

Specification of an Online Advisement System

Christian Rayburn James Hays Bryan Phillips Frederick C. Harris, Jr.

Department of Computer Science
University of Nevada, Reno, NV 89557
Email: fredh@cs.unr.edu

Abstract

This paper outlines the motivation for, and the development of, an online advisement scheduling system, OASIS, which will be used in the Computer Science Department at the University of Nevada, Reno. OASIS will allow students to schedule an advisement sessions with an advisor by way of the internet anytime day or night. This project has the application of being distributed to every university nation wide and ultimately generalized into a broad scale online scheduling system that can be used by many different businesses. It further discusses the specific requirements, functional and non-functional, with the aid of the Unified Modeling Language (UML). Directions of further development are also described.

Keywords: advisement, internet, online, scheduling, software, UML

1 Introduction

Advising of undergraduate students in Computer Science and Engineering is such an important topic that it is mentioned by ABET in their accreditation criteria [1]. Yet, as David Woolsten shows, it is not a topic covered in the educational literature [2]. One of the hassles regarding advising is the scheduling of the appointments between two busy groups, students and faculty.

Currently, the office staff in the Computer Science department does all department advisement scheduling. The staff records the student's appointment by hand in a logbook. This prevents the office staff from completing other important tasks. The Online Advisement Scheduling Information System (OASIS) provides a convenient way for students to schedule advisements with professors on the internet. Universities and their respective departments who use OASIS will reduce the office staff workload and provide more time for office staff to perform other important duties.

Students are already using the internet for countless activities (paying bills, making purchases, registering for classes, submitting homework – just to name a few!). Advisement scheduling is just another activity that lends itself well to the internet.

OASIS is designed to interact with three main types of registered users, advisees, advisors, and office staff (Figure 1). The office staff shall also be the system administrators, and will be responsible for other user accounts.

Once professors have an OASIS account, students can begin scheduling appointments with them. After entering the necessary data, students can also download the appropriate advisement forms with their personal information already filled. Further, office staff and advisors may log in, view the current report of scheduled advisements for a

given day, and view analytics of the system's usage.

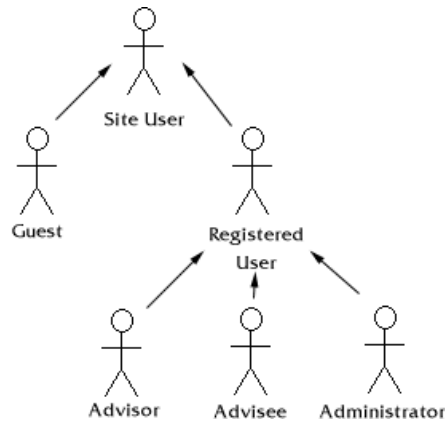


Figure 1: Actor Role Hierarchy

With the support of the Unified Modeling Language (UML) and the Unified Process (UP), we intend our software to be very precise and robust in order to easily accommodate the implementation of additional components or logic. UML helps to “specify, visualize, construct, and document” the artifacts of the project direction and application [3]. UP helps to show us, the developers, what artifacts we need in order to “utilize, perform, or create” the software model [4].

It is possible for every post-secondary school to benefit from this type of online software. OASIS is immediately distributable as an advisement scheduling system for other departments. It can also be easily modified into any other type of time scheduling software for institutions, and in due time can be a scheduling utility for many different types of enterprises.

2 Requirements

The UP shows that before you can begin working on project analysis and

design, there must be notions of what the software will be required to do [4].

2.1 Functional

The functional requirements of a system entail what the system will actually do [4]. For example, the below functional requirements are the authentication requirements of the OASIS structure.

- 1.1.1 The OASIS system shall validate usernames upon login.
- 1.1.2 The OASIS system shall validate passwords upon login.
- 1.1.3 The OASIS system shall authenticate users upon login.
- 1.1.4 The OASIS system shall only allow authenticated users to view web pages registered to their group type.
- 1.1.5 The OASIS system shall logout users upon logout.

2.2 Non-Functional

The non-functional requirements of a system outline the properties and constraints of the system. For example, the below non-functional requirements facilitate what internet based mechanisms shall be used for our software, see Figure 2.

- 1.1.6 The OASIS system code shall be written in PHP & HTML.
- 1.1.7 The OASIS system shall use a MySQL database.
- 1.1.8 The OASIS system be accessible via any standard web browser.

3 Use Case Modeling

UML integrates use case modeling to define how a system performs with

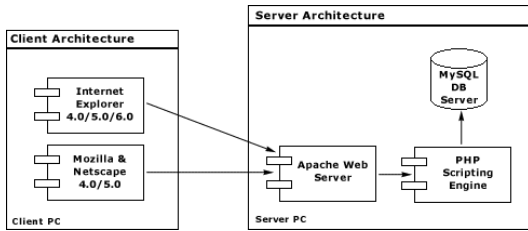


Figure 2: Application Architecture

respect to the interaction with outside sources such as system users, which are also known as actors [5]. Use case modeling helps OASIS in defining the necessary functionality and interactions needed in order to be useful software.

3.1 Use Case Diagram

The use case diagram helps demonstrate the use case model. The system is represented as a box, with its actors on the outside and its use cases on the inside [4]. This is illustrated in our use case diagram shown in Figure 3. It depicts the overview of the main website's functionality and layout, and how each actor relates to the OASIS system. It houses the components for each actor, a method of logging in and authorizing the actors, and illustrates the ability to view both printer friendly and html graphically rich web-pages.

3.2 Use Case Specification

UML use case specification is used to describe specific details and attributes of each use case. In the use cases below (see Tables 1 and 2), the specification consists of a natural language description that provides the details of that particular use case and how it incorporates into the system. The list of descriptions also includes relationships the use case has with other system use

cases, as well as the actors that the use case interacts with.

Table 1: View Public Pages Use Case

Use case: View Public Pages
ID: UC1.1
Description: These are the pages viewed even if not logged into the system. These pages are for displaying news, notices etc. They can show how to use the system when used in a demo site.
Relationships: 1. Communicate with the actor Site Visitor 2. Extended by UC 1.1.1 Render Printable Public Pages 3. Extended by UC 1.1.2 Render HTML Public Pages

Table 2: Login Use Case

Use case: Login
ID: UC1.4
Description: If an actor wishes to utilize the private functionality of the site the actor must be a registered user. The user must select login, which will be on the left side of the home page. The user must enter their "User Name" and "Password." The system will then search for the actor and if authenticated the system will let the user utilize the OASIS web site designed for that type of actor.
Relationships: 1. Communicate with the actor Advisee. 2. Communicate with the actor Advisor. 3. Communicate with the actor Admin. 4. Include UC 1.8 Authentication User.

3.3 Scenarios

UML defines a scenario as a specific path through the flow of events of a particular a use case [4]. UP helps express the need for each scenario to document which use case it belongs to,

the actors that act upon it, the system's state before the scenario, the actual steps of going through the scenario, and the state of the system after the scenario has completed [4]. A primary scenario is the expected normal path taken through the use case, as presented in Table 3. Subsequently, an alternate scenario is the unexpected, or error driven path, for the particular use case, as shown in Table 4.

Table 3: Scenario for Successful Login

Scenarios for Use case: Login
ID: UC1.4
Actors: Advisors, Staff, Students
Preconditions: 1. Actor finds web site using an Internet browser.
Flow of Events: 1. Actor requests the website using the browser. 2. System displays front page with "Login" option. 3. Actor selects "Login." 4. System Displays "Login" screens. 5. Actor enters "User Name" and "Password." 6. System "Authenticates" the user name and password. 7. System accepts user name and password. 8. System displays "Main Menu" and available options.
Postconditions: 1. System waits for request from Actor on "User Menu" screen

4 Web Site Road Map

UML uses road maps to depict all possible flow of states and events that are required to define all possible steps in going from one system state to another. The road map for OASIS depicted in Figure 4, is a portion of the advisee's specific web site interface. Due to its dimensions, Figure 4 is placed near the end of this paper.

Table 4: Scenario for Login Failure: Incorrect Password

Scenarios for Use case: Login
ID: UC1.4
Actors: Advisors, Staff, Students
Preconditions: 2. Actor finds web site using an Internet browser.
Flow of Events: 1. Actor requests the website using the browser. 2. System displays front page with "Login" option. 3. Actor selects "Login." 4. System Displays "Login" screens. 5. Actor enters "User Name" and "Password." 6. System "Authenticates" the user name and password. 7. System accepts user name and password. 8. System displays "Main Menu" and available options.
Postconditions: 1. System waits for request from Actor on "User Menu" screen

5 Future Extensions

There are numerous possible extensions to OASIS that would enhance its usability and capabilities. Implementation and usage of OASIS within the Computer Science Department at the University of Nevada, Reno is expected to begin this summer. Once this implementation has occurred we expect to receive feedback on what extensions are most needed to be implemented next.

OASIS migration to other departments at the University of Nevada is expected shortly after that, beginning with the other departments in the College of Engineering. Once a high level of performance and service satisfaction has been obtained, it is planned to market OASIS to other post-secondary schools.

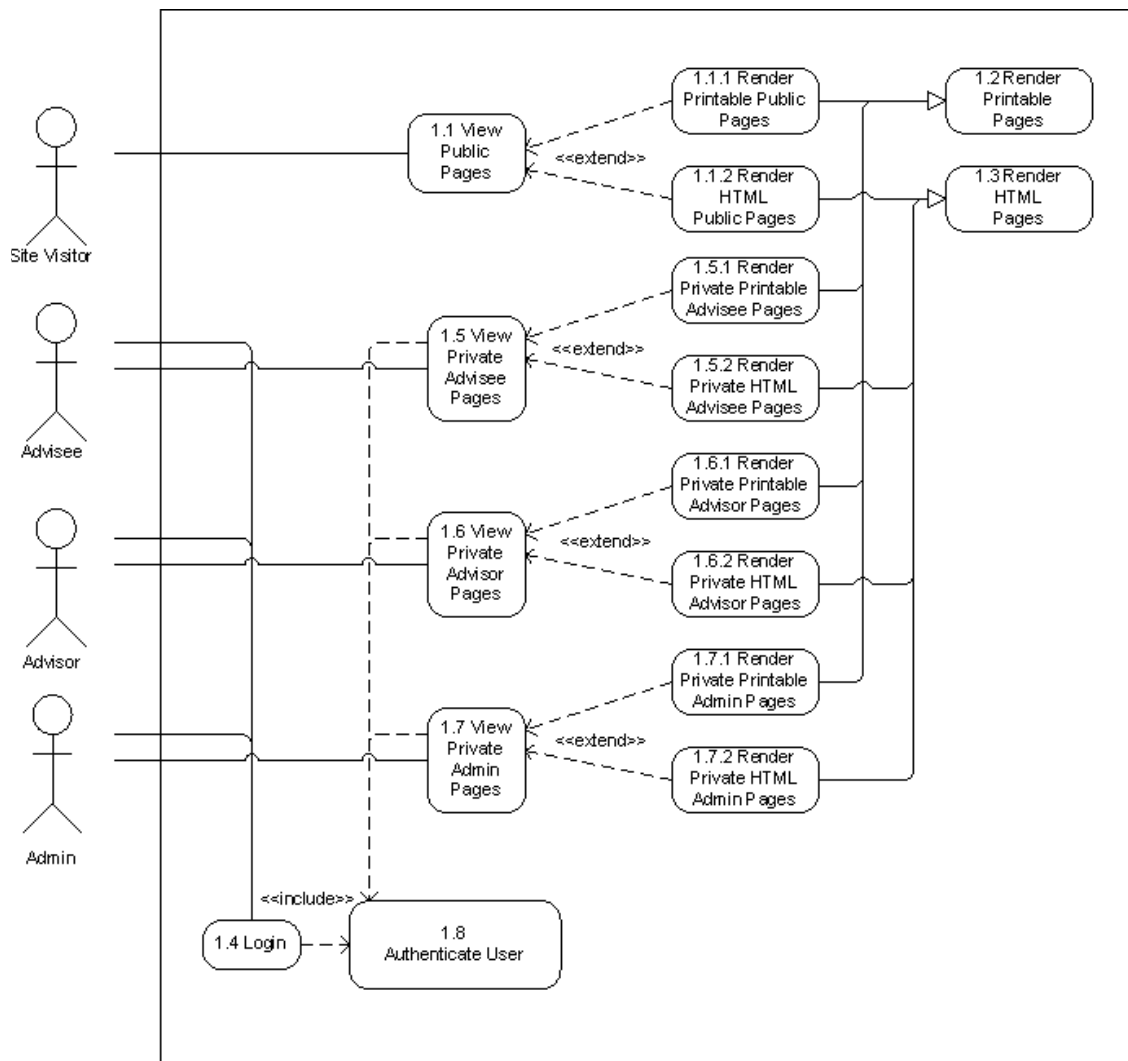


Figure 3: Main Web Site Use Case Diagram

6 Conclusions

We have presented in this paper an Online Advising Scheduling and Information System. As described above, OASIS is designed to make advisement appointment scheduling as easy as possible for students, advisors, and office staff. By converting to computers and the internet, there is less human workload & user analytics are immediately available.

With the aid of UML and UP, OASIS has been seamlessly designed to support a large functionality set and work fluidly on the UNR Computer Science department website.

At this point in time, OASIS's UML analysis model has been completed and the tool's implementation is undergoing. The first working version of this tool is expected to be available in a few months.

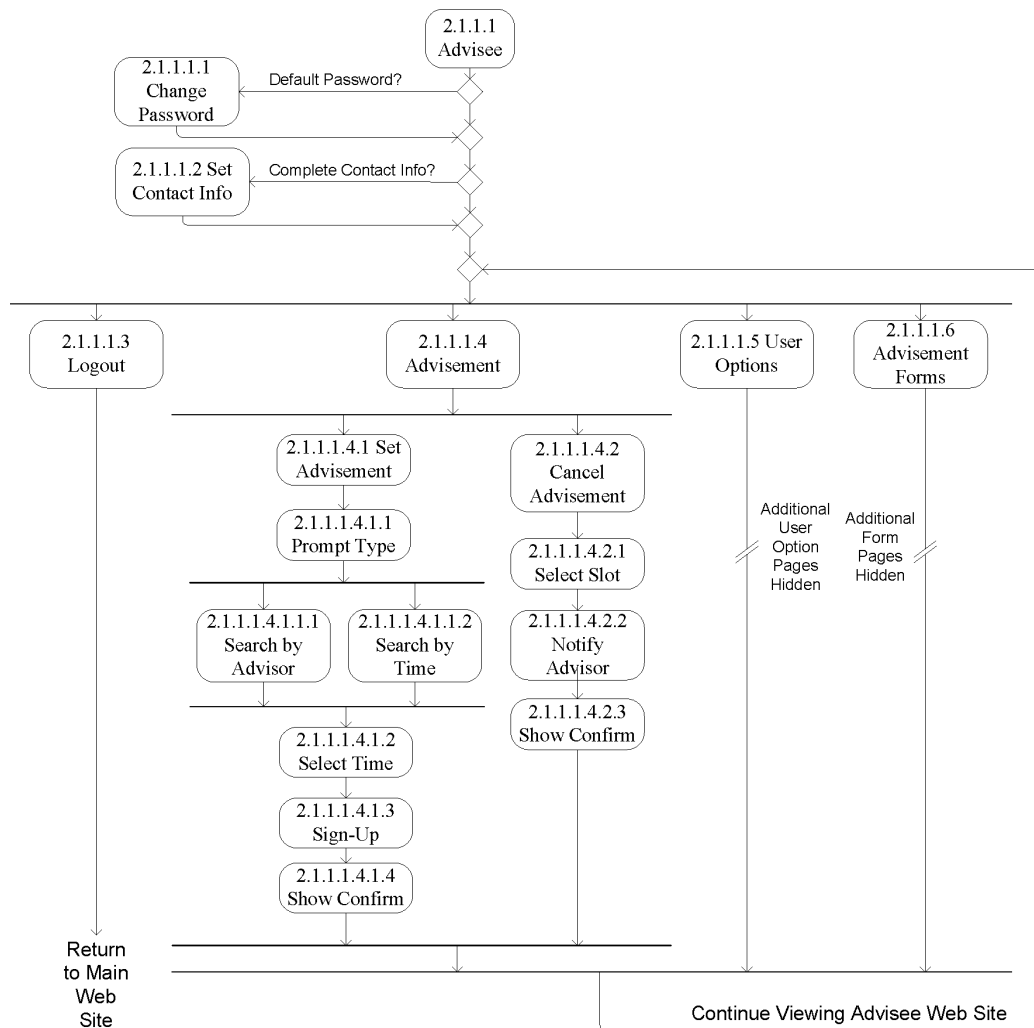


Figure 4: Advisee Web Site Road Map

References

- [1] Accreditation Board for Engineering and Technology. *2002-2003 Criteria for Accrediting Engineering Programs*. Available on the web: <http://www.abet.org/images/Criteria/2002-03EACCcriteria.pdf> Accessed April 25, 2003.
- [2] Donald C. Woolston, "Improving undergraduate academic advising in engineering: It's not rocket science," in *Proceedings of FIE 2002*, November 6-9, 2002, Boston, MA.
- [3] Dictionary.com. An online dictionary. Lexico Publishing Group, LLC. 2003 Available online at: www.dictionary.com. Accessed April 25, 2003.
- [4] Jim Arlow & Ila Neustadt. *UML and the Unified Process: Practical Object-Oriented Analysis & Design* Addison-Wesley: Great Britain. 2002.
- [5] James Rumbaugh, Ivar Jacobson, & Ggrady Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley. 1998.