



# Afstudeerverslag herontwikkeling Eduarte Studentenapp

Afstudeerproject Saxion HBO-ICT Software Engineering

Auteur	Jesse Roessink
Datum	17 juni 2023
Klasse	Openbaar
Versie	2.0

# Details

Bedrijf:	Topicus Onderwijs
Adres:	Keizerstraat 43, 7411 HE Deventer
Telefoon:	0570 - 203 093
Bedrijfsbegeleider:	Matthijs de Vrijs, Elise Puijk
Functie:	Applicatie ontwikkelaar
Email:	matthijs.de.vries.topicus.nl, elise.puijk@topicus.nl
Student:	Jesse Roessink
Studentnummer:	481030
Email:	481030@student.saxion.nl
Telefoon:	+31681939066
Afstudeerbegeleider	Eelco Jannink
Email	e.h.a.jannink@saxion.nl

# Voorwoord

Dit verslag is geschreven door Jesse Roessink naar aanleiding van de afstudeerstage bij Topicus in Deventer. Het project is uitgevoerd binnen de Eduarte businessline van Topicus. In dit verslag beschrijf ik mijn bevindingen uit het onderzoek naar verschillende frameworks die voor de Eduarte Studentenapp gebruikt kunnen worden, advies over ontwerpkeuzes en een prototype van de app in een nieuw framework. De stage is uitgevoerd van 13 februari tot 7 juli 2023.

Binnen Topicus ben ik begeleid door Matthijs de Vries, Elise Puijk en het ontwikkelteam Davo. Vanuit Saxion Hogeschool ben ik begeleid door Eelco Jannink. Bij deze wil ik iedereen bedanken die mij gedurende het project heeft ondersteund.

# Definities

Term	Definitie
Framework	Een framework is een stuk software dat de basis functionaliteiten bevat.
Hybrid development	Een applicatie die in één keer gebouwd wordt voor verschillende platforms
DOM	Het Document Object Model is de structuur en inhoud van een document op het internet.
MVP	Minimum Viable Product
IDE	Integrated development environment
MVC+S	Model View Controller + Service
CLI	Command Line Interface
AES	Advanced Encryption Standard

Tabel 1 - Definities

# Inhoud

Details .....	2
Voorwoord .....	3
Definities .....	4
Samenvatting .....	7
1. Inleiding .....	8
1.1 Topicus .....	8
1.2 Eduarte .....	8
1.3 Aanleiding .....	8
1.4 Probleemstelling .....	9
1.5 Doelstelling .....	9
1.6 Scope .....	10
2. Projectaanpak .....	11
2.1 Proces .....	11
2.2 Werkwijze .....	11
2.3 Retrospective .....	12
2.4 Planning .....	13
2.5 Tools .....	13
3. Onderzoek .....	14
3.1 Vooronderzoek .....	14
3.2 Huidige situatie .....	14
3.3 Hoofd- en deelvragen .....	17
3.4 Welke eisen stelt Topicus aan een nieuwe app? .....	18
3.5 Welke eisen stelt Topicus aan een toekomstig framework? .....	21
3.6 Welke risico's neemt het ontwikkelen van de app in een nieuw framework met zich mee? .....	23
3.7 Wat zijn de belangrijkste aspecten waar het nieuwe framework aan moet voldoen om dezelfde functionaliteiten als de huidige Eduarte Studentenapp te bieden? .....	27
3.8 Welk framework kan ik Topicus het beste adviseren? .....	30
3.9 Hoe kan ik de authenticatie zo veilig mogelijk maken in het nieuwe framework? ..	32
3.10 Op welke manier kan het beste veilig data opgeslagen worden binnen het gekozen framework? .....	35
3.11 Welke functionaliteiten moeten er in het prototype zitten om aan te tonen dat het framework voldoet aan de eisen? .....	37

4. Prototype.....	39
4.1 Ontwerppatroon.....	39
4.2 Keuze scherm school .....	40
4.3 Inloggen OAuth 2.0 .....	41
4.4 Opslag.....	43
4.5 Thema's.....	44
4.6 Agendapagina.....	45
4.7 Resultatenpagina .....	46
4.8 Pushnotificaties.....	47
5. Acceptatietest .....	49
5.1 Conclusie .....	51
6. Conclusie.....	52
7. Reflectie .....	53
7.1 Planning .....	53
7.2 Analyseren .....	53
7.3 Adviseren.....	53
7.4 Realiseren .....	54
7.5 Resultaat .....	54
8. Bibliografie .....	55
9. Versiebeheer .....	57
10. Bijlagen .....	58
Bijlage 1 interview met Ton.....	58
Bijlage 2 overleg met Robert .....	59

# Samenvatting

Topicus Onderwijs is een tak van Topicus dat zich bezighoudt met het maken van software voor het basisonderwijs, voortgezet onderwijs en middelbaar beroepsonderwijs. Tijdens het afstuderen wordt er gewerkt aan de Eduarte Studentenapp, dit is een onderwijsinformatiesysteem voor studenten op MBO scholen. De studenten kunnen in de app de resultaten, agenda, berichten, werklijst en personalia inzien. Ook kunnen studenten zich absentmelden via de app mits de student 18 jaar of ouder is en de school deze instelling heeft geactiveerd. In dit verslag wordt beschreven hoe de Eduarte Studentenapps (iOS en Android) kunnen worden omgebouwd naar één app met behulp van een hybride framework.

## Probleem

Topicus heeft recent Eduarte overgenomen van Iddink Group en wil de verouderde studentenapp herontwikkelen. Om dit te realiseren moet er een nieuw framework gezocht worden. Het doel van dit afstudeerproject is onderzoeken welk framework het meest geschikt is om de nieuwe Eduarte Studentenapp in te ontwikkelen. Hiervoor is de volgende hoofdvraag opgesteld: *“Wat is het meest geschikte hybride framework om de Eduarte applicatie toekomstbestendig in te herontwikkelen?”*.

## Onderzoek

Om een antwoord te kunnen geven op de hoofdvraag zijn er verschillende deelvragen opgesteld die helpen bij het beantwoorden van deze vraag. Er is onderzoek gedaan naar verschillende frameworks, advies gegeven op ontwerpkeuzes en een prototype ontwikkeld. Het onderzoek is uitgevoerd door middel van library research. Al deze onderdelen zorgen ervoor dat de onderzoeksvraag wordt beantwoord.

## Resultaat

Uit het onderzoek bleek dat Flutter het meest geschikte framework is voor de Eduarte Studentenapp. Ter ondersteuning is er een prototype gemaakt in het gekozen framework. Op basis hiervan wordt aanbevolen om de nieuwe studentenapp te ontwikkelen in Flutter. Eventueel vervolgonderzoek zou zich kunnen richten op verschillende extra functies die mogelijk zijn binnen het framework.

# 1. Inleiding

## 1.1 Topicus

Topicus is in 1998 opgericht door Harry Romkema en Leo Essink. Het bedrijf start in Enschede als een adviesbureau en lanceert in 2003 een applicatie voor het basisonderwijs genaamd ParnasSys. In de loop van de jaren groeit Topicus steeds sneller en in 2008 hebben ze 104 medewerkers.



*Figuur 1 - Topicus Eduarte gebouw*

In 2013 splitst Topicus af van 18 cellen naar drie divisies: Finance, Onderwijs en Zorg, later kwam hier nog een afdeling bij genaamd Sociaal domein. Tegenwoordig is Topicus een bedrijf dat marktleider is in verschillende sectoren (Topicus, 2023).

## 1.2 Eduarte

Eduarte is een onderwijsinformatiesysteem voor het MBO. In deze applicatie kunnen studenten, docenten, administratie en applicatiebeheerders informatie zien die ze nodig hebben tijdens hun opleiding of werk. Studenten kunnen in Eduarte overzichtelijk zien wat de examenuitslagen zijn, wat er op het huidige rooster staat, welke lessen er zijn uitgevallen en informatie over stages. Voor docenten is Eduarte een online omgeving waar al het materiaal staat om goed les te kunnen geven. Hierbij kun je denken aan studieplanning, lesmateriaal, opdrachten, resultaten en communicatie.

## 1.3 Aanleiding

Sinds 1 oktober heeft Topicus de rechten over Eduarte gekocht van Iddink Group. In het begin van 2023 is er een nieuw team opgezet dat de applicatie gaat verbeteren en onderhouden. Eduarte wordt dagelijks door veel scholen gebruikt, hierdoor is het niet wenselijk dat de app er verouderd uit ziet. Hiernaast is de code verouderd en slecht onderhouden waardoor het op dit moment rendabeler is om de huidige applicatie te herbouwen. Het huidige Eduarte onderwijsinformatiesysteem bestaat nu uit meerdere



applicaties waardoor het lastig is om deze te onderhouden en ook meer kosten met zich mee brengt.

## 1.4 Probleemstelling

De huidige studentenapp bestaat nu uit twee applicaties, Android en iOS. De apps bestaan uit Java(Android) en Swift(iOS). Op het moment dat de apps gelanceerd werden was het de enige manier om applicaties te ontwikkelen voor beide platforms.

Tegenwoordig zijn er andere manieren om dit probleem op te lossen. Het aanpassen van twee apps is voor Topicus geen wenselijke optie meer. De strategie van Eduarte gaat richting mobile first waardoor de app een grotere rol zal krijgen binnen het bedrijf. Hierdoor zullen meer nieuwe functionaliteiten moeten worden toegevoegd. Naast dat de app slecht te onderhouden is, zou dan ook kennis in huis moeten komen om native apps te bouwen.

Het probleem waar Topicus momenteel tegenaan loopt, is het kiezen van het juiste framework. De apps moeten herontwikkeld worden zodat ze allebei uit één codebase bestaan. De herontwikkeling hangt af van verschillende factoren. Er moet onderzocht worden of de huidige functionaliteiten blijven werken. Ook moeten de mogelijkheden voor toekomstige uitbreidingen onderzocht worden, het gekozen framework mag namelijk niet de toekomstige functionaliteiten beperken.

Een mooie bijkomstigheid is dat Topicus ook de portalen wil uitfaseren in de toekomst en dat nu mogelijk wordt door het in de webversie van de app te ontwikkelen. Het is daarom belangrijk dat er zorgvuldig een framework uitgezocht wordt die de ontwikkelingskosten terugbrengt en toekomstbestendig is.

## 1.5 Doelstelling

Het doel van dit afstudeerproject is om duidelijk in kaart te brengen welk framework het beste past bij de nieuwe Eduarte Studentenapp. Er wordt onderzoek gedaan naar verschillende eisen vanuit Topicus, op deze onderzoeken wordt advies uitgebracht richting Topicus.

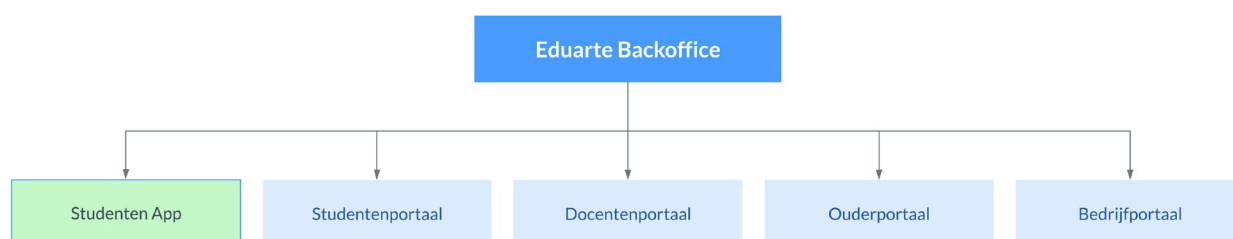
Er wordt een prototype opgeleverd met verschillende functionaliteiten om aan te tonen dat het framework ook echt gebruikt kan worden voor de nieuwe app. Tijdens het

realiseren van het prototype wordt er ook advies gegeven over verschillende keuzes die gemaakt zijn tijdens het ontwikkelen, bijvoorbeeld welke packages gekozen zijn en welke andere mogelijkheden er op dat gebied beschikbaar zijn. Uiteindelijk kan op basis van het onderzoek en het werkende prototype met zekerheid vastgesteld worden dat het framework aan de eisen voldoet en bij de nieuwe Eduarte Studentenapp past.

## 1.6 Scope

Het Eduarte systeem bestaat uit een backoffice, portalen en tot nu toe één mobiele app. De mobiele app is een verlengde van het studentenportaal, dit betekent dat de app een deel van de functionaliteiten van het studentenportaal bevat. Het project dat uitgevoerd wordt in opdracht van Topicus richt zich alleen op de Eduarte Studentenapp. De applicatie werkt op Android en iOS.

In de onderstaande afbeelding is te zien hoe Eduarte is opgebouwd uit verschillende lagen. In het groen is weergegeven waar dit project zich op richt.



*Figuur 2 - Project scope*

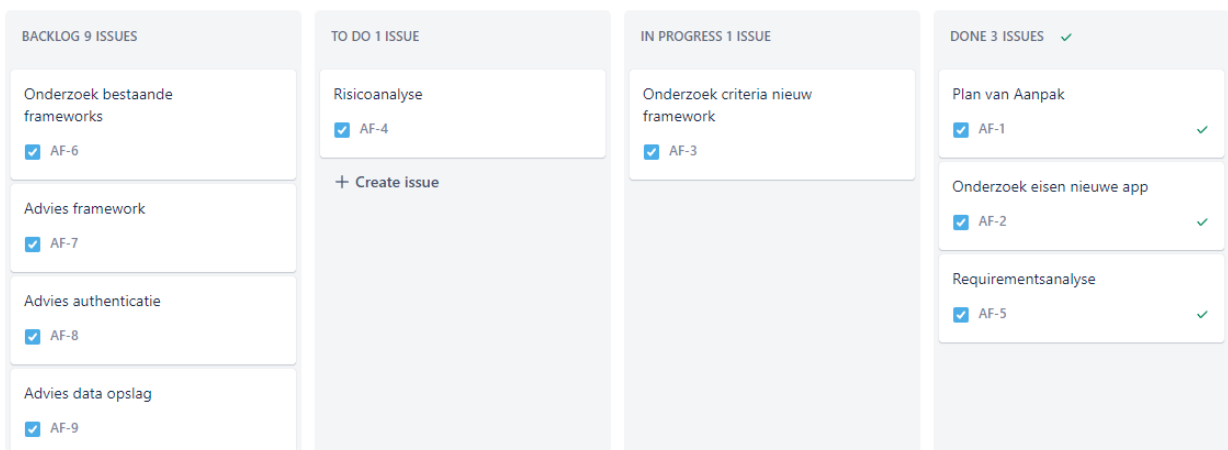
## 2. Projectaanpak

### 2.1 Proces

Om het project gestructureerd te laten verlopen is er gekozen voor de Scrum methode (Wardt, n.d.). In principe wordt het project individueel uitgevoerd maar er is wel een dagelijkse stand-up met het team. In de afstudeerperiode wordt het team beïnvloed door ontwerpkeuzes die tijdens het afstudeerproces zijn gemaakt. Deze worden elke dag besproken in de stand-up waardoor er een samenwerking ontstaat tussen het afstudeerproject en de huidige ontwikkeling van de Eduarte Studentenapp. Na elke sprint(twee weken) wordt er een demo gegeven van het prototype aan het Davo team. In de toekomst is het de bedoeling dat er door teamleden een code review wordt uitgevoerd.

### 2.2 Werkwijze

In de beginfase van het project is er een planning opgesteld, uit deze planning worden verschillende issues opgemaakt die uitgevoerd moeten worden tijdens het project. Deze issues worden in de backlog van een scrumboard gezet. Aan het begin van elke sprint worden er issues in "To Do" gezet. Wanneer er gewerkt wordt aan een issue wordt deze in "In Progress" gezet en als een issue klaar is in "Done". Wanneer een issue klaar is wordt deze getest om de kwaliteit te behouden. Om aan te tonen hoe dit er precies uit ziet, is er in de onderstaande foto een momentopname van een sprint getoond. Op deze manier kan een sprint overzichtelijk en succesvol worden uitgevoerd.

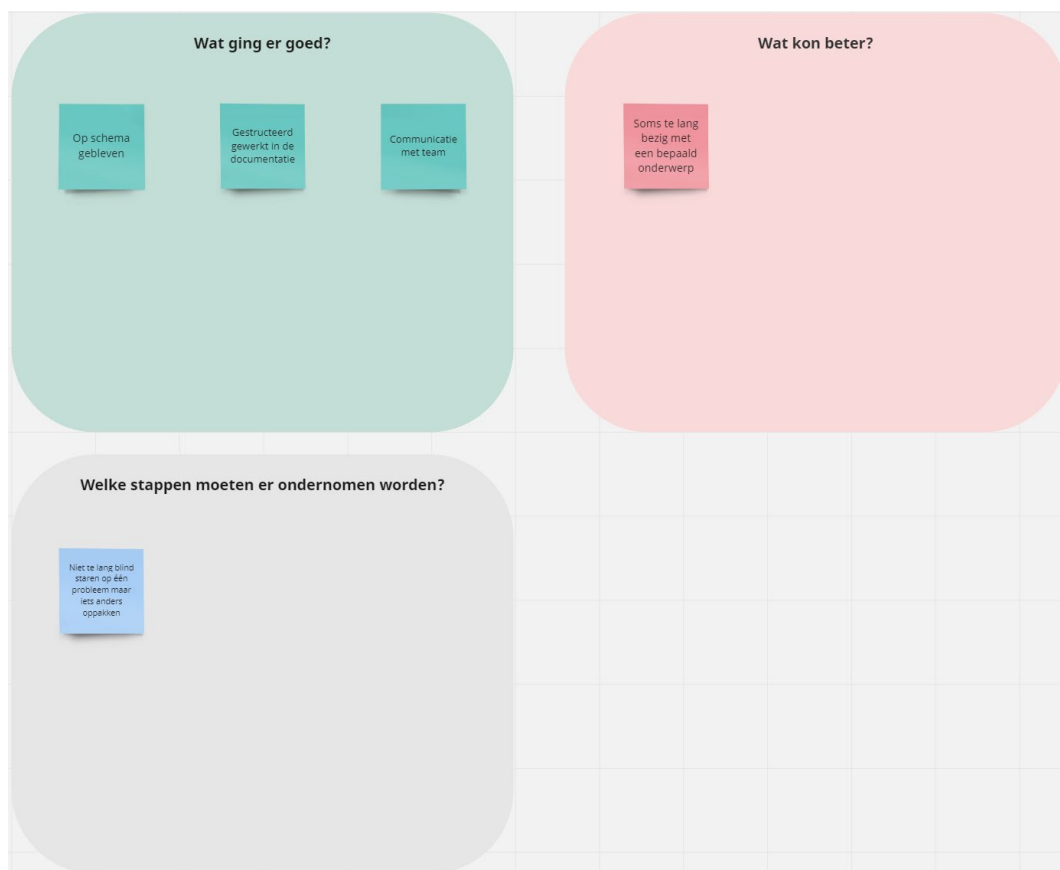


Figuur 3 - Momentopname werkwijze

## 2.3 Retrospective

Aan het eind van elke sprint wordt er een retrospective gehouden (Retrospective Scrum, n.d.). In deze retrospective wordt er ingegaan op wat er goed ging, wat beter kan en welke acties er ondernomen moeten worden. Voor dit onderdeel wordt er gebruikgemaakt van de What Went Well methode (Easyretro, n.d.). Er zijn verschillende methodes die gebruikt kunnen worden voor een retrospective maar omdat er bij Topicus al gebruikgemaakt wordt van deze manier is er voor deze methode gekozen. Tijdens de retrospective wordt er gekeken of er aan de actiepunten voldaan is en of de onderwerpen die beter konden ook echt verbeterd zijn. Het doel van de retrospective is om fouten die zijn gemaakt tijdens de vorige sprint te verbeteren en positieve punten voort te zetten.

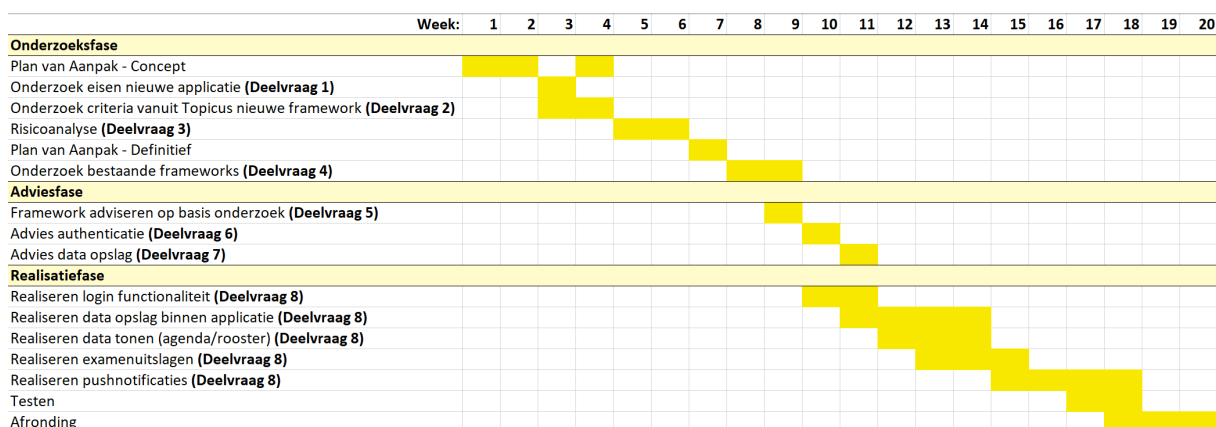
In de onderstaande foto wordt een voorbeeld getoond van een retrospective die tijdens dit afstudeerproject is gemaakt.



*Figuur 4 - Scrum retrospective*

## 2.4 Planning

Het project wordt uitgevoerd tussen 13 februari 2023 en 7 juli 2023. In onderstaande planning worden de verschillende fases van het project weergegeven. De planning is opgedeeld in fases om overzicht en controle te krijgen over de loop van het project. Op deze manier kan er snel gehandeld worden op het moment dat er van de planning afgeweken wordt.



Figuur 5 - Planning afstudeerproject

## 2.5 Tools

Tijdens het afstudeerproject zijn verschillende tools gebruikt. In de onderstaande tabel zijn deze tools opgesteld met bijbehorende functies.

Tool	Functie
Visual Studio Code	Realiseren app
Postman	Testen API calls
Android Studio	Onderzoeken oude applicatie
Teams	Stand-up en meetings binnen Topicus
Slack	Schriftelijk overleg met Topicus Onderwijs
Jira	Kanban bord
Miro	Kanban retrospective

Tabel 2 - Tools

De tools die gebruikt worden zijn voornamelijk aangeraden door Topicus omdat ze deze al gebruiken. De keuze voor de IDE mag zelf bepaald worden, er is voor Visual Studio Code gekozen omdat de meeste medewerkers in het team dit ook gebruiken en gratis is.

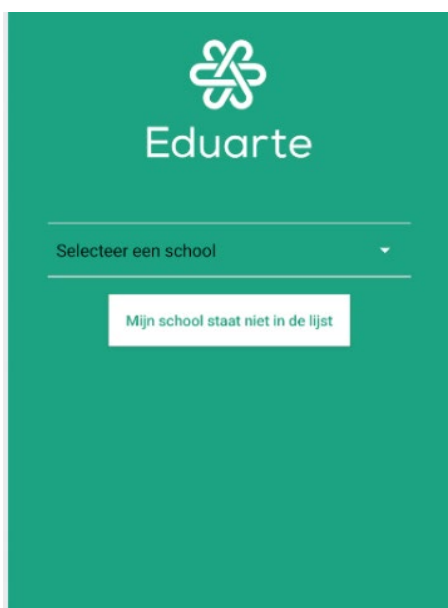
# 3. Onderzoek

## 3.1 Vooronderzoek

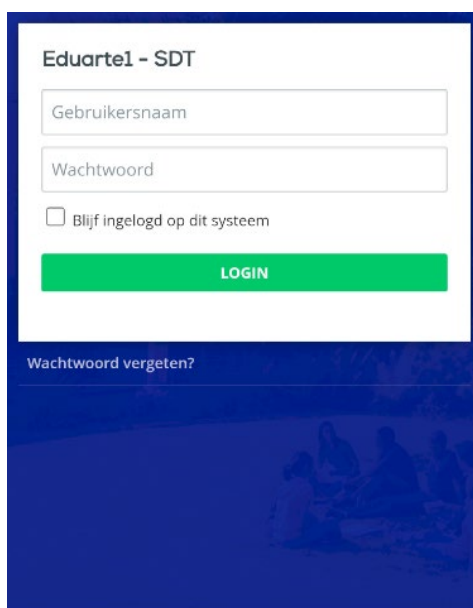
Er is een vooronderzoek uitgevoerd om de hoofd- en deelvragen in kaart te brengen. In dit onderzoek is een demoversie van de huidige applicatie opgezet om te kijken over welke functionaliteiten de applicatie beschikt. Het doel van het vooronderzoek is om vooral duidelijk te krijgen hoe het inlogproces verloopt en er wordt gekeken welke API calls er uitgevoerd worden die naar de juiste data van de student wijzen. In dit vooronderzoek wordt ook in kaart gebracht wat het ontwerp van de app is.

## 3.2 Huidige situatie

Het huidige systeem wordt gebruikt door MBO studenten. De studenten kunnen inloggen met hun schoolaccount en komen zo in een omgeving waar ze het rooster, toets uitslagen, examen uitslagen, personalia en berichten kunnen bekijken. Ook kunnen ze via de app, als ze boven de 18 zijn en dit geactiveerd is vanuit de school, een absentiemelding doen als ze door omstandigheden niet aanwezig kunnen zijn. Om een beeld te krijgen hoe de flow werkt in de app wordt aan de hand van een paar screenshots getoond hoe dit er in de huidige applicatie uit ziet:



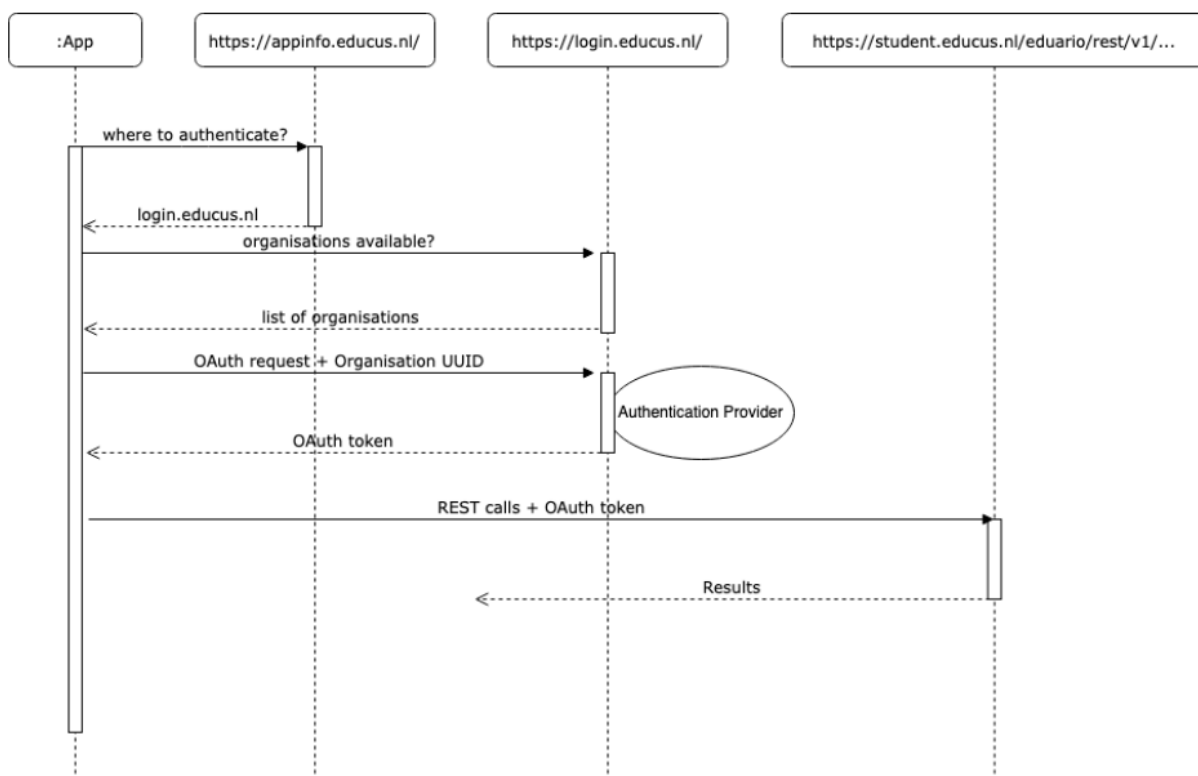
*Figuur 6 - Eduarte school keuzescherf huidige app*



*Figuur 7 - Inlogpagina school huidige app*

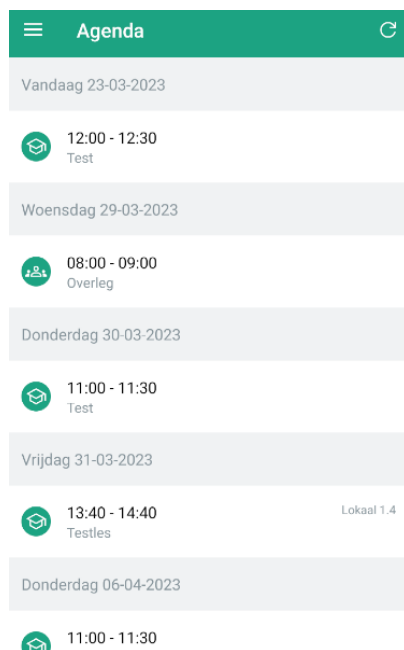
Om erachter te komen hoe het inlog proces precies werkt, is er door middel van library research onderzocht wat er gebeurt op het moment van inloggen. In de eerder opgezette demo versie van de huidige app is er via debugging gekeken welke calls er worden uitgevoerd. Omdat er een test account beschikbaar is vanuit Topicus, is de hele flow te reproduceren zoals een echte student.

Na het onderzoek is er een sequence diagram opgesteld die de flow van het inloggen in kaart brengt. In de onderstaande foto is te zien hoe dit in zijn werk gaat.

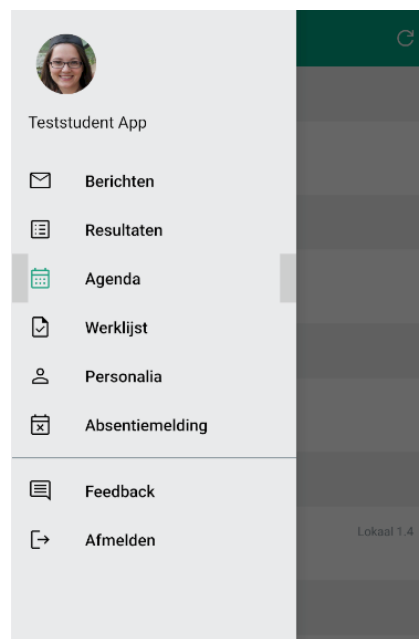


*Figuur 8 - Login proces sequence diagram*

Op het moment dat er ingelogd is komt de student standaard op de agendapagina. Hier is voor gekozen omdat de app het meest gebruikt wordt om op het rooster te kijken. De data wordt opgehaald met een token die is verkregen tijdens het inloggen. Op deze manier kan de app gevuld worden met data die voor de student bestemd is. Wanneer de student uitlogt wordt de token weer verwijderd.



*Figuur 9 - Agendapagina huidige app*



*Figuur 10 - Menu huidige app*

De huidige applicatie bestaat uit nog meer schermen dan bovenstaande figuren, maar omdat de andere schermen ook data laten zien van de student is dit te vergelijken met de agendaweergave en is het niet nodig om op de andere schermen in te gaan. In de onderstaande afbeelding worden alle functionaliteiten van de app getoond:



*Figuur 11 - Functionaliteiten huidige app*

Nu er een aantal schermen bekeken zijn in de huidige app kunnen er aan de hand van deze informatie deelvragen opgesteld worden.



## 3.3 Hoofd- en deelvragen

### Hoofdvraag

Wat is het meest geschikte hybride framework om de Eduarte applicatie toekomstbestendig in te herontwikkelen?

### Deelvragen

1. Welke eisen stelt Topicus aan een nieuwe app?
2. Welke eisen stelt Topicus aan een toekomstig framework?
3. Welke risico's neemt het ontwikkelen van de app in een nieuw framework met zich mee?
4. Wat zijn de belangrijkste aspecten waar het nieuwe framework aan moet voldoen om dezelfde functionaliteiten als de huidige Eduarte Studentenapp te bieden?
5. Welk framework kan ik Topicus het beste adviseren?
6. Hoe kan ik de authenticatie zo veilig mogelijk maken in het nieuwe framework?
7. Op welke manier kan het beste veilig data opgeslagen worden binnen het gekozen framework?
8. Welke functionaliteiten moeten er in het prototype zitten om aan te tonen dat het framework voldoet aan de eisen?

## 3.4 Welke eisen stelt Topicus aan een nieuwe app?

### 3.4.1 Werkwijze

Om vast te stellen welke eisen Topicus stelt aan een nieuwe applicatie moet er eerst onderzocht worden wie de stakeholders zijn. Samen met de bedrijfsbegeleider is er gekeken naar medewerkers binnen Eduarte die invloed hebben op het product. Met deze medewerkers zal een interview gehouden worden of overlegd worden om zo de eisen vast te stellen. Wanneer deze eisen vastgesteld zijn kan er een requirementsanalyse opgesteld worden, de requirements worden hier volgens de MoSCoW methode (Mulder, 2017) geprioriteerd. In de requirementsanalyse worden ook de acceptatiecriteria opgesteld waaraan het requirement moet voldoen om geaccepteerd te worden.

### 3.4.2 Stakeholders

De Eduarte Studentenapp is een product dat in de Eduarte omgeving bestaat. Zo zijn er meerdere stakeholders maar uiteindelijk is de product manager (Ton) van Eduarte de stakeholder die het meeste invloed heeft. Ton representeert de klanten en is goed op de hoogte wat er speelt in de markt en bij scholen. De andere stakeholders zijn het Davo team en aanbestedingen maar deze zijn in dit geval niet van toepassing omdat de huidige app wordt nagemaakt. Omdat het niet gebruikelijk is dat de requirements na één interview vaststaan, zijn er ook verschillende overleggen geweest met medewerkers van Topicus.

### 3.4.3 Resultaat

Uit het interview en verschillende overleggen met medewerkers (Bijlage 1 en 2) is een lijst met requirements opgesteld. Uit het interview met Ton kwam naar voren dat de nieuwe app dezelfde functionaliteiten moet bevatten als de oude app. In de bijlage "Requirementsanalyse" is een uitgebreidere versie te vinden waar de acceptatiecriteria voor elk requirement in behandeld wordt. In de onderstaande tabellen staan de requirements voor de nieuwe Eduarte Studentenapp. Aan de hand van de kolom genaamd "prototype" wordt aangegeven welk van deze requirements in het prototype uitgewerkt gaan worden.

## Functionele requirements

#	Requirements	MoSCoW	Prototype
<b>F1</b>	De student moet een lijst te zien krijgen van verschillende scholen waar hij/zij mee kan inloggen. <b>Acceptatiecriteria</b> <ul style="list-style-type: none"> <li>Lijstweergave van verschillende scholen</li> <li>Instructies andere pagina wanneer school niet in lijst staat</li> </ul>	Must	Ja
<b>F2</b>	De student moet kunnen inloggen met zijn/haar school account.	Must	Ja
<b>F3</b>	De student moet zijn/haar agenda kunnen zien.	Must	Ja
<b>F4</b>	De student moet per afspraak meer detail kunnen zien op een detailpagina	Must	Ja
<b>F5</b>	De berichten van de student kunnen ingezien worden op de berichten pagina.	Must	Ja
<b>F6</b>	De student moet de toets uitslagen kunnen zien.	Must	Ja
<b>F7</b>	De student moet de examen uitslagen kunnen inzien.	Must	Ja
<b>F8</b>	De student moet een werkljst overzicht zien waar hij/zij opdrachten en deadlines in kan zien.	Must	Nee
<b>F9</b>	De persoonsgegevens van de student kunnen ingezien worden op de personalia pagina	Must	Ja
<b>F10</b>	De persoonsgegevens van de student kunnen worden aangepast	Must	Nee
<b>F11</b>	De absentie van de student kan worden doorgegeven via de app	Must	Nee
<b>F12</b>	De student kan feedback geven op de app.	Must	Nee
<b>F13</b>	De student kan in een hamburger navigeren door de app	Must	Ja
<b>F14</b>	De student moet zich kunnen uitloggen in de app	Must	Ja
<b>F15</b>	Er kunnen andere thema's ingesteld worden in de app	Could	Ja
<b>F16</b>	Er kunnen pushnotificaties naar de student gestuurd worden	Could	Ja

Tabel 3 - Functionele requirements app

## Niet-functionele requirements

#	Requirements	MoSCoW	Prototype
<b>NF1</b>	Het systeem moet via OAuth2 in kunnen loggen	Must	Ja
<b>NF2</b>	Het systeem moet data kunnen ophalen via een API call	Must	Ja
<b>NF3</b>	Het systeem moet data lokaal kunnen opslaan	Must	Ja
<b>NF4</b>	Het systeem moet werken op Android en iOS	Must	Nee
<b>NF5</b>	Het systeem moet de native functies van de telefoon gebruiken	Must	Nee
<b>NF6</b>	Het systeem moet offline gebruikt kunnen worden	Must	Ja

Tabel 4 - Niet-functionele requirements app

### 3.4.4 Conclusie

Uit het resultaat kan geconcludeerd worden dat de app vooral moet werken zoals de huidige app. De requirements hoeven naderhand niet gevalideerd te worden omdat deze precies hetzelfde zijn als de huidige app. Er zijn wensen dat er andere thema's ingesteld kunnen worden per school en pushnotificaties gestuurd kunnen worden, maar dit zijn de enige veranderingen ten opzichte van de huidige situatie.

Niet alle functionaliteiten zullen gerealiseerd worden in het prototype omdat dit niet haalbaar is binnen de gegeven tijd. De uitwerking hiervan is te vinden in hoofdstuk 3.11.

## 3.5 Welke eisen stelt Topicus aan een toekomstig framework?

### 3.5.1 Werkwijze

Buiten de eisen die bij een app horen zijn er ook eisen die bij een framework horen. Om deze eisen in kaart te brengen moet er met de bedrijfsbegeleider gekeken worden naar onderwerpen die van belang zijn bij een nieuw framework voor Eduarte. Ook moet er, door middel van een interview of overleg, met medewerkers die buiten het team werkzaam zijn gekeken worden naar deze eisen. In de requirementsanalyse bijlage worden deze eisen opgesteld als requirement volgens de MoSCoW methode.

### 3.5.2 Stakeholders

De stakeholders voor dit onderdeel zijn dezelfde belanghebbenden als bij de vorige deelvraag. Met Ton (Product Manager), Robert (Manager Software Development) en het ontwikkelteam Davo zijn, net als voor de eisen van de app, een interview en verschillende overleggen gehouden over de eisen van het toekomstige framework.

### 3.5.3 Resultaat

In de onderstaande tabel is een lijst gemaakt van de requirements waar het framework aan moet voldoen. Deze lijst is voorgehouden aan de stakeholders en geverifieerd.

#	Requirements	MoSCoW
<b>NF7</b>	Het framework moet een hybride ontwikkel methode beschikbaar stellen	Must
<b>NF8</b>	De applicatie gemaakt in het nieuwe framework moet makkelijk uit te breiden zijn	Must
<b>NF9</b>	Het framework moet niet meer kosten dan de huidige native implementatie	Must
<b>NF10</b>	Het framework moet een goed toekomstperspectief hebben (minimaal 3 jaar)	Must
<b>NF11</b>	Het framework moet ondersteund worden door duidelijke documentatie	Must

Tabel 5 - Niet-functionele requirements framework

### 3.5.4 Conclusie

Tijdens de gesprekken zijn er meerdere requirements naar voren gekomen waar het framework aan moet voldoen. In eerste instantie zijn er niet heel veel nieuwe functionaliteiten nodig omdat er zo snel mogelijk een MVP (Minimum Viable Product) uitgebracht moet worden. Maar het moet het niet zo zijn dat er in de nabije toekomst weer een nieuw framework uitgezocht moet worden omdat gewenste functionaliteiten vanuit Topicus niet mogelijk zijn.

## 3.6 Welke risico's neemt het ontwikkelen van de app in een nieuw framework met zich mee?

### 3.6.1 Werkwijze

Om Topicus zo goed mogelijk voor te bereiden op eventuele risico's die zich voor kunnen doen bij het wisselen van framework zijn er verschillende stappen ondernomen. Als eerste worden er risico's geïdentificeerd die een rol spelen bij deze overstap, deze risico's worden bepaald met het Davo team. Daarna worden de risico's beoordeeld door de kans en impact te bepalen, hier wordt ook een toelichting op gegeven. Wanneer deze stappen gedaan zijn kan er een behandelplan per risico worden opgesteld in het geval dat dit risico zich voordoet tijdens de ontwikkelfase.

### 3.6.2 Resultaat

De resultaten uit de bijlage 'Risicoanalyse' worden in onderstaande tabel samengevat. Voor uitgebreide informatie kan de bijlage worden geraadpleegd.

Risico	Kans	Impact	Gevolg	Behandelplan
Het framework ondersteund geen OAuth2.	1	5	De gebruikers kunnen niet inloggen via de gewenste functionaliteit.	Er moet voor het kiezen van een framework onderzocht worden of dit mogelijk is. Wanneer dit niet het geval is, moet er overwogen worden om een andere inlogmethode te gebruiken of een ander framework te kiezen.
De native functies van het apparaat werken niet in het nieuwe framework.	2	4	De native functionaliteiten kunnen niet gebruikt worden in de nieuwe applicatie.	Tijdens het onderzoek moet er gekeken worden naar de native functionaliteiten die het framework aan

				kan bieden en welke functionaliteiten uiteindelijk nodig zijn voor de app.
Packages/functies werken op één platform wel en op de andere niet,	3	2	De applicatie werkt goed op een toestel maar op een ander niet waardoor de kwaliteit van het product niet goed is	Op het moment van ontwikkelen wordt er een keuze gemaakt voor een bepaalde package, voordat deze keuze gemaakt wordt moet er onderzocht worden of alle platformen die van toepassing zijn ondersteund worden.
Ontwikkelen duurt langer dan verwacht,	3	1	Het ontwikkelen van de app in een nieuw framework neemt te veel tijd op waardoor de kosten te hoog worden.	Wanneer het ontwikkelen te lang duurt omdat er te weinig kennis is of omdat de documentatie niet duidelijk genoeg is, moet er overwogen worden om extra kennis in te huren vanuit andere teams van Topicus.
Ontwikkelaar stopt met onderhouden gekozen framework,	1	5	De applicatie wordt niet meer ondersteund en krijgt geen updates meer.	Op het moment dat de ontwikkelaar stopt met het onderhouden van het framework moeten er meteen stappen ondernomen worden om een nieuw framework te kiezen



				en de app opnieuw te herontwikkelen.
Performance issues in nieuw framework.	2	3	Framework werkt slecht op telefoons waardoor de nieuwe app niet beter is dan de oude.	Wanneer de nieuwe app niet goed werkt moet er gekeken worden waar het probleem zich bevindt (bijv performance test).
Security issues in nieuw framework.	1	3	Framework bevat security problemen waardoor de veiligheid niet gewaarborgd kan worden.	Op het moment dat het framework security issues bevat moet er vanuit de backend een mogelijkheid zijn om de werking van de app te stoppen totdat het beveiligingslek verholpen is. Ook moeten er aan de backend kant extra beveiliging zijn zodat het probleem niet groter wordt.
Requirements veranderen waardoor framework niet meer geschikt is.	1	4	Requirements zijn misschien niet mogelijk om te ontwikkelen in gekozen framework.	Wanneer er een verandering plaats vindt in de requirements zal er een nieuw onderzoek gestart moeten worden of dit wel haalbaar is. Eventuele vervolg stappen zijn requirements aanpassen of schrappen.

Kosten framework hoger dan huidige implementatie.	4	1	Het nieuwe framework is minder rendabel ten opzichte van de huidige native implementatie	Op het moment dat de kosten van het framework niet acceptabel zijn voor Topicus zal er een ander framework gezocht moeten worden. In het onderzoek wordt hier aandacht aan besteed en daarom is dit vooraf bekend.
---	---	---	--	--

Tabel 6 - Samenvatting risicoanalyse

### 3.6.3 Conclusie

Op basis van deze risicoanalyse kan er geconcludeerd worden dat de meeste risico's een kleine kans hebben. De elementen waarbij de risico kans het grootst is, is de impact over het algemeen het laagst waardoor een eventueel risico geen groot gevolg heeft. Wanneer het behandelplan goed wordt uitgevoerd zijn de risico's opgelost en kan het systeem succesvol ontwikkeld worden.

## 3.7 Wat zijn de belangrijkste aspecten waar het nieuwe framework aan moet voldoen om dezelfde functionaliteiten als de huidige Eduarte Studentenapp te bieden?

### 3.7.1 Werkwijze

Om aan te tonen of het nieuwe framework het oude systeem kan vervangen moeten er verschillende criteria onderzocht worden. Het onderzoek wordt opgesteld in een onderzoeksdocument, deze is te vinden in de bijlage genaamd "Onderzoek frameworks". Om de requirements uit 3.4 en 3.5 te realiseren zijn er ook nog andere aspecten die invloed hebben op welk framework het meest geschikt is. Deze aspecten zijn overkoepelend en niet specifiek aan requirements van de app gekoppeld met uitzondering van de pushnotificaties.

### 3.7.2 Resultaat

Uit de volgende behandelde deelvragen zijn door middel van fieldresearch verschillende eisen naar voren gekomen waar het nieuwe framework aan moet voldoen:

- Welke eisen stelt Topicus aan een nieuwe app?
- Welke eisen stelt Topicus aan een toekomstig framework?

De punten die per framework onderzocht moeten worden door middel van library research zijn:

#### **Hybrid development**

In dit onderdeel wordt gekeken of het framework voor alle gewenste platformen beschikbaar is. Er wordt ook onderzocht in hoeverre het mogelijk is om de applicatie uit één codebase te laten bestaan.

#### **Schaalbaarheid**

In dit onderdeel wordt gekeken in hoeverre code hergebruikt kan worden tijdens de ontwikkeling, bijvoorbeeld door componenten. De tijd voor de ontwikkeling moet verbeterd worden ten opzichte van de ontwikkelingstijd die nu bij het onderhouden van Android en iOS geldt.

### **Kosten**

Een eis vanuit Topicus is dat het framework niet meer mag kosten dan de huidige native implementatie. Per framework wordt onderzocht wat de kosten zijn.

### **Toekomstperspectief**

Om uit te sluiten dat het toekomstige framework niet meer wordt ondersteund binnen aanzienlijke tijd moet er onderzocht worden hoe populair en hoeveel draagvlak er voor het framework is. Het framework moet nog minimaal drie jaar ondersteund worden. Het is belangrijk dat een framework door veel developers gebruikt wordt omdat dit de levensduur en beschikbare informatie (bijvoorbeeld probleem oplossingen) bevordert.

### **Kennis in huis**

In dit onderdeel wordt per framework gekeken of er al eerdere projecten mee zijn gemaakt. Dit om te onderzoeken of er ondersteuning van andere teams beschikbaar is tijdens het ontwikkelproces. Wanneer er één of meer ontwikkelteams binnen Topicus met het framework werken of gewerkt hebben, heeft dit de voorkeur.

### **Documentatie**

Omdat het framework dat gekozen gaat worden waarschijnlijk nieuw gaat zijn voor alle developers in het team is het noodzakelijk dat hier probleemloos mee te werken is. In het onderzoek moeten de verschillende documentaties met elkaar vergeleken worden. Hier wordt gekeken of de documentatie standaard functionaliteit zoals het installeren en opzetten van de app bevat, daarnaast wordt gekeken naar de kwantiteit.

### **Pushnotificaties**

Het toekomstige framework moet de mogelijkheid bieden om pushnotificaties te versturen op het moment dat er roosterwijzigingen of cijfers bekend zijn. Er moet daarom per framework gekeken worden of dit mogelijk is.

## **3.7.3 Conclusie**

Door de verschillende aspecten uit het eerdere resultaat per framework te onderzoeken, kan er geconcludeerd worden dat de nieuwe Eduarte Studentenapp dezelfde functionaliteiten zal bevatten als de huidige app. Uit het interview met Ton en eerder genoemde deelvragen is gebleken dat de beschreven requirements aanwezig moeten zijn

in het nieuwe framework. Daarnaast zijn er requirements vanuit Topicus benoemd die ze graag zouden terugzien in het nieuwe framework, bijvoorbeeld de pushnotificaties. Op basis van deze criteria kan een weloverwogen keuze worden gemaakt voor het meest geschikte framework.

## 3.8 Welk framework kan ik Topicus het beste adviseren?

### 3.8.1 Werkwijze

Om een zo compleet mogelijk advies te geven over het toekomstige framework voor Eduarte, moet het onderzoek dat is opgesteld in de deelvraag *“Wat zijn de belangrijkste aspecten waar het nieuwe framework aan moet voldoen om dezelfde functionaliteiten als de huidige Eduarte Studentenapp te bieden?”* uitgevoerd en gedocumenteerd worden. In het onderzoeksdocument (Onderzoek frameworks) wordt een conclusie opgesteld met de uitkomsten van het onderzoek. Op basis van deze conclusie kan een advies gegeven worden over het framework dat het beste past bij Eduarte.

### 3.8.2 Uitvoering

Er is een vooronderzoek gedaan naar de populairste frameworks. Met deze informatie wordt al een deel van alle beschikbare frameworks uitgesloten. Om aan de wensen van Topicus te voldoen m.b.t het toekomstperspectief zijn het vaak de grootste frameworks die ondersteund blijven worden door developers. Uit deze bron kan geconcludeerd worden dat React Native, Ionic, Xamarin, Flutter en Native Script het meest populair zijn in 2023 (Sogbesan, 2022). Cordova is ook een van de grootste maar deze wordt vanaf 2020 al niet meer ondersteund en daarom is deze niet meegenomen in het onderzoek.

Voorafgaand aan dit onderzoek kan er geconcludeerd worden dat basisfunctionaliteiten zoals data ophalen en gebruiken vanuit een backend, lokaal data opslaan en inlogfunctionaliteit in elk framework mogelijk zijn. Deze zullen daarom ook niet worden meegenomen in het onderzoek maar bij de ontwikkeling van het prototype worden deze aspecten wel meegenomen. De onderwerpen die zijn opgesteld in de eerdergenoemde deelvraag zijn door middel van library research onderzocht.

### 3.8.3 Resultaat

In het onderzoek zijn de eerder opgestelde criteria per framework onderzocht. Na het onderzoek is een MCA-tabel opgesteld (Janse, 2018). In de tabel zijn cijfers gegeven per onderwerp van het onderzoek.

	React Native	Ionic	Xamarin	Flutter	Native Script
Hybrid development	6	7	5	8	6
Schaalbaarheid	7	6	5	8	6
Kosten	10	10	7	10	10
Toekomstperspectief	8	4	0	8	6
Kennis in huis	0	2	0	2	0
Documentatie	7	5	9	8	4
Pushnotificaties	5	5	5	5	5
<b>Totaal</b>	<b>43</b>	<b>37</b>	<b>31</b>	<b>49</b>	<b>37</b>

Tabel 7 – MCA-tabel Framework

### 3.8.4 Conclusie

In de Multicriteria-analyse komt Flutter het best uit het onderzoek, vooral op het gebied van hybrid development, schaalbaarheid en toekomstperspectief stijgt Flutter boven de concurrerende frameworks uit. De doorslaggevende factor in vergelijking met React Native is de kennis in huis. Het is voor Topicus erg belangrijk dat ontwikkelteams kennis onder elkaar kunnen uitwisselen en elkaar kunnen ondersteunen waar dat nodig is. Kort geconcludeerd, Flutter is een vooruitstrevend framework met een goed toekomstperspectief dat aansluit op de wensen van Topicus. Daarom is het advies aan Topicus om Flutter te gebruiken voor de nieuwe Eduarte Studentenapp.

Om de hoofdvraag te kunnen beantwoorden moet blijken of Flutter ook in de praktijk de beste keuze is. Dit zal in de verdere deelvragen worden onderzocht.

## 3.9 Hoe kan ik de authenticatie zo veilig mogelijk maken in het nieuwe framework?

### 3.9.1 Werkwijze

De authenticatie binnen Eduarte is een belangrijk onderdeel omdat er met gevoelige studentgegevens gewerkt wordt. Het huidige systeem maakt gebruik van OAuth 2.0 en er is daarom gekozen om géén advies uit te brengen over het inlogmechanisme maar een advies over hoe dit het beste kan ontwikkeld kan worden in Flutter.

### 3.9.2 Onderzoek authenticatie

Om de gebruiker zo veilig mogelijk te laten inloggen wordt er gebruikgemaakt van OAuth 2.0. Wanneer OAuth 2.0 wordt gebruikt wordt de gebruiker buiten de applicatie naar een inlogsysteem gestuurd van een andere instantie. Hierdoor kan de gebruiker zien waar hij inlogt en zo ook de veiligheid controleren. Nadat de code van de huidige Android applicatie onderzocht is, zijn de volgende endpoints en headers naar voren gekomen:

Type	
Clientid	3cdf9da3-705a-4899-b3c9-393db1fdc8b7
Clientsecret	587c2c26-cead-4155-b19c-70a83042270a
Callback URL	http://student.eduarte.local/oauth-callback
Authorization URL	https://d-login.educus.nl/oauth2/authorize
Accesstoken URL	https://d-login.educus.nl/oauth2/token

Tabel 8 - Informatie app

De velden "clientid" en "clientsecret" worden vanuit de applicatie gebruikt om de app te verifiëren. Hierdoor kan een andere applicatie zich niet voordoen als legitiem en wordt de veiligheid gewaarborgd. In Flutter is het zo dat er in de callback URL de applicatiennaam meegegeven moet worden, zo weet de webview waar deze naar terug moet navigeren. Een webview is een browser die gebruikt kan worden vanuit de app om te navigeren op het internet. De callback ziet er als volgt uit:

nl.topicus.myapplication://http://student.eduarte.local/oauth-callback.

De server van Topicus is op dit moment niet afgesteld om dit te accepteren.

```
{"error_description": "Client authentication failed: Callback url does not match one of the registered values", "error": "invalid_client"}
```

Figuur 12 - Error backend callback URL



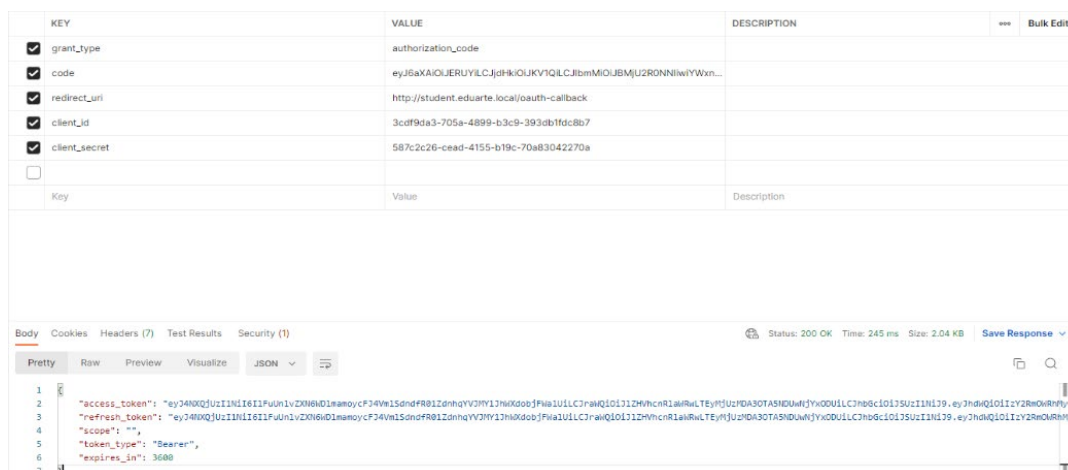
Er is een veiligheid filter aangebracht waardoor de server alleen een vaste URL accepteert, geen tekens voor of na de URL. Omdat de mogelijkheid er nu niet is om de server aan te passen is er een oplossing bedacht om hier omheen te werken.

Er wordt in plaats van een OAuth 2.0 package zelf een webview gestart. Via deze webview kan het login scherm getoond worden zonder de eerder genoemde applicatienaam mee te sturen. Op het moment van inloggen komt er van de server een URL terug met daarachter een token. Met deze token kan er vervolgens een accesstoken en een refreshtoken opgehaald worden. Vanuit de code kan er geluisterd worden naar wat er in de webview gebeurt. Op het moment dat de server een URL teruggeeft met de authorization token wordt deze opgevangen. Zoals eerder aangegeven wordt deze response gestuurd samen met een token, hieronder een voorbeeld van de reactie van de server.

```
http://student.eduarte.local/oauth-callback?
code=eyJ6aXAiOiJERUYiLCJjdHkiOiJKV1QiLCJlbmMiOiJBMjU2R0NNIiwiaWF0IjoiZGlyIn0..Vn_KmFSM6U2W02Pz.Zm25ekPri3UmPHYpQJuII8ARvAquz
fSPLgs849ioxp04MeFqoBgFoPBha_zcYij0yibWya5t7jyILEckbPQaz5RmboqTiuEhZ9lJNjzhs9myL8C6QaSqzT9BjyBRAP_maAJSBDejbqDvliaz24CVieM1
wZ063fAFozsXZjwHa7KkUPnc-BYlmrSH1KBqPU3tRlW1adHoSELPi3zdMLdI9Fb1P8gHl-s72PJy5Y5TIyFKGzBHUA01Z-QJ5vB7Vdhj4FKs25e4GoqQQhy-
8myYa89-
8XpK_Z12yZsne3s6yt8D9xowfz_axcfD64aIvA04h9Gahoc4hBtFSj1La_JGfnqOPgHj1E9gGTLmC0pw62G7e36QM59fxJgR2isISUavIovzCnqGad16eBHD28w
emprrWUNmZtkntppqoh5iJM-y4Rbyc1PTX5H_V0nOkqrTRNLAYVza0QVgkFZu8kuXnUkfsn9XWxzAuLUPfP1_1klcT8nxuCaouSXM-
4EaZJIdRIWV5HTEi2B4NI2fcx0ajIY4HrgGfPKC2hdwkjUybnj6zHegfkw3bQMazf2_vXH238tttPmNb5B75L_SoaEB270MRq_DVZwH5xOcyDLNSfhfjyk1Lch1C
94Hwu2qK5DWNxro0PGDm_bIxb_7ue91wkL6DYGQGMf0yY-f5bVX-pmvvsSDcRP4PTHK_wF0uuh3SyEgAsokmdY3eXvncKy11-
041nvJlLr6qgZH5g2pnrD7cY3P4yfl4vweINBmYKXTXSd8WpLc_euFZHLdW7aLgtNlZF1w775g1ZYNP1EgW3Xqha6b3x_bcnLddvtovf99hbbe12FLp8PKidBC
9fz6KNQHU_9-wM4CqwuPq5vzq0Bky_6.sc8wiwtwzfCo0GFVSWlmaq
```

**Figuur 13 - Voorbeeld authorization token**

Nu de authorization token is opgevangen, kunnen de accesstoken en refresh tokens worden opgehaald. Om dit te doen moeten er headers meegestuurd worden. In figuur 8 in hoofdstuk 3.2 is een sequence diagram afgebeeld hoe het inlog proces precies werkt. In onderstaande afbeelding is te zien hoe dit gedaan wordt en wat de server terug stuurt. Om een duidelijk beeld te geven wat er gebeurt is dit gedaan in Postman.



**Figuur 14 - Token request postman**

### 3.9.3 Conclusie advies

#### **Backend aanpassen**

Het probleem wat zich op dit moment voordoet, is dat de Eduarte backend geen callback accepteert met extra parameters eraan vast. Dit is wel nodig om met een OAuth 2.0 package te werken in Flutter. Daarom is het advies aan Topicus om zo snel mogelijk de backend te veranderen zodat deze ook werkt vanuit Flutter.

Na dit aan te hebben gegeven bij de bedrijfsbegeleider is hier meteen werk van gemaakt en is hier een issue van gemaakt dat momenteel wordt opgepakt door een ander team. Ook wordt de oplossing die bedacht is voor dit probleem gebruikt door Topicus voor de eerste versie van de app.

#### **Vervolg stappen**

Wanneer de backend is aangepast zijn er verschillende packages die gebruikt kunnen worden om in te loggen. Ik adviseer om de officiële package van Flutter te gebruiken genaamd "oauth2". Deze is uitgebracht door de ontwikkelaars van Flutter en krijgt daarom vaak een update. Dit is belangrijk omdat er met gevoelige inloggegevens van studenten wordt gewerkt.

## 3.10 Op welke manier kan het beste veilig data opgeslagen worden binnen het gekozen framework?

### 3.10.1 Werkwijze

Om een advies uit te brengen over de opslag binnen Flutter moet er eerst onderzocht worden welke mogelijkheden er allemaal beschikbaar zijn. Om hierachter te komen wordt er gekeken naar verschillende packages die beschikbaar zijn. In dit onderdeel moet rekening gehouden worden met de eis dat de app offline gebruikt moet kunnen worden. Wanneer er geen internet beschikbaar is, moet de opgeslagen data geraadpleegd kunnen worden.

### 3.10.2 Onderzoek data opslag

Het is belangrijk om de data lokaal op te slaan omdat live data niet altijd beschikbaar is, of de student heeft niet altijd een (stabiele) internetverbinding. Door het beschikbaar stellen van offline data kan de student altijd gebruik maken van de app. Op het moment dat het internet beschikbaar is zal de nieuwste data altijd voorrang hebben. Na een vooronderzoek om te kijken op welke manieren er data opgeslagen kan worden (Oleinik, 2021) zijn deze opties naar voren gekomen:

#### **Shared Preferences**

De Shared Preferences package (Flutter, sd) in Flutter is bedoeld om simpele data op te slaan binnen de app. Er wordt gebruikgemaakt van key-value pairs, dit houdt in dat data(value) opgeslagen kan worden aan de hand van een key. Met deze key kan er overal in de code de juiste data opgehaald worden. De package beschikt momenteel alleen over de opslag van een int, double, bool, String en List, er kunnen dus geen objecten of mappen worden opgeslagen.

#### **Flutter Secure Storage**

Flutter Secure Storage werkt in principe hetzelfde als de Shared Preference package zoals eerder besproken. Het is ook een key-value systeem waarmee via een key een waarde opgehaald kan worden. Het enige verschil tussen deze twee packages is dat Flutter Secure Storage gebruikmaakt van encryptie. Op Android wordt er AES-encryptie gebruikt en op iOS gebruiken ze Keychain, dat ook gebruik maakt van AES maar door Apple zelf ontwikkeld is.

## Hive

Ook Hive is een key-value data opslag methode. Dit systeem werkt met een zogenoemde "box", in deze box wordt de data opgeslagen. Op het moment dat er data opgeslagen moet worden, moet de box eerst geopend worden. Wanneer er zeker is dat de eerder opgeslagen data niet meer nodig is, kan de box gesloten worden. De data die in deze box zat wordt op dat moment verwijderd uit het geheugen. Hive ondersteunt String, int, Map, List, DateTime en Uint8List. Ook is het mogelijk om met een adapter een object te converteren van en naar binaire vorm (Dhakal, 2022).

## Dio cache interceptor

Een iets minder bekende methode is Dio Cache Interceptor. Op dit moment is ervoor gekozen om data op te halen van een API met Dio. Dit is een package dat automatisch foutmeldingen kan afvangen wanneer deze voorkomen, er kan per foutafhandeling bepaald worden welke actie er ondernomen moet worden. De cache interceptor is een uitbreiding op Dio, het zorgt ervoor dat alle data die op wordt gehaald automatisch gecached wordt. Ook kan er met deze package offline gebruikgemaakt worden van de data die eerder is opgehaald. De package ziet wanneer er wel of geen verbinding is en wisselt zo tussen opslag en live data.

### 3.10.3 Conclusie advies

Na de verschillende manieren onderzocht te hebben kan geconcludeerd worden dat de Dio Cache Interceptor de beste keus is voor dit project. Er wordt namelijk ook al gebruik gemaakt van normale Dio package bij het maken van de API calls.

Shared Preferences is een lokale opslag die niet heel erg veilig is, bij de andere is dit wel veilig. Offline gebruik is een belangrijk onderwerp voor de studentenapp, bij elke methode kan dit maar bij de Dio Cache Interceptor gaat dit automatisch. Bij Shared Preferences, Secure Storage en Hive is het niet mogelijk om in te stellen hoelang offline data beschikbaar mag zijn. In de Dio Cache Interceptor kan ingesteld worden hoelang offline data geldig mag zijn, er is ook een optie om de data te encrypten waardoor dit de veiligste opslag methode is.

## 3.11 Welke functionaliteiten moeten er in het prototype zitten om aan te tonen dat het framework voldoet aan de eisen?

### 3.11.1 Werkwijze

Om aan te tonen dat het framework kan voldoen aan alle eerder opgestelde eisen is ervoor gekozen om een prototype te maken. In dit onderdeel wordt er omschreven welke functionaliteiten dit zijn en op basis hiervan wordt het prototype gemaakt.

### 3.11.2 Prototype functionaliteiten

De Eduarte Studentenapp is niet groot en complex. De app bestaat voornamelijk uit data ophalen en tonen aan de student. Het complexe onderdeel zit dus vooral in het inloggen en de juiste data van de student opslaan en tonen. In onderstaande tabel zijn de functionaliteiten van het prototype beschreven.

Functionaliteit	Functioneel / Non-functioneel	MoSCoW
School keuze scherm	F	M
Inlog pagina in webview	F	M
Inlogmechanisme OAuth2.0	NF	M
Data opslag	NF	M
Offline gebruik	NF	S
Agenda weergave en detail	F	M
Berichten weergave en detail	F	C
Resultaten en detail	F	M
Personalia pagina	F	M
Light & Dark theme	F	C
Pushnotificaties	F	M

Buiten het inloggen is het ook van belang dat de data van de student veilig wordt opgeslagen. Ook moet het mogelijk zijn om de app offline te gebruiken binnen een bepaalde periode (bijvoorbeeld 7 dagen na de laatste inlog). Omdat er nog wensen voor pushnotificaties en thema's zijn, is er gekozen om dit ook in het prototype te verwerken.

### 3.11.3 Conclusie

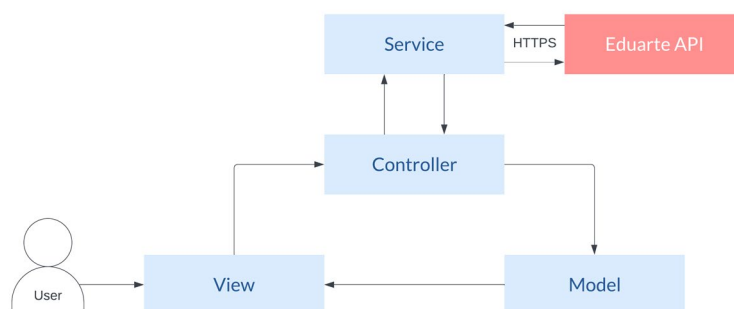
Door de verschillende bovenstaande functionaliteiten te realiseren in het nieuwe framework kan er geconcludeerd worden dat het framework voldoet aan de eisen die eerder opgesteld zijn in hoofdstuk 3.4. De functionaliteiten die niet zijn uitgewerkt, zijn met dezelfde methodiek en logica te realiseren. Hierdoor is het ontwikkelde prototype een goede weerspiegeling van de uiteindelijk te realiseren studentenapp.

# 4. Prototype

In dit hoofdstuk wordt ingegaan op de belangrijkste onderdelen van het prototype. Het prototype is gemaakt aan de hand van het voorgaande onderzoek en het advies. Het totale prototype bevat meer schermen maar zijn technisch hetzelfde als de onderdelen die hier besproken worden. In hoofdstuk 2 is de aanpak van het project al een keer behandeld. Het prototype is in sprints uitgevoerd en functionaliteiten zijn in issues opgesteld. Na elke issue wordt de applicatie getest om de kwaliteit te waarborgen.

## 4.1 Ontwerppatroon

Er is gekozen om het project op te bouwen door middel van het ontwerppatroon MVC+S (Model-View-Controller-Service) (Gürel, 2022). De Models worden gebruikt om een betekenis te geven aan de data door relaties tussen de opgehaalde data toe te voegen. De modellen zijn grotendeels gegenereerd door het resultaat van een API call in een "JSON to Dart Converter" te stoppen, er zijn verschillende websites en packages die dit kunnen. De Views zijn alle Widgets en pagina's in de Flutter app, hier wordt data getoond vanuit de modellen. In de Controllers worden de API calls gedefinieerd die via de Service uitgevoerd moeten worden, de controller verwerkt de data daarna in de modellen. De Service zorgt dat er data opgehaald kan worden van de Eduarte API. In deze Service worden ook fouten afgehandeld die zich kunnen voordoen bij het ophalen van data. Er is voor MVC gekozen omdat de app op deze manier overzichtelijker en beter te onderhouden is. Door het scheiden van het datamodel(model), datapresentatie(view) en applicatielogica(controller) wordt de leesbaarheid en herbruikbaarheid van de code bevorderd (Sofco, sd). In de onderstaande afbeelding is te zien hoe het MVC+S ontwerppatroon in het prototype werkt:

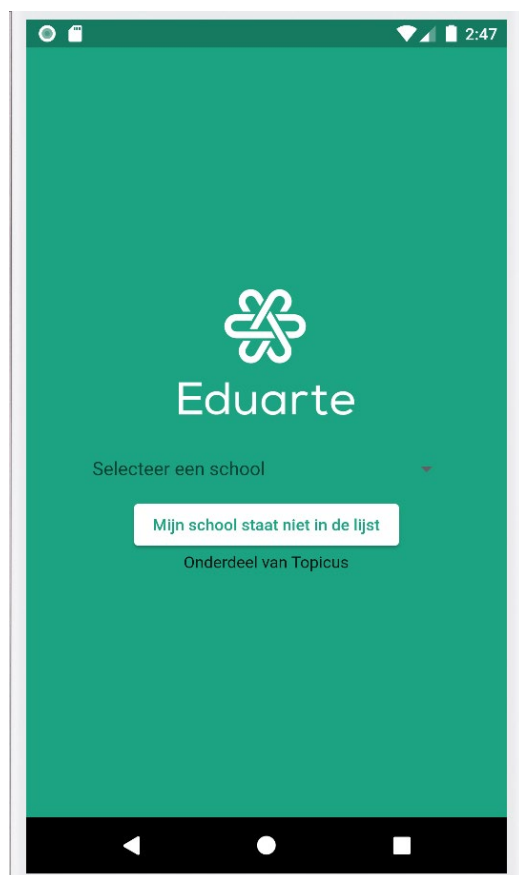


*Figuur 15 - MVC+S in Flutter*

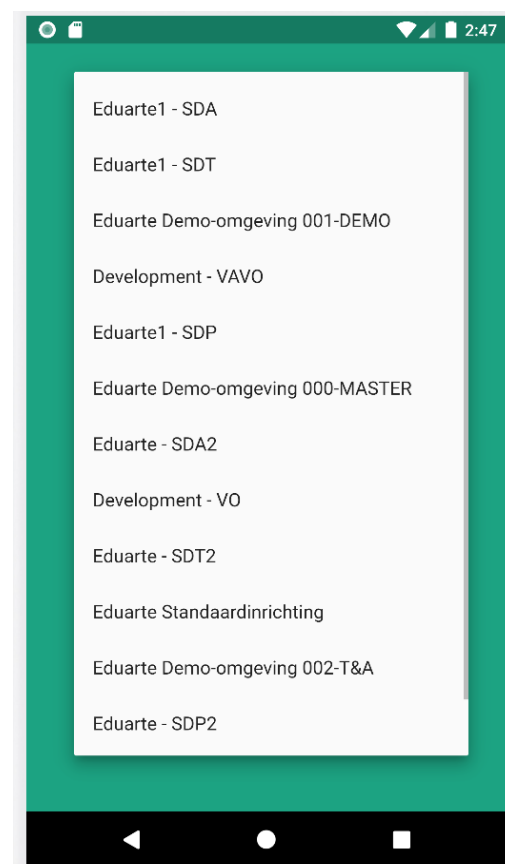
## 4.2 Keuze scherm school

Op het moment dat de applicatie gestart wordt kan er gekozen worden uit meerdere scholen. Via de Eduarte API kunnen er scholen opgehaald en weergegeven worden. Deze lijst wordt opgehaald vanaf het endpoint "<https://d-login.educus.nl/app/endpoints>".

Wanneer er een school gekozen is, wordt het bijbehorende OAuth login proces gestart. Dit proces kan per school verschillen, om deze reden zit er een organisatie id aan de school vast. Dit wordt meegestuurd bij het openen van de webview om in te loggen en hieraan kan er naar het juiste inlog scherm verwezen worden. Op het moment dat de school van de leerling niet in de lijst staat wordt de student doorverwezen naar een andere pagina. Op deze pagina wordt instructie gegeven dat de student contact op moet nemen met de school. In het prototype is het design van de huidige applicatie aangehouden. In onderstaande afbeeldingen wordt getoond hoe dit er in de Flutter app uit ziet:



Figuur 16 - Prototype startscherm



Figuur 17- Prototype schoolkeuze scherm

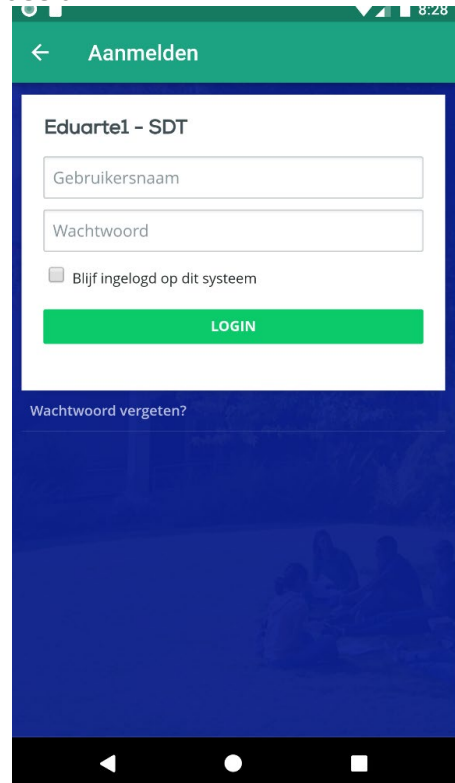


## 4.3 Inloggen OAuth 2.0

Zoals eerder besproken in de deelvraag *"Hoe kan ik de authenticatie zo veilig mogelijk maken in het nieuwe framework?"* is het inloggen niet op de gebruikelijke manier gerealiseerd. Dit komt doordat de backend van Eduarte anders is afgesteld dan verwacht. In figuur 8 in hoofdstuk 3.2 is uitgelegd hoe dit proces in zijn werk gaat. In de onderstaande foto is een deel van de uitwerking afgebeeld:

```
onNavigationRequest: (NavigationRequest request) async {  
  if (request.url  
    .startsWith('http://student.eduarte.local/oauth-callback')) {  
    String url = request.url.toString();  
    var accessToken = url.split("code=").last;  
    Response<dynamic> res = await httpService.request(  
      url: "https://d-login.educus.nl/oauth2/token",  
      method: Method.POST,  
      isAuthorized: false,  
      headers: {  
        "Content-Type": "application/x-www-form-urlencoded"  
      },  
      params: <String, dynamic>{  
        'grant_type': 'authorization_code',  
        'code': accessToken,  
        'redirect_uri':  
          'http://student.eduarte.local/oauth-callback',  
        'client_id': '3cdf9da3-705a-4899-b3c9-393db1fdc8b7',  
        'client_secret': '587c2c26-cead-4155-b19c-70a83042270a'  
      }  
    );  
  
    await LocalStorageManager().setAccessToken(res.data);  
    await LocalStorageManager().setRefreshToken(res.data);  
  
    await cookieManager.clearCookies();  
    await accountController.fetchAccountWithRedirect();  
  
    navigatorKey.currentState?.pushNamedAndRemoveUntil(  
      '/homepage', (Route<dynamic> route) => false);  
  }  
  return NavigationDecision.navigate;  
},
```

Figuur 18 - Webview code snippet



Figuur 19 - Prototype login scherm

Er wordt handmatig een webview gestart en in deze webview wordt geluisterd naar veranderingen in de URL. Op het moment dat de URL een bepaalde waarde bevat, in dit geval `"http://student.eduarte.local/oauth-callback"`, wordt de code die achter `"/oauth-callback?code="` staat afgesneden. Nu is de authorization token bekend, met deze token kan meteen de volgende API call uitgevoerd worden en de access token en refresh token van de gebruiker opgehaald worden. Deze tokens worden vervolgens opgeslagen in de lokale opslag van de applicatie.

Wanneer de gebruiker is ingelogd is het ook wenselijk dat hij/zij ingelogd blijft wanneer de applicatie afgesloten wordt. Hier komen de opgeslagen access token en refresh token

uit de storage bij kijken. Op het moment dat de app wordt gestart wordt er gekeken of één van deze tokens aanwezig is. Als dit niet zo is, wordt de loginpagina meteen getoond aan de gebruiker. Wanneer dit wel zo is, wordt er geprobeerd data op te halen met de access token. Nu is het zo dat de access token kan verlopen. Als dit het geval is, wordt de refresh token geraadpleegd om een nieuwe access token op te halen. In de onderstaande foto wordt afgebeeld hoe dit in zijn werk gaat.

```
onError: (err, handler) async {  
  if (err.response?.statusCode != null &&  
      err.response!.statusCode == 401) {  
    try {  
      await refreshToken().then((value) async {  
        RequestOptions origin = err.response!.requestOptions;  
        return handler.resolve(await _dio!.request(origin.path,  
            options: Options(  
              method: origin.method,  
              headers: <String, String>{  
                'Authorization':  
                  'Bearer ${await LocalStorageManager().getAccessToken()}',  
                "Accept": "application/json"  
              },  
            ));  
      });  
    }  
  });  
}
```

*Figuur 20 - Refresh token code snippet*

Op het moment dat de access token verlopen is, geeft de server een error. Er wordt een functie aangeroepen die met de refresh token een nieuwe access token ophaalt. Als dit niet lukt wordt er meteen doorverwezen naar de inlogpagina. Wanneer dit wel lukt wordt de eerdere API call, waar de foutmelding zich voordeed, opnieuw uitgevoerd met de nieuwe access token. Nadat dit proces succesvol doorlopen is kan de gebruiker meteen doorgestuurd worden naar de app zonder handmatig in te loggen. Dit gebeurt in de achtergrond en de gebruiker merkt hier niks van.

## 4.4 Opslag

Het opslaan van de studentgegevens wordt gedaan een combinatie van Flutter Secure Storage en Dio Cache Interceptor. De Flutter Secure Storage wordt gebruikt om data zoals access en refresh tokens op te slaan. Deze package maakt gebruik van AES encryptie en is daarom veilig. De Dio Cache Interceptor maakt gebruik van de normale Dio package, deze package wordt gebruikt voor het maken van de API calls en het afhandelen van fouten die daar in voor kunnen komen zoals het niet hebben van een geldige token etc.

Op het moment dat er data binnenkomt als JSON, wordt het eerst via het eerder beschreven MVC+S ontwerppatroon in een model gezet. Daarna wordt het opgeslagen in de HiveCacheStore. Dit is een package dat een plek in het geheugen beschikbaar stelt om data op te slaan. Er is voor deze CacheStore gekozen omdat deze op alle platformen beschikbaar is. In de onderstaande foto is te zien hoe deze extra package bovenop de huidige Dio package werkt.

```
getTemporaryDirectory().then((dir) {  
  CacheStore cacheStore = HiveCacheStore(dir.path);  
  
  var cacheOptions = CacheOptions(  
    store: cacheStore,  
    policy: CachePolicy.forceCache,  
    hitCacheOnErrorExcept: [401, 403],  
    maxStale: const Duration(days: 7),  
    priority: CachePriority.high,  
    cipher: null,  
    keyBuilder: CacheOptions.defaultCacheKeyBuilder,  
    allowPostMethod: false,  
  ); // CacheOptions
```

*Figuur 21 - Data opslag code snippet*

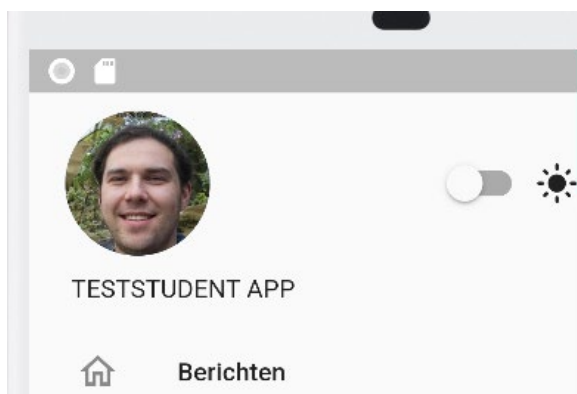
### Offline gebruik

Een belangrijk onderdeel van het opslaan van de data is dat de app ook gebruikt kan worden wanneer de gebruiker geen internet heeft. In de eerder genoemde Dio Cache Interceptor is het ook mogelijk om de opgeslagen data te gebruiken wanneer het internet niet beschikbaar is. In de cache opties kan er ingesteld worden, zoals in de data opslag code snippet te zien is, hoe lang de data gebruikt mag worden voordat er nieuwe data

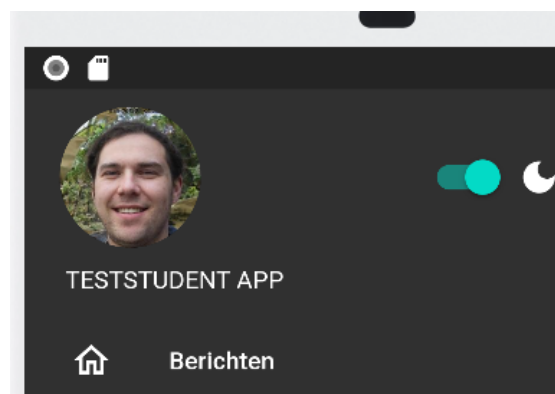
opgehaald moet worden. In theorie zou een student dus een week het internet uit kunnen zetten en nog steeds gebruik kunnen maken van de app.

## 4.5 Thema's

In de onderstaande figuren is te zien dat er via het menu een thema ingesteld kan worden. Omdat Topicus in de toekomst thema's per school wil aanbieden door middel van bijvoorbeeld andere kleuren of andere logo's, is er in dit prototype gekozen voor een 'Light en Dark' thema. Op deze manier wordt er aangetoond dat dit mogelijk is en dit geen obstakel zal zijn in de toekomst.



*Figuur 22 - Flutter app light theme*



*Figuur 23 - Flutter app dark theme*

Er is een ThemeProvider class aangemaakt waar via een boolean wordt gekeken of de knop aan of uit is. In deze ThemeProvider kunnen meerdere thema's worden aangemaakt en het "Dark Theme" dat daar bij hoort.

Om ervoor te zorgen dat het thema onthouden wordt wanneer de app wordt afgesloten is ervoor gekozen om dit op te slaan in de lokale opslag. Zo hoeft de gebruiker niet elke keer wanneer hij/zij de app opent opnieuw te kiezen.

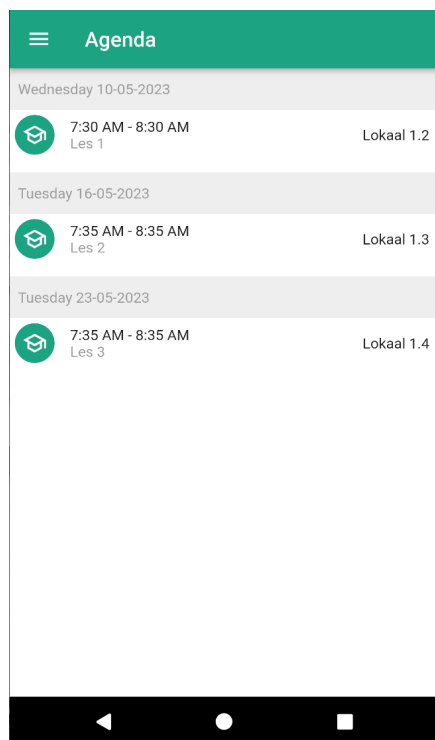
Er is ook een mogelijkheid om het thema te laten bepalen door het OS. In de main class van het prototype is dit in te stellen maar de keuze hiervoor ligt nog open.

## 4.6 Agendapagina

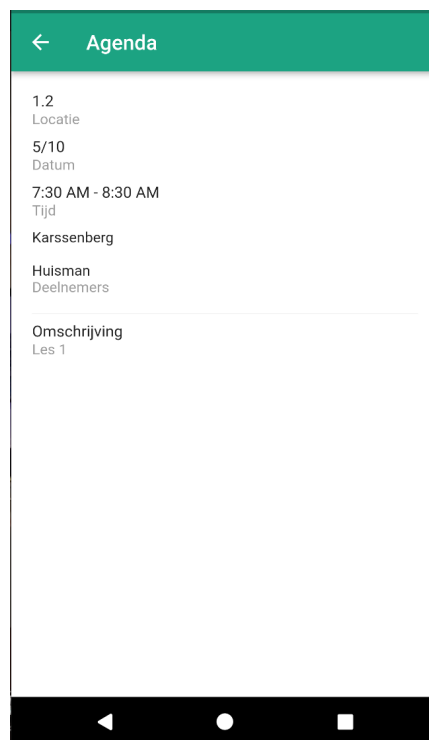
De agendapagina is de eerste pagina waar de gebruiker op terechtkomt na het inloggen. Tijdens het inloggen is de juiste gebruiker al ingeladen in de cache, hierdoor kan de agenda van de student meteen getoond worden zonder problemen. Het ophalen gebeurt met deze API call: "https://sdt-rest.educus.nl/eduario/rest/v1/afspraak?deelnemer="

Buiten het id van de deelnemer worden ook de begin- en eindtijd meegestuurd. Deze zijn in te stellen in de code en betekenen dat er twee weken voor en na de huidige tijd agendapunten opgehaald moeten worden.

Wanneer de student op de pagina komt, wordt de data automatisch geladen. De student kan ook handmatig de data verversen door middel van een RefreshIndicator, dit kan uitgevoerd worden door naar beneden te swipen. De laatste data wordt op dat moment opnieuw van de API opgehaald. Door het ontwerp te volgen uit de oude app ziet de app er in Flutter hetzelfde uit. In de onderstaande foto wordt dit getoond.



Figuur 24 - Flutter app agenda overzicht



Figuur 25 - Flutter app agenda detail

## 4.7 Resultatenpagina

In de resultatenpagina kunnen studenten de cijfers van toetsen en examens zien.

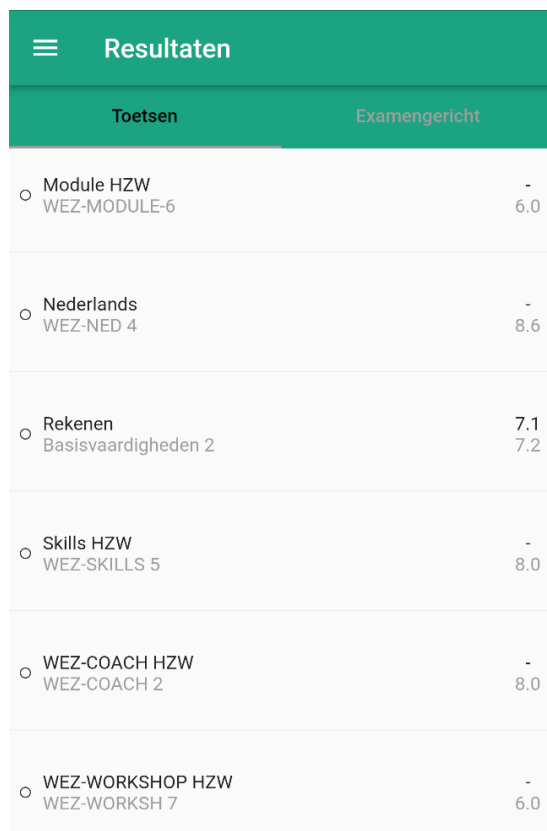
Wanneer de student op de pagina komt worden de volgende API calls uitgevoerd:

<https://sdt-rest.educus.nl/eduario/rest/v1/resultaat/overzicht/userID?type=FORMATIEF>

<https://sdt-rest.educus.nl/eduario/rest/v1/resultaat/overzicht/userID?type=SUMMATIEF>

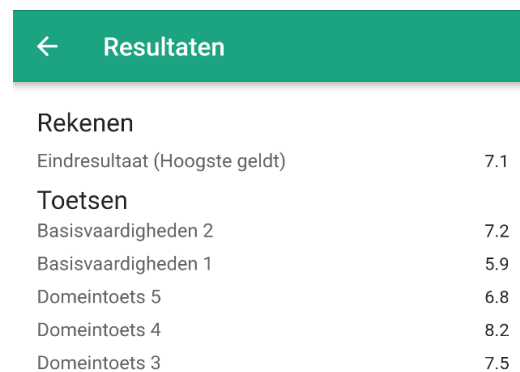
Formatief is voor toetsresultaten en summatief is voor examenresultaten. Door deze informatie apart op te halen kunnen de pagina's makkelijk gescheiden worden. De pagina is opgebouwd uit twee schermen. Er kan door middel van swipen of door op de knop te drukken genavigeerd worden tussen de pagina's.

Ook kan de student handmatig de data verversen op deze pagina. In beide pagina's is een RefreshIndicator toegevoegd die de API calls opnieuw kan uitvoeren. Hierdoor kan de student de laatst beschikbare cijfers ophalen door de pagina naar beneden te slepen. Elk resultaat beschikt over een detailpagina waar verdere informatie over de gemaakte toets te vinden is.



Resultaten	
Toetsen	Examengericht
Module HZW WEZ-MODULE-6	- 6.0
Nederlands WEZ-NED 4	- 8.6
Rekenen Basisvaardigheden 2	7.1 7.2
Skills HZW WEZ-SKILLS 5	- 8.0
WEZ-COACH HZW WEZ-COACH 2	- 8.0
WEZ-WORKSHOP HZW WEZ-WORKSH 7	- 6.0

Figuur 26 - Flutter app toetsresultaten pagina



Resultaten	
Rekenen	
Eindresultaat (Hoogste geldt)	7.1
Toetsen	
Basisvaardigheden 2	7.2
Basisvaardigheden 1	5.9
Domeintoets 5	6.8
Domeintoets 4	8.2
Domeintoets 3	7.5

Figuur 27 - Flutter app resultaten detailpagina

## 4.8 Pushnotificaties

In het onderzoek naar de verschillende frameworks in hoofdstuk 3.8 is ook onderzocht op welke manieren pushnotificaties afgehandeld kunnen worden in een framework. Het was al snel duidelijk dat dit via Firebase het beste gerealiseerd kan worden. Het belangrijkste punt van Firebase is dat dit gratis te gebruiken is ten opzichte van bijvoorbeeld Airship, Amazon SNS of OneSignal. Om deze reden is hiervoor gekozen tijdens het realiseren van dit onderwerp in dit project. Als eerste moet het project aan de Firebase Console worden toegevoegd, dit kan simpel worden uitgevoerd door de Firebase CLI te installeren en in te loggen met het Google account waar het aan gekoppeld moet worden. Door het command "flutterfire configure" uit te voeren kan het project gekozen worden en deze is daarna automatisch zichtbaar in de Firebase Console.

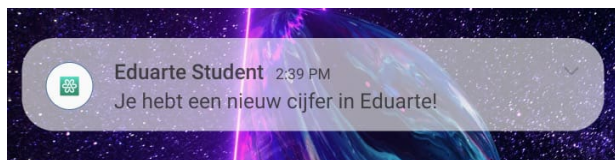
Om een pushnotificatie te sturen wordt er vanuit Firebase een tekst ingesteld. Daarna kan er door middel van een FCM (Firebase Cloud Messaging) token een toestel gekozen worden waar de pushnotificatie heen moet. Deze token is voor dit prototype tijdens het opstarten van de app in de console geprint zodat dit getest kan worden.

Ook moeten er verschillende packages worden toegevoegd aan de code om dit te laten werken, Firebase Core, Firebase Messaging en Firebase analytics. In de main van het project moet Firebase geïnitieerd worden om dit goed te laten werken. In onderstaand figuur is te zien hoe dit is opgelost.

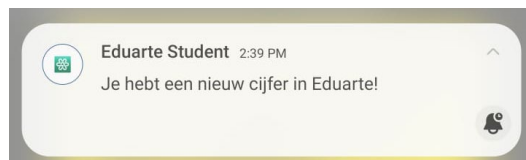
```
await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform);  
final fcmToken = await FirebaseMessaging.instance.getToken();  
// ignore: avoid_print  
print(fcmToken);
```

*Figuur 28 - Firebase code snippet*

In de onderstaande foto worden twee situaties op een Samsung telefoon getoond van een notificatie. Dit onderdeel is niet getest op iOS omdat er tijdens de stageperiode geen toegang is tot een Mac computer en iPhone.



*Figuur 29 - Notificatie vergrendeld*



*Figuur 30 - Notificatie ontgrendeld*

In een latere ontwikkeling zal er ook een backend gemaakt moeten worden die per student een notificatie kan sturen wanneer er een nieuw cijfer, bericht of agenda punt is.



## 5. Acceptatietest

In dit hoofdstuk worden de requirements getest op de verschillende acceptatiecriteria die zijn opgesteld in de Requirementsanalyse bijlage. Er wordt gekeken of de criteria volledig, gedeeltelijk of niet geïmplementeerd zijn.

#	Acceptatiecriteria	Status
<b>F1</b>	Lijstweergave van verschillende scholen. Instructies andere pagina wanneer school niet in lijst staat.	Volledig
<b>F2</b>	Een inlogpagina van de gekozen school. De student moet ingelogd zijn om andere pagina's in de app te kunnen zien.	Volledig
<b>F3</b>	Lijstweergave per dag. In de app wordt de vroegst bekende afspraak getoond Agenda opent standaard aan het begin van de huidige week. Gegevens per item: Begintijd en eindtijd, Omschrijving (vak), Locatie. Dag van vandaag wordt als 'Vandaag' aangeduid, morgen als 'Morgen' en overmorgen als 'Overmorgen' , de rest met datum.	Gedeeltelijk, dagen worden nog niet als 'Vandaag, morgen en Overmorgen' getoond
<b>F4</b>	Voor elk agenda-item zijn de volgende detailgegevens zichtbaar: Locatie, Datum, Tijd, Deelnemers (studenten, groepen, docenten), Omschrijving van agenda-item.	Volledig
<b>F5</b>	Lijst weergave met alle berichten. Gegevens per bericht in de lijstweergave: Onderwerp, afzender, datum, (alle zonder label) eerste regels van bericht. Detailpagina per bericht. Gegevens op de detailpagina: veld 'Van' (afzender, vrij tekstveld), Datum, Onderwerp. Prullenbak om bericht te verwijderen (lokaal).	Gedeeltelijk, geen optie voor lokaal verwijderen
<b>F6</b>	Lijst weergave met toets uitslagen.	Volledig

	Gegevens per toets uitslag in lijstweergave: eindresultaat (gemiddelde), het laatst behaalde resultaat, Omschrijving van het vak. Detailpagina per vak: alle gemaakte toetsen + cijfers. Detailpagina per toets: naam, resultaat, toets code, weging, datum, toets soort, status, pogingen	
<b>F7</b>	Lijstweergave met gemaakte examens. Gegevens per examen: Naam, omschrijving en cijfer Detailpagina per examen: Naam, cijfer, gemaakte examens	Volledig
<b>F8</b>	Lijst weergave met taken. Gegevens per taak: Naam, beschrijving en deadline Taken gesorteerd op datum, hoogste bovenaan. Detailpagina per taak: Type taak, deadline en omschrijving	Gedeeltelijk, module wordt niet meer verkocht vanuit Topicus
<b>F9</b>	Lijst/tabel weergave met persoonsgegevens. Gegevens: Profiel foto, voornaam + achternaam, geboortedatum, studentnummer, telefoon mobiel, telefoon algemeen, email algemeen, email school, adres 1, adres 2	Volledig
<b>F10</b>	Wijzigen knop op de personaliapagina. Telefoon mobiel, telefoon werk, telefoon algemeen, email algemeen mogelijk om te wijzigen in tekstvelden	Niet geïmplementeerd
<b>F11</b>	Absentie reden in een dropdown menu (ziekte of kort verzuim) Startdatum en tijd Einddatum en tijd (optioneel) Absentiemelding versturen knop Informatie wanneer een absentie reden geldig is tonen	Niet geïmplementeerd
<b>F12</b>	Op de feedbackpagina staat een tekstveld waar de student zijn/haar feedback kan invullen. Stuur feedback knop actief wanneer feedback is ingevuld .	Gedeeltelijk, geen POST mogelijkheid
<b>F13</b>	Links bovenaan een knop om menu te openen. Menu bevat student profiel foto en student naam Pagina's in menu: Berichten, resultaten, agenda, werkljst, personalia, absentiemelding, feedback en afmelden	Volledig
<b>F14</b>	In het menu staat onderaan een afmeld knop.	Volledig

<b>F15</b>	Een schakelaar in het menu waar gekozen kan worden tussen de twee thema's.	Volledig
<b>F16</b>	Vanuit Firebase kan een pushnotificatie gestuurd worden en deze kan op een emulator of telefoon ontvangen worden.	Volledig

Tabel 9 - Functionele acceptatietest

#	Acceptatiecriteria	Status
<b>NF1</b>	Er kan via het bestaand OAuth2 systeem ingelogd worden.	Volledig
<b>NF2</b>	Er kan data opgehaald worden van de bestaande Eduarte API en getoond worden.	Volledig
<b>NF3</b>	De opgehaalde data van de API kan lokaal worden opgeslagen.	Volledig
<b>NF4</b>	De app functioneert naar behoren op beide platforms Android en iOS.	Gedeeltelijk, de app is voornamelijk getest op Android omdat er geen Mac beschikbaar is op dit moment
<b>NF5</b>	De app moet in de toekomst gebruik kunnen maken van functies zoals de native agenda van de telefoon.	Volledig
<b>NF6</b>	De app moet in vliegtuigstand nog goed functioneren om rooster/cijfers te bekijken.	Volledig

Tabel 10 - Non-functionele acceptatietest

## 5.1 Conclusie

De requirements die eerder zijn opgesteld gelden voor de hele Eduarte Studentenapp, het prototype richt zich op verschillende onderdelen van de huidige app. Omdat het prototype niet alle functionaliteiten bevat, en de acceptatietest is gebaseerd op de huidige app, zijn sommige testen niet behaald. Uit de acceptatietest kan er geconcludeerd worden dat de meeste tests zijn geslaagd en aanwezig zijn in het prototype. De niet behaalde testen zijn geen onderdeel van het prototype en zijn voorafgaand al uitgesloten door het gebrek aan tijd. De belangrijkste functionaliteiten zitten wel in het prototype en zijn ook getest.

## 6. Conclusie

Als antwoord op de vraag *"Wat is het meest geschikte hybride framework om de Eduarte applicatie toekomstbestendig in te herontwikkelen?"*. Er is onderzoek gedaan naar de huidige applicatie, risico's bij een nieuw framework, verschillende frameworks en ontwerpkeuzes.

Door het opstellen van verschillende criteria en deze per framework te onderzoeken, kan er geconcludeerd worden dat Flutter het beste framework is voor de Eduarte Studentenapp. Door het ontwikkelen van een prototype wordt deze conclusie onderbouwd en bewezen. Zo bevat het prototype functies zoals OAuth 2.0, data ophalen en tonen, cache opslag, secure storage, thema's en pushnotificaties. Dit zijn de functionaliteiten die vereist zijn voor de keuze van het nieuwe framework.

Een minder punt van het prototype is dat er niet op alle vormen van data getest is. Ook wordt sommige data uit de modellen niet gebruikt omdat het in dit prototype niet van toepassing is. In het uiteindelijke product zal dit wel allemaal toegevoegd en getest moeten worden. Ook kan er een vorm van logging worden toegepast door bijvoorbeeld Firebase, op deze manier kan de app goed gemonitord worden.

Tijdens de afstudeerperiode was er niet genoeg tijd om alle functionaliteiten van de app te ontwikkelen. Door de eerder genoemde functionaliteiten toe te voegen aan het prototype is bewezen dat de volledige studentenapp gerealiseerd kan worden in het nieuwe framework.

Verschiedende aspecten van het prototype zijn al zo goed als klaar om overgezet te worden naar de live versie van een nieuwe app. Op dit moment worden verschillende onderdelen van dit prototype ook daadwerkelijk gebruikt door Topicus.

# 7. Reflectie

In dit hoofdstuk ga ik in op verschillende onderdelen van mijn afstudeerstage. In de bijlage "Reflectieverslag" zijn meerdere STARR-reflecties te vinden waar op verschillende situaties wordt gereflecteerd.

## 7.1 Planning

Tijdens de stageperiode heb ik mij goed aan de planning kunnen houden die opgesteld was aan het begin van het project. Het project was opgedeeld in sprints van twee weken waardoor het bijhouden van taken overzichtelijk is verlopen. Op deze manier kon er snel gehandeld worden als ik achter zou gaan lopen maar dit is niet voorgekomen.

Er zijn momenten geweest dat ik te lang met hetzelfde onderdeel bezig was waardoor de concentratie iets achteruit ging. Dit heb ik opgelost door iets anders op te pakken waardoor ik even in een ander ritme zat. Uiteindelijk heeft dit geresulteerd in een goede aanpak om op de planning te blijven.

## 7.2 Analyseren

Het afstudeerproject richt zich op meerdere competenties, waaronder analyseren. In dit onderdeel heb ik verschillende aspecten van de huidige app onderzocht zoals requirements van de huidige app, wensen van Topicus voor een nieuw framework en de risico's die daarbij horen. Ook heb ik onderzoek uitgevoerd waarin ik de eerder genoemde aspecten per framework onderzocht heb. Als ik terugkijk op dit onderdeel heb ik duidelijk antwoord kunnen geven op de verschillende deelvragen.

## 7.3 Adviseren

De adviezen die ik heb uitgebracht richten zich op functionaliteiten binnen het nieuwe framework. Ik heb verschillende keuzes moeten maken tijdens het realiseren, om dit goed te kunnen onderbouwen is er onderzoek gedaan naar verschillende mogelijkheden en heb ik adviezen uitgebracht die daar op aansluiten. Tijdens de stageperiode heb ik veel invloed gehad op het ontwikkelteam dat zich bezig houdt met de daadwerkelijke

app, zo zijn verschillende ontwerpkeuzes en adviezen aangenomen en worden op dit moment ook gebruikt.

## 7.4 Realiseren

De realisatie van het prototype is goed verlopen. Hoewel het inlogsysteem wat stroever ging omdat de backend bepaalde acties niet accepteerde, heb ik dit goed op weten te pakken en af te ronden. De methode die ik bedacht heb om dit tijdelijk te omzeilen wordt op dit moment ook zelf door Topicus gebruikt. Ook de ontwikkel keuze voor Dio en Dio Cache Interceptor worden na aanbevelingen van mijn onderzoek gebruikt in de toekomstige app van de Eduarte studentenapp.

## 7.5 Resultaat

Ik ben zeer tevreden met het resultaat van dit project. Het prototype bevat bijna alle functionaliteiten die de huidige app ook heeft. Hierdoor kan straks snel de overstap van native naar Flutter uitgebracht worden. Ik ga hier na de stage zelf aan meewerken om dit te realiseren. Ook heb ik veel geleerd op deze stage over het bedrijfsleven en hoe bepaalde functionaliteiten tot stand komen om ontwikkeld te worden. Het framework Flutter, waar ik het prototype in ontwikkeld heb, was een fijn framework om mee te werken. Ik heb hier tijdens de opleiding al eerder mee gewerkt en het was daarom leuk dat ik deze kennis kon gebruiken tijdens het afstuderen.

## 8. Bibliografie

- (sd). Opgehaald van Easyretro: <https://easyretro.io/templates/went-well-to-improve-action-items/>
- Dhakal, N. (2022, 08 04). *Hive in Flutter*. Opgehaald van Medium: <https://medium.com/codex/hive-in-flutter-7a2b320eb1dc>
- Flutter. (sd). *Shared Preferences*. Opgehaald van Pub.dev: [https://pub.dev/packages/shared\\_preferences](https://pub.dev/packages/shared_preferences)
- Gürel, T. (2022, 01 30). *Mvc-s Design Pattern In Flutter*. Opgehaald van Itnext: <https://itnext.io/mvc-s-design-pattern-in-flutter-6eba15169413>
- Janse, B. (2018, 01 21). *Multicriteria Analyse*. Opgehaald van Toolshero: <https://www.toolshero.nl/besluitvorming/multicriteria-analyse-mca/>
- Mulder, P. (2017, 06 06). *MoSCow Methode, een praktische uitleg met voorbeelden*. Opgehaald van Toolshero: <https://www.toolshero.nl/project-management/moscow-methode/>
- Oleinik, A. (2021, 04 13). Opgehaald van <https://levelup.gitconnected.com/the-4-ways-to-store-data-locally-in-your-flutter-app-that-youre-going-to-need-abdafa991ae3>
- Retrospective Scrum*. (sd). Opgehaald van ScrumGuide: <https://scrumguide.nl/retrospective-scrum/>
- Sofco. (sd). *wat-is-mvc-en-waarom-wij-het-gebruiken*. Opgehaald van Sofco: <https://sofco.nl/nl/blog/wat-is-mvc-en-waarom-wij-het-gebruiken>
- Topicus. (2023). *werkenbijtopicus*. Opgehaald van <https://werkenbijtopicus.nl/over-topicus>
- Wardt, R. v. (sd). *Wat is scrum methode*. Opgehaald van agileScrumGroup: <https://agilescrumgroup.nl/wat-is-scrum-methode/>

Figuur 1 - Topicus Eduarte gebouw .....	8
Figuur 2 - Project scope .....	10
Figuur 3 - Momentopname werkwijze .....	11
Figuur 4 - Scrum retrospective .....	12
Figuur 5 - Planning afstudeerproject.....	13
Figuur 6 - Eduarte school keuzescherf huidige app.....	14
Figuur 7 - Inlogpagina school huidige app .....	14
Figuur 8 - Login proces sequence diagram .....	15
Figuur 9 - Agendapagina huidige app .....	16

Figuur 10 - Menu huidige app.....	16
Figuur 11 - Functionaliteiten huidige app .....	16
Figuur 12 - Error backend callback URL .....	32
Figuur 13 - Voorbeeld authorization token.....	33
Figuur 14 - Token request postman.....	33
Figuur 15 - MVC+S in Flutter .....	39
Figuur 16 - Prototype startscherm .....	40
Figuur 17- Prototype schoolkeuze scherm .....	40
Figuur 18 - Webview code snippet .....	41
Figuur 19 - Prototype login scherm .....	41
Figuur 20 - Refresh token code snippet.....	42
Figuur 21 - Data opslag code snippet .....	43
Figuur 22 - Flutter app light theme .....	44
Figuur 23 - Flutter app dark theme .....	44
Figuur 24 - Flutter app agenda overzicht.....	45
Figuur 25 - Flutter app agenda detail .....	45
Figuur 26 - Flutter app toetsresultaten pagina .....	46
Figuur 27 - Flutter app resultaten detailpagina.....	46
Figuur 28 - Firebase code snippet .....	47
Figuur 29 - Notificatie vergrendeld.....	48
Figuur 30 - Notificatie ontgrendeld.....	48
 Tabel 1 - Definities .....	 4
Tabel 2 - Tools.....	13
Tabel 3 - Functionele requirements app .....	19
Tabel 4 - Niet-functionele requirements app .....	20
Tabel 5 - Niet-functionele requirements framework .....	21
Tabel 6 - Samenvatting risicoanalyse .....	26
Tabel 7 – MCA-tabel Framework .....	31
Tabel 8 - Informatie app.....	32
Tabel 9 - Functionele acceptatietest .....	51
Tabel 10 - Non-functionele acceptatietest .....	51



## 9. Versiebeheer

<b>Versie</b>	<b>Gewijzigd door</b>	<b>Wijzigingen</b>
<b>0.1</b>	Jesse Roessink	Initiële versie
<b>0.2</b>	Jesse Roessink	Opzet hoofdstukken
<b>0.3</b>	Jesse Roessink	Inleiding
<b>0.4</b>	Jesse Roessink	Projectaanpak
<b>0.5</b>	Jesse Roessink	Onderzoek
<b>0.6</b>	Jesse Roessink	Feedback stagebegeleider verwerkt
<b>0.7</b>	Jesse Roessink	Feedback schoolbegeleider verwerkt
<b>1.0</b>	Jesse Roessink	Concept inlevering
<b>2.0</b>	Jesse Roessink	Definitieve inlevering

# 10. Bijlagen

## Bijlage 1 interview met Ton

Het doel van dit interview is om meer inzicht te krijgen in de eisen en toekomstvisie van de Eduarte studenten app. Het interview is opgesteld als een ongestructureerd interview waarbij een aantal onderwerpen wordt besproken. Deze onderwerpen zijn voor het interview bedacht, de punten die behandeld worden zijn:

- Waarom overstappen van Native naar Hybrid?
- Welke functionaliteiten moet het framework ondersteunen?
- Wat is de toekomstvisie van studenten in Eduarte?

### **Waarom overstappen van Native naar Hybrid?**

Het uitfaseren van de native apps is geen doel op zich. Het is wel zaak om te gaan ontwikkelen op een toekomstbestendige manier: De apps/ portalen moeten in de toekomst gemakkelijk en kostenefficiënt (door)ontwikkeld kunnen blijven worden.

### **Welke functionaliteiten moet het framework ondersteunen?**

De functionaliteit die op korte en middellange termijn in de apps beschikbaar zou moeten zijn betreft de functionaliteit die nu ook beschikbaar is. Wel zou deze functionaliteit herijkt moeten worden. Dat wil zeggen dat dezelfde functionaliteit moet worden voorzien van een betere look and feel (UX).

Gewenste uitbreidingen op de huidige functionaliteit zijn:

- Pushnotificaties
- De app kunnen aanpassen naar de look and feel van de school
- Functionaliteit aan en uit kunnen zetten per school.

### **Toekomstvisie Studenten in Eduarte:**

We zijn een studenten managementsysteem en geen studenten leeromgeving. We hebben ook niet de ambitie om een leeromgeving te zijn. Dit betekent dat Eduarte voor studenten vooral een informerend karakter zal hebben. Slechts enkele zaken moeten worden teruggeschreven vanuit de studenten omgeving naar Eduarte backoffice. Denk aan: verzuimmeldingen, het inschrijven voor toetsmomenten en het kiezen van (onderwijs) producten.

## Bijlage 2 overleg met Robert

Robert geeft aan dat het niet zijn voorkeur heeft om met native apps door te gaan. Dit heeft verschillende redenen:

### **Kwetsbaarheid:**

Wanneer we zowel IOS als Android zouden willen ondersteunen, dan hebben we van beide een ontwikkelaar nodig. Met het bijkomende probleem dat wanneer er iemand uitvalt, we een probleem hebben.

### **Resources en toekomstbestendigheid:**

Na ontwikkeling van de apps voor de verschillende doelgroepen hebben we wellicht niet voldoende werk om zowel een IOS- als een Android ontwikkelaar in dienst te houden.

### **Eenzelfde codebase voor IOS/Android en Web**

Het ontwikkelen zal nu en in de toekomst sneller gaan, omdat je niet drie keer (IOS, Android, Web) hoeft te ontwikkelen maar één keer. Dit zal niet drie keer zo snel gaan, maar wel sneller.