7-6-2017

Patrick van Batenburg,

Steven Logghe

Samen Sterk

Documentatie van de aanpassingen

Inhoudsopgave

[Inleiding 2](#_Toc485591949)

[Aanpassingen 2](#_Toc485591950)

[Controllers.AppointmentController 2](#_Toc485591951)

[Controllers.GradeController 2](#_Toc485591952)

[Controllers.RepeatingTaskController 3](#_Toc485591953)

[Controllers.SubjectController 3](#_Toc485591954)

[Controllers.TaskController 4](#_Toc485591955)

[Controllers.UserController 4](#_Toc485591956)

[Views.DeleteUsers 4](#_Toc485591957)

[Views.EditSubject 4](#_Toc485591958)

[Views. EditTask 5](#_Toc485591959)

[Views.Login 5](#_Toc485591960)

[Views.Register 5](#_Toc485591961)

[Views.Shedule 5](#_Toc485591962)

[Revisies 6](#_Toc485591963)

# Inleiding

In dit document worden de aanpassingen naar aanleiding van de testbevindingen gedocumenteerd.  
Deze aanpassingen waren afhankelijk geweest van de testen van de ontwikkelaars en andere collega’s die het programma hebben getest.

# Aanpassingen

## Controllers.AppointmentController

In de methode Details(uint) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Create(Appointment) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Edit(Appointment) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Delete(Appointment) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen. Verder moest de appointment ook op een makkelijke manier de juiste rij te kunnen verwijderen, we hebben dus die rijen een verstopte ID nummer gegeven zodat we zo

Met het opvangen van de onverwachtse database fouten kunnen we hiermee achterhalen wat er fout is gegaan met de database, mocht dat weer van toepassing zijn. En op die manier kunnen we die fout sneller gaan opzoeken en dan oplossen.  
  
Dit geval was alleen wel het geval geweest in elke methode van elke controller, dus dit moest dus overal aangepast worden zodat we daar zeker van zijn dat er geen crashes zullen gebeuren in de applicatie.

## Controllers.GradeController

In de methode Details(uint) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Create(Grade) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Edit(Grade) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Delete(Grade) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

Er waren fouten ontdekt bij het verwijderen van een tabel van een vak/onderwerp, dat de cijfers die bij het onderwerp horen niet volledig verwijderd werden, Hiervoor hadden we een query gebruikt die alle cijfers ophaalt van de tabel die bij het verwijderde vak horen.

## Controllers.RepeatingTaskController

In de methode Details(uint) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Create(RepeatingTask) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Exceeds(RepeatingTask) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Edit(RepeatingTask) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Delete(RepeatingTask) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen. Verder moest er gecheckt worden op de parameter of dat deze wel geïnitialiseerd is.

## Controllers.SubjectController

In de methode Details(uint) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Create(Subject) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Edit(Subject) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Delete(Subject) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

Bij het onderwerp moest er ook zeker van zijn dat, als er een wijziging is plaatsgevonden, dat de cijfers nog steeds bij het onderwerp horen, we hebben dit opgelost om het te laten baseren op ID nummer.

## Controllers.TaskController

In de methode Details(uint) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Create(Task) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Exceeds(Task) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Edit(Task) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Delete(Task) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen. Verder moest er gecheckt worden op de parameter of dat deze wel geïnitialiseerd is

## Controllers.UserController

In de methode Details(User) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Register(User) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Edit(User) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode Delete(User) moest er een try-catch statement komen voor de connectie met de database, om zo onverwachte database fouten te kunnen afvangen.

In de methode LogOff(Type) moest er gecheckt worden of dat de parameter niet van type null was.

Bij het overzetten van de online database naar de offline versie, moest de usercontroller met alle verbindingen eruit gehaald te worden om zo geen fouten of lekken met de gebruikers te veroorzaken.

## Views.DeleteUsers

Nadat fouten in de UserController opgelost waren, moest de waarde waarop gecheckt wordt of dat iets juist of onjuist is aangepast worden.

## Views.EditSubject

Nadat fouten in de SubjectController en GradeController opgelost waren, moest de waarde waarop gecheckt wordt of dat iets juist of onjuist is aangepast worden.

## Views. EditTask

Nadat fouten in de TaskController en RepeatingTaskController opgelost waren, moest de waarde waarop gecheckt wordt of dat iets juist of onjuist is aangepast worden.

## Views.Login

Nadat fouten in de UserController opgelost waren, werkte de methodes weer naar behoren.

## Views.Register

Nadat fouten in de UserController opgelost waren, werkte de methodes weer naar behoren.

## Views.Shedule

Nadat fouten in de de TaskController, RepeatingTaskController, SubjectController, GradeController, en AppoinmentController opgelost waren, werkte de methodes weer naar behoren.

# Revisies

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Wat | Wie | Versie |
| 12-06-2017  08:45 CEST | Documentatie van de aanpassingen gedocumenteerd. | Patrick van Batenburg | 1.0.0 |
| 14-06-2017  23:23 CEST | Documentatie van de aanpassingen verbeterd. | Patrick van Batenburg | 2.0.0 |
| 18-06-2017  23:28 CEST | Extra aanpassingen toegebracht aan het document. | Steven Logghe | 3.0.0 |