2012De Gelder Raf

2013Van Den Branden Kurt

***Mobile reader applicatie en webcomponent voor nieuwssite***

Ondernemingsproject voorgedragen tot het behalen van het diploma van Gegradueerde in de Informatica

Promotor:

De heer Johan Geybels

Copromotor:

De heer Gilles Van Mol *Twipe Mobile Solutions*

Campus Vesalius - Vesaliusstraat 13 - B-3000 Leuven - Tel. +32 16 30 10 30 - Fax +32 16 30 10 40 - E-mail: groept@groept.be

2012De Gelder Raf

2013Van Den Branden Kurt

***Mobile reader applicatie en webcomponent voor nieuwssite***

Ondernemingsproject voorgedragen tot het behalen van het diploma van Gegradueerde in de Informatica

Promotor:

De heer Johan Geybels

Copromotor:

De heer Gilles Van Mol *Twipe Mobile Solutions*

Content

[1 Inleiding 8](#_Toc356899867)

[1.1 Algemeen 8](#_Toc356899868)

[1.2 Opdrachtgever 8](#_Toc356899869)

[1.3 Projectdefinitie 9](#_Toc356899870)

[2 Scope – Probleem analyse – Business analyse 10](#_Toc356899871)

[2.1 Scope 10](#_Toc356899872)

[2.1.1 High level event – response list 10](#_Toc356899873)

[2.1.2 High level context diagramma 12](#_Toc356899874)

[2.1.3 High level functional decomposition diagram 13](#_Toc356899875)

[2.1.4 Scope volgens de MoSCoW-methode 14](#_Toc356899876)

[2.2 Probleem analyse & business analyse 16](#_Toc356899877)

[2.2.1 Problem statement table 16](#_Toc356899878)

[2.2.2 Cause & effect analyse voor de webcomponent 18](#_Toc356899879)

[2.3 Uiteindelijke werkwijze 21](#_Toc356899880)

[3 Analyse en ontwerp database 22](#_Toc356899881)

[3.1 Entity – Relationship Diagram 22](#_Toc356899882)

[3.2 Logisch ERD model 22](#_Toc356899883)

[3.2.1 RSS 2.0 Standaard voor een rss-feed 23](#_Toc356899884)

[3.2.2 Eerste Logisch ERD model 24](#_Toc356899885)

[3.2.3 Tweede Logisch ERD model 25](#_Toc356899886)

[3.3 Fysiek database model van een nieuwsblad rss-feed 26](#_Toc356899887)

[3.4 Database definitie 27](#_Toc356899888)

[3.4.1 Channel 28](#_Toc356899889)

[3.4.2 Article 29](#_Toc356899890)

[3.4.3 Source 30](#_Toc356899891)

[3.4.4 Enclosure 30](#_Toc356899892)

[3.4.5 GlobalUniqueIdentifier 31](#_Toc356899893)

[3.4.6 Cloud 31](#_Toc356899894)

[3.4.7 Image 32](#_Toc356899895)

[3.4.8 TextInput 33](#_Toc356899896)

[3.4.9 HoursToBeSkipped 33](#_Toc356899897)

[3.4.10 DaysToBeSkipped 34](#_Toc356899898)

[3.4.11 HoursToBeSkippedForChannel 34](#_Toc356899899)

[3.4.12 DaysToBeSkippedForChannel 34](#_Toc356899900)

[3.4.13 Category 35](#_Toc356899901)

[3.4.14 CategoriesPerChannel 35](#_Toc356899902)

[3.4.15 CategoriesPerArticle 35](#_Toc356899903)

[3.4.16 User 36](#_Toc356899904)

[3.4.17 ChannelsPerUser 36](#_Toc356899905)

[3.4.18 Preference 37](#_Toc356899906)

[3.4.19 Layout 37](#_Toc356899907)

[3.4.20 Font 38](#_Toc356899908)

[3.4.21 Effect 38](#_Toc356899909)

[3.4.22 Color 39](#_Toc356899910)

[3.4.23 History 39](#_Toc356899911)

[4 Requirement analyse 40](#_Toc356899912)

[4.1 Use cases en storyboards 40](#_Toc356899913)

[4.1.1 High level use cases 40](#_Toc356899914)

[4.1.2 StoryBoards 42](#_Toc356899915)

[4.2 Object georiënteerde analyse & ontwerp van de use cases 55](#_Toc356899916)

[4.2.1 Class diagramma 55](#_Toc356899917)

[4.2.2 Sequence diagramma’s 57](#_Toc356899918)

[4.2.3 Activity diagramma’s 57](#_Toc356899919)

[5 Decision analyse en physical design 59](#_Toc356899920)

[5.1 The meaning of offline 59](#_Toc356899921)

[5.1.1 Android en SQLite 59](#_Toc356899922)

[5.1.2 Browser persistence 60](#_Toc356899923)

[5.1.2.1 WebStorage 60](#_Toc356899924)

[5.1.2.2 WebSQL & IndexedDB 60](#_Toc356899925)

[5.1.2.3 Application cache 62](#_Toc356899926)

[5.1.2.4 De bootstrap startup procedure 65](#_Toc356899927)

[5.2 Ontwikkelingsomgeving en tools 69](#_Toc356899928)

[5.2.1 Notepad++ 69](#_Toc356899929)

[5.2.2 Eclipse Indigo 69](#_Toc356899930)

[5.2.3 Visual Paradigm for UML 69](#_Toc356899931)

[5.2.4 Uml Factory 69](#_Toc356899932)

[5.2.5 Oracle VM Virtualbox 70](#_Toc356899933)

[5.2.6 Github 70](#_Toc356899934)

[5.2.7 Excel 70](#_Toc356899935)

[5.2.8 Adobe Photoshop 70](#_Toc356899936)

[5.2.9 Powerpoint Storyboarding 71](#_Toc356899937)

[5.2.10 Putty 71](#_Toc356899938)

[5.2.11 Nano 71](#_Toc356899939)

[5.2.12 ADB & ADB wireless 71](#_Toc356899940)

[5.2.13 Google Chrome developer tool 72](#_Toc356899941)

[5.3 Gebruikte technologieën 73](#_Toc356899942)

[5.3.1 Java 73](#_Toc356899943)

[5.3.2 Android 73](#_Toc356899944)

[5.3.3 SQLite 73](#_Toc356899945)

[5.3.4 WebSQL & IndexedDB 73](#_Toc356899946)

[5.3.5 Javascript 74](#_Toc356899947)

[5.3.5.1 De javascript jungle & the difference of the language 74](#_Toc356899948)

[5.3.5.1 Google feed api 74](#_Toc356899949)

[5.3.6 Linux Ubuntu 12.04.1 Server 74](#_Toc356899950)

[5.3.7 Apache HTTP Server Version 2.2 75](#_Toc356899951)

[5.3.8 Vsftpd FTP Server 3.0.2 75](#_Toc356899952)

[5.4 Gebruikte patterns 75](#_Toc356899953)

[5.4.1 Model View Controller (MVC) (Native android) 75](#_Toc356899954)

[5.4.2 Data Access Object (DAO) 76](#_Toc356899955)

[5.5 Eigen gemaakte Listeners (Native android) 76](#_Toc356899956)

[5.5.1 ArticleListener 76](#_Toc356899957)

[5.5.2 ImageListener 77](#_Toc356899958)

[5.5.3 Swipe gesture 77](#_Toc356899959)

[5.6 Eigen gemaakte scrollers (Native android) 77](#_Toc356899960)

[5.7 Dynamische layout (Native android) 78](#_Toc356899961)

[5.8 Database in Native android 78](#_Toc356899962)

[5.8.1 Update database 79](#_Toc356899963)

[5.8.2 Uitlezen van de feeds 79](#_Toc356899964)

[5.9 Project structuur 80](#_Toc356899965)

[5.9.1 BackgroundActivity 82](#_Toc356899966)

[5.9.2 Statechart diagram activities 82](#_Toc356899967)

[6 Testen 84](#_Toc356899968)

[6.1 Native Android app 84](#_Toc356899969)

[6.2 Webcomponent 84](#_Toc356899970)

[7 Conclusie 85](#_Toc356899971)

[8 Appendix 88](#_Toc356899972)

[8.1 Work breakdown structure 88](#_Toc356899973)

[8.2 Gantt chart 89](#_Toc356899974)

[8.3 Verslagen meetings 89](#_Toc356899975)

Dankwoord

Dit project is niet alleen tot stand gekomen door ons, maar we werden ondersteund door een klein team van behulpzame mensen.

Graag willen we enkele personen in het bijzonder bedanken:

Gilles Van Mol, bedrijfspromoter, gaf ons de kans om te proeven van mobile development. Hij heeft meegeholpen met zijn technische advies en kennis ter zake. Johan Geybels, Ace GroepT promotor, begeleidde ons in dit project. Wij waarderen zijn advies op het vlak van planning en zijn controle op het ontwikkelingsproces.

Ten slotte willen we ook onze familie en vrienden bedanken die het tijdens dit academiejaar meerdere malen zonder ons hebben moeten stellen.

# Inleiding

## Algemeen

In opdracht van de firma Twipe Mobile hebben we een android component ontwikkeld dat RSS feeds leest en deze als snelnieuws in een readermodule beschikbaar stelt, voor de krant Het Nieuwsblad.

De nadruk ligt hierbij op een androidmodule en een crossbrowser html5 versie ervan die als webcomponent makkelijk integreerbaar is in andere mobiele platformen zoals iOS.

Mobile development is de dag van vandaag één van de belangrijkste it-trends waarbij vooral mobiele webapplicaties populair zijn. Dankzij hun gemakkelijke integratie op verschillende platformen zullen zij vermoedelijk een belangrijke positie in de mobiele wereld opeisen.

Het spreekt uiteraard voor zich dat we deze trein niet willen missen en dat we met dit project voor een enorme uitdaging staan. Buiten het aanwenden van onze verworven kennis in de afgelopen jaren, krijgen we de kans om in de mobiele wereld te duiken en hierdoor nieuwe architecturen en technologieën te ontdekken.

## Opdrachtgever

Opdrachtgever van dit afstudeerproject is Twipe Mobile Solutions NV. Het bedrijf is gevestigd in Heverlee en is gespecialiseerd in het publiceren van kranten en magazines in digitaal formaat. Gilles Van Mol is de copromotor die ons bijstaat, één van de drie project managers binnen Twipe.

De voornaamste producten van Twipe Mobile NV. zijn het aanbieden van replica’s van dagdagelijkse kranten, zowel in pdf-formaat als in een daarvoor speciaal tablet/smartphone optimized newspaper reader. Daarbuiten verzorgen zij nog tal van andere mobile news applications.

Een kleine greep uit hun cliënteel: De Standaard, Het Nieuwsblad, De Gazet van Antwerpen, Het Belang van Limburg, Humo...

Organigram van Twipe Mobile Solutions NV

## Projectdefinitie

De mobiele applicaties schieten als paddenstoelen uit de grond. Zo is ook het Nieuwsblad geïnteresseerd in een mobiele elektronische krant. Zij hebben deze opdracht in handen gegeven van Twipe Mobile Solutions N.V. in Heverlee. Wij zullen hen hierbij helpen.

Met deze nieuwe technologische toepassing probeert het Nieuwsblad de bestaande lezers en nieuwe lezers aan zich te binden. De applicatie toont de verschillende krantenartikels, net zoals een lezer een papieren krant kan lezen en doorbladeren. Maar je kan ook meer met een elektronische krant. Wat zou je er als lezer van vinden om je eigen krant te maken? In de schoenen te staan van de redacteur? Om zelf te beslissen over de opmaak, de kleur, de inhoud, de looks van je krant? Wel, daar willen wij wat aan doen.

Twipe Mobile Solutions N.V. heeft ons een aantal duidelijke opdrachten meegegeven. Ze willen een applicatie geschreven voor het Android besturingssysteem. Maar wat dan voor de smartphones en tablets met andere besturingssystemen? Daar gaan we een webcomponent voor maken zodat ook hun eigenaars kunnen genieten van hun krant.

# Scope – Probleem analyse – Business analyse

## Scope

Elk project heeft zijn grenzen waarin een oplossing en de uitwerking ervan verwezenlijkt moet worden. Deze grenzen dienen in een eerste fase grondig afgebakend te worden. In dit hoofdstuk bekijken we de scope van naderbij en doen dit aan de hand van een vooronderzoek, een probleem analyse en een business analyse. We belichten hierbij verschillende technieken.

### High level event – response list

Uit de eerste gesprekken met copromoter Gilles is snel duidelijk gebleken wat de belangrijkste vereisten zijn in dit project. We hebben ze in eerste plaats langs de kant van de gebruikers bekeken en in een high-level event–response list gegoten.

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | Event | Trigger | Response |
| Gebruiker | Lees een artikel | Klik op de overzichtpagina | Artikeldetail |
| Gebruiker | Zoek een artikel | Trefwoord | Artikellijst |
| Gebruiker | Configureer de reader | Het aanklikken van de configuratiebutton | Configuratiescherm |
| Gebruiker | Kies een kleurenthema | Kleurkeuze | Kleuraanpassing |
| Gebruiker | Kies een font thema | Fontkeuze | Font thema aanpassing |
| Gebruiker | Kies een layout | LayoutKeuze | Layout aanpassing |
| Gebruiker | Kies een blader-effect | Effectkeuze | Effect aanpassing |
| Gebruiker | Selecteer feeds | Keuze van feeds | Aanpassing van de feeds |

Omdat één van de grote vereisten het offline aanbieden is van feeds en artikels, is in een later stadium echter gebleken dat ook grote delen van het systeem zich ook als een actor gedragen. Ze worden voornamelijk veroorzaakt door interne en externe events. We hebben ze eveneens in een event – response list verzameld.

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | Event | Trigger | Response |
| Webcomponent | Start de bootstrap methode (intern event) | Applicatie opstart | Bootstrap opstart |
| Webcomponent/native | Bouw de database tabellen (intern event) | Allereerste applicatie opstart | Eerste opstart |
| Nieuwsblad feed | Download de feeds (extern event) | Nieuwe artikels | Download van artikels |
| Nieuwsblad feed | Parse de feeds (extern event) | Nieuwe artikels | Parsen van de feeds |
| Nieuwsblad feed | Bewaar de feeds (extern event) | Nieuwe artikels | Persisten van de feeds |
| Webcomponent/native | Verwijder oude artikels (intern event) | Time | Opkuis |

### High level context diagramma

Het Data Flow Diagram (DFD), is een grafische representatie van de gegevensstroom, de dataflow, door een informatiesysteem. Het is een veel gebruikte techniek binnen de gestructureerde analyse. We gebruiken het om een beter inzicht te krijgen in de in- en outputs van het systeem, opnieuw vanuit een high-level standpunt.



### High level functional decomposition diagram

Aan de hand van een high level functional decomposition diagramma breken we het systeem open en definiëren we de belangrijkste processen met hun subsystemen. Op deze manier krijgen we een overzicht van het systeem en welke functionaliteiten er uiteindelijk ontwikkeld moeten worden. Uit deze processen zullen de use cases en storyboards uit hoofdstuk vier vloeien.



### Scope volgens de MoSCoW-methode

Het vastleggen van de scope volgens de Moscow-methode gebeurt door het stellen van prioriteiten aan de verschillende requirements, problemen en eisen. Het een populaire methode die gebruikt wordt bij rapid application development en agile projecten. Het is een afkorting waarbij de letters voor het volgende staan. De kleine o’s hebben geen betekenis maar zijn een handig geheugensteuntje.

**M : must haves**

Deze requirements moeten ontwikkeld worden en in het eindprodukt voorkomen.

**S : should haves**

Deze eisen zijn gewenst, maar zonder is het product zeker bruikbaar.

**C : could haves**

Deze eisen komen enkel in aanmerking om ontwikkeld te worden wanneer er tijd over schiet.

**W : won’t haves**

Deze requirements komen niet aan bod in het project, maar zijn misschien interessant om in een opvolgingsproject opgenomen te worden.

In onderstaande lijst hebben we requirements, opportuniteiten en probleemstellingen nogmaals verzameld en een prioriteit gegeven op basis van het MoSCoW principe. Dit is uiteindelijk ook de manier geworden waarop we de scope hebben vastgelegd.

**Must:** uitlezen van XML file van het Nieuwsblad

**Must:** De reader presenteert de artikels in een optimale vorm (Verschillende templates, op basis van de artikels wordt een passende template gekozen)

**Must:** De reader is configureerbaar (themes, fonts, blader effecten, kleur ...) Themes(fonts, kleur, ...) Bladereffect en (keuze uit verschillende animaties) Verschillende, meerdere RSS feed(s)

**Must:** De reader is adaptive aan de verschillende schermresoluties

**Must:** De reader module en download module kunnen als blackbox makkelijk geïntegreerd worden in andere apps

**Must:** Aanmaken van een 5-tal verschillende templates. Dit is wegens het vele grafische design werk achteraf teruggeschroefd tot responsive design.

**Must:** De reader kan elk soort 2.0 Rss-feed inlezen

**Must:** De artikels moeten offline leesbaar zijn

**Wouldn't:** De reader kan nog andere populaire vormen van feeds inlezen en bijhouden (Atom, custom made xml ...)

**Must:** Zoekfunctie naar artikels op trefwoord in titel en/of artikeltekst

**Should:** Uitgebreide zoekfunctie met verschillende parameters (feed, datum, auteur, trefwoord ...)

**Could:** Zoekfunctie met autocomplete

**Should:** De reader aanpassen adhv de historiek van de gebruiker. vb. gebruiker leest veel sport, dan wordt er een soort van "sport"reader gemaakt en daarnaast de gewone reader. -> content driven

**Should:** Meerdere users per reader met historiek (feeds, theme, effect, leesgeschiedenis ...)

**Should:** Per gebruiker [min-max] aantal feeds, themes, effecten

**Could:** Per gebruiker, per feed [min- max] bijhouden artikels (tijdsduur, aantal) + instelbaar

**Could:** Geolocatie gebruiken voor een weerberichtmodule, georeclame, regionieuws

Voor de webcomponent dachten we nog een gebruikers community te maken. Dit lijkt ons een interessante toevoeging

**Won’t:** Inschrijven, uitschrijven, herinschrijven, password recovery

**Won’t:** Per artikel (commentaren/reacties, social plugin buttons, polls, analyse, ratings, recensies, mail dit artikel

**Won’t:** Polls

**Won’t:** Reclame

**Won’t:** User driven artikels aan de hand van leesgeschiedenis

**Won’t:** Blok => Meest gelezen, hoogste rating, recenste ...

## Probleem analyse & business analyse

### Problem statement table

Met deze techniek bekijken we de requirements, problemen en opportuniteiten opnieuw vanuit een high-level niveau. We vertrekken van een korte omschrijving per probleem, opportuniteit of requirement en geven er een dringendheid en ranking aan mee.

|  |  |  |  |
| --- | --- | --- | --- |
| Brief statements of problem, opportunity or directive | Urgency | Priority | Proposed solution |
| 1. De reader moet rss-feeds van het Nieuwsblad kunnen uitlezen | High | 1 | Xml-parser |
| 1. De reader moet elk soort rss-feed kunnen uitlezen | Med - low | 3 | Xml-parser |
| 1. De reader presenteert de artikels in optimale vorm | High | 1 | Responsive design |
| 1. De reader moet configureerbaar zijn | High | 1 | Development |
| 1. De artikels moeten offline gelezen kunnen worden | High | 1 | WebSQL & SQLite database |
| 1. Een zoekfunctie op basis van een trefwoord | Low | 3 | Development |
| 1. De reader kan als blackbox element geïntegreerd worden in andere android applicaties. | Med | 2 | Development |
| 1. Een gebruiker kan kiezen tussen verschillende font-thema’s | High - med | 2 | Development |
| 1. Een gebruiker kan kiezen tussen verschillende kleur layouts | High - med | 2 | Development |
| 1. Een gebruiker kan tussen verschillende blader-effecten kiezen. | High - med | 2 | Development |
| 1. Een gebruiker kan een layout kiezen | High - med | 2 | Development |

### Cause & effect analyse voor de webcomponent

Een andere techniek die bij een probleemanalyse gebruikt wordt is het opstellen van een causes & effects tabel. Hierbij worden ook onmiddellijk de doelstellingen van het nieuwe systeem geschetst, samen met de beperkingen die gepaard gaan met de gestelde oplossing. In ons geval legt het problemen van technische aard bloot. Deze tabel is dan ook vooral tijdens het ontwikkelen en het testen gegroeid, en brengt heel wat nieuwe technische termen met zich mee. We lichten deze voorlopig niet verder toe. Dit doen we uitgebreid in hoofdstuk vijf.

|  |  |  |  |
| --- | --- | --- | --- |
| Cause & effect analyse | | System improvement objectives | |
| Probleem of opportuniteit | Causes & effects | System objective | System constraint |
| Uitlezen van xml-feeds | * Cross domain javascript probleem. Javascript kan uit veiligheidsoverwegingen geen files uitlezen die afkomstig zijn van een ander domein. De rss-feeds draaien op een nieuwsblad webserver. Het parsen van deze xml-feeds zou alleen werken als onze javascripts op dezelfde server zou draaien. | Het systeem moet in staat zijn de externe feeds te kunnen parsen en verwerken. |  |
| Elk soort feed uitlezen | * Het grote probleem is dat een rss-feed enorm veel optionele elementen kan bevatten. Hierbij weten we op voorhand niet wat we zullen parsen. * Veel publishers lappen de officiële rss 2.0 feed aan hun laars. * Soms zijn velden leeg, soms niet, wat bij het aanmaken van een initiëel database model problemen kan opleveren. |  |  |
| Het offline lezen van artikels | * Alles moet clientside worden bijgehouden. * Application cache beperkingen * Asynchrone WebSQl database * WebSQL beperkingen in browsers zoals de pragma queries, database updates. * WebSQL is deprecated en wordt niet langer ondersteund door het W3C. | Het systeem moet artikels offline kunnen aanbieden. |  |
| Het presenteren van de artikels in optimale vorm | * Er is een enorme variëteit in toestellen en resoluties. | Het systeem moet in staat zijn de artikels in de juiste vorm te presenteren. Responsive design. |  |
| De reader moet kunnen fungeren als blackbox | * De applicatie moet als blackbox in andere androidtoepassingen ingebouwd kunnen worden. * Application cache probleem * Het Mime-type probleem van de application cache. |  |  |
| Snelheid van het presenteren van de artikels | * WebSQL is soms trager dan het netwerk * Vaak zijn de enclosure, de media objecten die bij een artikel horen enorm groot. * De asynchrone WebSQL vereist callback queries. |  |  |

## Uiteindelijke werkwijze

Op aanraden van promotor Johan hebben we gekozen om dit project agile-matig uit te werken. In tegenstelling tot de klassieke waterval methode splitsen we het project op in kleinere delen en werken deze iteratief uit in periodes van een viertal weken. Hierbij worden de stappen in het ontwikkelingsproces per iteratie herhaald.

Op deze manier kunnen we in korte periodes kleine deelstukken analyseren, ontwikkelen en testen, en maandelijks kleine proof of concepts (POC) opleveren. Het voordeel van deze POC’s is dat er op regelmatige tijdstippen kleine resultaten geboekt worden waardoor er steeds een gedeelte van een gevraagde functionaliteit opgeleverd kan worden.

Gezien de frameworks, java for android en html5-javascript, waarop de applicatie moet draaien voor ons beiden compleet nieuw waren, is achteraf gebleken dat het opleveren van deze POC’s niet altijd even evident was. Deze frameworks en het gebruik van onbekende technologieën hebben ons niet alleen verplicht onszelf nieuwe vaardigheden aan te leren door zelfstudie, ze hebben tevens voor het nodige opzoekingswerk gezorgd en verschillende pitfalls bloot gelegd. Achteraf, en tijdens onze queeste is duidelijk gebleken waarom een dergelijke project nog niet bestond.

Aan de hand van de verschillende technieken de we gebruikt hebben in dit hoofdstuk voor ons vooronderzoek, de probleem analyse en de business analyse hebben we besloten om de scope vast te leggen aan de hand van de Moscow-methode en deze uit werken in een iteratief proces. Hierbij hebben we ons hoofdzakelijk gefocust op de must haves.

# Analyse en ontwerp database

## Entity – Relationship Diagram

Een entity-relationship model of kortweg ERD genoemd, is een abstracte beschrijving van een database. Het is een grafische voorstelling van een database die bestaat uit verschillende tabellen met bijhorende relaties en beperkingen. We gebruiken dit model om de uiteindelijke fysische database te bouwen.

Een kleine toelichting over belangrijke termen die hierbij gehanteerd worden.

***Entiteit****:* een entiteit kan worden gezien als een abstract "iets" dat deel uitmaakt van het datamodel. Een entiteit heeft een eenduidige naam met een duidelijke kernachtige omschrijving. Voorbeelden hiervan zijn: klant, product, bestelling... Entiteiten worden in het fysieke database model uiteindelijk tabellen.

***Attribuut***: attributen beschrijven eigenschappen van een entiteit. Een klant heeft een naam, een adres, een telefoonnummer... Attributen worden kolommen in een tabel.

***Relatie***: een relatie beschrijft het verband tussen verschillende entiteiten met de nodige kardinaliteit. Er zijn drie soorten van kardinaliteit. Een één-op-één relatie, een één-op-veel-relatie en een veel-op-veel-relatie.

***Sleutel***: een sleutel is een attribuut, of een combinatie van meerdere attributen van een entiteit die een unieke waarde representeert. Sleutels zorgen ervoor dat we verschillende entiteiten van een zelfde soort van elkaar kunnen onderscheiden en indentificeren.

## Logisch ERD model

Een logisch ERD model bevat enkel de verschillende entiteiten en de onderlinge relaties. We hebben in totaal twee versies van het logische ERD gemaakt. De eerste versie is gebaseerd op het officiële datamodel van een RSS 2.0 Feed. Het tweede model beschrijft de processen die we uit de use cases en de storyboards hebben gehaald. De many-to-many relaties zijn hierbij nog niet uitgewerkt tot associatietabellen.

### RSS 2.0 Standaard voor een rss-feed[[1]](#footnote-1)

De nieuwsberichten die we te verwerken krijgen zijn van het type rss, wat staat voor ‘*Really Simple Syndication’.* Dit zijn xml-bestanden die volgens een bepaalde standaard zijn opgebouwd en die hoofdzakelijk gebruikt worden om updates van nieuwe content naar buiten toe te publiceren. Sinds 2002 is de standaard rss 2.0. Deze bestaat uit twee grote elementen, een channel en een artikel. Artikels horen steeds bij een channel.





### Eerste Logisch ERD model

In een eerste ERD model hebben we de officiële standaard voor een rss-feed uit elkaar getrokken en hieruit het volgende diagramma afgeleid.



### Tweede Logisch ERD model

In een tweede model hebben we de must haves onder de loep genomen. Hier vinden we voornamelijk de processen uit de use cases en storyboards terug.



We hebben beide logische ERD modellen samengevoegd en hieruit de volgende fysieke database geconstrueerd. Hierbij zijn de many-to-many relaties wel uitgewerkt tot associatietabellen.

## Fysiek database model van een nieuwsblad rss-feed



## Database definitie

Zowel de native app als de webcomponent gebruiken een SQLite database. SQLite ondersteunt slechts een beperkt aantal datatypes. We sommen ze daarom even op.

***Null:*** het datatype heeft een null-waarde

***Integer:*** een geheel getal

***Real:*** floating point waarde

***Text:*** representatie van een string

***Blob:*** Binary Large Object, dit is een - potentieel groot – gegevenselement

SQLite kent geen ***boolean*** type. Boolean waarden worden daarom opgeslagen als de integers 0 (false) en 1 (true).

SQLite kent evenmin ***Date*** of ***Time*** datatypes. Er zijn drie manieren om een Date of Time datatype voor te stellen.

***Text*** as ISO8601 strings ("YYYY-MM-DD HH:MM:SS.SSS").

***Real*** as Julian day numbers, the number of days since noon in Greenwich on November 24, 4714 B.C. according to the proleptic Gregorian calendar.

***Integer*** as Unix Time, the number of seconds since 1970-01-01 00:00:00 UTC.

### Channel

Een channel representeert een rss-feed die nul tot veel artikels kan bevatten.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary Key |
| title | Text | Not null |  | Naam van de channel. |
| description | Text | Not null |  | Omschrijving van de channel. |
| selected | Integer | Not null | 1 | Boolean object om een channel al dan niet te tonen aan de gebruiker. |
| ttl | Integer |  |  | Time to live. Een nummer in minuten om aan te geven hoelang een channel in cache wordt bijgehouden vooraleer te refreshen. |
| language | Text |  |  | Taal waarin de feed is geschreven. |
| pubDate | Text |  |  | Date object. Moment van publicatie, in RFC 822[[2]](#footnote-2) formaat. |
| lastBuildDate | Text |  |  | Date object. De laatste keer dat content is veranderd geweest. |

### Article

Artikels zijn de nieuwsberichten die door de gebruiker gelezen kunnen worden. Volgens de officiële rss 2.0 standaard zijn alle elementen optioneel, er moet alleen een titel *of* een link zijn. Omdat deze regelgeving ietswat ambigu is hebben we beide velden mandatory[[3]](#footnote-3) gemaakt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary key |
| title | Text | Not null |  | Titel van het artikel. |
| link | Text | Not null |  | Link naar de online versie van het artikel. |
| description | Text |  |  | Artikelinhoud. |
| counter | Integer |  |  | Teller die we bijhouden om populaire artikels en channels te identificeren. |
| guid | Text |  |  | Link naar het artikel. |
| pubDate | Text |  |  | Date object. Moment van publicatie, in RFC 822 formaat. |
| channelID | Integer |  |  | Foreign key die verwijst naar de feed waarbij het artikel hoort. |

### Source

De naam van de channel, afgeleid van de titel. De bedoeling van dit element is credits voor links te voorzien, voornamelijk om de bron te vermelden waarbij het artikel hoort. Dit is meestal de naam van de channel. Elk element is mandatory.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary key |
| url | Text | Not null |  | Url van de feed waarbij het artikel hoort. |
| title | Text | Not null |  | Titel van de feed waarbij het artikel hoort. |
| articleID | Integer | Not null |  | Foreign key die verwijst naar het nodige artikel. |

### Enclosure

Een media element dat bij een artikel hoort. Dit kan elk soort standaard MIME-type[[4]](#footnote-4) zijn. Doorgaans zijn dit images die bij een artikel horen. Alle velden zijn verplicht.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary key |
| url | Text | Not null |  | Url naar het bestand. |
| length | Text | Not null |  | Grootte van het bestand in bytes. |
| type | Text | Not null |  | Het mime-type. |
| articleID | Integer | Not null |  | Foreign key naar het artikel waarbij het media element hoort. |

### GlobalUniqueIdentifier

GlobalUniqueIdentifier ofwel kortweg guid genoemd. Vaak is dit een string in de vorm van een url dat naar een artikel verwijst.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary key |
| guid | Text | Not null |  | Link naar een artikel. |
| isPermalink | Integer |  | 0 | Boolean object. Indien true is de guid waarde een effectieve link naar het artikel. |
| articleID | Integer | Not null |  | Foreign key naar het artikel waarbij de guid hoort. |

### Cloud

Dit element specifieert een webservice waarbij processen zich kunnen inschrijven om op de hoogte te worden gehouden van updates.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary key |
| domain | Text |  |  | Url van de webservice. |
| port | Integer |  |  | De port waarbij de webservice hoort. |
| path | Text |  |  | Folder waarin de webservice draait. |
| registerProcedure | Text |  |  | Procedure die opgeroepen wordt. |
| protocol | Text |  |  | Protocol. |
| channelID | Integer | Not null |  | Foreign key naar channel |

### Image

Dit is een beeld bestand van het type gif, jpeg of png dat bij een feed hoort. Het representeert meestal het logo van de feed of channel.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary key |
| url | Text | Not null |  | Url naar het bestand |
| title | Text | Not null |  | Beschrijft het bestand, in html is dit het ALT-attribuut dat bij een <img> tag hoort. |
| link | Text | Not null |  | Link naar de channel waarbij het bestand hoort. |
| description | Text |  |  | Bevat een tekstuele beschrijving van het bestand. Dit komt vaak overeen met het title attribuut. |
| width | Integer |  | 88 | Breedte van het bestand in pixels. De maximum waarde is 144. |
| height | Integer |  | 31 | Hoogte van het bestand in pixels. De maximum waarde is 400. |
| channelID | Integer |  |  | Foreign key waarbij het bestand hoort. |

### TextInput

Een rss-feed kan optioneel een textInput veld bevatten waarin feedback kan gegeven worden. De bedoeling van dit element is eigenlijk een misterie. Het wordt daarom ook zelden tot nooit gebruikt. Alle velden zijn verplicht.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary key |
| title | Text | Not null |  | Label van de submit-button. |
| description | Text | Not null |  | Omschrijving van het input veld. |
| name | Text | Not null |  | Name van het tekst object in het inputveld. |
| link | Text | Not null |  | Url naar het script dat input verwerkt. |
| channelID | Integer | Not null |  | Foreign key naar channel. |

### HoursToBeSkipped

Een xml-element dat de uren beschrijft die kunnen geskipt worden. Indien het element bepaalde uren bevat, wordt de channel op die uren ook niet uitgelezen. Het is een getal tussen 0 en 23.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary key |
| value | Integer | Not null |  | [0-23] |

### DaysToBeSkipped

Een xml-element dat zeven mogelijke <day> tags bevat. De mogelijke waarden voor de tags zijn Monday, Tuesday, Wednesday, Thursday, Friday, Saturday or Sunday. De feed wordt voor de dagen die in het element voorkomen, op die dagen niet uitgelezen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary key |
| value | Text | Not null |  | [Monday-Sunday] |

### HoursToBeSkippedForChannel

Associatietabel tussen channel en hoursToBeSkipped.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| channelID | Integer | Not null |  | Foreign key |
| hoursToBeSkippedID | Integer | Not null |  | Foreign key |

### DaysToBeSkippedForChannel

Associatietabel tussen channel en daysToBeSkipped.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| channelID | Integer | Not null |  | Foreign key |
| daysToBeSkippedID | Integer | Not null |  | Foreign key |

### Category

Een categorie waarbij channels en artikels behoren. Zowel channels als artikels kunnen behoren tot meerdere categorieën.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary key |
| name | Text | Not null |  | Benaming van de categorie. |
| domain | Text |  |  | Url naar een beschrijving. |

### CategoriesPerChannel

Associatietabel tussen category en channel.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| categoryID | Integer | Not null |  | Foreign key |
| channelID | Integer | Not null |  | Foreign key |

### CategoriesPerArticle

Associatietabel tussen article en category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| categoryID | Integer | Not null |  | Foreign key |
| articleID | Integer | Not null |  | Foreign key |

### User

De gebruikersgegevens van users wanneer de applicatie meerdere users toelaat. Dit is een van de opportuniteiten die we op papier hebben uitgewerkt, maar niet hebben omgezet in broncode.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary key |
| name | Text |  |  | Echte naam van de user |
| username | Text |  |  | Gebruikersnaam |
| password | Text |  |  | Paswoord van de gebruiker. |
| preferenceID | Integer |  |  | Foreign key naar de instellingen die bij een gebruiker horen. |

### ChannelsPerUser

Associatietabel tussen channel en user. Default zijn dit de vijf feeds die meegegeven zijn door Twipe. Deze kunnen in de applicatie uit- of aangevinkt worden. Een gebruiker krijgt enkel de channels te zien die zijn aangevinkt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| channelID | Integer | Not null |  | Foreign key |
| userID | Integer | Not null |  | Foreign key |

### Preference

Dit zijn de instellingen die bij een gebruiker horen. Het bevat vijf foreign keys die naar de verschillende instellingen en thema’s verwijzen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary key |
| layoutID | Integer | Not null |  | Foreign key |
| fontID | Integer | Not null |  | Foreign key |
| effectID | Integer | Not null |  | Foreign key |
| colorID | Integer | Not null |  | Foreign key |
| historyID | Integer | Not null |  | Foreign key |

### Layout

De layout van een view, scherm. Dit is in een later stadium gereduceerd naar responsive design. De schermen passen zich automatisch aan aan de grootte van de display van het apparaat dat de artikels toont.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary key |
| Name |  |  |  | Naam van de layout. |
| description | Text |  |  | Verwijzing naar het bestand, view, layout die gebruikt moet worden. |

### Font

Het font dat gebruikt wordt om de artikels te tonen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary key |
| fontName | Text | Not null |  | Naam van het font: |
| fontType | Text | Not null |  | [TrueType – openType] |

### Effect

Het blader effect tussen verschillende artikels.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary key |
| Name | Text |  |  | Naam van het effect: swipe-effect, booklet-effect... |
| Description | Text |  |  | Verwijzing naar het bestand dat het effect verwezenlijkt. |

### Color

De kleuren die gebruikt worden om de artikels weer te geven.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Primary key |
| colorCodeInHex | Text | Not null |  | Overbodig veld omdat we nu met kleurenthema’s werken in plaats van echte kleuren. |
| Name | Text | Not null |  | Naam van de kleur: dark, grey, white |

### History

De tabel die het formaat bijhoudt hoelang artikels bewaard moeten worden.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Veld | Type | Null | Default Value | Comments |
| id | Integer | Not null |  | Foreign key |
| ttl | Integer | Not null |  |  |
| timeFormat | Text |  |  | Minutes, hours, days, weeks... |

# Requirement analyse

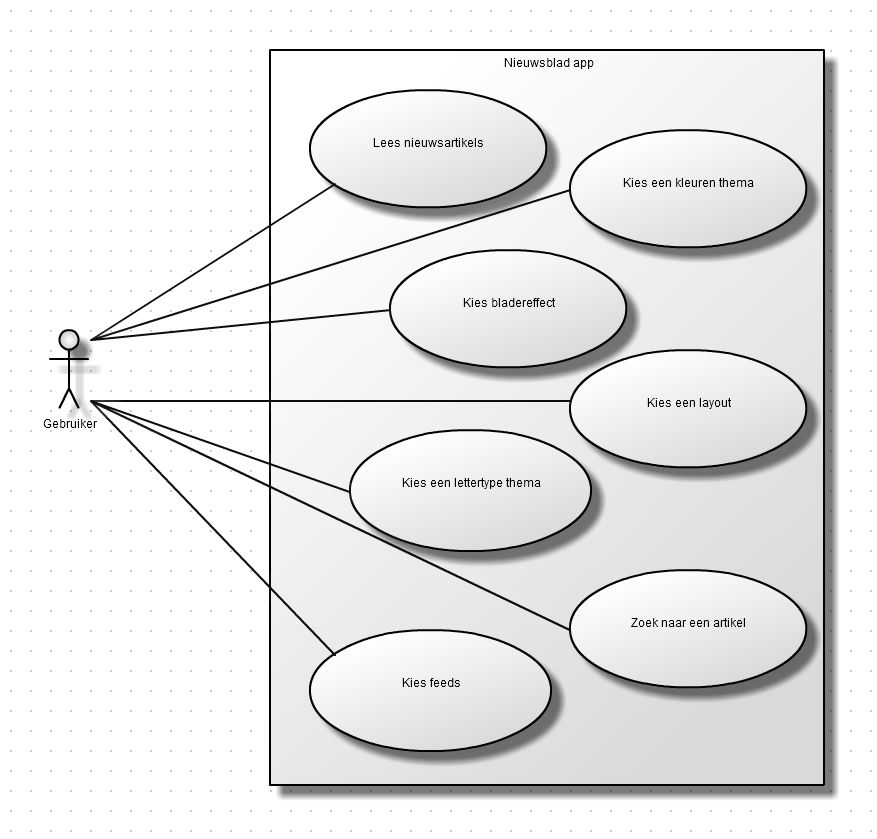
## Use cases en storyboards

De processen die we in hoofdstuk twee hebben geïdentificeerd aan de hand van oa het functional decomposition diagramma werken we nu verder uit in use cases en storyboards. Use cases laten ons toe om deze processen verder te verfijnen en uit te werken in de vorm van een grafisch en beschrijvend gedeelte.

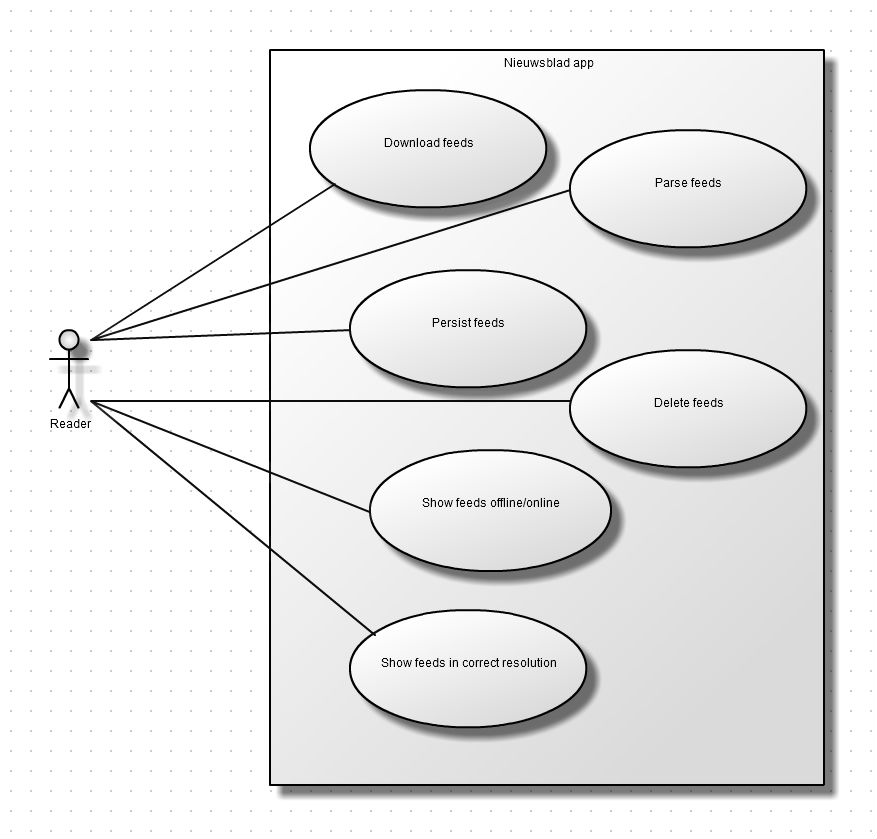
Van de meest eenvoudige use cases maken we storyboards. Deze storyboards zijn een grafische voorstelling van prototypes met een korte beschrijvende uitleg. Het grote voordeel van deze twee technieken is dat ook niet-ontwikkelaars de functionele requirements van het te bouwen systeem visueel gemakkelijk kunnen vatten en begrijpen. Gezien de agile-matige aanpak die we hanteren hebben we grotensdeels gebruik gemaakt van storyboards.

### High level use cases

*De gebruikers vereisten in een high level omschrijving*



*De reader requirements*



### StoryBoards

































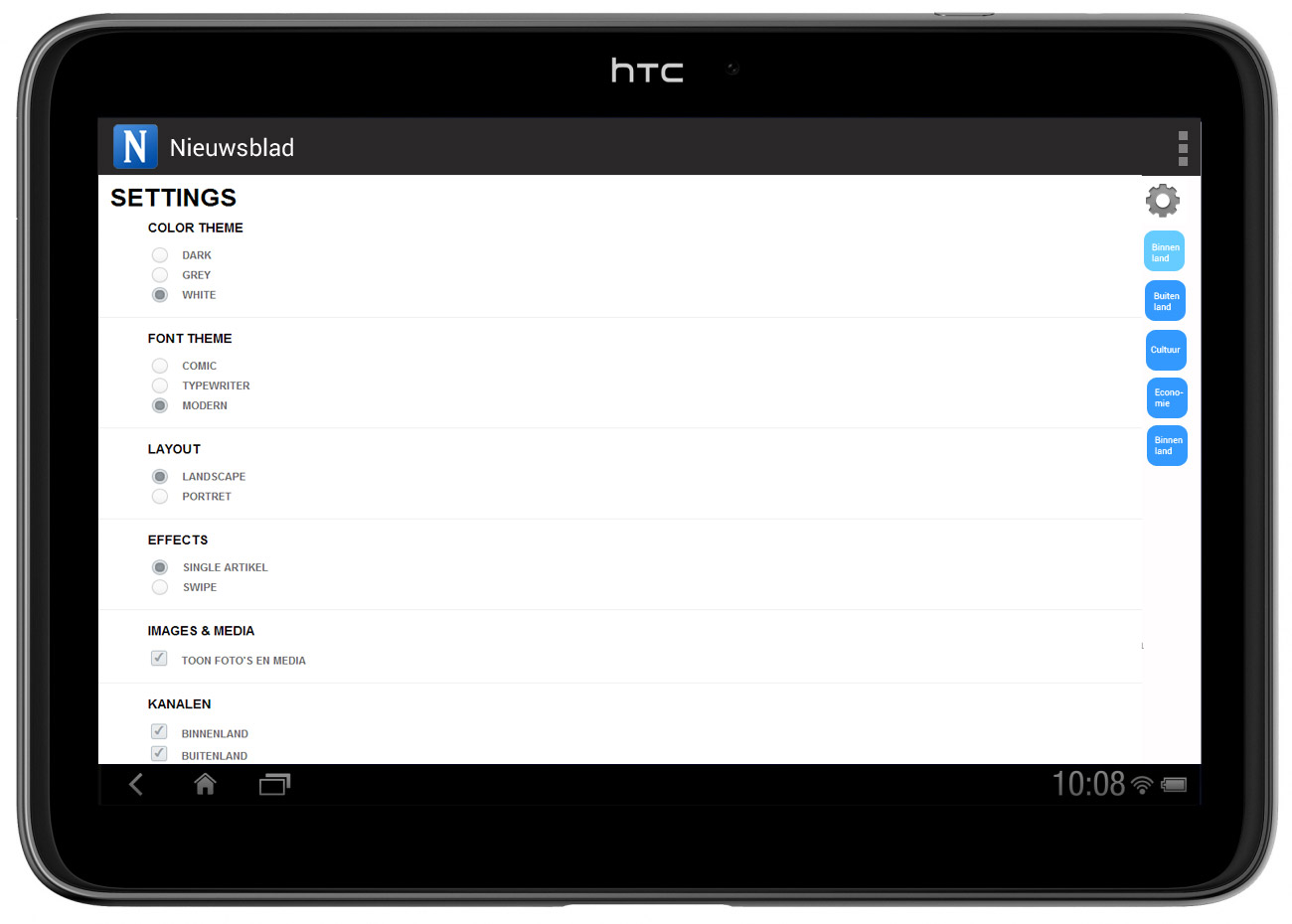




Enkele mockups voor tablets die al wat gedetailleerder zijn uitgewerkt.



Een overzicht van vijf channels met een aantal artikels die beknopt worden weergegeven.



Het configuratiescherm



Een gekozen kleuren thema (grey) zonder media elementen.

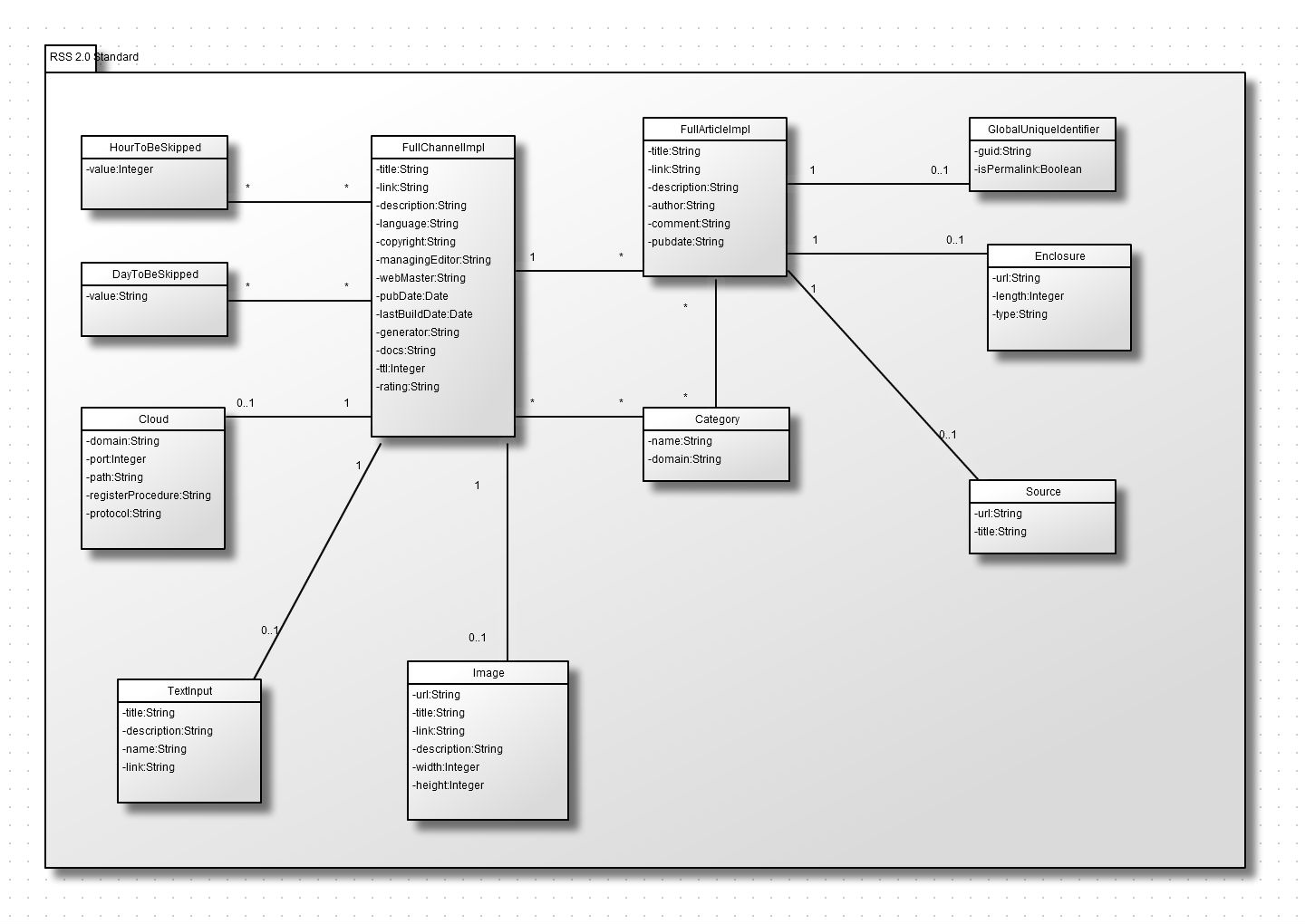


Een detailpagina van een artikel.

## Object georiënteerde analyse & ontwerp van de use cases

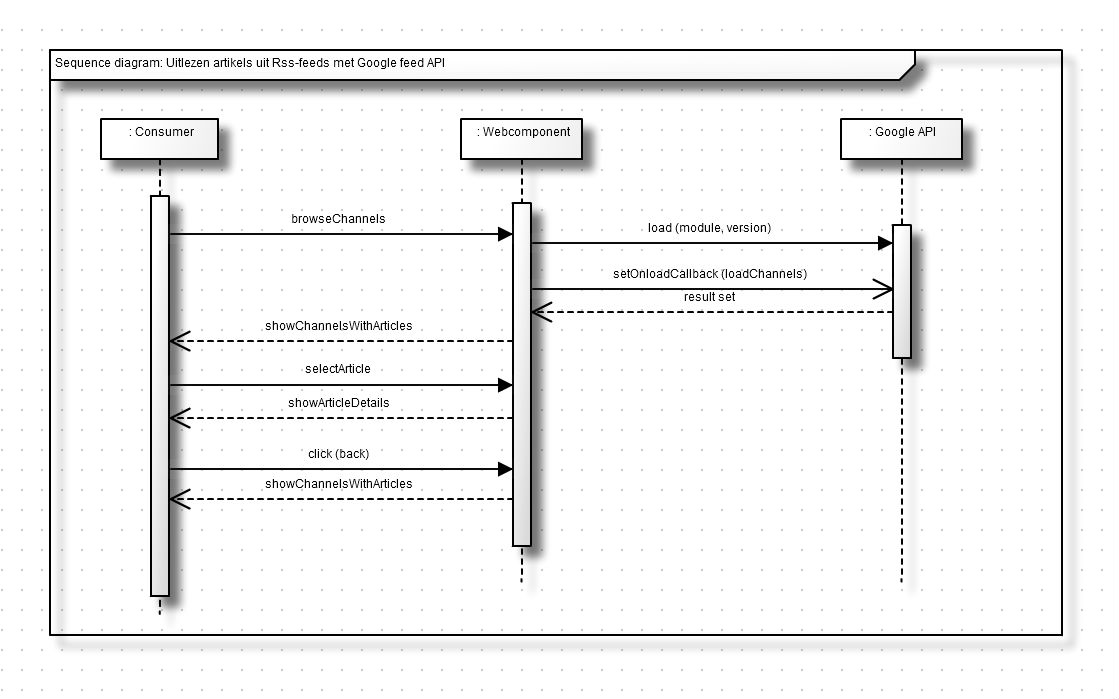
### Class diagramma

Uit de use cases, de storyboards en het erd-model uit de vorige hoofdstukken hebben we het bovenstaande class-diagram afgeleid. Dit diagramma geeft een overzicht van het domain class diagram. Alle one-to-many relations moeten hierbij nog geintegreerd worden als list<> objecten.



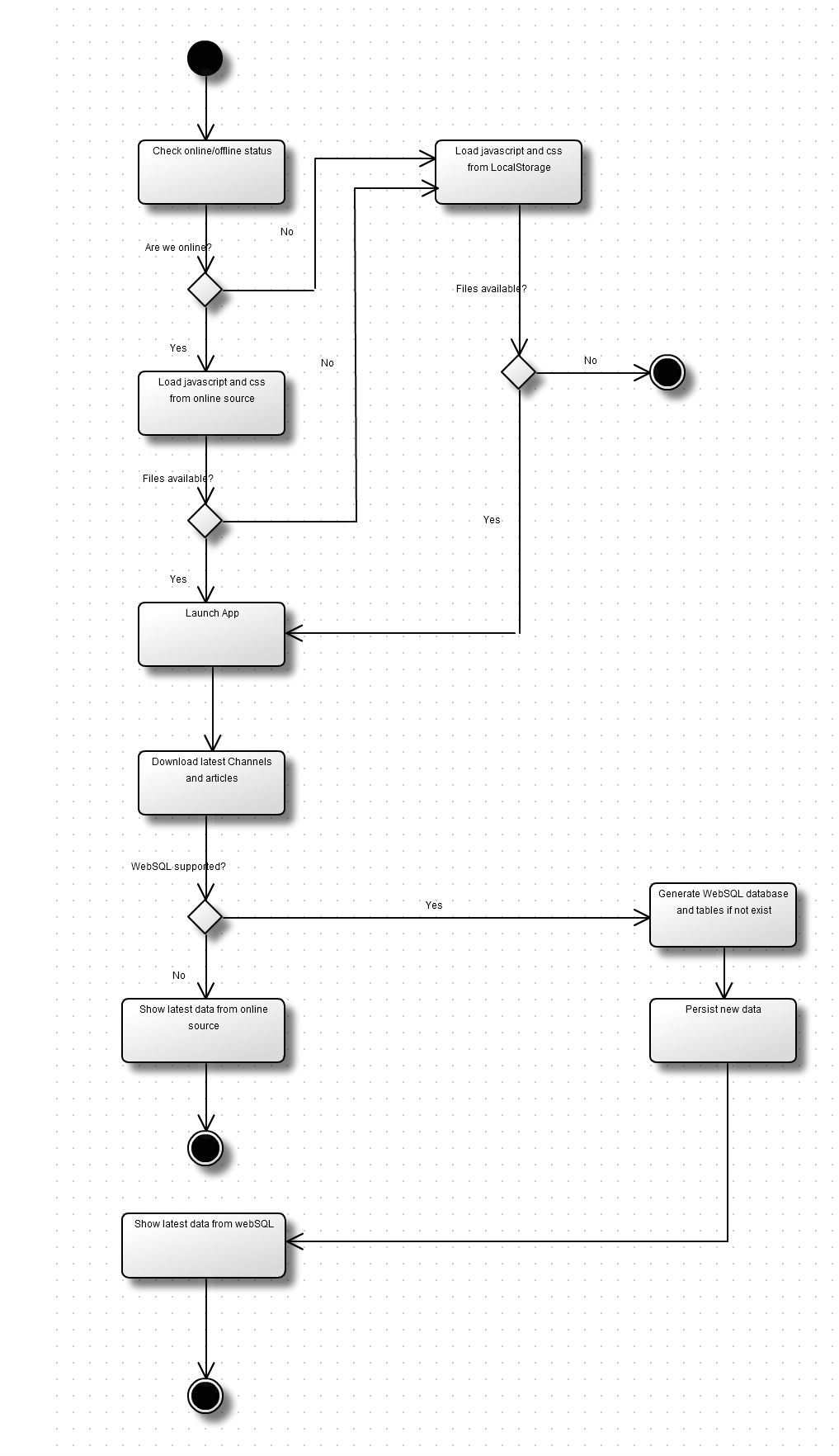
### Sequence diagramma’s

Onderstaand sequence diagramma toont het uitlezen van artikels.



### Activity diagramma’s

Onderstaand activity diagramma toont het opstarten van de bootstrap procedure. Deze procedure wordt in hoofdstuk vijf uitgebreid uitgelegd aan de hand van een volledige use case met de nodige beschrijvingen.



# Decision analyse en physical design

## The meaning of offline

Een van de grote requirements van de applicatie is dat alle artikels offline gelezen kunnen worden. Dit houdt in dat alle data client-side moet worden bijgehouden, zowel voor de native app als voor de webcomponent. Achteraf is dit een enorme queeste gebleken.

Wat houdt de term offline nu precies in? De termen web, internet en online liggen zeer kort bij elkaar en zijn voor veel mensen zelfs synoniemen. Om het web te gebruiken moeten we immers online zijn, waarom zouden we dan willen spreken over ‘*offline web’* technologieën?

Enerzijds kennen we de complete offline toepassingen, die data aanspreken op dragers als usb, harddisk, CD's, DVD's ... en anderzijds kennen we de toepassingen die data van het web, internet of netwerk halen. Wij moeten een applicatie hebben die beide functionaliteiten aankan, en bij voorkeur één die ons ERD-model uit hoofdstuk drie respecteert.

### Android en SQLite

Android is een linux-based besturingssysteem dat bestaat uit een aantal componenten en libraries. Een van die componenten is een lijst van libraries waaronder oa een SQLite database is in terug te vinden.

SQLite is een relationeel database management systeem en implementeert de meeste SQL standaards. SQLite wordt vooral gebruikt als embedded database langs clientside en is dus ook terug te vinden in het android besturingssyteem. Het is meer dan logisch dat we voor het native gedeelte deze oplossing hebben gebruikt.

### Browser persistence

Voor de webcomponent liggen de zaken anders. Om data client-side in een browser bij te houden hebben we een pak dieper moeten graven. Er zijn namelijk verschillende manieren om content offline te bewaren.

Verouderde technieken:

1. Cookies: Dit zijn zeer kleine bestanden die een minimum aan gegevens bevatten. Ze worden door websites gegenereerd en opgeslagen in een map of submap van een browser.
2. Plugin Based Storage: Flash macromedia, java, google gears … Ze werken niet overal en worden vaak uit veiligheidsoverwegingen op enterprise niveau geblokkeerd.
3. Browser-specific features: Deze zijn vendor specifiek en werken niet overal.

Nieuwere technieken: Webstorage, Application cache, WebSQL en IndexedDB

#### WebStorage

WebStorage vinden we terug in twee vormen, localStorage and sessionStorage. Het is één van de nieuwe features die html5 met zich meebrengt. WebStorage laat ons toe om data in een browser als key-value pairs op te slaan. LocalStorage slaat data permanent op, sessionStorage enkel tijdens een browser sessie.

Het is nu mogelijk om grotere hoeveelheden van data op te slaan, zonder dat dit de performantie en snelheid van een website in gevaar brengt. De hoeveelheid varieert van vendor tot vendor en gaat van 2.5mb tot 10mb. Alle moderne browsers ondersteunen ondertussen dit gebruik.

#### WebSQL & IndexedDB

WebSQL[[5]](#footnote-5) is een lightweight SQLite database die we terug vinden in de Google Chrome en Apple Safari browser. Dit zijn ook de browsers die respectievelijk gesupporteerd worden door Android en IOs, de twee belangrijkste besturingssystemen die we terug vinden in de wereld van tablets en smartphones.

IndexedDB is eveneens een API die gebruikt wordt voor client-side storage maar is geen echte database. Het maakt gebruik van key-values pairs in combinatie van een index en is terug te vinden in Firefox en Microsoft browsers. We sommen de voornaamste voor- en nadelen van beiden in onderstaande tabel op.

|  |  |  |  |
| --- | --- | --- | --- |
| WebSQL | | IndexedDB | |
| Voordelen | Nadelen | Voordelen | Nadelen |
| Gesupporteerd door Safari en Chrome, de twee browsers die we terugvinden op IOs en android, de belangrijkste besturingssystemen voor tablets en smartphones | Niet gesupporteerd door Firefox en Microsoft browsers | Gesupporteerd door Firefox en Microsoft browsers | Niet gesupporteerd door Safari en Chrome |
| Echte relationele SQL database | Het is een light SQLite versie, geen volledige SQL support. |  | Geen SQL, key-value pair met een index. Joins moeten bijvoorbeeld manueel gebeuren. |
| Queries in transacties met rollbacks |  |  |  |
|  | Deprecated door het W3C sinds oktober 2010. De specificaties worden niet meer ondersteund. |  | Lijkt maar niet van de grond te komen. |
| Asynchroon: Ideaal voor performantie | Asynchroon: Verplicht gebruik van callback functies wat ingewikkelde code veroorzaakt | Asynchroon: Ideaal voor performantie |  |
|  | Sommige pragma-queries werken uit veiligheidsoverwegingen niet. Zo kan je bvb geen database droppen, kan je er de versie niet van opvragen |  |  |
|  | Zeer weinig documentatie over terug te vinden. |  | Weinig informatie over terug te vinden. |
| Geen html5, werkt op sommige oudere browsers. |  |  |  |

De keuze om WebSQL te gebruiken is snel gemaakt, ook al zijn er vele nadelen aan verbonden. Het is echter de enige database die ondersteund wordt door Chrome en Safari. Achteraf zal bovendien ook nog blijken dat het asynchrone karakter ervan ons vaak in moeilijkheden heeft gebracht.

#### Application cache

Naast Webstorage introduceert html5 ook de Application cache, wat betekent dat eens een website is gecached, deze ook beschikbaar is zonder een internetconnectie.

Application cache biedt ons drie grote voordelen:

1. Offline browsing.
2. Speed. Cached resources laden sneller in.
3. Reduced server load.

Op A List Apart[[6]](#footnote-6) , een zeer toonaangevende website op gebied van webdesign & webdevelopment vonden we onderstaande quote terug die de eerste aanzet is geweest om het offline gedeelte voor de webcomponent met de application cache aan te pakken. Achteraf is meermaals gebleken waarom dit artikel zo belangrijk was.

I was at a party, one where the guests were mostly strangers to one another. I was part of a little huddle that was awkwardly trying to make introductions. A rather pretty lady turned to one of the shyer members of the group, introduced herself as “Dev,” and asked “So, what do you do then?”

“Oh, I’m the LocalStorage” he replied, shuffling uncomfortably. “I provide a scripting interface for text storage maintained across pages and browser sessions.”

“Yeah, he’s basically a shelf!” interrupted another. The group tittered at LocalStorage’s expense. I stayed silent, as I’d come to know this guy pretty well.

Another member of the group piped up. “Hi, I’m ApplicationCache,” he said, as he reached over to shake Dev’s hand. “I turn your offline experience from sucks-ass, to success. Just one extra file, and bosh! It works. No fuss, no ‘scripting’ necessary.” Yes, he did finger-quotes while saying ‘scripting.’ I’m gritting my teeth at this point, because I know he’s greatly exaggerating his abilities and the others don’t see it. However, if I call “bullshit” on it *I’ll* seem like the jerk.

I felt bad for not doing anything on that completely real evening. It’s painful to see articles that praise ApplicationCache’s ease of use, written by people who’ve clearly only met him in passing. I must set the record straight: I’m here to tell you ApplicationCache is a douchebag.

Now, I don’t mean he’s useless or should be avoided, you just have to be very careful when and how you work with him. If you get it wrong, the douchebaggery oozes through onto the end user. By reading through my own painful experiences with AppCache, you’ll know what to expect from AppCache and how to deal with it.

Application cache is vereist om websites offline weer te kunnen geven langs client-side. Dit is een additionele cache die sinds html5 beschikbaar is. Het laat ons toe om webpagina’s te cachen en deze offline beschikbaar te stellen. Offline betekent in dit geval geen toegang tot internet.

De werking van application cache is gebaseerd op een whitelist. Deze whitelist bevat een reeks van files die gecashed moeten worden en die dan offline beschikbaar zijn. De file beschrijft vier delen waarmee rekening moet worden gehouden: cache, network, fallback & settings. De naming convention voor een application cache file is het gebruik van de extensie .appcache, al is de keuze hierin vrij. Een typische layout voor application cache file is de volgende.

CACHE MANIFEST NIEUWsBLAD  
##### versie 13-03-2023

index.html  
content/scripts/jquery-min-1.8.1.js  
  
  
NETWORK:  
\*  
  
FALLBACK:  
fallback.html

- Cache: Dit zijn de files die gecached moeten worden om offline getoond te kunnen worden.

- Network: Hier vinden we alle files terug die niet gecached mogen worden. Indien ze getoond willen worden moet er bijgevolg een internetconnectie zijn.

- Fallback: Dit is een htmlpagina die getoond wordt wanneer een pagina offline niet terug te vinden is.

-Settings: Specifieert de appcache gedragingen. De enige mogelijke setting is appcache available. De bedoeling ervan is onduidelijk en dit wordt bijgevolg ook zelden geïmplementeerd.

Een tweede manier om in de whitelist te worden opgenomen is om <html manifest="nieuwsblad.appcache"> in de html declaratie van een pagina op te nemen.

Deze jonge veelbelovende techniek brengt echter een pak van beperkingen met zich mee. Net zoals de schrijver van het artikel op ‘A List part’ hebben ook wij tal van problemen ondervonden om deze techniek werkend te krijgen. De grootste problemen die we zijn tegen gekomen sommen we hier even op.

1. MIME Type: De application cache heeft een MIME Type[[7]](#footnote-7) nodig, een internet media type. Web browsers supporteren niet alleen tekstformaten, ze bieden ook ondersteuning voor images, video’s, audio... en tal van andere formaten. De verzameling van deze formaten heet MIME Types. Deze moeten echter gesupporteerd worden door een webserver. Enkel webpagina’s die op een webserver staan te draaien die het MIME Type ‘application cache’ kent, kunnen offline worden geraadpleegd. Het grootste gevolg hiervan is dat we zelf een webserver hebben moeten opstellen en hosten.
2. Files die niet in de lijst terug te vinden zijn worden niet opgeroepen. Dit kan dramatische gevolgen hebben wanneer het gaat over scripts of images. De scripts werken niet en de images worden niet getoond. Dit geldt zowel voor files die in de application cache staan als voor de files die gebruik maken van de manifest declaratie in hun html tag. Elke file die we offline willen tonen of gebruiken moet bijgevolg deel uitmaken van de application cache.
3. Wanneer er server-side veranderingen aan de website gebeuren wordt dit niet automatisch overgenomen door de appcache. De application cache moet zelf veranderd zijn eer de browser die veranderingen zal downloaden.
4. Wanneer de appcache file is veranderd, krijgt de gebruiker nog steeds niet de nieuwe versie te zien. Er is een refresh in de browser vereist om de nieuwe content te tonen. Dit kan gevolgen hebben wanneer een gebruiker reeds in interactie is met de oude versie.
5. Wanneer er een update gebeurt en nieuwe versies worden gecached is het belangrijk dat de volledige update lukt. Faalt de update van één onderlinge file, dan faalt de ganse update.

Conclusie: We zetten enkel zaken die nooit veranderen in de appcache.

#### De bootstrap startup procedure

Gezien de nodige site-effects die de appcache veroorzaakt, hebben we naar een mogelijke oplossing gestreefd die zo weinig mogelijk last heeft van deze bijwerkingen. We hebben daarom geopteerd om enkel een lichtgewicht javascript startup procedure te maken die wordt bijgehouden in de application cache. Een soort van bootstrap.

Alle andere files, zoals css- en javasript files, die regelmatig kunnen veranderen houden we bij in de localStorage. Pure data slaan we op in de WebSQL database.

De bootstrap opstart procedure maakt zo weinig mogelijk gebruik van de application cache. Het laadt een minimale lichtgewicht pagina in vanop een webserver (online) of de application cache(offline). Vervolgens wordt de startpagina met de nodige javascript-files en css aangemaakt die eveneens van de webserver(online) of uit de localStorage(offline) worden geladen. Als derde stap wordt de data van het web geladen(online) of uit de WebSQL database.

Onderstaande use case en activity diagramma’s verduidelijken meer.

1. Use case: Startup bootstrap
2. Use case omschrijving: Opstartprocedure door een lichtgewicht javascript dat zichzelf up to date houdt
3. Primary actor: Systeem
4. Preconditions: Geen
5. Trigger: Applicatie startup
6. Flow of events:

A1. De use case begint wanneer de applicatie wordt opgestart door een gebruiker

A2. Het systeem controleert of er een internetconnectie is

B1. Er is geen internetconnectie

A3. Het systeem start de online bootstrap opstartprocedure

A4. De online bootstrap procedure downloadt de nodige javascript en css-files van de webserver en bouwt de webpagina.

C1. De nodige javascript- en css files kunnen niet online geladen worden.

A5. Het systeem schrijft de net gedownloade css- en de javascriptfiles weg in de localStorage

A6. Het systeem kijkt of er online nieuwe artikels zijn.

A7. Het systeem downloadt de laatste artikels

A8. Het systeem parset de xml-file

A9. Het systeem schrijft de laatste artikels weg in de database.

A10. Het systeem laadt de laatste artikels uit de database en toont deze aan de gebruiker.

A11. De gebruiker krijgt de startpagina met de laatste artikels te zien.

B1. Er is geen internet verbinding

B2. Het systeem start de offline opstart procedure

B3. De offline opstart procedure haalt de nodige javascript- en css-files uit de localStorage.

D1. De javascript- en css-files zijn niet terug te vinden in de localStorage.

B4. Het systeem bouwt de startpagina aan de hand van de gevonden files in de localStorage.

B5. Het systeem kijkt of er artikels in de WebSQL database steken

E1. Er zijn geen artikels terug te vinden in de WebSQL database

B6. Het systeem laadt de artikels in uit de WebSQL database en toont ze aan de gebruiker.

B7. De gebruiker krijgt de artikels te zien.

C1. De nodige javascript files kunnen niet online ingeladen worden.

C2. We gaan naar stap B1

D1. De javascript- en css-files zijn niet terug te vinden in de localStorage.

D2. Het systeem geeft een melding dat de applicatie niet kan opstarten.

D3. De gebruiker krijgt de foutmelding te zien.

E1. Er zijn geen artikels terug te vinden in de WebSQL database.

E2. Het systeem geeft een foutmelding dat er geen artikels zijn gevonden.

E3 De gebruiker krijgt een algemene foutmelding te zien dat hij offline is en dat er geen artikels zijn terug gevonden.

1. Postconditions: Gebruiker kan nieuws consumeren.



In een tweede fase is de bootstrap procedure uitgebreid met het ophalen van data uit de webSQL database.



## Ontwikkelingsomgeving en tools

### Notepad++

Notepad++ is een open source code editor die heel wat programmeertalen ondersteunt. Deze gratis editor is in ons geval bijzonder geschikt om grote delen van de webcomponent (javascript, html, css) te ontwikkelen. Quote van de vendor: ‘*This is a hugely versatile tool which will prove incredibly useful for any coder, but also operates well as a more powerful text editor*.’

Meer info op <http://notepad-plus-plus.org/>

### Eclipse Indigo

Voor de ontwikkeling van de native app heeft Raf gebruik gemaakt van de ontwikkelingsomgeving Eclipse Indigo (versie 3.7.0). De app is gebouwd, getest en gedebugged in de Android Software Development Kit (SDK) versie 2.3.3 (API 10).

### Visual Paradigm for UML

De UML modellen zijn gemaakt in Visual Paradigm. Na een korte inleiding tijdens ons 2e jaar leek ons dit het meest geschikt om onze analyse te documenteren.

Meer info op <http://www.visual-paradigm.com/>

### Uml Factory

Uml Factory is een online macromedia flash based webapplicatie die toelaat diverse visuele uml modellen te tekenen. Deze modellen zijn exporteerbaar in verschillende formaten (xml, jpeg, png ...) zodat deze achteraf op andere dragers nog bewerkt kunnen worden. Er bestaat tevens een betalende, zeer goedkope android versie die dezelfde functionaliteiten biedt als de online versie, en die uitstekend werkt op de grotere tablets. We hebben van beide gebruik gemaakt.

De online tool is terug te vinden op <http://www.umlfactory.com/>

### Oracle VM Virtualbox

Met Oracle VM (versie 4.2.2) heeft Raf een tablet omgeving nagebootst. Zo kan hij testen of zijn gemaakte app ook in tablet formaat werkt.

Meer info op <https://www.virtualbox.org/>

### Github

Git is een versie controle systeem. Omdat software schrijven vaak te complex is om in één keer af te werken, schrijven we vaak verschillende versies. Deze bewaren we op een versie controle systeem, zodat we deze achteraf steeds kunnen raadplegen. Een versie controle systeem laat toe om verschillende versies van documenten, programma’s .. bij te houden en deze bij wijzigingen, indien nodig terug te draaien. Git is het versie controle systeem, github is het website gedeelte ervan.

We hebben Github for Windows (versie 1.0.46.0) gebruikt om onderling code en analyses uit te wisselen. Ook al hebben we aan verschillende onderdelen gewerkt, was het interessant om elkaars vooruitgang en code te bekijken. We hebben Github ook gebruikt om aan de bundel te werken. Onze publieke account is terug te vinden op <https://github.com/readerAppRafKurt>. We hebben 2 repositories aangemaakt, één voor de info en code van de webcomponent, de andere voor de native android app.

Meer info op <http://windows.github.com/>

### Excel

De alom bekende spreadsheat van Microsoft voor wie wellicht niemand extra uitleg hoeft. We hebben deze gebruikt om oa onze Work Breakdown Structure en Gantt Chart uit te werken.

### Adobe Photoshop

Eveneens een zeer gekende tool van macromedia. Gezien de goede skills van Kurt in deze omgeving hebben we geopteerd om eerst alle storyboards, mockups en prototypes op papier uit te tekenen, en deze daarna in photoshop snel in digitaal formaat om te zetten. Alternatieven om de storyboards uit te tekenen waren Axure, MockFlow, Balsamic...

### Powerpoint Storyboarding

Om onze analyses te verduidelijken/vereenvoudigen gebruiken we naast Uml Factory ook nog PowerPoint Storyboarding. Het programma krijg je bij de aankoop van Visual Studio Premium 2012, VS Ultimate 2012 of VS Test Professional 2012.

Meer info op <http://msdn.microsoft.com/en-us/library/vstudio/hh409276.aspx>

### Putty

PuTTY is een telnet en SSH client geschreven door Simon Tatham. Het is software waarmee men een verbinding kan maken met een andere computer, meestal zijn dit servers met een op UNIX gebaseerd besturingssysteem.

PuTTY is een vervanging voor een terminal (ook wel Teletype, TTY genoemd), een machine die voor de introductie van de personal computer gebruikt werd om te verbinden met een server. De verbinding tussen de terminalsoftware en de server verloopt tegenwoordig via het SSH-protocol, dat als veiligere opvolger wordt gezien voor het verouderde telnetprotocol.

We hebben deze tool voornamelijk gebruikt om manueel kleine aanpassingen te doen aan de webcomponent die staat te draaien op de apache http web server. In plaats van telkens kleine veranderingen via een trage ftp-verbinding naar de webserver te uploaden maken we gebruik van een SSH-sessie om rechtstreeks broncode aan te passen.

<http://www.chiark.greenend.org.uk/~sgtatham/putty/>

### Nano

Nano is een terminal-based tekst editor voor linux. We hebben deze vooral gebruikt om kleine aanpassingen te doen in broncode van css-, html- en javascript-files die staan de draaien op de apache web server.

### ADB & ADB wireless

Android Debug Bridge is een command line tool waarmee je met een android device kan communiceren. Dit kan zelfs een virtueel device (emulator) zijn.

<http://developer.android.com/tools/help/adb.html>

### Google Chrome developer tool

Google Chrome is een WebKit gebaseerde browser. WebKit is een verzameling van softwarebibliotheken die het mogelijk maakt om webpagina's te renderen. WebKit werd door Apple op de MacWorld Expo in San Francisco op 7 januari 2003 geïntroduceerd samen met de Safari-browser. WebKit is in eerste instantie ontwikkeld als back-end voor Safari. Het eerste doel van het WebKit-project was het verbeteren van de compatibiliteit van Safari met websites.

Op 7 juni 2005 werd heel WebKit open source. Sindsdien is de ontwikkeling van WebKit snel gevorderd. Tal van producenten en vendors van browsers hebben ondertussen de bibliotheken geïmplementeerd, oa ook Google Chrome.

Een van de tools die Google Chrome, evenals de meeste andere vendors voorziet, is een developer-plugin. Het is een zeer handige tool om tal van zaken te onderzoeken en te debuggen. Naast het feit dat deze plugin de mogelijkheid biedt om de html, css en javascript structuur van webpagina’s te onderzoeken, geeft het ook de mogelijkheid om de localStorage te raadplegen en sql-queries uit te voeren.



## Gebruikte technologieën

### Java

Java is een object georiënteerde programmeertaal. Gekende voordelen van object georiënteerde talen zijn encapsulatie, abstractie, overerving en polymorfisme. Android applicaties worden uitsluitend in java geschreven, waardoor er geen keuze is voor andere hogere talen zoals C# of visual basic.

### Android

Android is een linux-based besturingssysteem ontwikkeld door Android Inc. voor touchscreen devices zoals tablets en smartphones. Voordat het in 2005 opgekocht werd door Google, werd het door datzelfde Google al financieel gesteund. Android heeft wereldwijd een marktaandeel van 75% op gebied van smartphones. Omdat we nog nooit gewerkt hadden met Android, hebben we eerst een online cursus doorgeworsteld. De meest interessante kan je terugvinden op <http://developer.android.com/training/index.html> en <http://thenewboston.org/list.php?cat=6>.

### SQLite

SQLite is een relationeel database management systeem en implementeert de meeste SQL standaards. SQLite wordt vooral gebruikt als embedded database langs clientside. We vinden deze database bijvoorbeeld terug in application software zoals als het android platform. Omdat we alle data langs clientside moeten bijhouden hebben we geen andere keus dan deze database te gebruiken.

### WebSQL & IndexedDB

WebSQL is een lightweight SQLite database die we terug vinden in de Google Chrome en Apple Safari browser. Dit zijn ook de browsers die respectievelijk gesupporteerd worden door Android en IOs, de twee belangrijkste besturingssystemen die we terug vinden in de wereld van tablets en smartphones.

IndexedDB is eveneens een API die gebruikt wordt voor client-side storage maar is geen echte database. Het maakt gerbuik van key-values pairs in combinatie van een index en is terug te vinden in de firefox en microsoft browser.

### Javascript

#### De javascript jungle & the difference of the language

Het verplichte gebruik van javascript heeft een nieuwe wereld voor ons opengedaan. Ook al is dit een object georiënteerde taal, het verschil met ander OO talen, zoals Java of C# is enorm en in de verste verten niet te vergelijken. Enkele markante verschillen.

1. Javascript is zeer flexibel en zeker geen klassieke OO omgeving. Het kent enkel classless objecten. Er is bijgevolg nooit een blueprint van een object aanwezig.
2. Objecten kunnen op drie manieren aangemaakt worden. Prototyping, literal en singleton.
3. Javascript is asynchroon
4. Namespacing moet zelf geforceerd worden
5. Javascript code is moeilijk leesbaar, oa door het gebruik van de vele callback functies
6. Er bestaan geen goede tools zoals eclipse of visual studio die over een intellisense functie beschikken waardoor debuggen een pak gemakkelijker wordt.
7. Debugging, ondanks het gebruik van google chrome en safari developer blijft enorm pijnlijk.
8. Het is moeilijk om design patterns te implementeren.

#### Google feed api

Door het javascript cross domain probleem zijn we verplicht geweest om voor de webcomponent de xml-feeds extern te parsen. We konden ofwel onze parser zelfs schrijven, en deze op een webserver hosten, of op zoek gaan naar bestaande oplossing. Yahoo en google bieden een service aan om rss-feeds te parsen. We hebben uiteindelijk gekozen voor de google feed api. Het nadeel van de google feed api is dat deze asynchroon is, en dat deze om zijn eigen servers niet te fel te belasten zelf een cache bijhoudt. Hierdoor kan het zijn dat er reeds nieuwe artikels beschikbaar zijn, maar dat je die nog niet kan downloaden.

### Linux Ubuntu 12.04.1 Server

Ubuntu is een besturingssysteem dat is gebaseerd op Linux en GNU software. Ubuntu is gesponsord door het in het Verenigd Koninkrijk gevestigde bedrijf Canonical Ltd, dat inkomsten genereert door de verkoop van technische ondersteuning en diensten gekoppeld aan Ubuntu. Het besturingssysteem is en blijft gratis.

We hebben deze server opgezet en geconfigureerd om er een Apache webserver en de Vsftpd daemon op te laten draaien.

### Apache HTTP Server Version 2.2

De noodzaak van een webserver is er gekomen omdat we anders bepaalde zaken voor de webcomponent niet konden testen. De oorzaak ervan zijn volgende twee elementen.

1. De application cache, die instaat voor een deel van de offline functie heeft een MIME type nodig om te kunnen werken. Dit MIME type kan alleen gegenereerd worden door een webserver. Dit doen we door een .htacces file met de nodige elementen in de rootfolder van onze javascriptfiles te zetten.
2. Uit veiligheids overwegingen kunnen we lokaal (localhost of 127.0.0.1) op een android toestel geen javascript files runnen. Dit is een restrictie voorzien door het android besturingssysteem waar we niet omheen kunnen. Het serveren van de javascripts moet daarom van een echte webserver komen.

### Vsftpd FTP Server 3.0.2

Vsftpd is een linux FTP server. FTP staat voor File Transfer Protocol. Het is een protocol dat toelaat om files te downloaden en te uploaden. FTP werkt volgens het client/server model.. Het server gedeelte wordt een FTP daemon genoemd en draait in de achtergrond als een service op onze linux ubuntu server. De daemon luistert constant naar ftp requests van clients. Wanneer een request wordt ontvangen wordt er een sessie geopend. De client kan vanaf dan, naargelang de nodige rechten, files beginnen te downloaden, uploaden, deleten...

We hebben deze server opgezet om op een gemakkelijke manier alle bestanden die de webcomponent nodig heeft naar de apache webserver te uploaden.

## Gebruikte patterns

### Model View Controller (MVC) (Native android)

In android is een MVC structuur moeilijk te behalen. De built in procedure van een mvc gebruikt een xml file voor de views en de activities als debijhorende controller. Deze kunnen niet los van elkaar gekoppeld worden.

### Data Access Object (DAO)

|  |  |
| --- | --- |
|  | Voor de artikels, channels, fonts, layouts en themes hebben we een dao aangemaakt. Tussen de dao’s en de db staat de DatabaseHandler. Dit is een service klasse die overerft van SQLiteOpenHelper. Deze klasse bevat de verschillende queries naar de database. |

Verder bestaan ook de ActivityDao en ImagaDao. Deze hebben voeren geen selecties of bewerkingen uit op de database. De ActivityDao houdt de actieve activities bij in een static lijst. Deze lijst wordt bij afsluiten of update van de app leeg gemaakt.

Via ImageDao worden de images gesaved en opgehaald uit het filesysteem van het device.

Een poging om dit pattern in javascript te te implementeren is niet volledig gelukt. Doordat de webSQL database asynchroon is en met transactions werkt, zijn callback functies vereist binnen deze transactions. Callback functies zijn functies die als parameter worden meegeven in een andere functie, en die worden uitgevoerd binnen die functie. In het geval van WebSQL transactions worden deze callback functies pas uitgevoerd wanneer een transactie gelukt is. Vanaf dan kan men met de query resultaten beginnen werken.

## Eigen gemaakte Listeners (Native android)

### ArticleListener

Raf heeft een eigen articleListener gemaakt. Deze wordt opgeroepen in de activities van Phone 3 en tablet. Deze listener kijkt wanneer de lezer op een artikel klikt en zal een nieuwe activity starten. Deze nieuwe opgestarte activity toont één enkel artikel op het scherm.

### ImageListener

Raf heeft een eigen imageListener gemaakt. Deze Listener erft over van de OnClickListener. Zo kan de gebruiker na een klik op een foto, de foto op full screen bekijken. Met de klik wordt een volgende activity gestart (FullScreenImageActivity) en wordt het pad van de foto meegegeven. Raf heeft geprobeerd om de ganse Bitmap mee te geven maar voor sommige foto’s gaf dit de volgende foutmelding “FAILED BINDER TRANSACTION”. Foto’s die te groot zijn kunnen niet doorgegeven worden tussen activities.

De foto’s in portrait/landscape formaat worden fixed in portrait/landscape getoond, De lezer kan de foto niet meer draaien.

### Swipe gesture[[8]](#footnote-8)

De lezer wilt graag tussen de verschillende artikels kunnen bladeren. Hiervoor hebben we een eigen SimpleOnGestureListener gemaakt. Deze overloopt de volgende stappen in het scherm waarin een artikel wordt getoond.

Stap 1: Vaststellen van een veegbeweging. Ja, dan stap 2.

Stap 2: Is er een horizontale beweging. Ja, dan stap 3.

Stap 3: Is de beweging voldoende snel. Ja dan stap 4.

Stap 4: Controle van de richting. Indien naar rechts, dan wordt het volgende artikel getoond. Indien naar links, dan kan de lezer naar het vorige artikel terugkeren.

De opeenvolging gebeurt aan de hand van de id’s van de artikels van éénzelfde channel in de database. Indien de lezer in het scherm van het eerste artikel staat, dan is het vorige artikel datgene met de hoogste id. Het tegengestelde gebeurt wanneer de lezer in het laatste artikel van de database staat.

## Eigen gemaakte scrollers (Native android)

Voor de artikels en de channels heeft Raf eigen scrollers gemaakt. Deze worden in de tablet versies gebruikt en staan aan de rand van en onderaan het scherm. Voor de artikelscroller haalt een lijst van de beschikbare artikels, rekening houdend met het actieve channel, uit de database en toont vervolgens de verschillende artikels. De Channelscroller haalt op zijn beurt een lijst van de geactiveerde channels uit de database.

## Dynamische layout (Native android)[[9]](#footnote-9)

Zowel de kleurenthema's als de lettertypes kunnen door de lezer aangepast worden. De lezer kan de lettertypes en de thema's afzonderlijk van elkaar aanpassen. Hiervoor heeft Raf gewerkt met de setTheme methode van de Activity class. Deze themes worden gedefinieerd binnen de styles.xml.

Voor de lettertypes worden de volgende kenmerken gespecifieerd: textSize, textColor, textStyle, paddingBottom en typeface. Omdat een titel niet dezelfde lettergrootte heeft als de tekst van het artikel moet dit gespecifieerd worden. Zo hebben de knoppen, titelvelden en tekstvelden van de artikelteksten elk een aparte style.

Voor de kleurenthema's worden de volgende kenmerken gespecifieerd: backgroundTheme, buttonTheme, listviewTheme, spinnerTheme en backgroundTitleBar.

Deze werkmethode werkt spijtig genoeg niet voor objecten die at runtime aangemaakt worden. Daarvoor moet je eigen xml aanmaken. Zo heeft Raf voor de knoppen, titelvelden als tekstvelden eigen xml's aangemaakt. Deze worden in de activity opgeroepen. In Android wordt dit principe inflate genoemd[[10]](#footnote-10).

## Database in Native android[[11]](#footnote-11)

De artikels moeten ook offline te lezen zijn. Daarom heeft Raf geopteerd om de artikels bij te houden in een SQLite database. Tijdens de eerste opstart van de app worden de verschillende tabellen aangemaakt. Dit gebeurt in de onCreate methode van de DatabaseHandler klasse. In deze klasse staan ook de verschillende queries. Verder staan er in de DatabaseHandler klasse methodes die door de DAO’s (Data Access Object) gebruikt worden om info uit de database (vb. lijst van artikels) naar de views te sturen.

### Update database



### Uitlezen van de feeds

Aan de hand van de url van de channels, die worden bijgehouden in een Enum type (ChannelType) worden de geselecteerde feeds uitgelezen. De verschillende tags van de feed worden uitgelezen met behulp van Document Object Model (DOM) element. Het alternatief is SAX parser. Deze uitwerking staat in de initialize methode van ArticleDao. Na het uitlezen van elk artikel uit de feed wordt gecontroleerd of dit artikel nog niet in de database staat. Er wordt geen rekening gehouden met artikels die reeds in de database staan.[[12]](#footnote-12)

## Project structuur

De source code van de native android app is ingedeeld in 5 folders.

Hieronder kan je een overzicht terugvinden van de viewelementen.







### BackgroundActivity[[13]](#footnote-13)

Tijdens de activities wordt een achtergrond activiteit gestart. Dit is een Asynctask die in de achtergrond zijn werk doet. De lezer kan gerust verder bladeren of lezen. Hij/zij zal niet moeten wachten tot de activiteit volledig gedaan heeft. Het onderstaande diagram toont de verchillende activities die doorlopen worden.



### Statechart diagram activities[[14]](#footnote-14)

Voor deze app heeft Raf enkel de states onCreate en onRestart gebruikt. In de specifieke activity kan je deze methodes overriden.

De onCreate wordt gebruikt de eerste maal een activity wordt gestart.Tijdens deze fase worden de componenten van het scherm gemaakt en de backgroundactivity[[15]](#footnote-15) wordt gestart. Na het voltooien van de onCreate wordt het scherm aan de gebruiker getoond.

Tijdens de onCreate method van elke activity wordt deze toegevoegd in een static lijst van activities. Deze lijst wordt leeggemaakt elke keer de lezer de configuratie van de app aanpast.

Verder wordt ook de instellingen voor de fonts en themes gecontroleerd/ingesteld.[[16]](#footnote-16)



# Testen

## Native Android app

De sdk voorziet de mogelijkheid om de de app te testen in een eigen aangemaakte emulator. ‘ook de aanmaak van een virtual machine’ bracht geen soelaas. Na vele pogingen is Raf er niet in geslaagd om de snelheid van uitvoering op een deftig niveau te brengen. Daarom heeft Raf geopteerd om de app te testen op mijn eigen device.

Verder kan je de inhoud van variabelen ook testen aan de hand van Logcat. In de code kan je de inhoud van bepaalde veriabelen wegschrijven naar deze log.

## Webcomponent

Het testen van de webcomponent liep bijzonder moeilijk. Om de application cache te laten werken hebben we onder andere als extra een apache webserver moeten opzetten. Voorts is het testen en debuggen van javascript enorm lastig. Voor javascript bestaan er namelijk geen tools die beschikken over een intellisense feature zodat het vaak in het blinde weg werken is.

Ook de verschillen in bepaalde gedragingen tussen de verschillende browsers hebben een impact gehad. Een google chrome browser in een android webview activity gedraagt zich anders dan een native chrome browser op een android platform.

Zo zijn er tijdens het ganse project continu kleine problemen blijven opduiken waarvoor telkens nieuwe oplossingen moesten gezocht worden. Ook het niet weten wat voor data er telkens binnenkomt heeft ons meerdere malen met de handen in de haren doen zitten. Soms bestaan artikels uit slechts een titel en één enkele paragraaf waardoor er enorm veel witruimte onstaat. Omgekeerd worden dan vaak zeer grote media bestanden meegeleverd.

# Conclusie

We hebben gekozen voor de iteratieve methode van ontwikkelen. De methode sprak ons aan omdat we op deze manier op een snelle en eenvoudige manier aan de ontwikkelingsfase konden beginnen. We wisten op voorhand dat dit een intensieve methode is. Inzet en tijd zouden 2 termen zijn die wel heel rekbaar moesten zijn. Door schade en schande hebben we moeten vaststellen dat deze methode niet geschikt is om toe te passen voor een eindproject.

De voornaamste moeilijkheden die wij met deze methode hebben meegemaakt: je moet continue deliverables opleveren. Door de combinatie van werken en studeren weet je niet altijd wanneer je tijd zal hebben om aan het project te werken. Door ons gebrek aan ervaring in projectplanning en technische kennis is het moeilijk om op voorhand te weten hoe lang je aan een onderdeel zal werken. Dit heeft ervoor gezorgd dat we enkele malen onze deadline hebben overschreden. Hierin heeft Johan Geybels ons enkele malen op attent gemaakt.

Toch staan we niet negatief ten opzichte van de iteratieve methode en lijkt het ons in een professionele omgeving een interessante manier van werken. Communicatie is een niet te onderschatten onderdeel tijdens het proces. Tussen Kurt en Raf ging dit goed. We hadden wekelijks onderlinge contacten over de vooruitgang van het project en de problemen, zowel technisch als organisatorisch, die we zijn tegengekomen. Toch hebben we enkele problemen gehad met afgeschafte meetings. Een korte e-mail ter herinnering van de meeting kan hier als oplossing dienen. Zeker als de meeting doorgaat na een vakantie.

We hadden graag al veel vroeger een werkende versie willen tonen aan Gilles Van Mol. Zo hadden we met hem meer in detail kunnen gaan en problemen verder kunnen wegwerken. Ook hadden we Gilles meer moeten betrekken in de vooruitgang van het project. Een vooruitgangsrapport had Gilles kunnen helpen om het verloop van het project beter te volgen en indien nodig in te grijpen.

Nog een korte vergelijking van bevindingen tussen beide technieken:

|  |  |  |  |
| --- | --- | --- | --- |
| Native android | | Webcomponent | |
| Voordelen | Nadelen | Voordelen | Nadelen |
|  | Niet deployable op andere systemen zoals IOS, Blackberry | Porteerbaar naar andere systemen die html5, javascript, css3 en WebSQL ondersteunen |  |
|  |  |  | Gebruik van zowel de asynchrone google feed api als de asynchrone WebSQL database zorgt voor trage pagina loads. |
| Kan gebruik maken van hardware resources van het device |  |  | Kan slechts zeer beperkt gebruik maken van hardware resources , maar is daardoor afhankelijk van bovenliggende java-code en daardoor niet meer porteerbaar naar andere systemen. |
| Data kan zonder probleem bijgehouden worden in de SQLite database |  |  | Het offline gedeelte teert op de WebSQL database en de application cache. Deze laatste heeft een MIME Type nodig dat afkomstig moet zijn van een webserver. |
|  | De effecten zijn van technische aard en moeilijk in te bouwen |  | De effecten zijn zeer moeilijk te implementeren. |
| Ful SQL |  |  | SQL implementatie maar beperkt door de webkit-normen, eigen aan de browsers. |
| De feeds kunnen intern verwerkt worden |  |  | De feeds kunnen wegens het java cross-domain niet verwerkt worden en hebben bijgevolg een externe parser nodig zoals Google Feed api |
|  | Het gebruik van de enclosures, de media-objecten die bij artikels horen heeft een enorme vertraging tot gevolg |  | Enclosures, media-objecten hebben een enorme impact op de snelheid van inladen. |

Door de vele moeilijkheden die we hebben ondervonden is de uitwerking van dit project vaak een processie van Echternach geweest. Het was dikwijls drie stappen vooruit, twee achteruit waarbij tal van nieuwe problemen moesten opgelost. Ook al zit er functionaniteit in en een beetje design en usability, toch kent dit project kent nog vele tekortkomingen en is het zeker voor verbetering vatbaar.

Slotwoord

Chuchichästli. Een Zwitsers woord dat staat voor snoepjeskast.

Voor ons staat niet de betekenis maar wel de uitspraak ervan als een minimalistische metafoor die de moeilijkheden van het project omschrijft die we de ganse tijd hebben ondervonden. Het is er echter ook ééntje die duidelijk maakt dat deze moeilijkheden ervan geenszins minimalistisch waren. We nodigen iedereen uit om het eens uit spreken.

# Appendix

## Work breakdown structure

Work Breakdown Structure versie 2

1. Uitlezen artikels – reader module
   1. Analyse 6 d
   2. Programmeren: Android 6 d
   3. Testen: Android 4 d
   4. Programmeren: Webcomponent 6 d
   5. Testen: Webcomponent 4 d
2. Download module
   1. Analyse 6 d
   2. Programmeren: Android 6 d
   3. Testen: Android 4 d
   4. Programmeren: Webcomponent 6 d
   5. Testen: Webcomponent 4 d
3. Configuratie reader
   1. Analyse 6 d
   2. Programmeren: Android 6 d
   3. Testen: Android 4 d
   4. Programmeren: Webcomponent 6 d
   5. Testen: Webcomponent 4 d
4. "Optimaliseren presentatie artikels + 5 templates + verschillende schermresoluties"
   1. Analyse 6 d
   2. Programmeren: Android 6 d
   3. Testen: Android 4 d
   4. Programmeren: Webcomponent 6 d
   5. Testen: Webcomponent 4 d
5. Redesign en refactoring: Android en Webcomponent 30 d
6. Blackbox van reader en download module
   1. Analyse 6 d
   2. Programmeren: Android 6 d
   3. Testen: Android 4 d
   4. Programmeren: Webcomponent 6 d
   5. Testen: Webcomponent 4 d
7. Offline capaciteit
   1. Analyse 6 d
   2. Programmeren: Android 6 d
   3. Testen: Android 4 d
   4. Programmeren: Webcomponent 6 d
   5. Testen: Webcomponent 4 d
8. De reader kan elk soort 2.0 Rss-feed inlezen
   1. Analyse 6 d
   2. Programmeren: Android 6 d
   3. Testen: Android 4 d
   4. Programmeren: Webcomponent 6 d
   5. Testen: Webcomponent 4 d

## Gantt chart



## Verslagen meetings

**Verslag 1 : Kick-off Meeting**

**Datum:** 16/10/2012

**Start:** 18 u

**Duur:** 1,5 uur

**Plaats:** ACE-Groep T Leuven

**Aanwezig:**

|  |  |
| --- | --- |
| Johan Geybels | Docent ACE-Groep T  Interne promotor |
| Kurt Van den Branden | ACE-Groep T  Project Team |
| Raf De Gelder | ACE-Groep T  Project Team |

**Afwezig:** /

Tijdens de vergadering werden de onderstaande afspraken gemaakt.

**Topic 1:** Scope

Tijdens de volgende vergadering met Johan zullen we een uitgebreide scope voorstellen. Aan de hand van het MoSCoW (Must/Should/Could/Would have) principe zullen we de voorwaarden van Twipe Mobile Solutions en onze eigen extra functionaliteiten indelen.

Op 31 december 2012 zullen we dan verder in samenspraak met de interne en externe promotor evalueren of de scope nog uitgebreid kan worden.

**Topic 2:** Inschrijvingen begeleidende cursussen

Vanaf deze week kunnen we ons inschrijven voor de verplichte cursus *planmatige aanpak van je ondernemingsproject* en de vrijblijvende cursus *communicatieve aspecten van je ondernemingsproject infosessie 1*.

De opleiding *planmatige aanpak van je ondernemingsproject* gaan we beiden bijwonen op 23/10/2012.

**Topic 3:** POC uitlezen RSS feeds

We gaan een eerste test uitvoeren om een opgegeven RSS feed uit te lezen. Indien mogelijk reeds te tonen. De test doen we zowel in html5 als native in de app.

**Topic 4:** Planning project

Op basis van de voorlopige[[17]](#footnote-17) scope gaan we een planning uitwerken. Met de informatie uit de cursus *planmatige aanpak van je ondernemingsproject* gaan we deze planning verder uitwerken.

**Topic 5:** Werking en taakverdeling

Op voorstel van onze externe promotor hebben we besloten om ons project uit te werken met een iteratief ontwikkelingsmethode. Tijdens sprints van een 3 tal weken zullen we het iteratieve proces van analyse, programmatie, testen, evaluatie en deployment.

De studenten, Kurt en Raf, zullen afwisselend een stap binnen dit iteratieve proces op zich nemen. Zo zullen we beiden aan het einde van het project de verschillende stappen meerdere keren hebben doorlopen.

Om de communicatie met de interne en externe promotor te vereenvoudigen en te kanaliseren zal Raf de communicatie met de interne promotor op zich nemen en is Kurt verantwoordelijk voor de communicatie met de externe promotor Gilles Van Mol.

**Verslag 2: meeting interne promotor**

**Datum:** 13/11/2012

**Start:** 19 u 45

**Duur:** 45 min

**Plaats:** ACE-Groep T Leuven

**Aanwezig:**

|  |  |
| --- | --- |
| Johan Geybels | Docent ACE-Groep T  Interne promotor |
| Kurt Van den Branden | ACE-Groep T  Project Team |
| Raf De Gelder | ACE-Groep T  Project Team |

**Afwezig:** /

Tijdens de vergadering werden de onderstaande afspraken gemaakt.

**Topic 1:** Inhoud “analyse” gedeeltes van planning

Elke taak binnen onze scope heeft als subtaak een analyse gedeelte. Omdat onze tijd beperkt is, zullen we deze subtaak waar mogelijk is proberen in te korten. Voor eenvoudige taken lijkt het ons voldoende om enkel een storyboard te maken. Voor complexe taken kunnen we de analyse uitbreiden met uses cases, activity diagrams, state diagrams,…

**Topic 2:** Inhoud “testen“ gedeeltes van planning

Een testfase van een taak binnen de Gantt chart bevat zowel het testen door het projectteam zelf als het testen door de externe promotor. Zo zullen we Gilles Van Mol telkens de afgewerkte versies doorsturen. Deze versies zullen we hem doorsturen via https://www.wetransfer.com/.

Een test zal geslaagd zijn indien deze voldoet aan de functional requirements die vooraf worden opgesteld. We gaan deze vereisten bespreken met de externe promotor.

**Topic 3:** Aanpassing Gantt chart

We hebben gekozen om ons project volgens het iteratieve methode aan te pakken. Door het gebruik van deze methode zullen zaken zoals klasses en code in de loop van het project niet meer up to date zijn. Daarvoor gaan we een periode inlassen voor redesign van het klassendiagram en refactoring van de reeds geschreven code. Deze taak hebben we gepland voor januari 2013.

**Topic 4:** Proces oplossen van bugs

Tijdens de testfase kan zowel de externe promotor als het project team gevonden fouten melden via Mantis[[18]](#footnote-18). Door gebruik te maken van versienummering binnen Mantis en de versienummer te vermelden in de naam van de applicatie kunnen we de fout gemakkelijker identificeren. Via Mantis kunnen we ook de status van de bugs aanpassen en opvolgen.

**Topic 5:** Code sharing binnen project team

We gaan Github gebruiken om onderling programmacode uit te wisselen[[19]](#footnote-19). Zo zullen we beide steeds met de meest recente code kunnen werken.

**Topic 6:** Opstellen voorlopig datamodel

Dit model zal tijdens de het verloop van het project vergroten en met een grote waarschijnlijkheid worden aangepast tijdens het redesign proces. Deze taak is gepland tijdens de maand januari 2013.

**Verslag 3: meeting interne promotor**

**Datum:** 9/1/2013

**Start:** 19 u 00

**Duur:** 30 min

**Plaats:** ACE-Groep T Leuven

**Aanwezig:**

|  |  |
| --- | --- |
| Johan Geybels | Docent ACE-Groep T  Interne promotor |
| Raf De Gelder | ACE-Groep T  Project Team |

**Topic 1:** We hebben ons gesprek met Gilles Van Mol overlopen o.a. de manier van testen door ons/hem, de functional requirements, code uitwisseling…

**Topic 2:** Tegen 22/01/2013 zullen er deliverables gemaakt worden voor de tweede sprint. Dit mag een uitgeschreven use case zijn of een werkend deel applicatie.

**Verslag 4: meeting interne promotor**

**Datum:** 22/01/2013

**Start:** 19 u 30

**Duur:** 30 min

**Plaats:** ACE-Groep T Leuven

**Aanwezig:**

|  |  |
| --- | --- |
| Johan Geybels | Docent ACE-Groep T  Interne promotor |
| Kurt Van den Branden | ACE-Groep T  Project Team |
| Raf De Gelder | ACE-Groep T  Project Team |

**Afwezig:** /

Tijdens de vergadering werden de onderstaande afspraken gemaakt.

1. Tegen de volgende vergadering, die gepland is eind februari 2013, zullen er deliverables getoond worden voor het configuratie onderdeel. Hierin vallen de volgende punten: fonts, thema’s, layouts, (blader)effecten en keuze van de verschillende RSS feeds (dit punt is optioneel).
   1. **Fonts**

De lezer zal kunnen kiezen tussen 3 verschillende lettergroottes, lettertypes, kleur, bold en italic. Zo kan de lezer de look van de artikels aanpassen.

* 1. **Thema’s**

Aanmaken/gebruik van 3 verschillende sjablonen.

* 1. **Layouts**

Tegen begin februari zullen we onze voorstellen voor de verschillende layouts aan Gilles voorleggen. Er zullen 3 verschillende layouts worden gemaakt voor tablets en 3 voor phones. Raf maakt de voorstellen voor de phoneresolutie en Kurt voor de tablets.

* 1. **Effecten**

De lezer kan kiezen tussen 3 bladereffecten om tussen de verschillende artikels/binnen 1 artikel te bladeren.

* 1. **RSS feeds (optioneel)**

De lezer kan bepalen welke feeds (Binnenland, Buitenland, Economie, Cultuur en/of Life) hij wilt bijhouden en lezen. Dit is geen requirement voor de opdrachtgever maar wij willen dit toch graag uitwerken.

**Verslag 5: meeting interne promotor**

**Datum:** 12/03/2013

**Start:** 19 u 45

**Duur:** 30 min

**Plaats:** ACE-Groep T Leuven

**Aanwezig:**

|  |  |
| --- | --- |
| Johan Geybels | Docent ACE-Groep T  Interne promotor |
| Kurt Van den Branden | ACE-Groep T  Project Team |
| Raf De Gelder | ACE-Groep T  Project Team |

**Afwezig:** /

Tijdens de vergadering werden de onderstaande afspraken gemaakt.

1. Tegen de volgende vergadering, die gepland is voor 16 april 2013, zullen er deliverables getoond worden voor het onderdeel blackbox reader en download module.
2. Wijziging planning
   1. Tegen midden april zullen de te testen applicaties aan Gilles Van Mol worden overhandigd. Zo kan hij deze testen. Raf zal de webcomponent testen en Kurt zal de native applicatie testen.
   2. Vanaf midden april zullen we beginnen aan het uitschrijven van de bundel voor het eindproject. Deze bundel moet begin mei 2013 klaar zijn.
3. Op 20 maart 2013 is een meeting gepland met Gilles Van Mol.

**Verslag 6: meeting interne promotor**

**Datum:** 02/05/2013

**Start:** 19 u 00

**Duur:** 30 min

**Plaats:** ACE-Groep T Leuven

**Aanwezig:**

|  |  |
| --- | --- |
| Johan Geybels | Docent ACE-Groep T  Interne promotor |
| Kurt Van den Branden | ACE-Groep T  Project Team |
| Raf De Gelder | ACE-Groep T  Project Team |

**Afwezig:** /

Tijdens de vergadering werden de onderstaande afspraken gemaakt.

1. Deadlines bundel eindproject:

Tegen 07 mei 2013 moet een voorlopige bundel worden doorgestuurd naar Johan Geybels.

Tegen 14 mei 2013 moet de bundel af zijn en doorgestuurd worden naar Johan Geybels. Hij zal deze dan nalezen en zijn opmerkingen naar ons doorsturen op 17 mei 2013. We zullen zelf elkaars delen nalezen.

Tegen 21 mei 2013 moet de bundel (in 5 exemplaren en ingebonden) ingediend worden bij de Studentenadministratie. Eén exemplaar moet in .pdf formaat naar Johan Geybels worden doorgestuurd.

1. Presentatie

Op 28 mei 2013 is er een testpresentatie gepland. Daarvoor zullen een aantal slides worden voorzien.

Op 4 juni 2013 is de finale presentatie gepland.

1. Op 8 mei 2013 is een meeting gepland met Gilles Van Mol.
2. Op 22 mei 2013 is een afspraak gepland met Jan Stevens om de projectdefinitie en Gantt chart toe te lichten.

1. Rss-feed volgens de officiële standaard. Meer info op: http://cyber.law.harvard.edu/rss/rss.html [↑](#footnote-ref-1)
2. RFC 822 en andere datum formaten: http://en.wikipedia.org/wiki/Date\_%28Unix%29 [↑](#footnote-ref-2)
3. Mandatory: Verplicht [↑](#footnote-ref-3)
4. MIME-type: http://en.wikipedia.org/wiki/MIME [↑](#footnote-ref-4)
5. WebSQL: http://www.w3.org/TR/webdatabase/ [↑](#footnote-ref-5)
6. A list Apart, het volledige artikel: http://alistapart.com/article/application-cache-is-a-douchebag [↑](#footnote-ref-6)
7. MIME Type: http://en.wikipedia.org/wiki/MIME [↑](#footnote-ref-7)
8. Adneal, <http://stackoverflow.com/questions/10008053/android-trouble-with-swipe-gesture>, 4 april 2012 [↑](#footnote-ref-8)
9. Matt Quigley, <http://www.androidengineer.com/2010/06/using-themes-in-android-applications.html>, (December 2012),

   [Janarthanan](http://stackoverflow.com/users/803736/janarthanan) http://stackoverflow.com/questions/3241729/android-dynamically-change-style-at-runtime, (December 2012),

   Lalit Sharma, <http://www.androidseeker.blogspot.in/2012/09/how-to-apply-new-theme-to-whole.html>, (December 2012) [↑](#footnote-ref-9)
10. Dan K, <http://stackoverflow.com/questions/3142067/android-set-style-in-code>, 30 maart 2011, [RoflcoptrException](http://stackoverflow.com/users/329637/roflcoptrexception) , <http://stackoverflow.com/questions/5763366/how-to-create-multiple-buttons-at-runtime-android>, 23 april 2011 [↑](#footnote-ref-10)
11. Joe, http://javapapers.com/android/android-sqlite-database/, 21 februari 2013,

    Lars Vogel, <http://www.vogella.com/articles/AndroidSQLite/article.html>, 16 januari 2013,

    Ravi Tamada, <http://www.androidhive.info/2011/11/android-sqlite-database-tutorial/>, (februari 2013) VimalTuts, <http://vimaltuts.com/android-tutorial-for-beginners/android-sqlite-database-example>, (maart 2013)

    <http://android-er.blogspot.be/2011/06/simple-example-using-androids-sqlite_02.html>, (maart 2013) [↑](#footnote-ref-11)
12. Ravi Tamada, <http://www.androidhive.info/2011/11/android-xml-parsing-tutorial/>, (februari 2013)

    Mkyong, <http://www.mkyong.com/java/how-to-read-xml-file-in-java-dom-parser/>, 21 januari 2013,

    [Nithy Vasudevan](Nithy%20Vasudevan), <http://theopentutorials.com/tutorials/android/xml/android-simple-xml-dom-parser/>, (januari 2013) [↑](#footnote-ref-12)
13. Info : <http://www.mysamplecode.com/2012/11/android-progress-bar-example.html>, (januari 2013),

    Lars Vogel, <http://www.vogella.com/articles/AndroidBackgroundProcessing/article.html>, 19 mei 2013,

    Java Sri Lankan Support, <http://www.javasrilankansupport.com/2012/11/asynctask-android-example-asynctask-in.html> (januari 2013) [↑](#footnote-ref-13)
14. <http://developer.android.com/reference/android/app/Activity.html> [↑](#footnote-ref-14)
15. Verwijzen naar uitleg backgroundActivity [↑](#footnote-ref-15)
16. Zie verder [↑](#footnote-ref-16)
17. Zie topic 1 van dit verslag [↑](#footnote-ref-17)
18. Meer info http://www.mantisbt.org/ [↑](#footnote-ref-18)
19. Meer info https://github.com/ [↑](#footnote-ref-19)