# Keuze verantwoording

Wij hebben gekozen voor React Native en Quasar dit zijn twee frameworks waar wij de documentatie erg prettig van vinden. Wij zijn zelf heel erg fan van het web framework Vue en daarom is Quasar ook heel interessant omdat de structuur van programmeren ons daar heel erg in aanspreekt. React Native heeft een grote community waardoor het interessant is om te proberen. Ionic heeft ons ook een beetje een vies gevoel gegeven. Wanneer je de site voor het eerst bezoekt lijkt het dat het Angular, React en Vue volledig ondersteund worden. Dit is helaas niet het geval en wordt je eigenlijk in de richting van Angular geduwd. Angular is misschien wel het meest uitgebreide framework maar aangezien wij in 4 weken een applicatie uit de grond moeten stampen is de learning curve om dat goed te masteren het niet waard. Hier komt natuurlijk ook nog bij kijken dat je het hybrid framework moet onder de knie moet krijgen. Ook tijdens het opzetten van Ionic werkte één project keuze niet(app met camera). Dit is erg slordig en aangezien je camera gebruik altijd even wilt testen om kennis te maken met een framework.

# Quasar

## Difficulties

Zonder Iphone is er niet een makkelijke manier met Cordova om IOS te testen. Voor de rest ging Quasar, Vue en Cordova zonder problemen. Ook werken dingen zoals gps alleen goed op een https pagina of android app. Dit doen ze voor de veiligheid maar hierdoor wordt testen wel wat irritanter.

## Implementations

Om de opdracht werkend te krijgen is er gebruik gemaakt van de volgende vier plugins:

Er is van de Cordova Gps plugin gebruik gemaakt op de locatie op te halen.

Er is van de Cordova Camera plugin gebruik gemaakt om foto’s te maken en permissie te vragen voor de Camera.

Er is van de Vuex store gebruik gemaakt om data globaal op te slaan, aan te passen en te verwijderen hierdoor kan je die data overal makkelijk gebruiken.

Er is van de Vuex persistent state gebruik gemaakt om de data die in de Vuex store zit (Dat normaal alles in de sessie opslag doet) in de localstorage op te slaan. hierdoor blijft de data bestaan ook al sluit je de app.

Om de locatie van de Gps te reverse Geocoden (Van lat, long naar een adres) wordt er een call gedaan naar de open street maps api die het in JSON terug geeft.

# React native

## Difficulties

Expo komt met een mooie kit aan development tools en packages. Het nadeel is wel dat je afhankelijk bent van bepaald peer dependencies die werken op een bepaalde versie. De ondersteuning van native componenten is zeer groot. Het was ook geen probleem om toegang te krijgen tot de camera of gps. De grootste problemen zijn de componenten stylen. Sinds React Native werkt met flex containers is dit even wennen. Ik liep tegen het probleem aan om 2 kinderen diep een functie aan te laten spreken. Wanneer je dan de context aan wilt passen is de context verouderd als je dit normaal doet. Dit kun je oplossen door de es6 arrow functie syntax te gebruiken. Hierdoor wordt altijd de huidige context gebruikt. Ook het geocoderen van Latitude en longitude naar een plaats resulteert in een array in plaats van een object. Ik vroeg me af waarom de values steeds undefined waren omdat ik dit op een array probeerde te doen ipv een object. We missen de standaard store integratie vanuit expo. Quasar doet dit beter en heeft een fijne plugin waardoor je direct de local storage kunt raadplegen. React Native heeft wel opties om een store(Redux) te gebruiken maar dit zag er iets ingewikkelder uit. Het reloaden van de applicatie met behulp van de expo app werkt soms niet altijd even soepel.

## Implementations

## Voor het weergeven van de lijst maken we gebruik van de FlatList component van React Native.

## Voor het opslaan van data maken we gebruik van de AsyncStorage. Dit is een bibliotheek die het mogelijk maakt om asynchroon data op te slaan of op te halen van de opslag.

Voor het selecteren van een Image maken we gebruk van de expo-imagepicker. Dit is een bibliotheek die het mogelijk maakt om vanuit je camera of storage een foto te selecteren of maken.

Voor het ophalen van de locatie maken we gebruik van expo-location. Dit is een bibliotheek waarmee je de gps aan kunt spreken en vertaalslagen kunt maken naar een object met o.a. je adres.

Met behulp van expo-permissions kun je toestemming vragen aan de gebruiker voor o.a. de camera.

# Conclusie

React Native is een fijn framework om mee te wennen. Wij waren niet super bekend met het ReactJS framework maar het voelde snel vertrouwd aan. Ook de componenten die React Native biedt zijn zeer uitgebreid. De documentatie is prettig en uitgebreid. De problemen waar wij tegenaan liepen waren dan ook vooral kleine dingetjes die we niet wisten. Bij React kan de parent alleen zijn state aanpassen en niet de childs. Dit zorgt voor extra werk aangezien je dus een mutate functie door moet geven als prop om vanuit de child de parent zijn state aan te passen. Het is een prettig framework maar veel dingen moeten uitgebreider opgelost worden dan als met Quasar.

Het ontwikkelen met Quasar verliep zonder problemen. Daarom komt dit framework ook voor ons als beste uit de test. Één van ons had al ervaring met Quasar en zelf zijn we groot fan van Vue. Hierdoor liepen we tijdens ontwikkelen tegen weinig problemen aan. Met behulp van Cordova zorg je er makkelijk voor dat je native componenten kunt gebruiken. Tijdens het initialiseren van het project bieden ze gelijk aan om gebruik te maken van VueX(de store van vue). De combinatie van de store werkte ook goed met Cordova samen daarom konden we de opdracht makkelijk maken.

We denken wel dat bij grotere projecten React Native een handigere keuze kan zijn. Je hebt meer vrijheid om specifieke zaken te ontwikkelen. Wij krijgen een opdracht van 4 weken en daarom is Vue met Quasar zeer geschikt aangezien ook de styling minder tijd gaat kosten.