Universidade de Brasília
Curso: Engenharia de software
Disciplina: Técnicas de Programação
Professor: Maurício Serrano
Alunos: Karine Valença
Murilo Duarte
Vinícius Bandeira
Wesley Araujo

Assunto: Descrição das variáveis
Linguagem do Projeto: Ruby com framework Ruby on Rails
Software Utilizado: Transparência em Escalas Médicas do DF (TEM-DF)

Class: ApplicationController
File: app/controllers/application_controller.rb
Purpose of class: This class is a controller and contains action methods for general application view.

Method	**Variable/constant**	**Utility**
current_user	@current_user	Receives the user logged on system currently.
list_speciality	@medic	Store all medic object.
list_work_unit_name	@work_unit	Store all WorkUnit object.
get_by_speciality_or_work_unit	object1 (argument)	An argument which can be of type medic or work unit.
get_by_speciality_or_work_unit	@speciality	Specialities array.
get_by_speciality_or_work_unit	@work_unit_name	Work unit names array.
get_by_speciality_or_work_unit	medic	A boolean variable which is changed in case the argument object1 be medic.

Class: ParsersController
File: app/controllers/parsers_controller.rb
Purpose of class: Populates the database with file named schedule.csv.

Method	**Variable/constant**	**Utility**
upload	document	Receives a csv file to store in database

Class: CommentsController

File: app/controllers/comments_controler.rb

Purpose of class: Contains action methods for comments view

Method	**Variable/constant**	**Utility**
reports	@user	Receives an object of class User of current session
deactivate	@comment	Receives an object of class Comment
reactivate	@comment	Receives an object of class Comment
disable_report	@comment	Receives an object of class Comment

Class: MedicsController

File: app/controllers/medics_controler.rb

Purpose of class: This class is a controller and contains action methods for medics view.

Method	**Variable/constant**	**Utility**
(is a constant)	NOT_EXIST_RATING	Receives an number zero which indicate that do not exists ratings.
(is a constant)	NOT_EXIST_GRADE	Receives an string with number zero to indicate that do not exists grade.
results	@medics	Receives an object of Medic class that is find by params on search in home.
profile	@comment	Receives an object of Medic class that is find by id param.
profile	@averange	Assigns the method calculate_averange with @medic argument.
profile	@ratings	Assigns the ratings of that medic id.
work_units_graph	@medics_size	Array that store medics quantity.
work_units_graph	@unit_name	Array that store units name.
work_units_graph	@work_unit	Get all work units from database.
work_units_graph	quantity	Assigns an integer with a medics quantity.
array_speciality	work_unit_id (argument)	An integer argument with work unit id.
array_speciality	@medic	Get all medics with specific id work unit.
array_specialit	@speciality	Array which assigns medics specialities.

y		
array_medics_quantity	id_work_unit (argument)	An argument with a id work unit (integer).
array_medics_quantity	array_speciality (argument)	Array with speciality.
array_medics_quantity	@medic	Get all medics with specific is.
array_medics_quantity	@medics_size_speciality	Array which store medics quantity by speciality
array_medics_quantity	quantity	Store an integer with medic quantity by params received.
rating	medic_id	Instance medic object by id from params
rating	@user	User currently logged on system.
rating	@medic	Instance medic object from variable medic_id.
create_comment	@user	User currently logged on system.
create_comment	@medic	Instance medic object from params comments_id.
create_comment	@comment	Instance a new comment object.
create_relevance	@user	
create_relevance	@comment	An instance of comment object by from params comment_id.
create_relevance	@relevance	If exists comment, its instance an relevance object by user id and comment id, else, a new relevance is created.
to_report	@comment	Instance a new comment object from params.
ranking	@medics	Instance medic object by average order des. with the limit 10.
create_rating	user (argument)	An argument of user object.
create_rating	medic (argument)	An argument of medic object.
create_rating	@rating	Instance a new rating.
update_rating	rating (argument)	Case a grade exists, then rating will have your attributes altered.
update_rating	grade (argument)	It’s passed a grade.
calculate_average	medic (argument)	A medic object passed like argument.
calculate_average	@ratings	Instance all ratings of that medic which was passed like argument.
calculate_average	sum	Store a grades sum.
calculate_average	arithmetic_mean_averange	Store an average of grades sum divide by ratings quantity.

Class: TemdfMailer

File: app/mailers/temdf_mailer.rb

Purpose of class: Creates the body of emails according to standard, and requested type.

Method	Variable/constant	Utility
request_account	path_date_medical_file	Addressing the file
request_account	attachments["arquivo_medico.pdf"]	That defines attachments
confimation_email	link	Set an account confimação link

Class: UserMailer

File: app/mailers/user_mailer.rb

Purpose of class: Creates the body of emails to reset user account password.

Method	Variable/constant	Utility
password_reset	@user	Variable instantiates, used in view

Class: ContactController

File: app/controllers/contact_controller.rb

Purpose of class: Contact class between the user and administrators.

Method	Variable/constant	Utility
create	email	That receives method of TemdfMailer class

Class: ParsersController

File: app/controllers/parsers_controller.rb

Purpose of class: Populates the database with file named schedule.csv.

Method	**Variable/constant**	**Utility**
upload	document	Receives the parameter document belonging to user class

Class: SessionsController

File: app/controllers/sessions_controller.rb

Purpose of class: Create and destroy sessions when users perform, login or logout.

Method	**Variable/constant**	**Utility**
create	user	Receives authentication method of the User class
create	session[:remember_token]	Receives parameter of an object of class User
create	session[:user_id]	Receives parameter of an object of class User
destroy	session[:remember_token]	Parameter modifies the class user to null
destroy	session[:user_id]	Parameter modifies the class user to null

Class: UsersController

File: app/controllers/users_controller.rb

Purpose of class: Contai action methods for users view.

Method	**Variable/constant**	**Utility**
index	@user	Receives the user found by the user id logged in
index	@user	Receives all the users in

		database
create	@user	Receives a new user with the user’s params
create	@random	An instance of Random class.
new	@user	Receives a new user
edit	@user	Receives the user found by the user id logged in
edit_password	@user	Receives the user found by the user id logged in
update	@user	Receives the user found by the user id logged in
update	email	Receives the email from the user
update	user_from_email	Receives the user found by the user email
update	username	Receives the username from the user
update	user_from_username	Receives the user found by the user username
update_password	@user_session	Receives the user found by the user id logged in
update_password	@user	Receives the user authenticated by the username and password
update_password	@user	Receives the new password from user
desactivate	@user	Receives the user found by the user id
reactivate	@user	Receives the user found by the user id
confirmation_email	@user	Receives the user found by the user id and token email
confirmation_email	@message	Receives a string

Class: PasswordResetsController

File: app/controllers/password_resets_controller.rb

Purpose of class: This class is a controller and contains action methods for password resets view.

Method	**Variable/constant**	**Utility**
create	@user	Receives the user found by the user username and email
edit	@user	Receives the user found by the

		user password reset token
update	@user	Receives the user found by the user password reset token
update	new_password	Store an new passoword from params :user and :password.

Class: User
File: app/models/user.rb
Purpose of class: Responsible for processing, validating, associate, other tasks in the treatment of user's data.

Method	**Variable/constant**	**Utility**
self.authenticate	user	Receives the user found by the user username and account status
encrypt_password	self.password_salt	Receives the engine which generate password salt
encrypt_password	self.password_hash	Receives the engine which generate passaword hash
send_password_reset	self.password_reset_sent_at	Receives the time zone at atual moment

Attribute	**Utility**
username	Username from user
email	Email from user
password	Password from user
pasword_confirmation	Password confirmation from user
new_password	New password from user when he call method update_password
encrypt_password	Encrypt the password

Class: Comment
File: app/models/comment.rb
Purpose of class: Responsible for processing, validating, associate, other tasks in the treatment of comment's data.

Attribute	**Utility**
content	Content from comment