

Digibeter

Klaas-Dirk van Opstal & Timo de Vries, Universiteit Leiden

8 juni 2007

Colofon

Gemaakt door

Timo de Vries
0308161
tdevries@liacs.nl

Klaas-Dirk van Opstal
0326658
kvopstal@liacs.nl

Begeleid door

Dr.ir. F.J. Verbeek
fverbeek@liacs.nl

Met dank aan

Dr. Harmen Jousma

Webadres

Voorlopig is de applicatie te vinden op <http://digibeter.plexus.leidenuniv.nl>

De informatie uit dit rapport is alleen bestemd voor personen binnen het LIACS. Informatie mag zonder toestemming van Klaas-Dirk van Opstal en Timo de Vries niet gebruikt worden voor andere doeleinden. De applicatie “Digibeter” blijft eigendom van bovengenoemde personen.

Inhoudsopgave

| | | |
|----------|--|-----------|
| 1 | Introductie | 4 |
| 1.1 | Wat is Digibeter? | 4 |
| 1.2 | Doelgroepen | 5 |
| 1.3 | Wat is er vernieuwend? | 5 |
| 2 | Materialen en methoden | 7 |
| 2.1 | De interface | 7 |
| 2.2 | Programmeertalen en scripts | 9 |
| 2.2.1 | HMTL, CSS & Javascript | 9 |
| 2.2.2 | PHP | 10 |
| 2.2.3 | Flash & ActionScript | 10 |
| 2.2.4 | MySQL database | 11 |
| 2.2.5 | Apache2.2 | 13 |
| 2.3 | Beveiliging | 13 |
| 2.4 | Aannames en beperkingen | 13 |
| 2.4.1 | Beperking tot rekenoefeningen | 13 |
| 2.4.2 | Standaard in- en output van spellen | 14 |
| 3 | Resultaten | 15 |
| 3.1 | Werkwijze | 15 |
| 3.1.1 | Opstartfase | 15 |
| 3.1.2 | Implementatiefase | 16 |
| 3.1.3 | Testfase | 16 |
| 3.1.4 | Obstakels | 16 |
| 3.2 | Het eindproduct | 17 |
| 3.2.1 | Interface elementen | 17 |
| 3.2.2 | Flow | 19 |
| 3.2.3 | Modulairiteit of spelspecifieke informatie | 21 |
| 3.3 | Usability tests | 22 |

| | |
|--|-----------|
| <i>Inhoudsopgave</i> | 3 |
| 3.3.1 Test met leerlingen | 22 |
| 3.3.2 Test met leraar | 23 |
| 4 Conclusies & discussie | 26 |
| 4.1 Conclusies | 26 |
| 4.2 Digibeter vanuit een new-venture oogpunt | 27 |
| 4.2.1 Wie zijn de klanten? | 27 |
| 4.2.2 Aanpassen van het basisidee | 27 |
| 4.2.3 Toegankelijkheid | 28 |
| 4.3 Toevoegingen in de toekomst | 29 |
| 4.3.1 Generalisatie van spellen | 29 |
| 4.3.2 Ondersteuning voor alle resoluties | 29 |
| 4.3.3 Terugkoppeling spelresultaten | 30 |
| 4.3.4 Privacy van de gegevens | 30 |
| A Tabellen | 31 |
| B Stroomdiagrammen | 33 |
| C Usability test resultaten | 36 |

Hoofdstuk 1

Introductie

In dit eerste hoofdstuk introduceren we ons Bachelorproject. In sectie 1.1 vertellen we wat Digibeter is. Vervolgens laten we in sectie 1.2 zien wie de doelgroepen zijn voor onze applicatie. Tot slot sommen we in sectie 1.3 de punten op waarin DigiBeter volgens ons vernieuwend is ten opzichte van huidige educatieve software.

1.1 Wat is Digibeter?

Digibeter is een leeromgeving voor leerling en leraar, waarop leerlingen van alle klassen van de basisschool op een leuke manier educatieve spellen en oefeningen kunnen doen. Elke leerling zal via zijn of haar school een eigen account krijgen. Met behulp van dit account worden alle vorderingen van een leerling opgeslagen in de achterliggende database. Leraren zullen zelf de applicatie kunnen gebruiken om de scores en vorderingen van hun leerlingen te bekijken en zullen deze informatie kunnen gebruiken om te bepalen wat het niveau is van een specifieke leerling.

Doordat het om een internet-applicatie gaat, zullen leerlingen en leraren overal waar een computer met internet en een browser beschikbaar is gebruik kunnen maken van Digibeter. Leerlingen kunnen dus de opdracht om een bepaalde oefening of spel te doen mee krijgen als huiswerk. Omdat het om een internet-applicatie gaat zullen leraren thuis ook toegang hebben tot alle informatie over hun leerlingen.

1.2 Doelgroepen

Bij het project hebben we te maken met twee grote doelgroepen. De eerste groep bestaat uit de leerlingen die de spellen en oefeningen gaan doen. De tweede groep bestaat uit de leraren die zich hoofdzakelijk bezig zullen houden met het beheren van klassen met leerlingen. Er is een kleine derde groep, namelijk de administratoren. Aangezien wij dat voorlopig alleen zelf zijn, hebben we deze groep buiten beschouwing gelaten.

Voor de leerlingen is het volgens ons belangrijk dat het gedeelte van de site dat zij moeten gebruiken overzichtelijk is en dat ze geen toegang kunnen krijgen tot de rest van de site, zoals het leraar-gedeelte. Ze moeten dus duidelijk te zien krijgen waar ze in kunnen loggen en hoe ze vervolgens een spel of oefening kunnen starten. Daarnaast is het leuk om de leerlingen toegang te geven tot een lijst met highscores voor elk spel.

Voor de leraren is het gedeelte van de applicatie waar zij leerlingen en klassen kunnen beheren het belangrijkste. Er moet dus voor gezorgd worden dat dit gedeelte overzichtelijk is. Dit willen we doen door de leraar de optie te geven de leerlingen in de klas op allerlei manieren te sorteren en hem of haar vervolgens toegang te geven tot de informatie van een specifieke leerling door op de naam te klikken. De leraar heeft ook de bevoegdheden van een leerling.

1.3 Wat is er vernieuwend?

Van te voren zijn wij op het internet en bij een basisschool als deel van onze analyse een inventarisatie gemaakt van bestaande educatieve software. Hieronder staan onze bevindingen opgesomd.

- Grotendeels oefeningen en weinig ontspanning.
- Als er data wordt opgeslagen gebeurt dit meestal lokaal of op een lokale server en is deze data alleen beschikbaar op school.
- Er is een groot aanbod aan educatieve software.
- Veel software is erg uitgebreid en daardoor onoverzichtelijk.

Het belangrijkste punt van vernieuwing dat Digibeter heeft ten opzichte van het huidige aanbod is dat het online beschikbaar is. Dit geeft leraren en leerlingen een grotere vrijheid in het gebruik van Digibeter. Daarnaast willen we op een aantal andere punten vernieuwingen realiseren. We willen

proberen in de spellen die we maken de educatieve en de ontspannende kant samen te voegen. Ook willen we een poging doen om de leraar alleen de belangrijke data te presenteren en zo Digibeter voor de docent overzichtelijk en duidelijk te houden. Als laatste willen we de mogelijkheid aanbieden om eenvoudig nieuwe spellen toe te voegen.

Hoofdstuk 2

Materialen en methoden

In dit hoofdstuk leggen we uit welke materialen we gebruikt hebben, en hoe deze materialen gebruikt zijn voor het maken van Digibeter. In sectie 2.1 laten we zien wat het startidee is van de interface. Daarna kijken we in sectie 2.2 naar de gebruikte programmeertalen en in sectie 2.3 naar stappen die we genomen hebben om de site te beveiligen. Als laatste lichten we in sectie 2.4 de aannames die we gedaan hebben en de beperkingen die we opgelegd hebben toe.

2.1 De interface

Om een idee te krijgen van hoe de interface eruit moest zien, zijn we begonnen met een *low-fidelity prototype*. We gaven een docent en een leerling beiden een stapeltje papieren interface-elementen en lieten hen daar zelf een interface van samenstellen. De resultaten hiervan zijn te zien in de onderstaande figuren.

Om te beginnen hebben we een leerling het loginscherm laten maken. Dit scherm werd erg simpel, alleen een box om het wachtwoord in te vullen en een knop werden toegevoegd. Dit leek ons een goed idee, daarom hebben we het zo gehouden.

Na het maken van het loginscherm zijn we verder gegaan naar de leerling interface, te zien in figuur 2.1. Dit scherm heeft meerdere elementen. Ten eerste is er de groene startknop, deze is bedoeld voor het starten van de spellen. Rechtsboven staan twee knoppen, de linker is om naar de statistieken en highscores te gaan en de rechter is om naar een “schatkist” te gaan waar alle behaalde beloningen terug te vinden zijn. Als laatste zijn er onderaan acht verschillende knoppen te vinden. Deze knoppen zijn er om een categorie

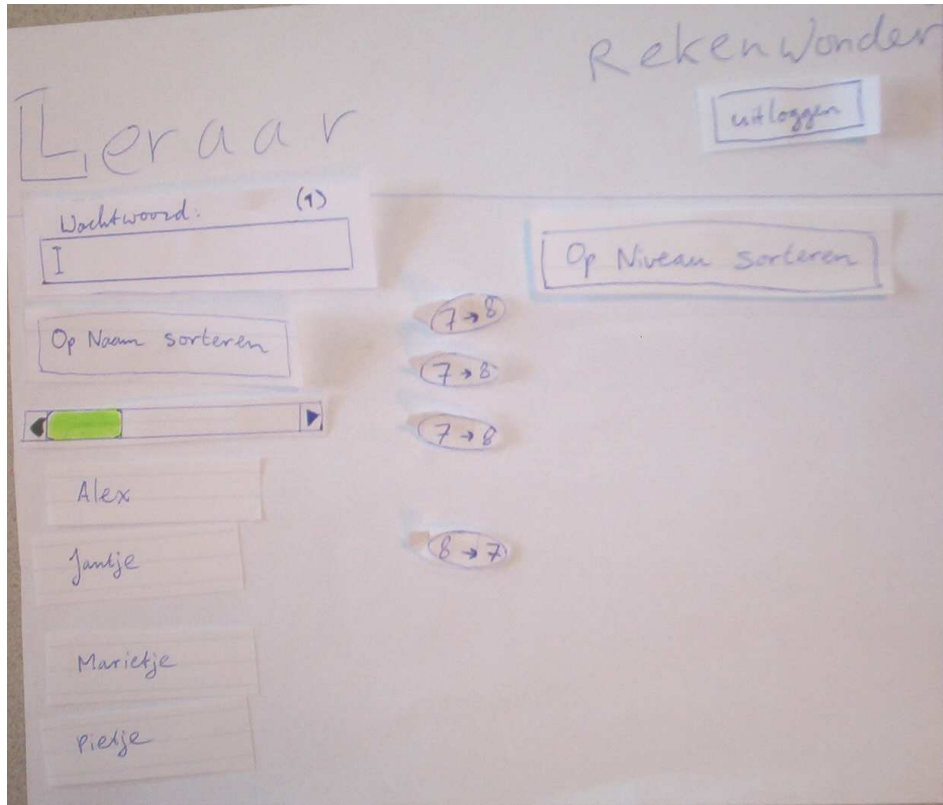


Figuur 2.1: Het low-fidelity prototype leerling-interface

van plaatjes te kiezen die als beloning verschijnen in de spellen. Blijkbaar hechten leerlingen hier veel waarde aan.

Als laatste hebben we een leraar gevraagd om te laten zien hoe de leraar interface er uit moet zien. Als eerste zou een leraar een link moet volgen die naar een apart leraargedeelte van de site verwijst. Hier is dan enkel het wachtwoord element linksboven te zien, dat verdwijnt na het inloggen. Zodra er ingelogd is verschijnt de rest van de leraar interface. Het belangrijkste onderdeel is de lijst met de leerlingen uit de klas van de betreffende leraar. Deze lijst moet te sorteren zijn op naam en op niveau. Daarnaast zijn er ook knoppen om leerlingen makkelijk naar een andere groep te verplaatsen.

De bovenstaande afbeeldingen laten zien ons idee van de interface was voor we begonnen met het maken van de applicatie. Zoals zal blijken is de interface tijdens het project flink veranderd. In sectie 3.2 is te zien hoe de interface geworden is.



Figuur 2.2: Het low-fidelity prototype leraar-interface

2.2 Programmeertalen en scripts

2.2.1 HTML, CSS & Javascript

Toen we begonnen met het opzetten van de applicatie zochten we een methode voor het maken van de statische delen van de website. De standaardomgeving waarin dit meestal gebeurt is HTML, hier hebben wij dan ook voor gekozen. Al onze HTML-code staat in *PHP* functies verwerkt, zie sectie 2.2.2. Op die manier konden we standaard onderdelen van een pagina, zoals de bovenbalk, elke keer opnieuw gebruiken.

Voor de layout hebben we gewerkt met een *Cascading Style Sheet* (CSS). CSS is een techniek voor de stijl en vormgeving van websites. De CSS informatie wordt automatisch toegevoegd aan de HTML-code. Deze informatie

kan in het HTML bestand staan of in een aparte .css-file. Met een stylesheet is het daardoor mogelijk om de inhoud en de opmaak van een document te scheiden, en om de opmaak consistent te houden over meerdere pagina's.

Wij hebben gekozen om alle CSS-informatie in een losse .css-file op te slaan. Met behulp van CSS hebben we onder andere de layout van de applicatie, de “tabbladen interface” en de overige menu's gemaakt. Daarnaast is CSS nog gebruikt voor basis dingen zoals het uiterlijk van kopjes en links.

We hebben in een beperkte mate ook gebruik gemaakt van *Javascript*. Javascript is een scripttaal die veel gebruikt wordt voor websites, de code wordt eerst naar de client gedownload en daar uitgevoerd. Wij hebben het enkel gebruikt voor het maken van de popup waarin de applicatie draait, en voor het dynamisch verwerken van sommige formulieren.

2.2.2 PHP

Aangezien een groot deel van de applicatie dynamisch moet worden is PHP een goede keuze. PHP staat voor *PHP: Hypertext Preprocessor* en is een scripttaal die speciaal bedoeld is voor het maken van dynamische webpagina's. In tegenstelling tot Javascript wordt de code op de server uitgevoerd, hierdoor stelt het lage eisen aan de computer van de gebruiker. PHP is goed te combineren met MySQL.

Bij ons project hebben we veel PHP-code gebruikt. Om deze hoeveelheid code overzichtelijk te houden hebben we een indeling gemaakt van verschillende bestanden. Alle code in een bestand heeft een duidelijke overeenkomst met elkaar, bijvoorbeeld alle code voor de leerlingmodule. Deze indeling is in tabel A.1 te vinden.

We wilden graag een makkelijke en efficiënte manier om bepaalde delen van onze HTML-code te kunnen hergebruiken. Hiervoor hebben we al onze HTML-code verwerkt in apart aan te roepen PHP-functies. Op deze manier is het makkelijk om bepaalde onderdelen van de HTML-code te beïnvloeden met variabelen die we meegeven aan de PHP-functies. Het is nu ook mogelijk om met if-statements en andere controlestructuren te werken. Dit laatste hebben we veel gebruikt om een bepaalde functie een andere output te laten geven, afhankelijk van het type gebruiker dat is ingelogd.

2.2.3 Flash & ActionScript

Gezien de vele mogelijkheden en het gebruikersgemak hebben we gekozen om Flash 8 te gebruiken voor het maken van de spellen. Het maken van dynamische elementen is met Flash een simpel proces. Aan deze elementen

kunnen vervolgens acties worden verbonden met behulp van een eenvoudige scripttaal. Alle spellen zijn in principe losstaande onderdelen, gebaseerd op dezelfde interface. Elk spel is dus een swf-bestand, het bestandstype waar Flash naar exporteert.

Flash MX heeft een eigen scripttaal genaamd ActionScript. Qua opbouw van code vertoont ActionScript grote overeenkomst met JavaScript, het leren omgaan met deze taal was daarom simpel en kostte weinig tijd. We hebben ActionScript gebruikt om acties te koppelen aan de widgets en voor algemene functies. Door al deze functies in een apart frame te zetten hebben we het geheel overzichtelijk gehouden.

Een belangrijk aspect van ons project was de communicatie tussen de spellen en de rest van de applicatie. Na enig onderzoek bleek ActionScript hiervoor een bruikbare oplossing te bieden, namelijk *LoadVars*. Bij het laden van een spel wordt automatisch een variable van het type *LoadVars* aangeemaakt, waarna met behulp van de methode *sendAndLoad* enkele waarden zoals de gebruikersnaam en het huidige level van de leerling worden opgehaald. Wanneer de leerling klaar is met een spel, wordt alle informatie waaronder de behaalde score met *sendAndLoad* teruggestuurd naar de applicatie. De applicatie gebruikt PHP om de informatie op te halen uit en te versturen naar de MySQL database.

2.2.4 MySQL database

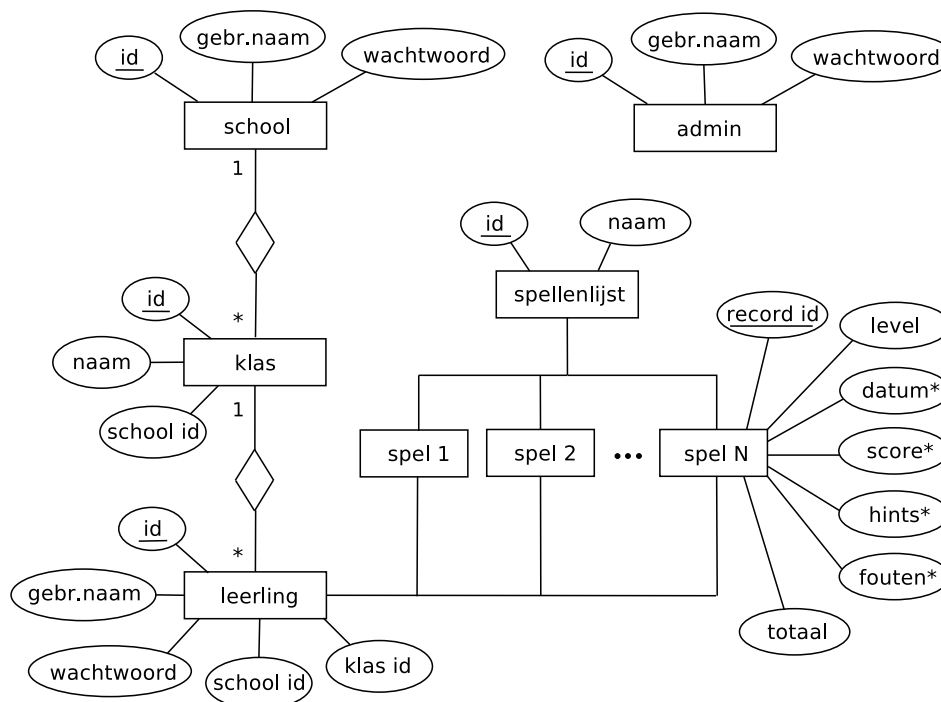
Om alle informatie die samenhangt met de applicatie op een handige manier op te slaan, is een database noodzakelijk. We hebben gekozen voor een MySQL database. Mede omdat MySQL, een *database management systeem* (DBMS), uitgebracht is onder de *GNU General Public License* (GPL). Daarnaast is MySQL op dit moment een populaire DBMS, waardoor er veel informatie over te vinden is. Omdat MySQL goed te draaien is in combinatie met Apache konden we gemakkelijk lokaal een server opzetten om de verschillende versies van ons project te testen.

De communicatie met de database verloopt via queries die in de PHP-code opgenomen zijn. Het grootste deel van de functies die communiceren met de database is ondergebracht in de file `database.php`. Bij het versturen van data naar de database zorgt de PHP-code dat de juiste gegevens in de query gebruikt worden. Bij het ontvangen van data zorgt de code voor het juist verwerken van de gegevens uit de database. Hieronder een voorbeeld van een query die we gebruikt hebben.

```
$sql = "INSERT INTO klassen (school_id, klas_naam)
```

```
VALUES(" . $school_id . ", '" . $klas_naam . "')";

mysql_query($sql, $dbh)
or die ('Invalid Insertion! ' . mysql_error() .
'  
>Druk op backspace om terug te gaan.');
```



Figuur 2.3: Database schema

Onze applicatie maakt onderscheid tussen drie verschillende typen gebruikers: beheerders, scholen en leerlingen. Eén school kan meerdere klassen bevatten, zoals aangegeven met de 1 en de * aan de twee kanten van de relatie (ruit). Een klas kan eveneens meerdere leerlingen bevatten. Een leerling heeft verschillende spellen-records, voor ieder spel één. In zo'n record wordt alle nuttige informatie opgeslagen terwijl de leerling een spelletje speelt. De “...” tussen Spel 2 en Spel N is om aan te geven dat de spellenlijst, en daarmee ook het aantal records per leerling, dynamisch kan groeien.

Achter sommige attributen van een spel staat een sterretje, om aan te geven dat het eigenlijk om zes attributen gaat, die genummerd zijn van 0 tot en met 5. Dit is noodzakelijk om de informatie van verschillende levels

apart op te kunnen slaan. In het kader van de overzichtelijkheid hebben we ervoor gekozen om deze niet allemaal apart weer te geven in het diagram.

2.2.5 Apache2.2

Apache is niet zozeer een programmeer- of scripttaal, maar is wel degelijk belangrijk voor het project. Apache is een webserver die wederom is uitgebracht onder de GPL. De naam Apache is gekozen uit respect voor de Apache indianenstam. Apache werkt met modules die bepaalde server-side talen en dergelijke ondersteunen. Het is de aangewezen kandidaat om te gebruiken samen met MySQL en een scripttaal zoals PHP. Als deze combinatie draait op een Linux OS, dan heet het *LAMP* (Linux, Apache, MySQL & PHP). Om op een server onze applicatie te kunnen draaien is er een Apache server met MySQL en PHP modules nodig.

2.3 Beveiliging

Tijdens het project is beveiliging niet onze hoogste prioriteit geweest, er is dus nog veel te verbeteren. Om toch enigzins aan de beveiliging te werken hebben we alle bestanden waarin code staat gescheiden gehouden in een aparte directory. Het enige bestand dat in de root directory staat is `index.html`. Deze afscherming van de code is de eerste stap in de beveiliging. Voor en overzicht van deze indeling, zie tabel A.2.

Een van de eerste aanpassingen die we kunnen doen om de site veiliger te maken, is het coderen van data die van en naar formulieren gestuurd wordt. Momenteel wordt alle data, dus ook de wachtwoorden, ongecodeerd opgeslagen. Dit brengt natuurlijk een groot risico met zich mee, als iemand bij de database weet te komen ligt alle informatie op straat. Hiervoor is een redelijk simpele oplossing. PHP heeft een standaard functie waarmee gegevens makkelijk gecodeerd kunnen worden voor ze naar de database gezonden worden.

Mochten we later besluiten onze applicatie verder uit te breiden, dan moet er eens goed naar de beveiliging gekeken worden.

2.4 Aannames en beperkingen

2.4.1 Beperking tot rekenoefeningen

Voor de spellen is er gekozen om ons te beperken tot het gebruik van reken spellen en -oefeningen. Bij het vak “Human Computer Interaction” hadden

wij al enkele van deze spellen gemaakt, deze spellen hebben we aangepast voor de site. We hebben deze beperking opgelegd om te voorkomen dat de database en het toevoegen van nieuwe spellen te ingewikkeld zouden worden. We willen in de toekomst de mogelijkheid creëren om de database en het proces van toevoegen te generaliseren, zodat er allerlei soorten spellen en oefeningen toegevoegd kunnen worden.

2.4.2 Standaard in- en output van spellen

Om de communicatie tussen de flashbestanden voor de spellen en de rest van de website overzichtelijk te houden, zijn we er van uit gegaan dat alle spellen die we tijdens het bachelorproject toevoegen met standaard in- en output werken. Het gaat hier om informatie als de naam van de leerling bij het opstarten, en de scores bij het afsluiten van het spel.

Hoofdstuk 3

Resultaten

In dit hoofdstuk zullen we meer vertellen over de resultaten van dit project. In sectie 3.1 bespreken we de manier waarop we te werk zijn gegaan, onder andere de planning en soortgelijke zaken. Vervolgens lichten we in sectie 3.3 de resultaten van onze usability tests, te vinden in appendix C, verder toe. Tot slot in sectie 3.2 laten we zien hoe het eindproduct er uit ziet.

3.1 Werkwijze

3.1.1 Opstartfase

Na enkele gesprekken met onze begeleider hadden we voor onszelf een idee gevormd van de applicatie. Om voor onszelf een leidraad te hebben, besloten we een planning te maken met een duidelijke verdeling van taken, uitgespreid over 4 maanden. Al snel werd echter duidelijk, dat we het niet zouden redden om het in december al af te hebben. Gelukkig hadden we daar al een beetje op gerekend.

In de eerste paar weken zijn we voornamelijk bezig geweest met brainstormen en oriënteren. Om erachter te komen wat voor soort educatieve software er momenteel veel gebruikt wordt op scholen, zijn we langs gegaan bij de Juliana van Stolberg in Hoofddorp. Hoewel het er best mooi uitzag, bleek al gauw dat het leraar-gedeelte veel te uitgebreid was, en daardoor onoverzichtelijk. De grote hoeveelheden informatie leek ons niet erg nuttig, maar eerder storend. We hebben er daarom voor gekozen om ons te beperken tot de belangrijkste informatie, zoals de behaalde scores en het aantal gemaakte fouten en gebruikte hints. Veel informatie lijkt leuk, maar de meeste leraren hebben toch geen zin om dat helemaal uit te pluizen. Ze zien liever in één oogopslag hoe het gaat.

3.1.2 Implementatiefase

Nadat we enkele schetsen hadden van de verschillende interfaces, zijn we begonnen met het bouwen van de website. Door lokaal een MySQL-server te installeren konden we ondertussen alvast wat queries testen op de database. Om de interface intuïtiever te maken, besloten we de knoppen bovenaan de pagina te veranderen in tabs, zodat elke pagina een tabblad werd. Vervolgens begonnen we met het schrijven van functies voor het ophalen en versturen van informatie van en naar de database. Hier hebben we de verschillende interfaces weer op aangepast, zodat de functionaliteit ook daadwerkelijk benut kon worden. Tenslotte besloten we dat het handig is als nieuwe spellen automatisch toegevoegd kunnen worden door een beheerder. In sectie 3.1.4 komen we hier nog op terug.

Aangezien het regelen van een server bij het LIACS te lang duurde en de geplande evaluatie-dag al naderde, is ons aangeboden om ons project via het Plexus online beschikbaar te maken.

3.1.3 Testfase

De eerste online test in groep 8 van de Juliana van Stolberg was zeer geslaagd. Er waren slechts enkele bugs en onvolkomenheden, waar de leerlingen over het algemeen weinig last van hadden. We hebben heel wat van deze evaluatie opgestoken. Zie sectie 3.3.1 voor meer informatie hierover. Ook de tweede test, in groep 7, waarbij we met name gekeken hebben hoe de docent haar weg vond in de leraar-interface, was nuttig. Meer informatie hierover staat in sectie 3.3.2.

3.1.4 Obstakels

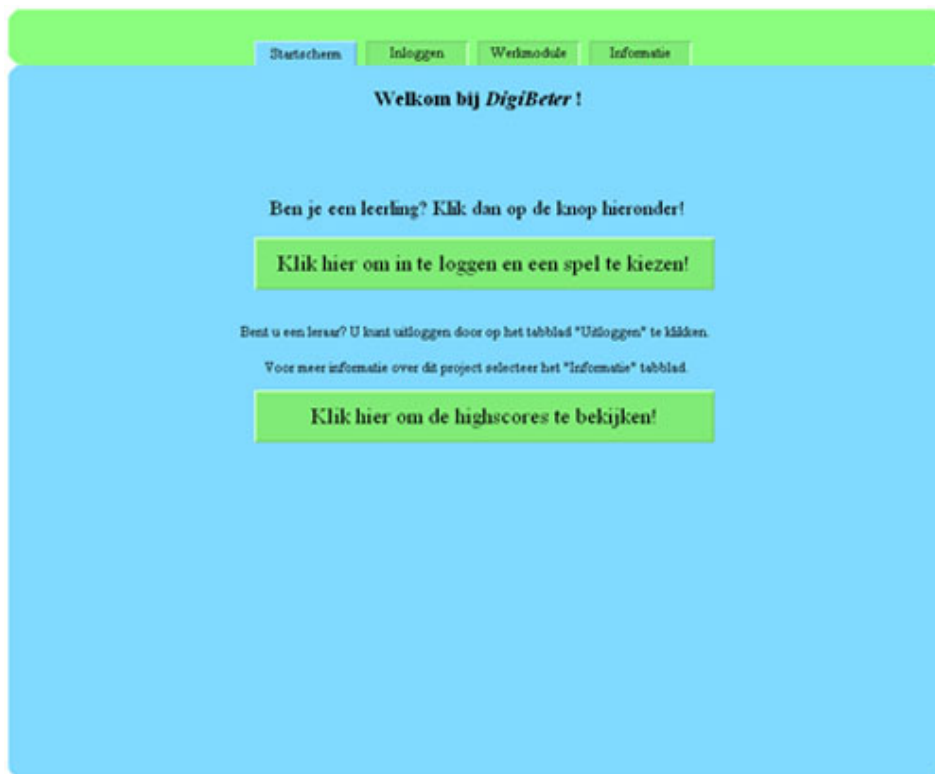
Een van de dingen waar we tegenaan liepen was het versturen van informatie uit de Database naar een Flash-spelletje en vice versa. Dit hebben we uiteindelijk opgelost met behulp van de functie LoadVars, zie sectie 2.2.3.

Het automatisch toevoegen van spellen bleek ook niet bepaald eenvoudig te zijn. Voor spellen met dezelfde variabelen, namelijk scores, hints en fouten op verschillende niveau's, is er geen probleem. Zodra je echter meer gedetailleerde informatie wilt opslaan over een spel, zoals bij 24-game de weg waarlangs een leerling een oplossing heeft gevonden, wordt het een stuk ingewikkelder. Uiteindelijk moet je een afweging maken tussen het makkelijk toevoegen van spellen en de gedetailleerdheid van de informatie.

3.2 Het eindproduct

3.2.1 Interface elementen

Het startscherm is het eerste scherm dat de gebruiker te zien krijgt wanneer de applicatie wordt geopend. We hebben er bewust voor gekozen dit scherm zo eenvoudig mogelijk te maken, zodat de gebruiker niet direct overdonderd wordt door de vele mogelijkheden. Zoals in figuur 3.1 te zien is, staan er bovenaan vier tabbladen, genaamd “Startscherm”, “Inloggen”, “Werkmodule” en “Informatie”. Het startscherm bevat tevens twee grote knoppen. Deze zijn bedoeld voor leerlingen om in te loggen of highscores te bekijken.



Figuur 3.1: Het startscherm van de applicatie

Figuur 3.2 toont het loginscherm, met verschillende opties zodat de verschillende typen gebruikers via hetzelfde scherm in kunnen loggen. Standaard zal de applicatie proberen als leerling in te loggen. Wanneer er als

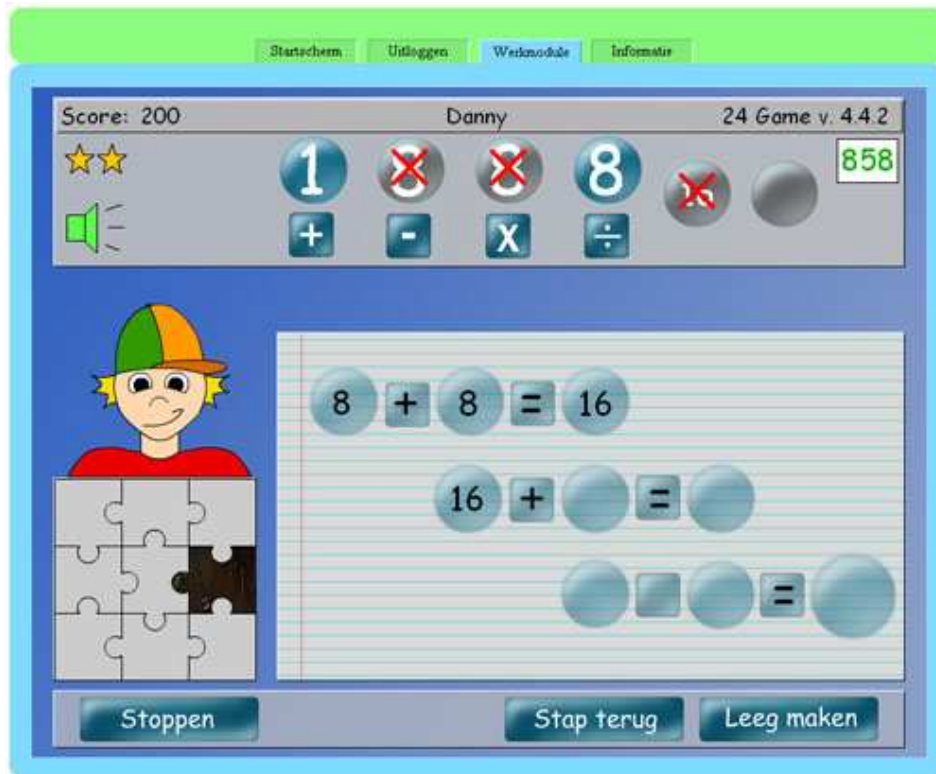
school of admin ingelogd moet worden moet die optie hier specifiek worden gekozen. De knop “Wissen” maakt de velden leeg.



Figuur 3.2: Het loginscherm met opties

Nadat een leerling ingelogd is, komt hij of zij automatisch in de werkmodule terecht. Daar kan een leerling een spel kiezen, zoals 24-Game, om die te gaan spelen. Het scherm ziet er dan uit zoals in figuur 3.3. Het aantal sterren linksboven geeft aan in welk level de leerling zit. Hoe meer sterren, hoe moeilijker de puzzels worden. Rechtsboven staat een tellertje dat aangeeft hoeveel seconden de leerling nog heeft om het level uit te spelen. Door op de knop “Stoppen” te drukken keert de leerling terug naar het spellenmenu, nadat de informatie van het spel is opgeslagen.

Wanneer een leraar inlogt, komt hij of zij ook in de werkmodule terecht. Aan de linkerkant verschijnen nu ook tabs. Via deze tabs kan de leraar klassen en leerlingen toevoegen en verwijderen, alsmede informatie opvragen van de verschillende leerlingen in een klas. Figuur 3.4 toont het scherm dat verschijnt wanneer de leraar informatie opvraagt van leerling Danny. Dit scherm bevat alle belangrijke informatie van de leerling, overzichtelijk weergegeven in tabellen. Via het dropdown-menu kan de docent voor elk spel de gewenste informatie opvragen en bekijken.



Figuur 3.3: Werkmodule voor een leerling

3.2.2 Flow

Om te laten zien hoe de gebruiker zijn weg gaat vinden in de applicatie hebben wij flowcharts gemaakt voor twee veel voorkomende situaties. De eerste situatie is die van een leerling die “24-game” gaat spelen en vervolgens uitlogt. De tweede is van een leraar die een leerling toevoegt en vervolgens weer uitlogt.

Figuur B.1 is de flowchart voor de eerste situatie. Dit voorbeeld laat “24-game” zien, het werkt echter hetzelfde voor elk willekeurig spel. De leerling start de applicatie en gebruikt de grote knop op het startscherm om naar het inlogscherm te komen. Hier vult de leerling zijn of haar eigen naam en wachtwoord in om in te loggen in de applicatie. Na het inloggen komt de leerling bij een menu waaruit spellen gekozen kunnen worden. De leerling kiest het spel “24-game”. Wanneer de leerling klaar is met spelen

The screenshot displays the 'Werkmodule' interface for a teacher. At the top, there are navigation tabs: 'Startscherm', 'Uitloggen', 'Werkmodule' (selected), and 'Informatie'. On the left sidebar, there are buttons for 'Klas toevoegen', 'Klassen beheren', 'Leerling toevoegen', and 'Leerlingen beheren'. The main content area shows the following information:

41: Danny
(Wachtwoord: p)
School ID: 1
Groep: groep7

game24 ▼

Spel: game24

| Level | Datum | Score | Hints | Fouten |
|-------|------------|-------|-------|--------|
| 0 | 23-03-2007 | 1755 | 6 | 21 |
| 1 | 23-03-2007 | 3496 | 2 | 6 |
| 2 | 23-03-2007 | 200 | 1 | 2 |
| 3 | ----- | 0 | 0 | 0 |
| 4 | ----- | 0 | 0 | 0 |
| 5 | ----- | 0 | 0 | 0 |

Totaalscore: 5451
Huidig level: 2

| Oplosmethoden: | | |
|----------------|-----|------|
| 4*6 | 5x | 23 % |
| 3*8 | 3x | 14 % |
| 2*12 | 2x | 9 % |
| rest | 12x | 55 % |

Figuur 3.4: Werkmodule voor een leraar

keert de applicatie automatisch terug naar het keuzemenu. Wil de leerling eerder stoppen met spelen, dan kan hij of zij altijd op de tab “Werkmodule” klikken. Vervolgens klikt de leerling op de tab “Uitloggen”, de applicatie gaat terug naar het Loginscherm zodat de volgende in kan loggen. Het is ook nog mogelijk om tijdens het spelen van een spel op de tab “Uitloggen” te klikken.

Figuur B.2 laat de flowchart zien voor de tweede situatie. De leraar start de applicatie en klikt op de tab “Inloggen” om in het inlogscherm te komen. Hier kiest de leraar voor “Inloggen als school” en vult de naam en het wachtwoord van de school in. Vervolgens verschijnt de “Werkmodule”, waarna de leraar op “Leerling toevoegen” klikt. In het formulier dat verschijnt vult de leraar een naam en wachtwoord in voor de nieuwe leerling. Hierna selecteert hij of zij de juiste klas en klikt op “Toevoegen”. De leraar krijgt een beves-

tiging en wordt automatisch naar het formulier teruggestuurd. Om terug te komen in de werkmodule kan de leraar altijd op de tab “Werkmodule” klikken. Om uit te loggen klikt de leraar op de tab “Uitloggen” en wordt dan naar het inlogscherf gestuurd. Dit uitloggen kan vanuit het formulier en vanuit de werkmodule.

3.2.3 Modulairiteit of spelspecifieke informatie

Tijdens het opbouwen van de site hebben we een belangrijke keuze moeten maken. Er moest gekozen worden tussen het makkelijk inbouwen van nieuwe spellen door de admin of gedetailleerdere informatie voor de leraar.

De keuze voor het makkelijk inbouwen van spellen gaat gepaard met een simpele databasetabel die voor elk spel hetzelfde eruit ziet. Dit komt doordat we momenteel een functie met een standaard MySQL-query gebruiken. Het resultaat is dat de docent slechts beperkte informatie over een spel te zien krijgt, dus geen details die specifiek zijn voor een bepaald spel. De beperkte informatie bestaat onder andere uit hoeveel punten een leerling op elk level gehaald heeft en hoeveel fouten er gemaakt zijn.

De tweede keuze is om meer details voor de leraar te verzorgen. De manier waarop dit bereikt kan worden is het handmatig opbouwen van een database tabel. Dat zou dan elke keer als er een nieuw spel toegevoegd wordt moeten gebeuren. Hoewel dit meer moeite zou kosten, heeft het wel als gevolg dat de tabel zo kan worden gemaakt dat er informatie in komt die specifiek is voor het spel. Als extra informatie denken we bijvoorbeeld aan het opslaan van specifieke fouten.

Voor het bachelorproject hebben we gekozen voor de eerste optie, het modulair invoeren van nieuwe spellen. Vanuit het admin-gedeelte kan een nieuw spel, dat al wel in de juiste directory op de server staat, ingevoerd worden. Deze keuze hebben we gemaakt om dit voor het bachelorproject niet te ingewikkeld te maken.

In de toekomst is het wel de bedoeling hier op verder te bouwen. Wat we uiteindelijk willen is dat de bestanden voor een nieuw spel via een file transfer op de juiste plaats gezet worden, en dat er bij het invoegen aangegeven kan worden hoe de database-tabel er uit moet gaan zien. Dit zit dus ergens tussen de twee bovengenoemde opties in, en zou volgens ons de beste methode zijn.

3.3 Usability tests

3.3.1 Test met leerlingen

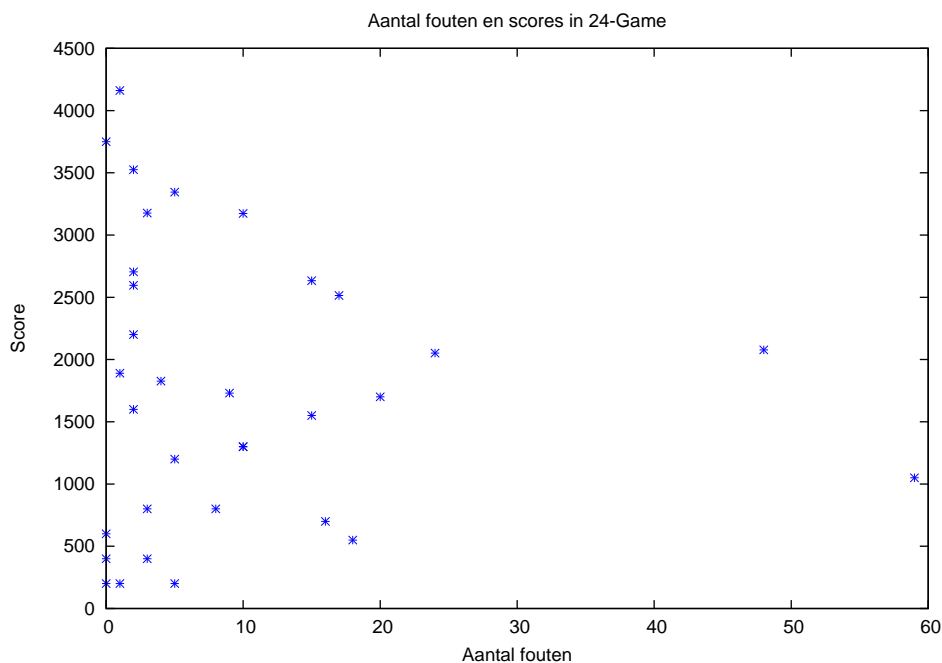
Toen we eenmaal een goed werkend prototype hadden van de applicatie, hebben we een gebruikersevaluatie gedaan met een klas basisschoolleerlingen. Om deze test te kunnen doen, hebben we ervoor gezorgd dat de applicatie via het internet te bereiken is. Een van de punten waar we extra op gelet hebben, was dan ook of de interactie met de webserver goed werkte. Tot dit moment hadden we namelijk alleen lokale tests uitgevoerd. We wilden zien of er vertraging was ten opzichte van de laadsnelheden bij de lokale tests. Daarnaast waren we extra beducht op foutmeldingen. Gelukkig leverde dit geen problemen op en draaide de applicatie zonder vertraging. We hebben ons dus kunnen concentreren op het testen van de applicatie zelf.

De test is uitgevoerd in groep 7 van de Juliana van Stolberg te Hoofddorp. Op deze school zijn er genoeg laptops om een hele klas tegelijk met de computer te kunnen laten werken. Voor de test hadden wij zelf alvast voor elke leerling een account aangemaakt. Daarna hebben we klassikaal uitgelegd wat de bedoeling was en gingen de leerlingen met de applicatie aan de slag. Tijdens dit half uur liepen wij rond om problemen op te lossen en aantekeningen te maken. We sloten af met de vraag wat de leerlingen ervan vonden en of ze nog suggesties hadden voor verbeteringen.

De opmerkingen die we gekregen en de bugs die we gevonden hebben tijdens de test zijn te vinden in tabel C.1. Over het algemeen vonden de leerlingen de applicatie duidelijk en de spellen leuk en leerzaam, vooral 24 game. Wel is er duidelijk gebleken, dat de leerlingen meer verschillende spellen wel op prijs zouden stellen. De overige opmerkingen zijn veelal gerelateerd aan onduidelijke details in de interface van de spellen. Al deze problemen zijn in principe eenvoudig op te lossen.

Aangezien 24 game het meest complete spel is in de applicatie, hebben we besloten om hier een aantal statistieken uit af te leiden. Hiervoor hebben we de informatie, die verkregen is tijdens de evaluatie, uit de database gehaald.

In figuur 3.5 hebben we het aantal gemaakte fouten tegen de behaalde scores afgebeeld, ieder sterretje representeert een leerling. Dit hebben we gedaan omdat we verwachtten dat hier een verband tussen zou bestaan. Uit de figuur blijkt dat het merendeel van de leerlingen onder de tien fouten blijft gedurende een half uur spelen. Daarnaast valt uit de figuur op te maken dat een leerling, die meer fouten maakt, ook een lagere score behaalt. Dit valt te verklaren doordat leerlingen die meer fouten maken over het algemeen meer moeite hebben met het spel. Het verband dat hier naar voren komt is



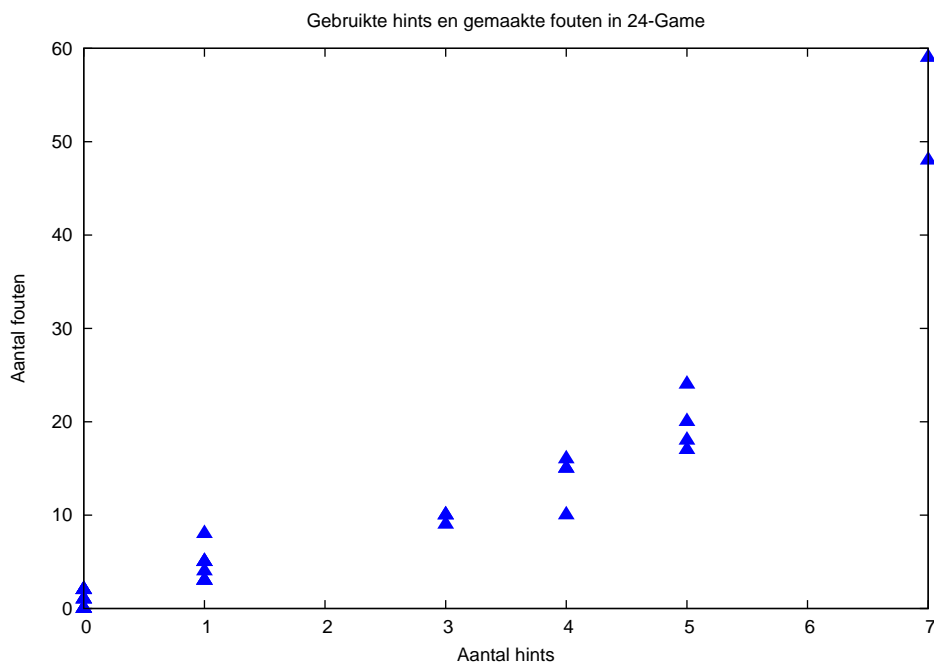
Figuur 3.5: Verband tussen fouten en score in 24 game

logisch en heeft verder geen gevolgen voor de implementatie van dit spel.

In figuur 3.6 is het aantal hints tegen het aantal gemaakte fouten afgezet. We wilden graag zien of het aantal gemaakte fouten samenhangt met het aantal gebruikte hints. In de figuur is te zien dat leerlingen die weinig hints gebruiken ook weinig fouten maken. Leerlingen die meer hints nodig hebben, maken ook meer fouten. Hieruit komt naar voren dat leerlingen die meer hints nodig hebben, meer moeite hebben met het spel. Zij maken daardoor ook meer fouten. Dit verband kan er op wijzen dat de hints onvoldoende steun bieden aan de leerlingen. Dit kunnen we oplossen door de manier van hints geven te verbeteren.

3.3.2 Test met leraar

Een week na de eerste evaluatie hebben we nog een tweede test gedaan. Ditmaal met de focus op de manier waarop de leraar met de applicatie omgaat. We hebben daarbij gelet op verschillende aspecten. Kan de leraar zijn of haar weg vinden in de interface? Is het duidelijk hoe klassen en leerlin-



Figuur 3.6: Verband tussen hints en fouten in 24 game

gen toegevoegd worden en is de informatie van de leerlingen gemakkelijk te bereiken en relevant?

Het viel gelijk op dat de leraar een soort vrees had om te leren werken met de nieuwe applicatie. Volgens ons kwam dit omdat de leraar weinig affiniteit had met computers in het algemeen. Het is echter wel iets om in de toekomst de interface op aan te passen. Het eerste echte probleem waar we tegen aan liepen was dat de optie “Inloggen als school” moeilijk te vinden is. Deze linkjes staan redelijk klein onder het inlog-formulier, een oplossing is het maken van een duidelijke knop voor deze optie.

Ook bij de leraarevaluatie kwam naar voren dat de basis van de applicatie goed in elkaar zit, en dat het een prima basis is voor een verdere uitbreiding. De leraar vond de informatie die weergegeven wordt over de leerlingen handig, maar wat specifiekere informatie is wel gewenst. Wederom kwam naar voren dat meer variatie in de spellen en oefeningen de volgende stap is die we moeten zetten. Suggesties waren onder andere een dictee-oefening, kruiswoordpuzzels, type-oefeningen en spellings-oefeningen.

Een overzicht van de opmerkingen en suggesties die naar voren kwamen uit de gebruikersevaluatie met een leraar is te vinden in tabel C.2.

Hoofdstuk 4

Conclusies & discussie

In dit laatste hoofdstuk zullen we onze conclusie toelichten, zie 4.1. Vervolgens in sectie 4.2 zullen we het hebben over de potentie van Digibeter om als product in de markt te worden gezet. Ten slotte noemen we in sectie 4.3 een aantal dingen die we in de toekomst nog toe willen voegen.

4.1 Conclusies

- Digibeter is een online applicatie voor zowel leerlingen als leraren. Ten opzichte van het huidige aanbod aan vergelijkbare applicaties heeft Digibeter een aantal vernieuwende onderdelen.
- De overloop tussen HTML en Flash verloopt naadloos, de gebruiker merkt er niks van. Ook het verzenden van informatie van en naar de database gebeurt in no-time.
- LAMP (Linux, Apache, MySQL & PHP) is een stabiele en gebruiksvriendelijk combinatie, die een perfecte ondersteuning biedt voor Digibeter.
- Alle gegevens worden opgeslagen op de server, zo kunnen ze niet per ongeluk worden verwijderd.
- We hebben regelmatig met onze doelgroepen tests uitgevoerd en onze applicatie aan de hand van de resultaten aangepast. Deze werkwijze heeft zijn vruchten afgeworpen.
- De interface ziet er aantrekkelijk uit, maar is simpel te navigeren. Leerlingen en leraren vinden binnen enkele minuten hun weg door de interface.

- De spellen in Digibeter worden door de leerlingen erg leuk gevonden. De ondersteuning voor verschillende niveaus zorgt dat de spellen leuk en uitdagend blijven.
- Alle informatie die voor de leraar interessant is, is eenvoudig te bereiken met slechts enkele muisklikken en wordt overzichtelijk weergegeven. Deze informatie is op een simpele manier te sorteren zodat de leraar snel kan zien hoe zijn leerlingen ten opzichte van elkaar presteren.
- Voor de beheerders is het mogelijk om in een oogwenk nieuwe spellen toe te voegen aan de applicatie.
- Het idee van Digibeter is in de loop van het project geëvolueerd. Uiteindelijk zijn wij heel tevreden met het eindproduct.

4.2 Digibeter vanuit een new-venture oogpunt

Wij zijn benieuwd in hoeverre het idee van Digibeter de potentie heeft om in de toekomst op de markt gebracht te worden. We hebben hierover een gesprek gevoerd met Dr. Harmen Jousma, Program manager van Science Based Business aan de Universiteit Leiden. Hij heeft met ons naar het project gekeken vanuit een new-venture oogpunt. De volgende zaken kwamen daarbij aan het licht.

4.2.1 Wie zijn de klanten?

We zijn bij dit project uitgegaan van twee grote doelgroepen, namelijk de basisscholen en hun leerlingen. Aan deze aanname hebben wij ons idee aangepast. Als we echter gaan kijken naar wie de betalende doelgroep is, oftewel de klant, komen we uit bij de basisscholen. Zij zullen degenen zijn die betalen voor ons product, dus is het belangrijk hun eisen voorop te zetten. Een andere groep klanten voor ons product zullen ook de ouders van basisschoolleerlingen zijn. Het is een goed idee om ouders ook in de mogelijkheid te stellen een abonnement te nemen voor hun kind. We moeten dus goed in de gaten houden welke van onze doelgroepen de betalende doelgroep is.

4.2.2 Aanpassen van het basisidee

Vanaf het begin is ons idee geweest om een *portal* te maken, waarin op een dynamische en eenvoudige manier spellen toegevoegd kunnen worden. Met dit doel maken we iets wat eigenlijk al bestaat, er zijn namelijk meerdere

sites op internet te vinden die hetzelfde voor ogen hebben. We moeten ons echter afvragen wat een school of ouder van een basisschoolkind als doel voor ogen heeft, en hoe wij met ons product aan dit doel kunnen voldoen.

Van het begin is het modulair toevoegen van spellen een onderdeel geweest van ons basisidee. Zo zou het makkelijk worden voor iedereen om spellen toe te voegen. Het resultaat zou een *portal* worden met veel educatieve spellen. Die grote hoeveelheid spellen kan alleen wel als gevolg krijgen dat we de kwaliteit en de educatieve waarde van de spellen uit het oog verliezen. Als we echter ons verplaatsen in de klanten (basisscholen, ouders, remedial teachers) zal de hoeveelheid spellen er weinig toe doen. Deze doelgroep is op zoek naar een *portal* met een selectie van kwalitatief goede spellen van grote educatieve waarde.

Het gevolg hiervan voor ons basisidee is dat we de focus af halen van het makkelijk toevoegen van spellen. Om de kwaliteit te waarborgen zal elk spel apart bekeken en aangepast moeten worden, vervolgens moet het handmatig toegevoegd worden zodat de leraar of ouder zoveel mogelijk gedetailleerde informatie krijgt. Ook dit proces zal voor een groot deel te automatiseren zijn. In plaats van het veel spellen beschikbaar stellen voor leerlingen gaan we zorgen voor veel goede informatie voor de scholen en ouders. Hierbij moeten we niet uit het oog verliezen dat de leerlingen nog steeds belangrijk zijn, en graag leuke spelletjes spelen. De uitdaging zal dus zijn om aan de eisen van deze verschillende doelgroepen te voldoen.

Ook lijkt het ons interessant om uit te gaan zoeken of ouders en scholen nog behoefte hebben aan andere informatie over hun leerlingen. Een voorbeeld dat genoemd werd is een programma dat kinderen in spelvorm een dagboek laat bijhouden. Het doel van dagboek is om de eetpatronen van het kind bij te houden en zo een eventuele voedselallergie te vinden. Het kan dus lonen om in de toekomst niet alleen spellen te leveren die voor een basisschool relevant zijn, maar ook spellen met een heel ander doel.

4.2.3 Toegankelijkheid

Een laatste punt wat aan het licht kwam is de bereikbaarheid en toegankelijkheid van Digibeter. Om bij het *portal* te komen zal er altijd een internetverbinding nodig zijn, dat zal simpelweg zo blijven. Er zullen echter ouders en scholen zijn die de kinderen niet zondermeer toegang geven tot het gehele internet.

Een van de oplossingen voor dit probleem is het aanbieden van een remote desktop. We zouden het project in de toekomst zo uit kunnen breiden dat we een mogelijkheid gaan bieden waarop kinderen zonder een internet

browser toch in kunnen loggen op Digibeter. Hiervoor zou de remote desktop client van Windows gebruikt kunnen worden. Het zou ook kunnen met een klein programma wat geïnstalleerd moeten worden, en vervolgens het inloggen verzorgt. Na het inloggen komen kinderen dan op een beschermde desktop terecht waar ze de spellen op kunnen spelen.

4.3 Toevoegingen in de toekomst

4.3.1 Generalisatie van spellen

Om het bachelorproject wat te beperken hebben we de keuze gemaakt om ons enkel bezig te houden met rekenspellen, waarvan we er al een aantal beschikbaar hadden. Het resultaat hiervan is een naar ons idee incomplete site. Daarom zouden we in de toekomst uit willen breiden met ondersteuning voor spellen en oefeningen voor andere basisschool vakken. Enkele ideeën die we hier voor hebben zijn te vinden in het lijstje hieronder.

- Topografie, hoofdsteden koppelen aan de juiste provincie.
- Taal, oefenen met typen.
- Taal, werkwoorden oefenen.

Voor het realiseren van deze generalisatie zal veel werk nodig zijn. Zo zal in ieder geval de manier waarop de databasetabellen aangemaakt worden uitgebreid moeten worden zodat de admin in de interface aan kan geven wat voor velden de tabel nodig heeft. Ook zal de site om moeten kunnen gaan met verschillende outputs vanuit de flash bestanden.

4.3.2 Ondersteuning voor alle resoluties

In het het eindproduct voor ons bachelorproject gebruiken we een vaste resolutie. Dit hebben we zo gedaan omdat onze spellen allemaal dezelfde afmeting in pixels hebben. We *embedden* de spellen in onze applicatie en daarom heb we de afmetingen van de applicatie hierop aangepast. Het grote nadeel hiervan is dat de computer van de gebruiker minstens op een resolutie van 1024x768 moet draaien om de applicatie volledig in beeld te krijgen. Op veel hoger resolutities word de applicatie daarentegen erg klein.

Een verbeterpunt in de toekomst is om ondersteuning te gaan bieden voor alle resoluties. De manier waarop we dit zouden kunnen doen is door de site op een andere manier op te bouwen. Door enkel de flash bestanden, de spellen, te starten in een pop-up en de rest van de applicatie op te bouwen

met percentages in plaats van vaste afmetingen is het probleem eenvoudig te verhelpen

4.3.3 Terugkoppeling spelresultaten

Op het moment doen we weinig met de informatie over leerlingen in de database. In de toekomst zou het een goed plan zijn om met de resultaten terug te gaan naar de leraar. We kunnen dan de informatie vergelijken met de kennis die de leraar over zijn leerlingen heeft. Zo kunnen we bijvoorbeeld uitzoeken wat de reden is achter een slechte score van een leerling. Doet de leerling het niet goed omdat hij of zij slecht is in een bepaald onderdeel, of begrijpt de leerling het spel slecht? Informatie van deze aard is erg belangrijk om de applicatie in de toekomst te kunnen verbeteren.

4.3.4 Privacy van de gegevens

Iedereen die beheerder is kan in principe bij de informatie over alle leerlingen van alle scholen. Momenteel zou een beheerder vergelijkingen kunnen maken tussen meerdere scholen en dit doorspelen aan ouders. Dit is echter niet de bedoeling, en zal in de toekomst eventueel moeten veranderen om de privacy van de gegevens richting de scholen te waarborgen.

Bijlage A

Tabellen

Tabel A.1: Overzicht van de bestanden

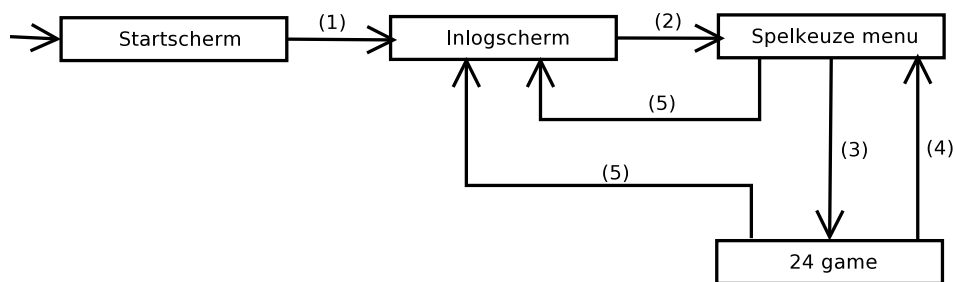
| Bestand | Omschrijving |
|-------------------|---|
| index.html | Aanroepen van main.php in een pop-up |
| main.php | Startscherm van de applicatie |
| login.php | Loginscherm voor verschillende soorten gebruikers |
| leerling.php | Leerlingmodule, te zien na inloggen als leerling |
| school.php | Schoolmodule, te zien na inloggen als school |
| admin.php | Adminmodule, te zien na inloggen als admin |
| anoniem.php | Voor verwerken uitloggen en doorverwijzen naar inloggen |
| info.php | Informatiepagina |
| htmlgen.php | Genereren van benodigde html voor verschillende modules |
| database.php | Interactie met de database |
| dbfuncties.php | Aanroepen van juiste functie uit database.php |
| tables.php | In database zetten van juiste tabellen indien nodig |
| conn.php | Informatie voor connectie met database |
| transact-user.php | Functies voor in- en uitloggen gebruikers |
| Main.css | Stylesheet voor de gehele applicatie |

Tabel A.2: Overzicht van de directories

| Directory | Omschrijving |
|-----------|---|
| .. | Rootdirectory, enkel index.html is hier te vinden |
| ../inc | Bevat alle bestanden met code |
| ../img | Bevat alle plaatjes en dergelijk |
| ../flash | Bevat alle flashbestanden voor spellen |

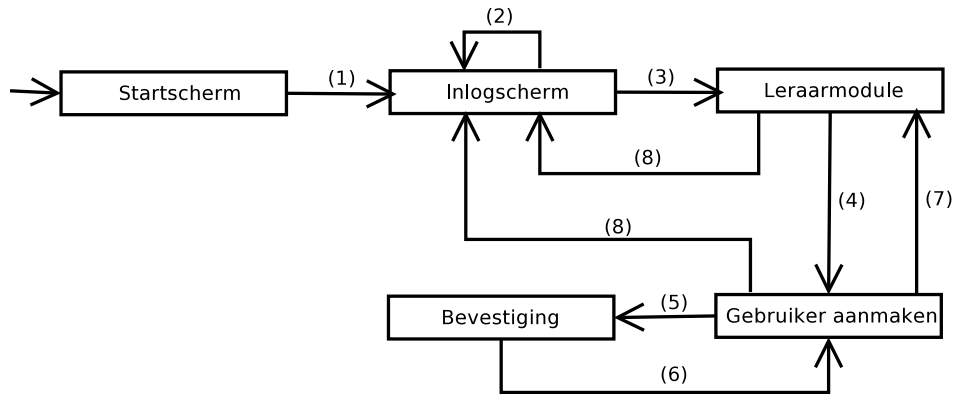
Bijlage B

Stroomdiagrammen



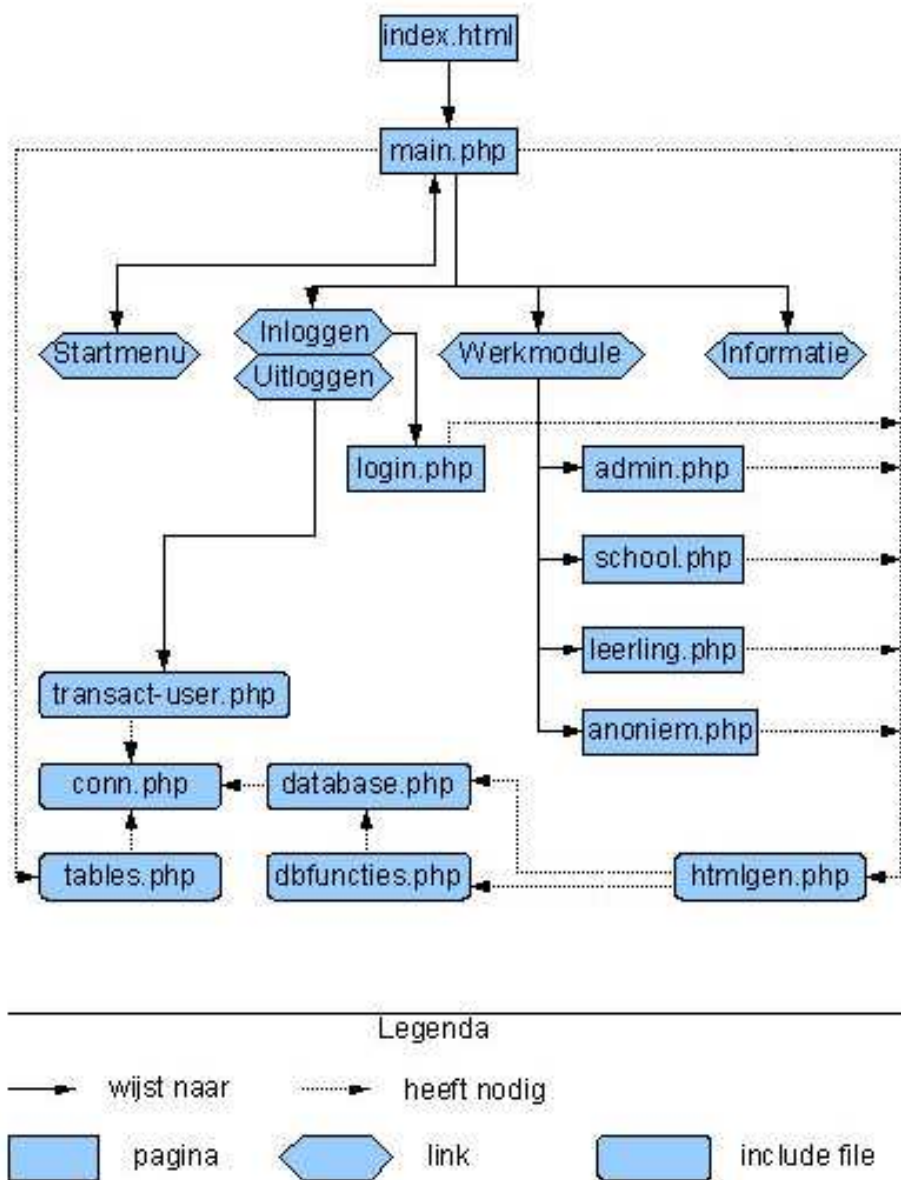
Figuur B.1: Leerling flowchart

1. Leerling klikt op de knop “Klik hier om in te loggen en een spel te kiezen!”
2. Leerling vult zijn naam en wachtwoord in en klikt op “Login”
3. Leerling kiest uit het menu de knop voor “24-game”
4. Leerling speelt 24 game tot het spel gestopt wordt of de leerling klikt op de tab “Werkmodule”
5. Leerling klikt op de tab “Uitloggen”



Figuur B.2: Leraar flowchart

1. Leraar klikt op de tab “Inloggen”
2. Om in te loggen als school klikt de leraar op “Inloggen als school”
3. Leraar vult zijn naam en wachtwoord in en klikt op “Login”
4. Leraar klikt op de tab “Leerling toevoegen”
5. Leraar vult naam en wachtwoord voor nieuwe leerling in, kiest de juiste klas en klikt op “Toevoegen”
6. Applicatie gaat automatisch terug naar het vorige scherm
7. Leraar klikt op de tab “Werkmodule”
8. Leraar klikt op de tab “Uitloggen”



Figuur B.3: De Samenhang van de verschillende bestanden

Bijlage C

Usability test resultaten

Tabel C.1: Feedback uit gebruikersevaluatie met leerlingen

| Opmerking | Verwerking |
|-----------------------------------|--------------------------------------|
| Opmerkingen van leerlingen | |
| Spellen zijn leuk en leerzaam | |
| 24 game onduidelijk | Na enige uitleg is het duidelijk |
| 24 game helpt met hoofdrekenen | |
| 24 game is het leukste spel | |
| Meer variatie in spellen | Taal, procenten etc. |
| Variatie in beloningsplaatjes | Voetbal, huisdieren etc. |
| Layout van de site is duidelijk | |
| Beloningspuzzel is leuk | |
| Rekentoets om niveau te testen | |
| De nul in spellen na de 9 | Zelfde als toetsenbordlayout |
| Ontdekte bugs | |
| Einde van spel onduidelijk | Groene knop duidelijker of vervangen |
| Strafpunten slecht zichtbaar | |
| Menu laten scrollen met een knop | Menu werkt nu erg onduidelijk |
| Duidelijke tijdsindicatie | bv grotere cijfers als tijd op raakt |

Tabel C.2: Feedback uit gebruikersevaluatie met leraren

| Opmerking | Verwerking |
|---|----------------------------------|
| Leraar moet wennen aan interface | Logisch bij iets nieuws |
| Klas onthouden bij leerlingen toevoegen | |
| Spaties in namen | Uitzoeken of dat wel of niet kan |
| Dictee-spel | Wel een koptelefoon nodig! |
| Leraar vindt gebruikersinfo handig | |
| Spellings-spel | |
| Kruiswoordpuzzels | |
| Fouten verbeteren in een tekst | Oefenen in fouten vinden |
| Typespel, overtikken van woorden/zinnen | |

Bibliografie

- [1] A. Hollander, CSS in 10 minuten, Pearson Education, 2005
- [2] M. Wandschneider, Web application development with PHP and MySQL, Prentice Hall, 2006
- [3] Ramakrishnan & Gehrke, Database Management Systems, McGraw-Hill Higher Education, 2003
- [4] Alan Dix et al., Human Computer Interaction (3rd edition), Pearson-Prentice Hall, 2003