04-02-2014

**Plan van aanpak**

Fifa: developers edition!



**Naam** : Anthony Peeters, Ricky van den Berg, Maikel Podean

**Groep** : Shining Unicorn Productions. *(Sup*.)

**Opdrachtgever** : Sietse Dijks, Fedde van Gils.

**Klas**: RIO4-APO1A + APO1B

**Plaats**: Radiuscollege, Breda

Inhoudsopgave

[Achtergronden 2](#_Toc387677597)

[Doelstellingen 2](#_Toc387677598)

[Project opdrachten 2](#_Toc387677599)

[Projectgrenzen 3](#_Toc387677600)

[Producten 3](#_Toc387677601)

[Kwaliteit 3](#_Toc387677602)

[Informatie 4](#_Toc387677603)

[Project Organisatie 4](#_Toc387677604)

[Planning 4](#_Toc387677605)

[Kosten en baten 5](#_Toc387677606)

[Risico’s 5](#_Toc387677607)

[Conventierapport 5](#_Toc387677608)

[Handtekening conventierapport 6](#_Toc387677609)

[Bereiksbaarheidslijst 7](#_Toc387677610)

[Rolverdeling 7](#_Toc387677611)

[Flowchart programflow 8](#_Toc387677612)

[Onderzoeksverslag database 9](#_Toc387677613)

# Achtergronden

Shining Unicorn Productions, of *Sup*, is gevormd op 7 April 2014 op het Radiuscollege te Breda.

Sup bestaat momenteel uit Peeters, Ricky van den Berg, en Maikel Podean.

Deze opdracht is gegeven door Sietse Dijks en Fedde van Gils.

Het doel van het project is om een Toto te maken voor een voetbal toernooi.

# Doelstellingen

Dit project is onderdeel van de opleiding Applicatieontwikkelaar, die wordt gegeven op het Radiuscollege Breda.

Hierbij oefenen en ontwikkelen wij onze kennis in de C#, en onderzoeken wij meerdere opties om ons project zo goed mogelijk af te krijgen.

De bedoeling is dat wij een geautomatiseerde en werkende Toto maken, en ervoor zorgen dat alle eisen eraan voldoen.

Voor *Einddatum*: 20-05-2014 moet het af zijn.

# Project opdrachten

*Projectnaam*: Het ontwikkelen van een geautomatiseerde Toto.

*Opdrachtgevers*: Sietse Dijks en Fedde van Gils.

Het maken van een Samenwerkingscontract.

Het maken van een Bereikbaarheidslijst, met handtekeningen van de deelnemende projectgenoten.

Een Plan van Aanpak maken, uitprinten, en in de gezamenlijke map stoppen.

# Projectgrenzen

Niet het hele programma schrijven, maar het aangeleverde stuk programma maken we af.

Het maken van een plan van aanpak.

Het opzetten en gebruikmaken van een database.

Het opzetten en gebruikmaken van Github.

Geen dropbox gebruiken.

Connecten naar een database.

Bij houden van de database.

Samenwerken met mediadevelopment.

# Producten

Opleverlijst:

* Schetsen van applicatie
* Verslag welke gegevens er in een database zouden moeten zitten
* Verslag van onderzoek naar database aan applicatie koppelen
* Flowchart van programflow op basis van bovengenoemde documenten
* Onderzoeksrapport applicatie(uitgebreide antwoorden op de leervragen)
* Klassendiagram uit Visual Studio
* Optimalisatieplan
* Optimalisaties doorgevoerd
* Technisch testplan
* Acceptatietest
* Verslag van tests
* Werkend eindproduct

# Kwaliteit

De kwaliteit van het opleverproduct hangt af van de kwaliteit van de deelopdrachten.

Deze proberen wij zo goed mogelijk te maken, zodat het eindproduct er beter uit komt te zien.

Per week kijken we of we de opdrachten voltooid hebben.

Ook wordt er een logboek gemaakt en wordt er opgeschreven hoe onze vorderingen zijn.

Er zullen ook regelmatig controles zijn met de opdrachtgevers, of we onze opdrachten in de goede richting maken.

Mochten er misverstanden over de kwaliteit van het opleverproduct zijn, dan schakelen wij de hulp van een leraar of een andere projectgroep in om ons verder te helpen.

# Informatie

Iedere werkdag bespreken we samen waar we aan (verder) werken.

Aan het einde van iedere werkdag slaan we ons werk op een USB en Github.

Alle bestanden die met dit project te maken hebben slaan we gezamenlijk op in een mapje.

De aanwezigheid van de projectgroep wordt door de opdrachtgevers bijgehouden in Comtak.

# Project Organisatie

**Team Leider & Lead Programmer:**

*Ricky van den Berg*

**Database Beheerder:**

*Maikel podean*

**Github Beheerder:**

*Anthony Peeters*

**Project beoordelaars:**

*Sietse Dijks*

*Fedde van Gils*

De leden zitten iedere Maandag en Dinsdag bij elkaar in een lokaal.

Buiten het lokaal om is er nog contact via Skype en Dropbox.

Ook zitten de projectbeoordelaars in hetzelfde lokaal, zodat er altijd contact is.

# Planning

*Begindatum*: 07-04-2014

*Einddatum*: 20-05-2014

*Einddatum Plan van aanpak*: 15-04-2014

# Kosten en baten

Kosten:

Er zijn geen kosten aan dit project verbonden.

Baten:

Kennis in het maken en aanpassen van een form applicatie in C#.

Kennis en handigheid in het maken van een Plan van aanpak

# Risico’s

Intern:

Database set up kan fout gaan. Deadline kan niet gehaald worden door mis communicatie.

Niet genoeg kennis waardoor deadline niet haalbaar is.

Bij het te laat beginnen aan het project zullen er meerdere complicaties oplopen, zoals het wachten op projectgenoten, of het wachten op iets wat nog niet af is.

Extern:

Internet doet het niet.

Computer doet het niet.

project tijd ingekort.

# Conventierapport

* Datum als eerst vervolgens een lage streep daarna een hoofdletter (jjjj/mm/dd) dus bijv. 20141404\_Sample,
* We gebruiken camelcase voor properties ,
* Methodes en classes beginnen met een hoofdletter,
* Commits in het Engels.
* User interface in het nederlands.
* Lettertype in documenten: Calibri (Hoofdtekst) en grootte lettertype 11 en grootte kop 14 dikgedrukt blauw.
* Comments gebruiken om de code uit te leggen en wie wat heeft veranderd en of toegevoegd,
* Logische naamgeving voor knoppen bijv. btnSend voor een verzend knop.

Tabellen conventie database rows en kolommen:

* Zorg ervoor dat de namen in de juiste Engels en dragen natuurlijk betekenissen. Deze namen moet zinvol en beschrijvend moeten zijn.
* Vermijd korte woorden en afkortingen als dat mogelijk is. Afkortingen worden met camelcase aangegeven

Afkortings tabel voor naamgeving van properties.

|  |  |
| --- | --- |
| btn | button |
| lbl | label |
| pb | picturebox |
| pnl | panel |
| gb | groupbox |
| nm | numericupdown |
| cb | checkbox |

# Handtekening conventierapport

Ricky van den Berg:

Anthony Peeters:

Maikel Podean:

Oliwier Grzelak:

# Bereiksbaarheidslijst

Bedrijf Naam: Shining Unicorn Productions

Naam: Ricky van den Berg

Tel: 06 10757168

Email: [d196626@edu.rocwb.nl](mailto:d196626@edu.rocwb.nl)

Naam: Anthony Peeters

Tel: 06 12335833

Email: [d193875@edu.rocwb.nl](mailto:d193875@edu.rocwb.nl)

Naam: Viorel Ion Podean

Tel: 06 41410194

Email: [mp150035@edu.rocwb.nl](mailto:mp150035@edu.rocwb.nl)

Naam: Oliwier Grzelak

Tel:

Email:

# Rolverdeling

Projectleider: Ricky van den Berg

Leidt het project. Is aansprakelijk voor opleveringen. Aanspreekpunt voor opdrachtgever.

lead developer: Anthony Peeters

Baas van de code, is verantwoordelijk voor de code.

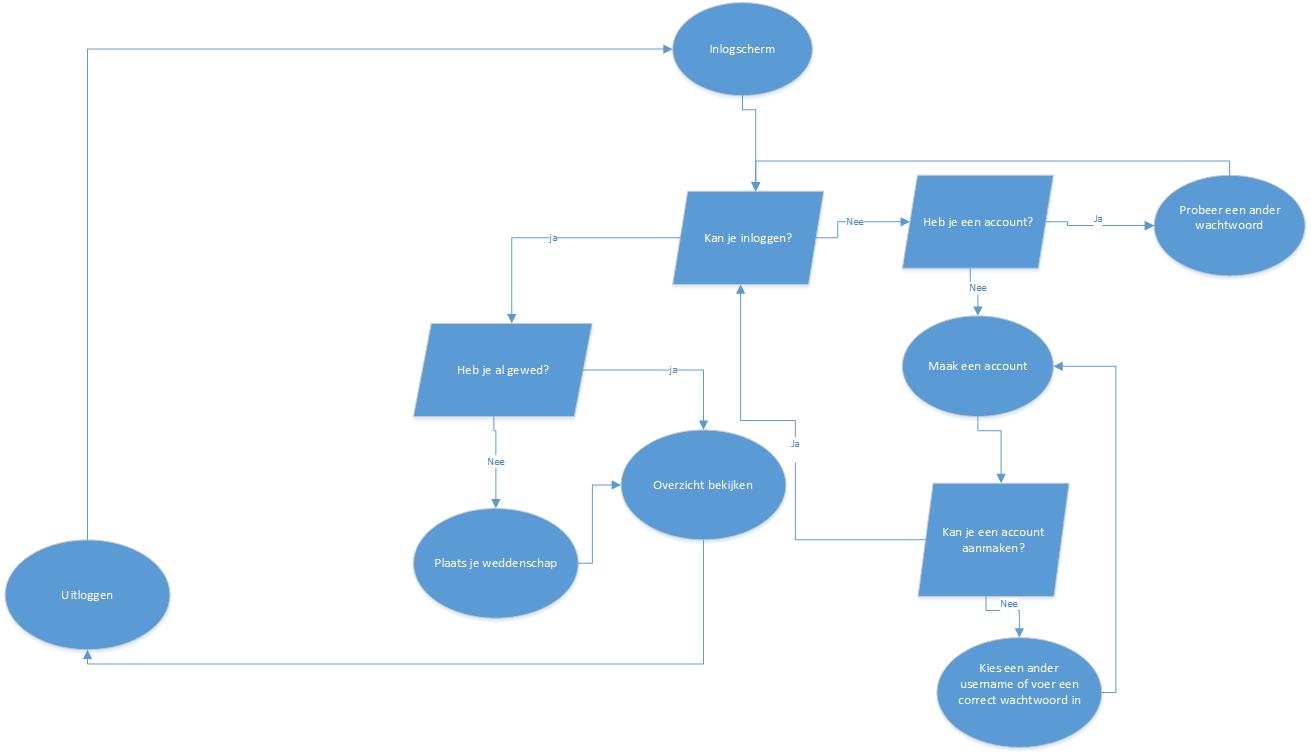
Database beheerder: Mikael Podean

Beheert de database. Maakt veranderingen in de database.

Github beheerder: Oliwier

Beheerd de github. Is verantwoordlijk voor github. Houdt github netjes.

# Flowchart programflow



+

# 

# Onderzoeksverslag database

Uitleg kolom:

Een kolom is een verzameling van data waarden van een bepaald type.

Uitleg veld:

Een rij van kolommen.

Uitleg rij:

In de context van een [relationele database](http://nl.wikipedia.org/wiki/Relationele_database), vertegenwoordigt een rij - ook wel een record of [tupel](http://nl.wikipedia.org/wiki/Tupel) genoemd - een enkel, impliciet gestructureerde [dataitem](http://nl.wikipedia.org/wiki/Gegeven) in een [tabel](http://nl.wikipedia.org/wiki/Tabel_(database)). In eenvoudige termen kan een databasetabel worden beschouwd als te zijn opgebouwd uit rijen en [kolommen](http://nl.wikipedia.org/wiki/Kolom_(database)) of velden. Elke rij in een tabel geeft een [verzameling](http://nl.wikipedia.org/wiki/Verzameling_(wiskunde)) van gerelateerde [data](http://nl.wikipedia.org/wiki/Gegeven) weer, en elke rij in de tabel heeft dezelfde structuur.

Uitleg record :

Verzameling van gegevens in een database die bij elkaar horen, bijvoorbeeld alle gegevens over iemands naam, adres en woonplaats.

Database Management System(Welke zijn er? Welke gebruikt Visual Studio?)

Met een databasemanagementsysteem (vaak afgekort tot DBMS) wordt het systeem aangeduid dat als [database](http://nl.wikipedia.org/wiki/Database) opgeslagen gegevens ontsluit, bewaakt en beheert. Een database bestaat soms uit drie onderdelen: de opgeslagen gegevens (in één of meer bestanden), het programma waarmee de gegevens worden onderhouden (DBMS) en eventueel de [gebruikersinterface](http://nl.wikipedia.org/wiki/Gebruikersinterface) ([client](http://nl.wikipedia.org/wiki/Client_(applicatie))) die het gebruikers mogelijk maakt om de gegevens te behandelen. Meestal is er een DBMS actief voor meerdere gebruikers. Bekende en veelgebruikte programma's zijn *relationele* [DBMS-en](http://nl.wikipedia.org/wiki/Relationele_database) (afgekort tot RDBMS) zoals [MySQL](http://nl.wikipedia.org/wiki/MySQL), [Microsoft SQL Server](http://nl.wikipedia.org/wiki/Microsoft_SQL_Server) en [Oracle](http://nl.wikipedia.org/wiki/Oracle_(software)) Database.

Voorbeelden van DBMS:

[Berkeley DB](http://nl.wikipedia.org/w/index.php?title=Berkeley_DB&action=edit&redlink=1)

[Caché](http://nl.wikipedia.org/wiki/Cach%C3%A9_(software))

[Clipper](http://nl.wikipedia.org/wiki/Clipper_(programmeertaal))

[DB2](http://nl.wikipedia.org/wiki/IBM_DB2)

[IDMS](http://nl.wikipedia.org/w/index.php?title=CA_IDMS&action=edit&redlink=1)

[dBase](http://nl.wikipedia.org/wiki/DBase)

[Filemaker](http://nl.wikipedia.org/wiki/Filemaker)

[Firebird](http://nl.wikipedia.org/wiki/Firebird_(database))

[FoxPro](http://nl.wikipedia.org/w/index.php?title=FoxPro&action=edit&redlink=1)

[Informix](http://nl.wikipedia.org/w/index.php?title=Informix&action=edit&redlink=1)

[MariaDB](http://nl.wikipedia.org/wiki/MariaDB)

[Microsoft Access](http://nl.wikipedia.org/wiki/Microsoft_Access)

[msSQL](http://nl.wikipedia.org/w/index.php?title=MsSQL&action=edit&redlink=1)

[MySQL](http://nl.wikipedia.org/wiki/MySQL)

[Microsoft SQL Server](http://nl.wikipedia.org/wiki/Microsoft_SQL_Server)

[Oracle](http://nl.wikipedia.org/wiki/Oracle_(software)) Database

[Paradox](http://nl.wikipedia.org/wiki/Paradox_(database))

[PostgreSQL](http://nl.wikipedia.org/wiki/PostgreSQL)

[SmallSQL](http://nl.wikipedia.org/w/index.php?title=SmallSQL&action=edit&redlink=1)

[SQLite](http://nl.wikipedia.org/wiki/SQLite)

[Sybase](http://nl.wikipedia.org/w/index.php?title=Sybase&action=edit&redlink=1)

[Turboveg](http://nl.wikipedia.org/wiki/Turboveg)

Virtual studio uses Microsoft SQL Sever as a DBMS.

DataType (Welke zijn er? Wat zijn hun eigenschappen?)

Een datatype, ook wel gegevenstype genoemd, is in de [informatica](http://nl.wikipedia.org/wiki/Informatica) een specifiek soort gegevens, zoals [integers](http://nl.wikipedia.org/wiki/Integer_(informatica)), [booleans](http://nl.wikipedia.org/wiki/Boolean), [reals](http://nl.wikipedia.org/wiki/Zwevendekommagetal), [karakters](http://nl.wikipedia.org/wiki/Karakter_(informatica)), [strings](http://nl.wikipedia.org/wiki/String_(informatica)). In een [programmeertaal](http://nl.wikipedia.org/wiki/Programmeertaal) wordt met iedere [variabele](http://nl.wikipedia.org/wiki/Variabele_(informatica)), en meer in het algemeen met iedere [expressie](http://nl.wikipedia.org/wiki/Expressie_(programmeren)), een datatype geassocieerd. Dit datatype bepaalt welke waarden de variabele of de expressie kan aannemen, hoe deze waarden in het geheugen worden opgeslagen en welke bewerkingen op de variabele of de expressie uitgevoerd kunnen worden.

Bij elk gegevenstype hoort een verzameling van de mogelijke waarden die een variabele of expressie van dat type kan aannemen. Ook hoort er een systeem bij voor de codering van de waarden. Een waarde kan alleen al binnen één programmeertaal soms door meerdere datatypes gerepresenteerd worden: zo kan het getal 5 gerepresenteerd worden door diverse typen integer en real (er zijn vaak meerdere types van elk, met verschillende aantallen bits).  
  
egevenstypes kunnen worden onderscheiden in primitieve (primitive), enkelvoudige (simple) en samengestelde (complex) types. Primitieve datatypes vormen de basis voor de definities van andere gegevenstypes.

Een primitief type wordt door de taal zelf gedefinieerd en kan niet beschreven worden in termen van een ander datatype. In C, bijvoorbeeld, zijn char, int en float primitieve types.

Een enkelvoudig type is een primitief type of een type dat op basis van een primitief type door de programmeur gedefinieerd is. Het criterium daarbij is dat een enkelvoudig gegevenstype uitsluitend als geheel kan worden gemanipuleerd en uitgelezen.

Bijvoorbeeld kan een type foo worden gedefinieerd als int.  
  
Een samengesteld (complex) type bestaat uit meerdere simpele types die afzonderlijk kunnen worden gemanipuleerd en uitgelezen. Hierbij valt te denken aan structuren, arrays en lijsten die samengesteld zijn uit elementen die op zichzelf ook kunnen bestaan uit samengestelde types: lijsten van lijsten, bijvoorbeeld, of geneste structuren. In tegenstelling tot primitieve types, die slechts in een beperkt aantal soorten voorkomen, is het aantal mogelijke samengestelde types in principe onbeperkt. Voorbeelden van benamingen voor samengestelde types in verschillende programmeertalen zijn: [array](http://nl.wikipedia.org/w/index.php?title=Array_(informatica)&action=edit&redlink=1), [class](http://nl.wikipedia.org/wiki/Klasse_(informatica)), struct en record.

Primary Key (Wat is het? Hoe herken je dit?)

De primaire sleutel van een relationele tabel een unieke identificatie van elke record in de tabel. Het kan een normaal kenmerk dat is gegarandeerd uniek te zijn (zoals Social Security Number in een tabel met niet meer dan een record per persoon) of het kan worden gegenereerd door het DBMS (zoals een Globally Unique Identifier of GUID ofwel, in Microsoft SQL Server). Primaire sleutels kan bestaan uit een enkel kenmerk of meerdere kenmerken in combinatie.

Foreign Key (Wat is het? Hoe herken je dit?)

Een veelgebruikte constraint in [relationele databases](http://nl.wikipedia.org/wiki/Relationele_database) is de foreign key. Deze zorgt er voor dat als gegevens in een record verwijzen naar gegevens in een andere record, er alleen verwezen kan worden naar werkelijk bestaande gegevens. Als er bijvoorbeeld in een tabel met bestellingen wordt verwezen naar een tabel met klanten, dan moet in iedere bestelling de desbetreffende klant zijn opgenomen in de tabel met klanten. Een bestelling van een niet-bestaande klant kan op deze manier nooit in de database terechtkomen. En als een bepaalde klant eenmaal in de tabel met bestellingen wordt gebruikt, dan kan de klant niet uit de tabel van klanten worden verwijderd zonder dat eerst de betreffende bestellingen zijn verwijderd.

Een datadictionary.(Wat is dit? En vul het?)

Gedetailleerde uniforme beschrijving van alle gegevens van een informatiesysteem.

Een data dictionary kan worden gezien als een geautomatiseerd docmentatiesysteem of als een woordenboek betreffende de definitie van alle entiteittypen, attributen en relaties. Afhankelijk van het specifieke product kunnen tevens de views of subschema's, de fysieke database kenmerken, de gebruikers en hun privileges en de programma's en hun privileges owrden gedefinieerd. De data dictionary is zelf ook een database.