KHOA CÔNG NGHỆ THÔNG TIN – TRƯỜNG ĐẠI HỌC CÔNG NGHỆ - ĐHQGHN

SOFTWARE ARCHITECTURE DOCUMENT

FOR

TUTORONLINE

Version 1.0

Prepared by

Group 10

Trần Trịnh Bình Thành K58CA thanhttb\_58@vnu.edu.vn

Trần Văn Liên K59CA 14020768@vnu.edu.vn

Nguyễn Hữu Hồng K59CA hongnguyenhuu96@gmail.com

Trần Như Thuật K59CA 14020457@vnu.edu.vn

TABLE OF CONTENT

[**1.   Introduction**](http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm#introduction)        [1.1 Purpose](http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm#Purpose)  
        [1.2 Scope](http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm#Scope)  
        [1.3 Definitions, Acronyms and Abbreviations](http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm#Definitions,%20Acronyms%20and%20Abbreviations)  
        [1.4 References](http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm#References)

[**2.   Architectural Representation**](http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm#Architectural%20Representation)

[**3.   Architectural Goals and Constraints**](http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm#Architectural%20Goals%20and%20Constraints)

[**4.   Use-Case View**](http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm#Use-Case%20View)        [4.1 Architecturally-Significant Use Cases](http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm#Architecturally-Significant%20Use%20Cases)

[**5.   Logical View**](http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm#Logical%20View)        [5.1 Architecture Overview – Package and Subsystem Layering](http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm#Architecture+Overview+%3F+Package+and+Subsystem+Layering)

[**6.   Process View**](http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm#Process%20View)         6.1 Web.php (Routes)

6.2 PagesController (Controller)

6.3 ProfilesController (Controller)

6.4 Auth (containing some controllers)

6.5 EditProfile(Views)

6.6 FindTutor(Views)

**7. Deployment view**

[**8.   Size and Performance**](http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm#Size%20and%20Performance)

[**9.   Quality**](http://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm#Quality)

**1. Introduction**

**1.1. Purpose**

This document provides an architecture overview of TutorOnline system, using a number of architectural view to depict aspects of the system.

**1.2. Scope**

This software architecture document (SAD) is about TutorOnline, which has been developed by group 10 – SE2016. It is generated directly from TutorOline Analysis & Design Model Implementation.

**1.3. Definition, Acronyms and Abbreviations**

SAD: Software Architecture Document

**1.4. References**

- The “4+1” view model of software architecture, Philippe Kruchten, November 1995

<http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf>

- IBM Rational Software Architect <http://www306.ibm.com/software/awdtools/architect/swarchitect/index.html>

- The IBM Rational Unified Process :

<http://www-306.ibm.com/software/awdtools/rup/index.html>

**2. Architecture Representation**

This document presents the architecture as a series of views; use case view, logical view, process view and deployment view. There is no separate implementation view described in this document.

**3. Architecture Goals and Constraints**

- This app is available on local PCs and mobile devices.

- All tutors, students and parents must be available from local PCs, mobile devices with the Internet Connection.

- The system implements some basic security behavior:

+ Authentication: login using user name and password.

+ Authorization: perform some actions according to user’s profile.

- Data persistence will be addressed using a relational database.

**4. Use-case view**

The tutoroneline website use cases are:

- login

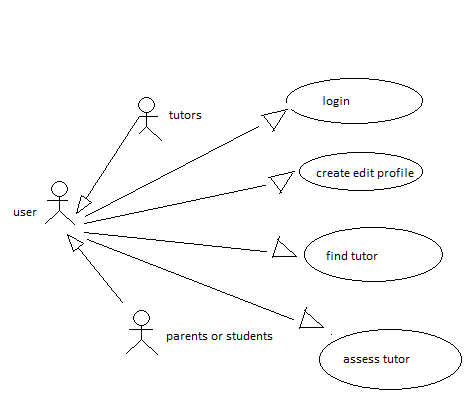
- create / edit profile

- find tutor for study purpose

- assess tutor via comment

These use cases are initiated by students, parents, and the tutor.

**4.1****Architecturally-Significant Use Cases**



**4.1.1 Login**

Brief Description: This use case describes how a user logs into the Tutor Online website. The actors starting this use case are tutor, parents, and students.

**4.1.2 Create edit profile**

Brief Description: This use case describes how a user create and edit his profile to show out for finding job or share information

**4.1.3 Find tutor**

Brief Desctiption: This use case describes how to find a tutor suit with the study purpose

**4.1.4 assess tutor**

Brief Description: give comment to share the emotion, experients about a tutor

**5. Logical view**

**5.1****Architecture Overview – Package and Subsystem Layering**

Using framework Laravel (version 5.3)

Backend :

Laravel PHP

Frontend

boostrap – flatly

agency – booswatch

select2

modal full screen

Jquery

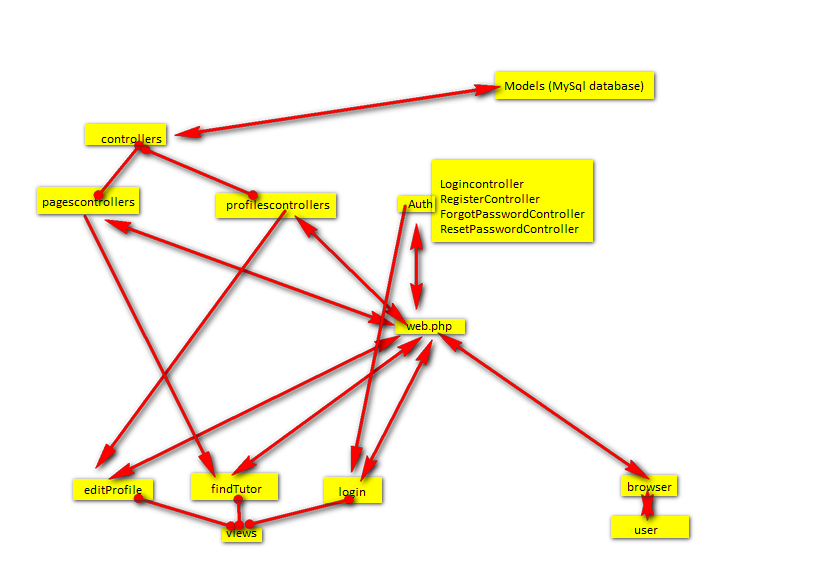
Ajax

Disqus

Database

My SQL

**6.****Processes view**



**6.1 Web.php (Routes)**

Containing all routes (information for get and post request)

**6.2 PagesController (Controller)**

This controller process jobs about showing the home page, get all of match tutors for finding tutor request from database via Models and sends all to the findTutor view

**6.3 ProfilesController (Controller)**

This controller used for create/ edit profile, get request from editProfile view via web.php and save it to database by models

**6.4 Auth (containing some controllers)**

These controllers were provided by laravel to process the login, register, forgot password, resetpassword of the users

**6.5 EditProfile(Views)**

This view used to show edit profile page in html to the browser, get request from the user and send to web.php for saving data by ProfilesController

**6.6 FindTutor(Views)**

This view used to send finding request (use filter) to get the results from PagesController (using Ajax for better UX)

**7. Deployment View**

Deploy on Heroku sever (tutoronline.herokuapp.com).

The tutor online webset can be accessed by any devices with iternet conection.

**8. Size and Performance**

The system can serve at most 2000 clients at one time.

Time to process action is less than 3 seconds

**9. Quality**

The system is supported by many platforms: Windows 95/98, XP, Vista, Win 7/8/10, …

Use Chrome to get the best performance.

The system contains step-by-step instructions on using this web-based system.

The system is available all time.