

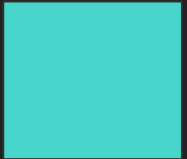


# HOE MAKEN WE TOERISME INTERACTIEVER MET BEHULP VAN AUGMENTED REALITY

Bachelorproef van Cedric Vanhaegenberg - NMD  
Arteveldehogeschool, Gent

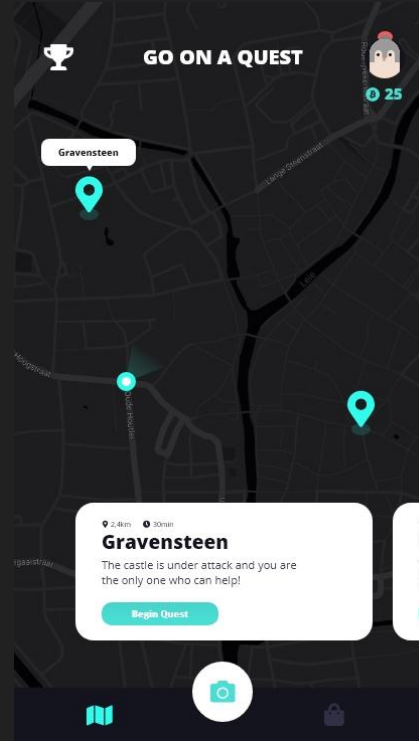
# Inhoudstafel

- De applicatie en het dashboard
- Onderzoek
  - Doelstellingen
  - Methoden
- Productieproces
  - Scope
  - Database Model
  - Models
- Demo



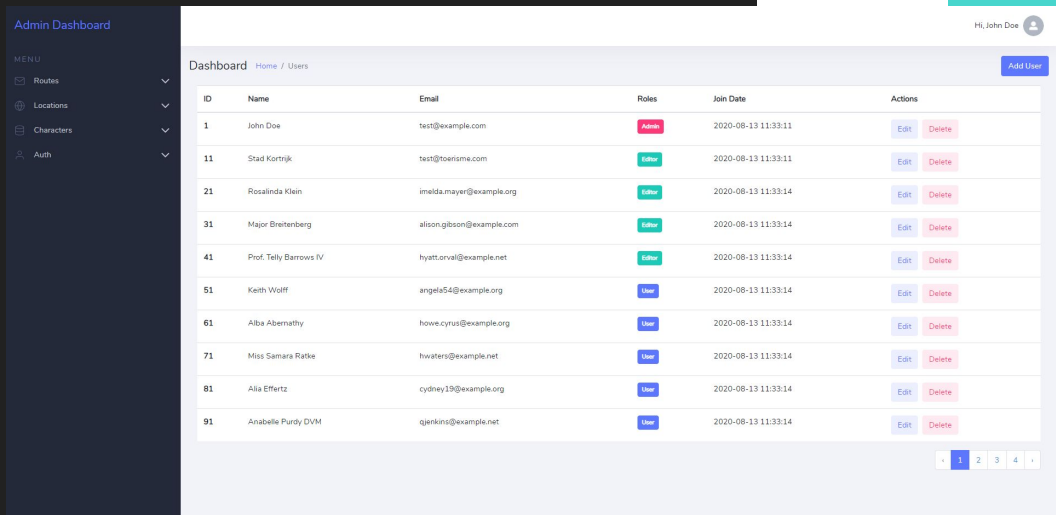
# Adventure Routes

- Scavenger Hunt met AR
- Geolocation
- Gamification (Character - Classes)
- Item Shop



# Admin Dashboard

- Beheer van routes en data
- Rollen
- CRUD
- Grafiek



The screenshot displays an Admin Dashboard with a dark sidebar and a light main content area. The sidebar contains a 'MENU' section with links for Routes, Locations, Characters, and Auth. The main content area shows a 'Dashboard' header with a breadcrumb 'Home / Users' and an 'Add User' button. Below this is a table listing users with columns for ID, Name, Email, Roles, Join Date, and Actions. The table contains 10 rows of user data. The 'Roles' column shows 'Admin' for the first user and 'Editor' for the next three, followed by 'User' for the remaining six. The 'Actions' column for each row contains 'Edit' and 'Delete' buttons. At the bottom right of the table is a pagination control showing '1' of 10 items.

Admin Dashboard

MENU

- Routes
- Locations
- Characters
- Auth

Dashboard Home / Users Add User

ID	Name	Email	Roles	Join Date	Actions
1	John Doe	test@example.com	Admin	2020-08-13 11:33:11	Edit Delete
11	Stad Kortrijk	test@toerisme.com	Editor	2020-08-13 11:33:11	Edit Delete
21	Rosalinda Klein	imelda.mayer@example.org	Editor	2020-08-13 11:33:14	Edit Delete
31	Major Breitenberg	alison.gibson@example.com	Editor	2020-08-13 11:33:14	Edit Delete
41	Prof. Telly Barrows IV	hyatt.ornall@example.net	Editor	2020-08-13 11:33:14	Edit Delete
51	Keith Wolff	angelaf54@example.org	User	2020-08-13 11:33:14	Edit Delete
61	Alba Abernathy	howe.cyrus@example.org	User	2020-08-13 11:33:14	Edit Delete
71	Miss Samara Ratke	hwaters@example.net	User	2020-08-13 11:33:14	Edit Delete
81	Alia Effertz	cydney19@example.org	User	2020-08-13 11:33:14	Edit Delete
91	Anabelle Purdy DVH	qjenkins@example.net	User	2020-08-13 11:33:14	Edit Delete

1 2 3 4

---

# Onderzoek

## Doelstellingen

- Toepassingen en huidige status van **Augmented Reality** **Gamification** en zijn technieken
- Verschil tussen Web AR en Native AR



# Onderzoek

## Onderzoeksmethode 1: Literatuurstudie

- Wat is Augmented Reality?
- Verschillende manieren van tracking
- Welke devices worden er gebruikt?
- Welke toepassingen heeft Augmented Reality?



# Onderzoek

## Onderzoeksmethode 2: Interviews Erfgoed bezoekers

- Erfgoed bezoekers zijn op zoek naar nieuwe unieke ervaringen
- Educatief en informatief

---

# Onderzoek

## Onderzoeksmethode 2: Interviews Scan4Stories

- Fauve Vanoverschelde - Researcher Howest
- Tetra Project - Vlaio & Howest (DAE)
- **Gamification** is belangrijk





---

# Onderzoek

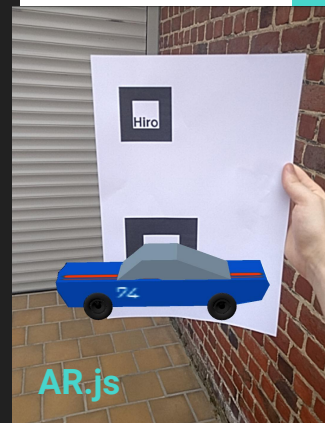
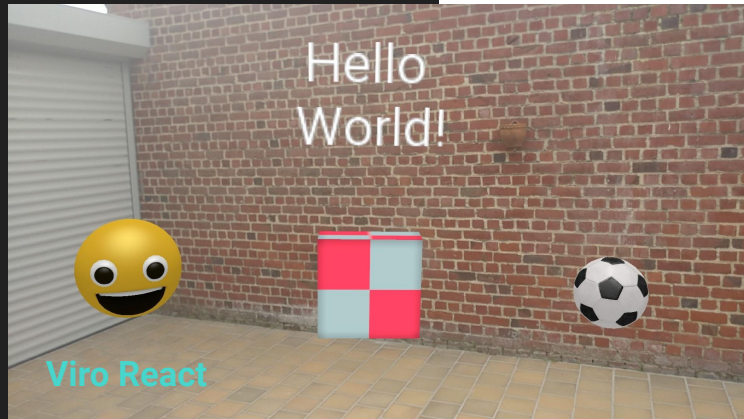
## Onderzoeksmethode 3: Benchmarks

- WebAR
  - AR.js, 8th Wall Web
- NativeAR
  - Unity (Vuforia), Viro React

# Onderzoek

## Onderzoeksmethode 4: Code Prototypes

- AR.js
  - Location & Marker based
- Viro React
  - Veel mogelijkheden





# Productieproces

## Technische scope

- **Adventure Routes - Mobile App**
  - React Native
  - Viro React
  - Lottie (Animaties)
  - Styled Components
  - Axios
  - Google Maps API



# Productieproces

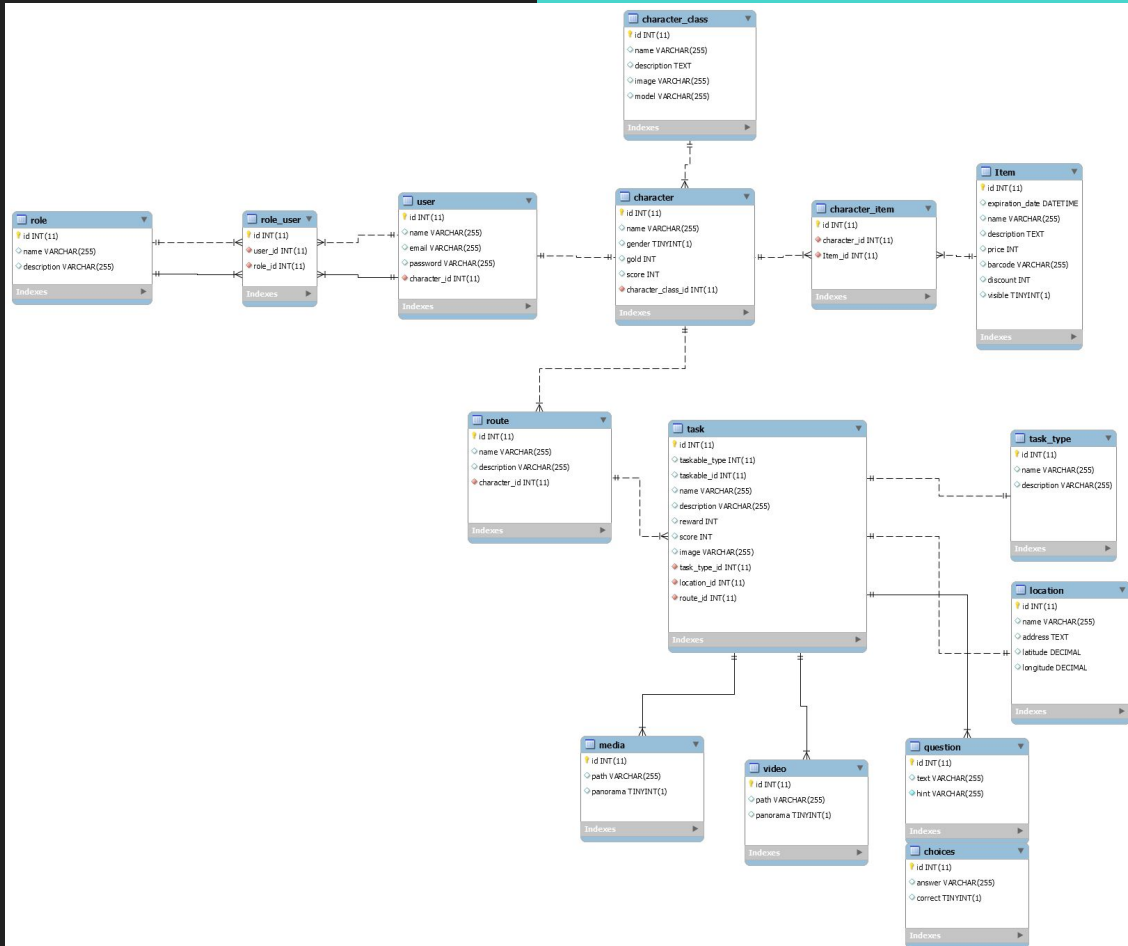
## Technische scope

- **Adventure Routes - Admin Dashboard**
  - Laravel met Blade Templating
  - MySQL Database
  - Volledig CRUD - Custom styling
  - Mapbox API
  - Laravel Faker & Geocoder
- **Adventure Routes - API**
  - Laravel API met resources
  - Laravel Passport

# Productieproces

## Database Model

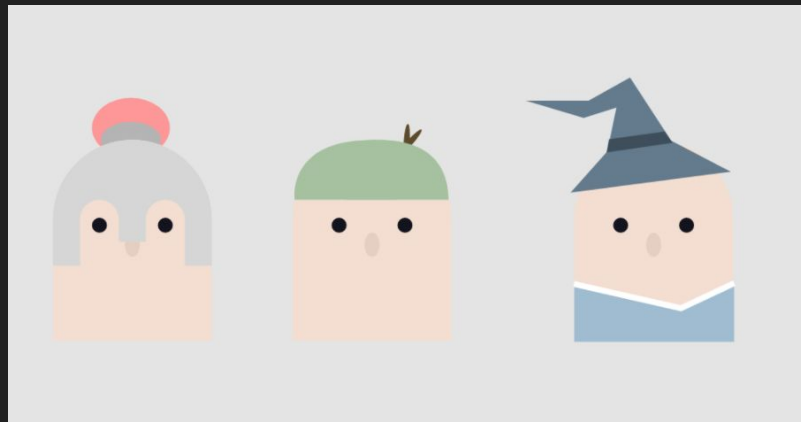
- Polymorfisme
  - Tasks
  - Task Types



# Productieproces

## Character Models

- **2D Illustraties - Character Classes**
  - Illustrator
  - Drie types (Knight, Ranger & Mage)
- **3D Models - Character Classes**
  - Sketchfab
  - Texturing in Blender
  - UV Mapping in Blender



# Productieproces

## Code Snippets

```
import Geolocation from '@react-native-community/geolocation';

const PERMISSION_DENIED = 1;
const POSITION_UNAVAILABLE = 2;
const TIMEOUT = 3;

export default usePosition = () => {
  const [position, setPosition] = useState(null);
  const [error, setError] = useState(null);

  const options = {
    enableHighAccuracy: true,
    timeout: 10000,
    maximumAge: 0,
    distanceFilter: 1, // Minimum distance before Location update
  };

  const onChange = async ({ coords }) => {
    await setPosition({
      latitude: coords.latitude,
      longitude: coords.longitude,
      heading: coords.heading, // User facing direction,
      speed: coords.speed // Velocity of device (m/s)
    });
  };

  // Catch Geolocation errors by status code
  const onError = (error) => {
    switch (error.code) {
      // Permission request has been denied
      case PERMISSION_DENIED:
        setError(error.message);
        break;
      // Location service of device is turned off
      case POSITION_UNAVAILABLE:
        setError(error.message);
        break;
      // Timeout when fetching for Location
      case TIMEOUT:
        setError(error.message);
        break;
      default:
        return;
    }
  };

  useEffect(() => {
    if(!Geolocation) {
      setError('Geolocation API is not available');
      return;
    };

    watcher = Geolocation.watchPosition(onChange, onError, options);

    return () => watcher && Geolocation.clearWatch(watcher);
  }, []);

  return {
    position,
    error
  }
}
```

```
// Permissions
// Check if Location permission have been granted
checkPermission = async () => {
  const status = await PermissionsAndroid.check( PermissionsAndroid.PERMISSIONS.ACCESS_FINE_LOCATION );
  setIsGranted(status);
}

checkRoute = async () => {
  let status = await _routeService.checkRouteStatus();
  setOnRoute(status);
}

getRouteInformation = async () => {
  if (onRoute && onRoute.status) {
    const { data } = await _routeService.getRouteById(onRoute.currentRoute);
    setCurrentRoute(data);
    setIsLoading(false);
  } else {
    const { data } = await _routeService.getRoutes();
    setRoutes(data);
    setIsLoading(false);
  }
}

onRouteStart = (item) => {
  navigation.navigate('Detail', {
    route: item
  });
}

onRouteCancel = async () => {
  console.log('Clicked');
  await _routeService.cancelRoute();
  setCurrentRoute({});
  setOnRoute({});
}

calculateDistanceNearTask = () => {
  // Let distance = getDistance(position, currentRoute.tasks[onRoute.currentIndex].Locations.coords);
  let distance = getDistance(position, {
    latitude: 50.84269204,
    longitude: 3.21211998,
  });
  return distance;
}

watchGeolocation = () => {
  return new Promise((resolve, reject) => {
    let distance = calculateDistanceNearTask();
    if (distance <= 10) resolve(true);
  });
}
```