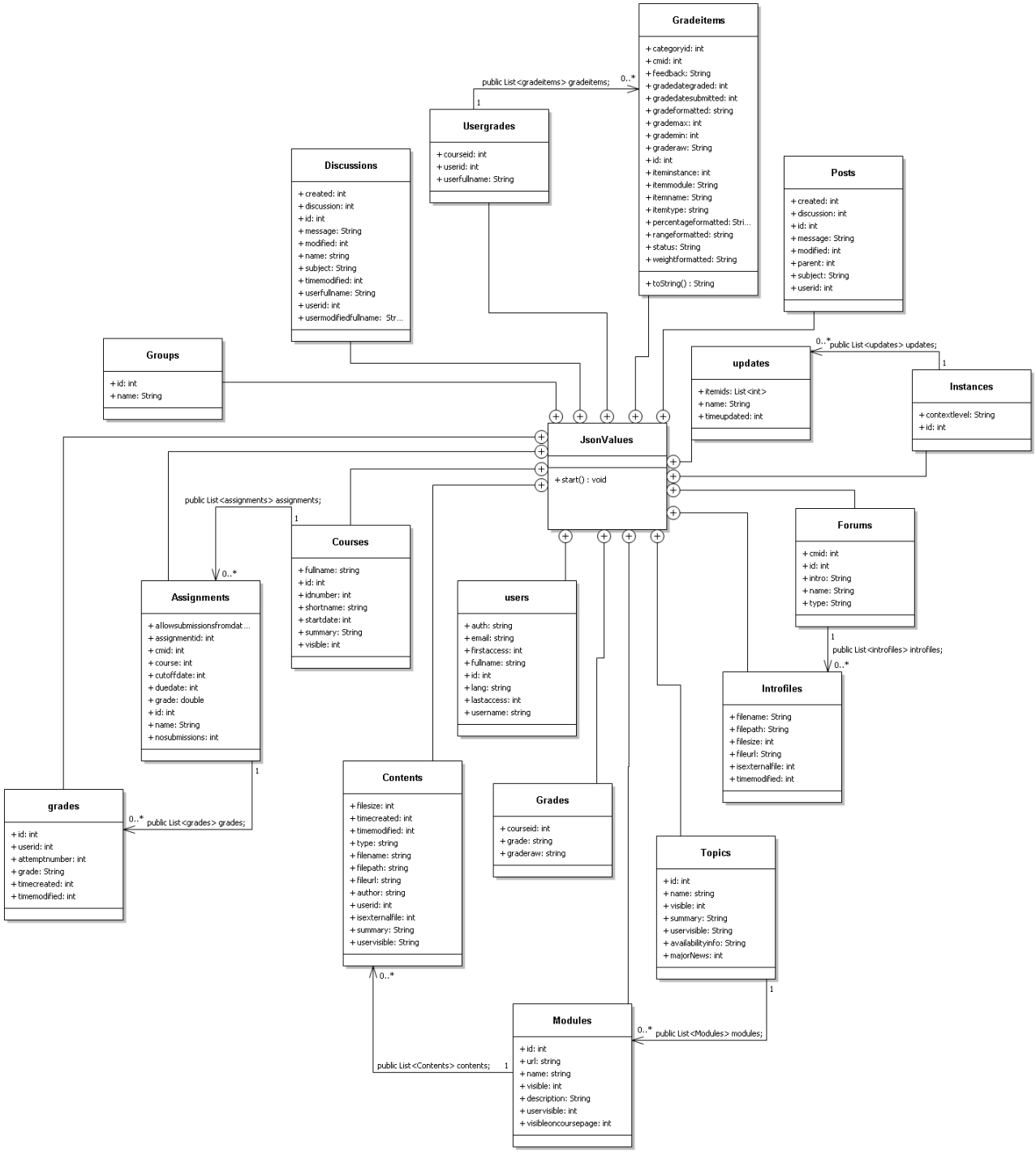

Figure B.13: JsonValues Class Diagram - structures from Moodle

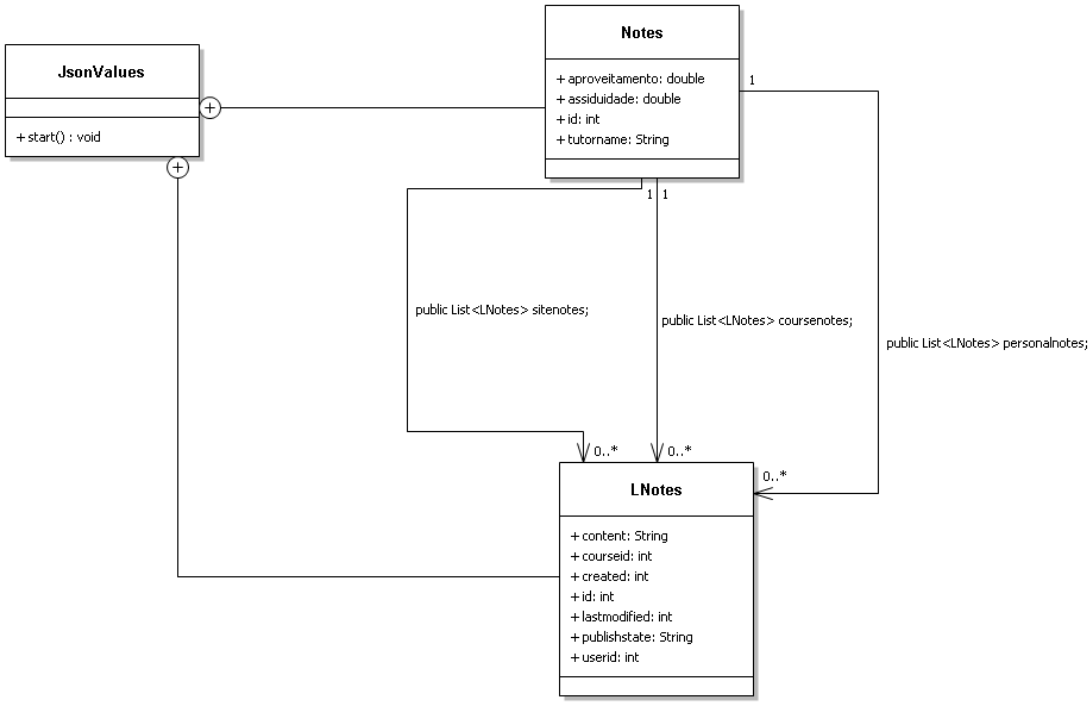

Figure B.14: JsonValues Class Diagram - Notes

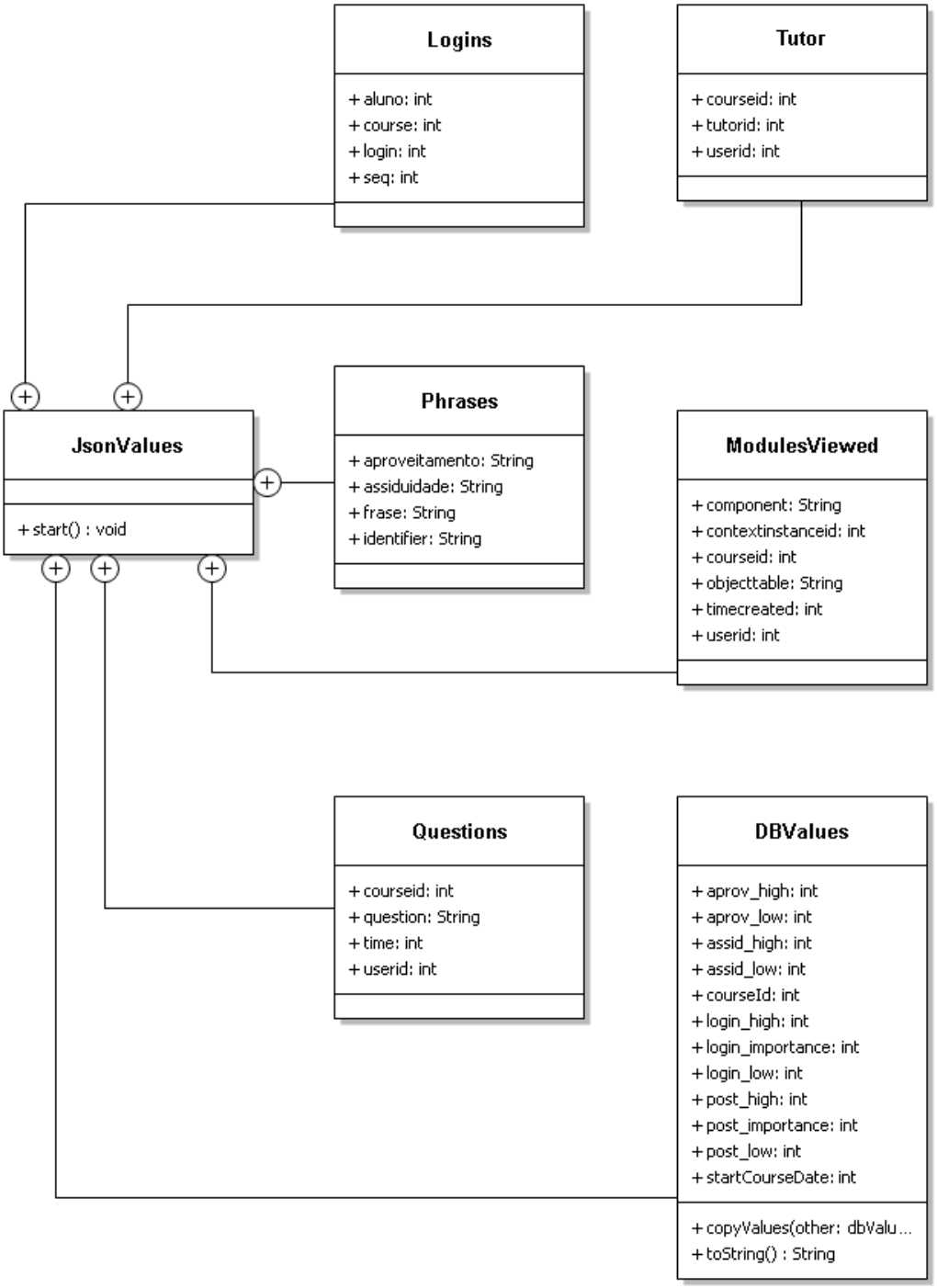

Figure B.15: JsonValues Class Diagram - Structures from External Database

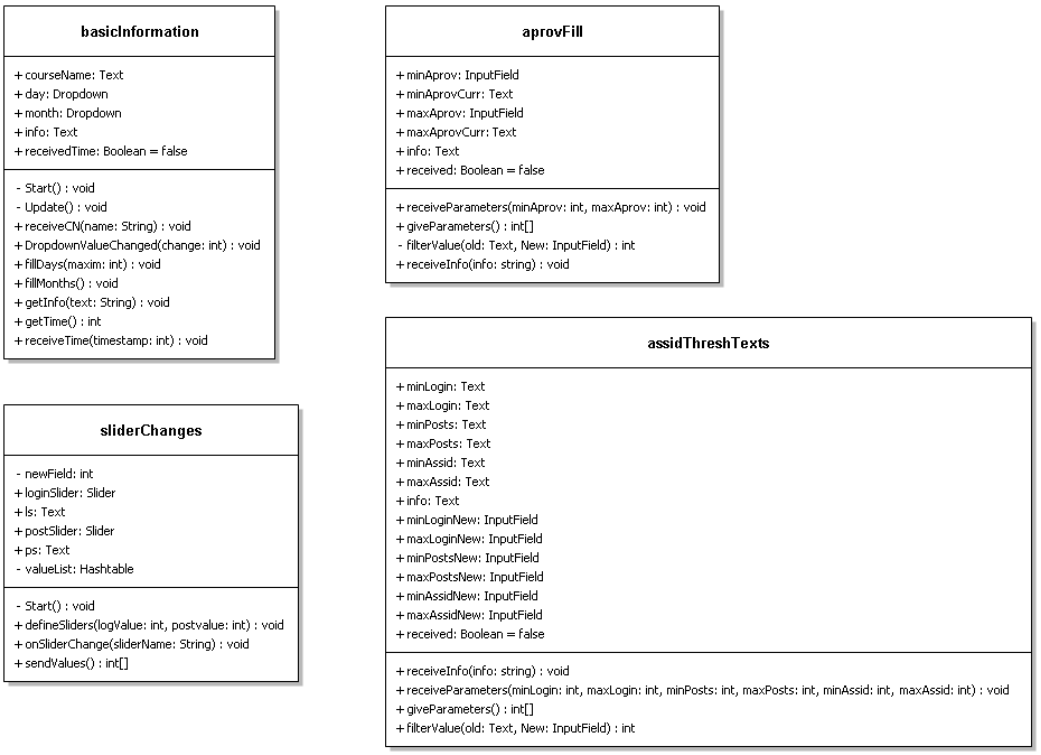

Figure B.16: Back office Class Diagram - Screen information retrieval

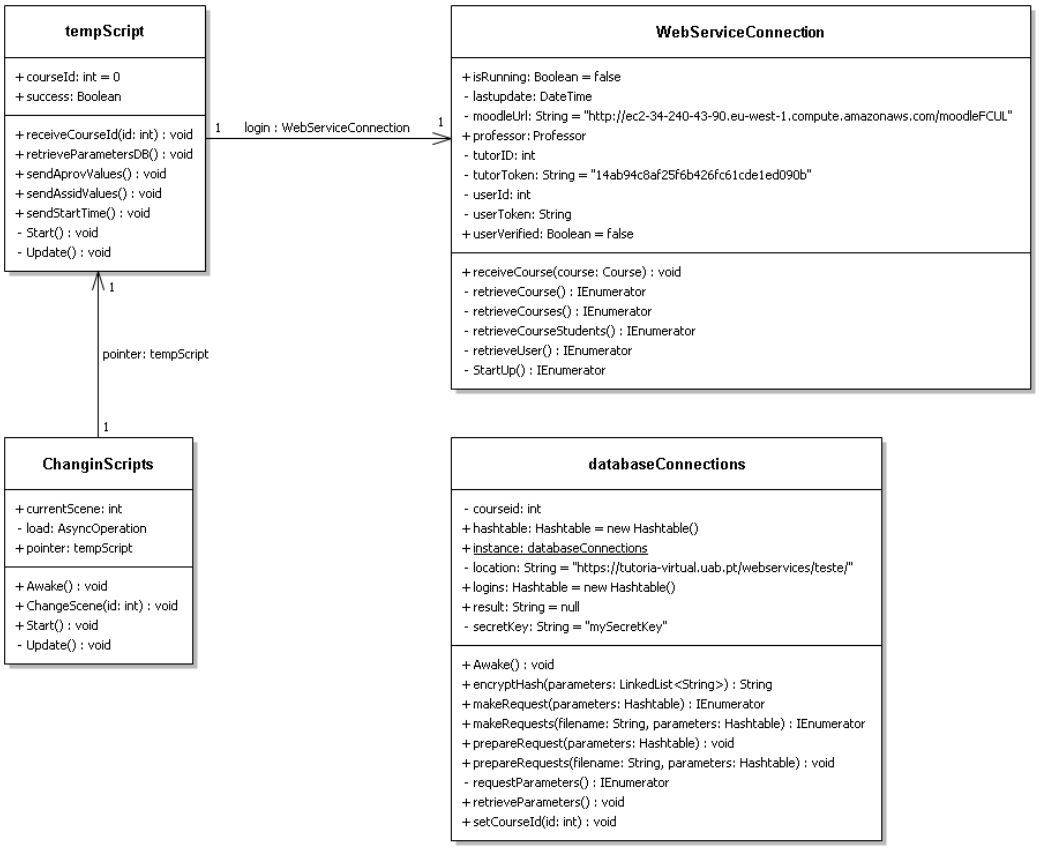

Figure B.17: Back office Class Diagram - Communication

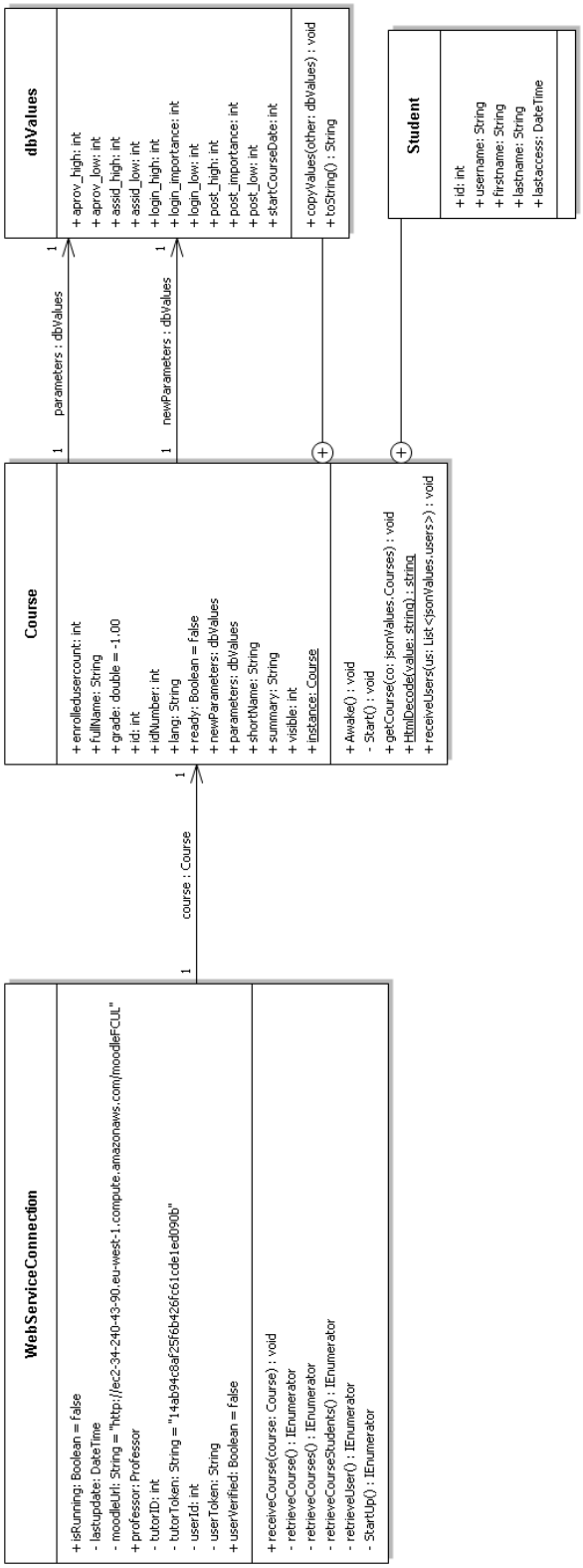

Figure B.18: Back office Class Diagram - Information Storage

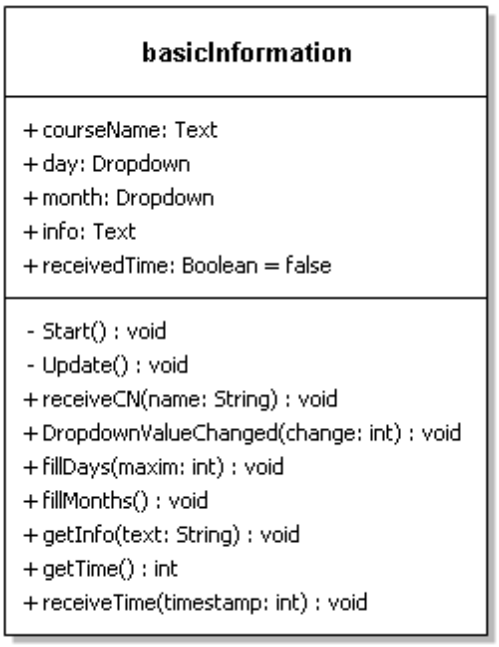

Figure B.19: Basic Information Class.

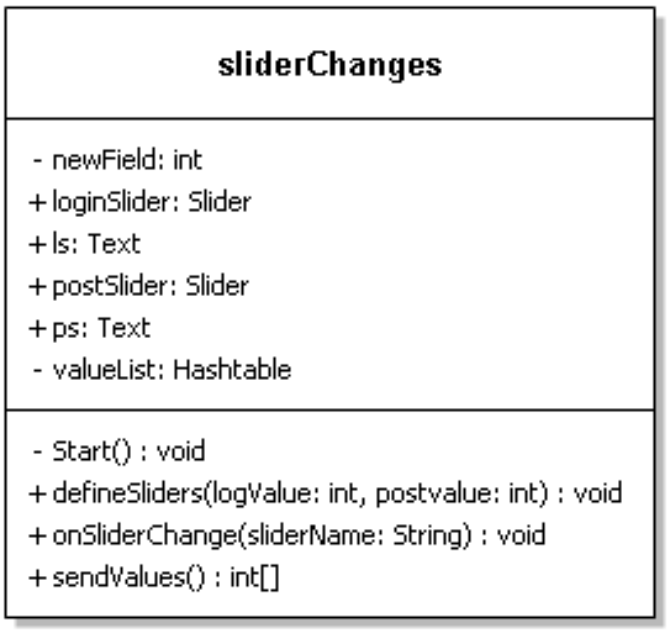

Figure B.20: Slider Changes class.

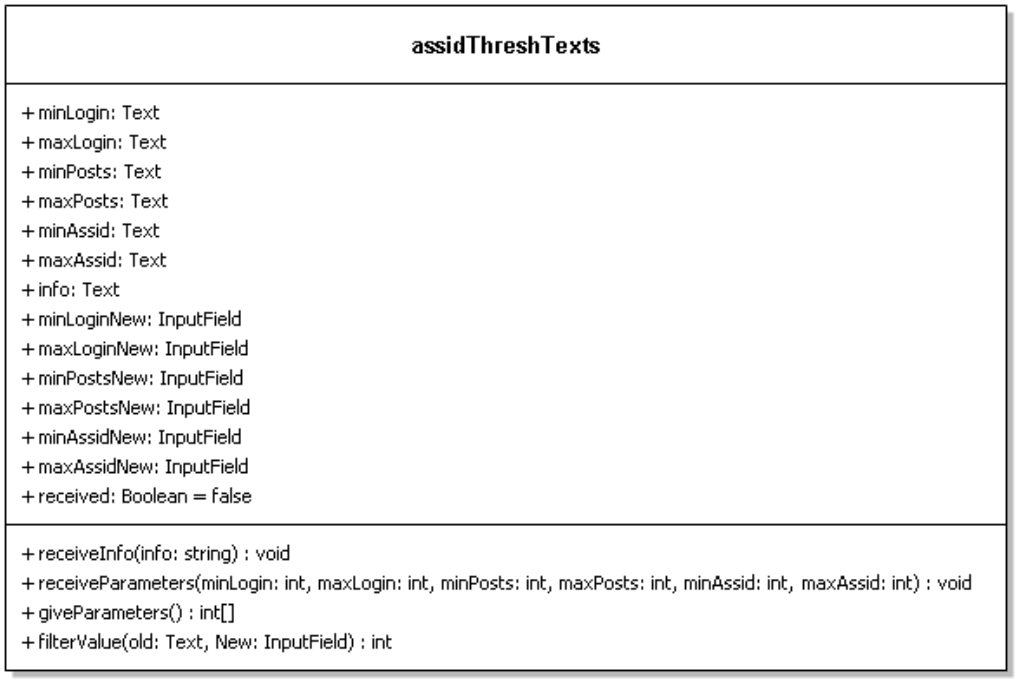

Figure B.21: Assid Thresh Texts class.

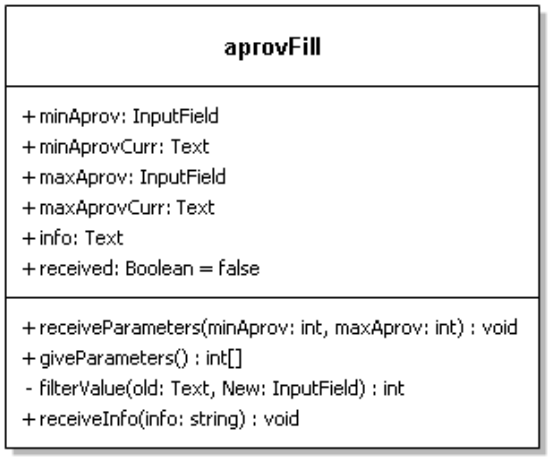

Figure B.22: aprov Fill Class.

Appendix C

Phrases stored in Database

id	identifier	aproveitamento	assiduidade	frase
1	fe1	LOW		As coisas vão certamente melhorar. Contacte o professor que está sempre disponível para colaborar!
2	fe2	LOW		As notas devem melhorar. Sabemos que consegue! Conte com a ajuda do professor.
3	fe3	LOW		Vai conseguir melhorar as suas notas. Estamos disponíveis para ajudar a ultrapassar os obstáculos.
4	fe4	LOW		Certamente vai conseguir melhorar! Sabe que pode contar com a ajuda do professor.
5	fe5	MIDDLE		Se investir mais nesta UC pode melhorar! Como sabe o professor está disponível para ajudar.
6	fe6	MIDDLE		Podemos ajudar a melhorar os resultados das avaliações. Veja com o professor o que não compreende.
7	fe7	MIDDLE		O seu desempenho está a melhorar, mas é possível ir mais longe! Esclareça as dúvidas connosco!
8	fe8	MIDDLE		Está no bom caminho e pode melhorar! Esclareça todas as dúvidas com o professor.
9	fe9	HIGH		Tudo indica que está a gostar desta UC: os resultados têm sido excelentes.
10	fe10	HIGH		Parabéns! Tem tido resultados muito bons!
11	fe11	HIGH		Parabéns pelos excelentes resultados! Continue o bom trabalho.
12	fe12	HIGH		Muito bem! Excelentes resultados!
13	ff1		LOW	A comunicação entre professor e estudantes é muito importante. Contacte-nos!
14	ff2		LOW	Interaja connosco! A interação na UC é fundamental para o sucesso académico.
15	ff3		LOW	A sua participação é muito importante para o sucesso académico. Contacte-nos!
16	ff4		LOW	O seu envolvimento na UC é um passo para o sucesso académico.
17	ff5		MIDDLE	Como estudante espera-se que seja activ@ na aprendizagem! Comunique connosco!
18	ff6		MIDDLE	Contacte-nos com frequência! É fundamental ser ativ@ na turma virtual!
19	ff7		MIDDLE	A assiduidade na UC é essencial para acompanhar as atividades!
20	ff8		MIDDLE	A sua assiduidade no acompanhamento da UC potencia o seu sucesso!
21	ff9		HIGH	Somos uma comunidade de aprendizagem. Continue a interagir na turma virtual.
22	ff10		HIGH	É fundamental que continue a interagir na turma virtual!
23	ff11		HIGH	O seu acompanhamento das atividades da UC é fundamental!
24	ff12		HIGH	A sua comunicação e colaboração na UC são essenciais para o seu progresso!
25	gp2		LOW	A interação na UC é muito importante. Contacte-nos! Use os botões em baixo para obter informações.
26	gp2		LOW	Interaja connosco! A interação na UC é muito importante. Eu posso dar-lhe informações, use os botões em baixo.
28	gp2		LOW	Não deixe de interagir connosco. Use os botões para verificar se há novidades.
29	gp2		LOW	A participação é muito importante para o sucesso académico. Participe! Interaja! Use os botões em baixo.
30	gp2		MIDDLE	Não deixe de comunicar connosco! A participação nos fóruns é fundamental! Use os botões em baixo para obter informações.
31	gp2		MIDDLE	Contacte-nos com frequência! Use os botões em baixo.
32	gp2		MIDDLE	Deve ser assíduo na página desta UC! Use os botões em baixo para obter informações.
33	gp2		MIDDLE	A sua assiduidade nesta página é essencial! Eu posso dar-lhe informações, use os botões em baixo.
34	gp2		HIGH	A sua frequência de acesso a esta turma virtual é elevada. Mantenha a assiduidade! Use os botões para verificar as novidades.
35	gp2		HIGH	Mantenha a frequência de acesso a esta UC. A sua interação e comunicação na turma virtual são essenciais! Use os botões.
36	gp2		HIGH	A sua frequência de acesso a esta UC é elevada. A sua interação e comunicação na turma virtual são essenciais! Escolha um botão.
37	gp2		HIGH	Mantenha a frequência de acesso a esta UC. Continue a interagir na turma virtual. Eu posso dar-lhe informações, use os botões em baixo.
38	pf1	LOW		O link para o e-fólio está assinalado na página. Certamente vai conseguir melhorar! Lembre-se que o professor está sempre disponível para ajudar!
39	pf2	LOW		O link para o e-fólio que escolheu está realçado na página. Sabemos que consegue melhorar as notas! Conte com a ajuda do professor.
40	pf3	MIDDLE		O link para o é-fólio que escolheu está agora destacado na página da UC. Parabéns, conseguiu ter nota positiva nesta componente de avaliação contínua!
41	pf4	MIDDLE		Use o link para o e-fólio que escolheu e que está destacado na página. Está no bom caminho e pode melhorar! Esclareça todas as dúvidas com o professor.
42	pf5	HIGH		Muitos parabéns, a sua nota nesta componente de avaliação contínua foi muito boa! Use o link destacado para consultar a sua nota que teve no e-fólio
43	pf6	HIGH		Excelente! O seu desempenho nesta componente de avaliação contínua foi muito bom! Use o link assinalado para verificar a sua nota.
44	fin1			Lembre-se que o cartão de aprendizagem tem a informação relevante sobre as suas avaliações. Pergunte ao professor se tiver alguma questão.
45	fin2			Recorde que o cartão de aprendizagem contém a informação importante sobre as suas notas. Fale com o professor em caso de dúvida.
46	fin3			No cartão de aprendizagem pode consultar todas as suas avaliações. O professor pode esclarecer em caso de dúvida.
47	fin4			Todas as suas avaliações estão registadas no cartão de aprendizagem. Se tiver dúvidas pergunte ao seu professor.

Figure C.1: Falas in database.

Appendix D

Attendance Tables for the tests

Day	1	2	3	4	5	6	7	8	9	10
Events		Release e-fólio 1 at 0:00	Due date e-fólio 1 at 23:59	grade e-fólio 1	Release e-fólio 2	Due date e-fólio 2	grade e-fólio 2		Release p-fóliodue date p-fólio	grade p-fólio
Students										
John1	Login	Login Post	LoginDeliver e-fólio 1	Login	Login	LoginPostDeliver e-fólio 2	Login	LoginPost	LoginDeliver p-fólio	Login
Attendance Value	50.00	76.67	76.67	76.67	76.67	88.33	88.33	100.00	100.00	100.00
Attendance	Middle	High	High	High	High	High	High	High	High	High
John2	Login		LoginDeliver e-fólio 1		LoginPost				LoginDeliver p-fólio	
Attendance Value	50.00	32.50	43.33	32.50	50.67	44.17	39.52	36.04	40.56	37.67
Attendance	Middle	Low	Low	Low	Middle	Low	Low	Low	Low	Low
John3	Login	LoginDeliver e-fólio 1	Login	Login		LoginPostDeliver e-fólio 2	LoginPost	Login	LoginDeliver p-fólio	Login
Attendance Value	65.00	65.00	65.00	65.00	52.00	65.83	79.05	80.21	81.11	81.83
Attendance	Middle	Middle	Middle	Middle	Middle	Middle	High	High	High	High
John4	Login				LoginPostDeliver e-fólio 2		Login		LoginDeliver p-fólio	Login
Attendance Value	50.00	32.50	21.67	16.25	37.67	33.33	39.52	36.04	40.56	44.17
Attendance	Middle	Low	Low	Low	Low	Low	Low	Low	Low	Low
John5	Login	LoginDeliver e-fólio 1		Login	Login	LoginDeliver e-fólio 2		LoginPost	LoginDeliver p-fólio	Login
Attendance Value	50.00	65.00	43.33	48.75	52.00	54.17	46.43	60.42	62.22	63.67
Attendance	Middle	Middle	Low	Low	Middle	Middle	Low	Middle	Middle	Middle
John6			LoginPostDeliver e-fólio 1			LoginDeliver e-fólio 2			LoginDeliver p-fólioPost	
Attendance Value	50.00	0.00	33.33	27.92	24.67	33.33	30.24	27.92	45.00	42.83
Attendance	Middle	Low	Low	Low	Low	Low	Low	Low	Low	Low
John7		LoginDeliver e-fólio 1				LoginDeliver e-fólio 2			LoginDeliver p-fólio	LoginPost
Attendance Value	50.00	32.50	21.67	16.25	13.00	21.67	18.57	16.25	21.67	37.67
Attendance	Middle	Low	Low	Low	Low	Low	Low	Low	Low	Low

Table D.1: Events and student action in e-fólio 2 course

Day	1	2	3	4	5	6	7	8	9	10
Events		Release e-fólio 1 at 0:00	Due date e-fólio 1 at 23:59	Release e-fólio 2grade e-fólio 1	Due date e-fólio 2	Release e-fólio 3Grade e-fólio 2	Due date e-fólio 3	grade e-fólio 3	Release p-fóliodue date p-fólio	grade p-fólio
Students										
Stan1	Login	Login	LoginDeliver e-fólio 1	LoginPost	LoginDeliver e-fólio 2	Login	LoginDeliver e-fólio 3	LoginPost	LoginDeliver p-fólio	Login
Attendance Value	50.00	65.00	65.00	76.67	76.67	76.67	76.67	88.33	88.33	88.33
Attendance	Middle	Middle	Middle	High	High	High	High	High	High	High
Stan2		Login	LoginDeliver e-fólio 1		LoginDeliver e-fólio 2Post		LoginDeliver e-fólio 3Post		LoginDeliver p-fólioPost	Login
Attendance Value	50.00	32.50	43.33	32.50	50.67	44.17	60.48	55.83	71.11	74.00
Attendance	Middle	Low	Low	Low	Middle	Low	Middle	Middle	Middle	Middle
Stan3	Login	LoginDeliver e-fólio 1Post			LoginDeliver e-fólio 2	LoginDeliver e-fólio 3			LoginDeliver p-fólioPost	Login
Attendance Value	50.00	76.67	55.00	44.17	50.67	55.00	48.81	44.17	59.44	62.33
Attendance	Middle	High	Middle	Low	Middle	Middle	Low	Low	Middle	Middle
Stan4			LoginDeliver e-fólio 1Post	LoginDeliver e-fólio 2Post			LoginDeliver e-fólio 3		LoginDeliver p-fólioPost	
Attendance Value	50.00	0.00	33.33	55.83	49.33	45.00	51.19	47.71	63.89	61.00
Attendance	Middle	Low	Low	Middle	Low	Low	Middle	Low	Middle	Middle
Stan5		LoginDeliver e-fólio 1	LoginPost	Login	Login	LoginDeliver e-fólio 3		Login	LoginDeliver p-fólioPost	Login
Attendance Value	50.00	32.50	55.00	60.42	63.67	65.83	58.10	60.42	73.89	75.33
Attendance	Middle	Low	Middle	Middle	Middle	Middle	Middle	Middle	Middle	High
Stan6	Login	LoginDeliver e-fólio 1			Login	LoginPostDeliver e-fólio 3			LoginDeliver p-fólio	Login
Attendance Value	50.00	65.00	43.33	32.50	39.00	55.00	48.81	44.17	47.78	50.67
Attendance	Middle	Middle	Low	Low	Low	Middle	Low	Low	Low	Middle
Stan7	Login	LoginPost	LoginDeliver e-fólio 1	Login	LoginDeliver e-fólio 2	Login	LoginDeliver e-fólio 3		LoginDeliver p-fólio	
Attendance Value	50.00	76.67	76.67	76.67	76.67	76.67	76.67	68.54	69.44	63.67
Attendance	Middle	High	High	High	High	High	High	Middle	Middle	Middle
Stan8		Login	LoginDeliver e-fólio 1		LoginDeliver e-fólio 2			Login	LoginDeliver p-fólio	
Attendance Value	50.00	32.50	43.33	32.50	39.00	32.50	27.86	32.50	36.11	32.50
Attendance	Middle	Low	Low	Low	Low	Low	Low	Low	Low	Low
Stan9	Login		LoginDeliver e-fólio 1		LoginDeliver e-fólio 2	LoginPost	LoginDeliver e-fólio 3Post		LoginDeliver p-fólio	Login
Attendance Value	50.00	32.50	43.33	32.50	39.00	55.00	69.76	63.96	66.67	68.83
Attendance	Middle	Low	Low	Low	Low	Middle	Middle	Middle	Middle	Middle
Stan10			LoginDeliver e-fólio 1	LoginDeliver e-fólio 2Post		LoginDeliver e-fólio 3			LoginDeliver p-fólio	
Attendance Value	50.00	0.00	21.67	44.17	37.67	44.17	39.52	36.04	40.56	37.67
Attendance	Middle	Low	Low	Low	Low	Low	Low	Low	Low	Low

Table D.2: Events and student action in e-fólio 3 course, part 1

Day	1	2	3	4	5	6	7	8	9	10
Events		Release e-fólio 1 at 0:00	Due date e-fólio 1 at 23:59	Release e-fólio 2grade e-fólio 1	Due date e-fólio 2	Release e-fólio 3Grade e-fólio 2	Due date e-fólio 3	grade e-fólio 3	Release p-fóliodue date p-fólio	grade p-fólio
Students										
Stan11	Login	LoginDeliver e-fólio 1		Login	Login	LoginDeliver e-fólio 3	Login	Login	LoginDeliver p-fólioPost	Login
Attendance Value	50.00	65.00	43.33	48.75	52.00	54.17	55.71	56.88	69.44	70.17
Attendance	Middle	Middle	Low	Low	Middle	Middle	Middle	Middle	Middle	Middle
Stan12	LoginPost		LoginDeliver e-fólio 1Post		LoginDeliver e-fólio 2	LoginDeliver e-fólio 3			LoginDeliver p-fólioPost	Login
Attendance Value	50.00	44.17	66.67	55.83	62.33	66.67	60.48	55.83	71.11	74.00
Attendance	Middle	Low	Middle	Middle	Middle	Middle	Middle	Middle	Middle	Middle
Stan13	Login	Login	LoginDeliver e-fólio 1	LoginPost	Login	LoginDeliver e-fólio 3		Login	LoginDeliver p-fólio	
Attendance Value	50.00	65.00	65.00	76.67	76.67	76.67	67.38	68.54	69.44	63.67
Attendance	Middle	Middle	Middle	High	High	High	Middle	Middle	Middle	Middle
Stan14	LoginPost		LoginDeliver e-fólio 1Post	LoginDeliver e-fólio 2			LoginDeliver e-fólio 3		LoginDeliver p-fólio	Login
Attendance Value	50.00	44.17	66.67	72.08	62.33	55.83	60.48	55.83	59.44	62.33
Attendance	Middle	Low	Middle	Middle	Middle	Middle	Middle	Middle	Middle	Middle
Stan15	Login	Login	LoginDeliver e-fólio 1	Login	LoginDeliver e-fólio 2	LoginDeliver e-fólio 3			LoginDeliver p-fólio	
Attendance Value	50.00	65.00	65.00	65.00	65.00	65.00	55.71	48.75	50.56	45.50
Attendance	Middle	Middle	Middle	Middle	Middle	Middle	Middle	Low	Middle	Low
Stan16			LoginDeliver e-fólio 1	LoginDeliver e-fólio 2	Login		LoginPostDeliver e-fólio 3		LoginDeliver p-fólio	Login
Attendance Value	50.00	0.00	21.67	32.50	39.00	32.50	48.81	44.17	47.78	50.67
Attendance	Middle	Low	Low	Low	Low	Low	Low	Low	Low	Middle

Table D.3: Events and student action in e-fólio 3 course, part 2

Appendix E

Expert test questionnaires

E.1 end questionnaire

Questionário para Docentes. Em todas as perguntas, a designação TUTOR VIRTUAL refere-se ao AVATAR.

Agora que usou o Tutor Virtual (avatar) assumindo o papel de 2 alunos fictícios de uma UC no moodle da Universidade Aberta, vimos pedir-lhe que responda a este questionário. A sua participação é fundamental!

Lembramos que:
- O projeto Tutoria Virtual é financiado pela Fundação para a Ciência e Tecnologia (referência PTDC/IVC-PEC/3963/2014) e envolve investigadores de 3 instituições: a Universidade Aberta, como instituição líder, e 2 instituições participantes da Universidade de Lisboa, a Faculdade de Ciências e o Instituto Superior Técnico.

- Neste projeto visamos analisar o impacto pedagógico resultante da integração de interfaces antropomórficas (isto é, com forma humana), também designados por humanos virtuais ou por "avatares", em ambientes de aprendizagem online.

- Uma das vertentes que estamos a testar é a utilização de avatares que são gerados com base em fotografias de pessoas. Os nossos avatares são designados por "Maria" e "João", os nomes verdadeiros das pessoas que serviram de modelos para a sua criação. Estes modelos são ainda o primeiro protótipo e o seu aspeto final irá ser melhorado.

*Obrigatório

1. **Endereço de email** *

2. **Departamento/Unidade a que pertence:** *
Marcar apenas uma oval.
- ◯ Ciências e Tecnologia
- ◯ Ciências Sociais e de Gestão
- ◯ Educação e Ensino a Distância
- ◯ Humanidades
- ◯ Aprendizagem ao Longo da Vida
- ◯ Outra: _____

3. **Faixa Etária** *
Marcar apenas uma oval.
- ◯ 25-40
- ◯ 41-55
- ◯ mais de 55

4. **Género**
Marcar apenas uma oval.
- ◯ feminino
- ◯ masculino

5. **Em que data começou a leccionar na Universidade Aberta (escolha dia 1 do mês e ano em que começou)**

Exemplo: 15 de dezembro 2012

6. **Já teve outras experiências de leccionação que não fossem 100% online?** *
Marcar apenas uma oval.

- ◯ Sim
- ◯ Não
- ◯ Outra: _____

7. **Já utilizou algum "assistente virtual" (por exemplo numa loja online ou numa plataforma de ensino, de treino ou de aprendizagem)?** *
Marcar apenas uma oval.

- ◯ Sim
- ◯ Não

8. **Se respondeu sim, diga qual**

9. **Com que frequência joga video-jogos com avatares?** *
Marcar apenas uma oval.

	1	2	3	4	5	
Nunca	◯	◯	◯	◯	◯	Muito frequentemente

Questionário de Usabilidade da Interface

Nas perguntas que se seguem, a designação TUTOR VIRTUAL refere-se ao AVATAR.

1. Gostaria de utilizar o Tutor Virtual frequentemente nas disciplinas. *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

2. Utilizar o Tutor Virtual é desnecessariamente complexo. *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

3. Senti que o Tutor Virtual é uma ferramenta simples de utilizar.
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

4. É preciso ajuda por parte do suporte técnico para conseguir utilizar o Tutor Virtual. *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

5. As funcionalidades do Tutor Virtual estão bem definidas/integradas. *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

6. Senti que o Tutor Virtual era inconsistente. *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

7. Qualquer pessoa utilizaria facilmente o Tutor Virtual. *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

8. O Tutor Virtual é muito difícil/desconfortável de utilizar. *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

9. Senti-me confiante ao usar o Tutor Virtual. *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

10. É necessário treinar muito antes de utilizar o Tutor Virtual. *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

Expectativas

Nas perguntas que se seguem, a designação TUTOR VIRTUAL refere-se ao AVATAR.

11. A utilização do Tutor Virtual: (assinale a(s) resposta(s) mais adequada(s)) *
Marcar tudo o que for aplicável.

- [ ] É apenas engraçado
- [ ] Torna menos monótono o acesso à página
- [ ] Faz com que me escape menos informação
- [ ] Auxilia-me a gerir as consultas na página
- [ ] É uma ferramenta para orientar o que devo fazer quando abro a página da UC
- [ ] Outra: _____

12. O Tutor Virtual correspondeu de um modo geral às minhas expectativas. *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

13. A indicação das novidades no "post-it" amarelo ajudou-me. *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

14. Verifiquei que os botões dos e-folios, dos fóruns e dos tópicos podem ter as seguintes cores: *
Marcar tudo o que for aplicável.

- [ ] branco
- [ ] vermelho
- [ ] amarelo
- [ ] azul
- [ ] verde
- [ ] Outra: _____

15. O tutor virtual muda de expressão e pode exibir uma expressão: *
Marcar tudo o que for aplicável.

- [ ] Neutra
- [ ] Contente
- [ ] Muito Contente
- [ ] Outra: _____

Impacto Pedagógico (1/2)

Nas perguntas que se seguem, a designação TUTOR VIRTUAL refere-se ao AVATAR.

16. A comunicação com o Tutor Virtual (Avatar)é fácil para o estudante. *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

17. A linguagem usada é clara em relação as orientações e informações dadas. *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

18. A interação com o Tutor Virtual (Avatar)é útil para orientar o estudante a explorar mais informações da U,C *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

19, O Tutor Virtual (Avatar) pode ajudar e / ou facilitar o trabalho colaborativo dos estudantes, *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

20, O Tutor Virtual (Avatar) agiliza os esclarecimentos de natureza pedagógica, *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

21. As notificações e lembretes dados pelo Tutor Virtual (Avatar) são úteis e adequadas. *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

22, O Tutor Virtual (Avatar) é auto motivante para os estudantes, *
Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

23. A interação com o Tutor Virtual (Avatar) contribui para que aceda mais rápido à informação da UC. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

24. O Tutor Virtual (Avatar) ajuda a compreender melhor as orientações de uso dos materiais de aprendizagem da UC. *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

25.O ensino foi facilitado com as dicas e apoio do Tutor Virtual (Avatar). *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo Totalmente	◯	◯	◯	◯	◯	Concordo Totalmente

Impacto Pedagógico (2/2)

Nas perguntas que se seguem, a designação TUTOR VIRTUAL refere-se ao AVATAR.

26- Na sua opinião quais as vantagens que o Tutor Virtual(Avatar) pode trazer na UC? para responder esta questão destaque as funções do Tutor Virtual (Avatar) que mais gostou.

. 27- Indique sugestões de funcionalidades que gostaria que o tutor realizasse para contribuir no trabalho docente em relação ao processo de ensino e aprendizagem.?

Muito Obrigad@

Agradecemos muito a sua colaboração neste estudo!

A sua opinião é indispensável para a evolução deste projeto de investigação!
Com a sua ajuda vamos continuar a trabalhar para introduzir melhorias neste protótipo.

Bibliography

[1] About moodle. https://docs.moodle.org/35/en/About_Moodle. Accessed at: 09-06-2018.

[2] About web services in moodle. https://docs.moodle.org/dev/Web_services. Accessed at: 09-06-2018.

[3] Course modules of moodle. https://docs.moodle.org/dev/Course_module. Accessed at: 09-06-2018.

[4] External services in moodle. https://docs.moodle.org/34/en/Using_web_services#Creating_a_service. Accessed at: 09-06-2018.

[5] Monobehaviour api description. https://docs.unity3d.com/ScriptReference/MonoBehaviour.html. Accessed at: 09-06-2018.

[6] tokens to web services in moodle. https://docs.moodle.org/27/en/Using_web_services#Create_a_token. Accessed at: 09-06-2018.

[7] Unity. https://unity3d.com/. Accessed at: 09-06-2018.

[8] update api description. https://docs.unity3d.com/ScriptReference/MonoBehaviour.Update.html. Accessed at: 09-06-2018.

[9] Web services in moodle. https://docs.moodle.org/34/en/Web_services. Accessed at: 09-06-2018.

[10] What is an lms? https://www.talentlms.com/what-is-an-lms/. Accessed at: 09-06-2018.

[11] What is elearning? http://www.elearningnc.gov/about_elearning/what_is_elearning/. Accessed at: 09-06-2018.

[12] Wsupdatedescription. https://tracker.moodle.org/browse/MDL-57395. Accessed at: 09-06-2018.

[13] Ygor Amaral, Alexandre Maciel, and Rodrigo Rodrigues. Development of a virtual assistant for alerts and notifications in a learning environment. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 26, page 742, 2015.

[14] Valentina Arkorful and Nelly Abaidoo. The role of e-learning, advantages and disadvantages of its adoption in higher education. *International Journal of Instructional Technology and Distance Learning*, 12(1):29–42, 2015.

[15] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[16] Pierre Dillenbourg, Daniel Schneider, and Paraskevi Synteta. Virtual learning environments. In *3rd Hellenic Conference" Information & Communication Technologies in Education"*, pages 3–18. Kastaniotis Editions, Greece, 2002.

[17] Maia Garau, Mel Slater, Vinoba Vinayagamoorthy, Andrea Brogni, Anthony Steed, and M Angela Sasse. The impact of avatar realism and eye gaze control on perceived quality of communication in a shared immersive virtual environment. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 529–536. ACM, 2003.

[18] W Lewis Johnson, L Friedland, Peter Schrider, Andre Valente, and Sean Sheridan. The virtual cultural awareness trainer (vcat): Joint knowledge online's (jko's) solution to the individual operational culture and language training gap. In *Proceedings of ITEC*. Clarion Events London, UK, 2011.

[19] W Lewis Johnson and Jeff Rickel. Steve: An animated pedagogical agent for procedural training in virtual environments. *ACM SIGART Bulletin*, 8(1-4):16–21, 1997.

[20] H Chad Lane and W Lewis Johnson. Intelligent tutoring and pedagogical experience manipulation in virtual learning environments. *The PSI handbook of virtual environments for training and education*, 3, 2008.

[21] Alexandre MA Maciel, Rodrigo L Rodrigues, and Edson CB Carvalho. Desenvolvimento de um assistente virtual integrado ao moodle para suporte a aprendizagem online. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 25, page 382, 2014.

[22] Maria Moundridou and Maria Virvou. Evaluating the impact of interface agents in an intelligent tutoring systems authoring tool. In *Proceedings of the Panhellenic Conference with International participation in Human-Computer interaction*, 2001.

[23] N Rajkumar and V Ramalingam. Cognitive intelligent tutoring system based on affective state. *Indian Journal of Science and Technology*, 8(24):1, 2015.

[24] Wayne Ward, Ron Cole, Daniel Bolaños, Cindy Buchenroth-Martin, Edward Svirsky, and Tim Weston. My science tutor: A conversational multimedia virtual tutor. *Journal of Educational Psychology*, 105(4):1115, 2013.