

HoGent

BEDRIJF
EN
ORGANISATIE

Opdracht: Wishlist app

Native Apps II: mobile apps voor Windows

Opdracht: Wishlist app

Niet-periodegebonden evaluatie

Omschrijving

Algemeen

Dit jaar is het de bedoeling om een wishlist app te maken. Deze app moet het aankopen van een cadeau vergemakkelijken. We kennen allemaal de problemen die hand-in-hand gaan met het kopen van cadeaus: niet weten welk cadeau je moet kopen, hopen dat je geen dubbel cadeau hebt gekocht, etc. Met deze app willen dit probleem oplossen. Je kan de app een beetje vergelijken met het "aanleggen van een geboortelijst".

Er wordt gewerkt in teams van 3 personen. In de Chamilo cursus via cursusgroepen dien je zich in te schrijven in een cursusgroep.

Een gebruiker (na registratie en inloggen) van de app kent 2 rollen: de "ontvanger" en de "koper". Zo kan de gebruiker een wishlist voor zichzelf aanmaken (ontvanger), maar kan hij ook de wishlist van één of meerdere vriendelijk raadplegen (koper). Een ontvanger kan uitnodigingen verzenden naar potentiële kopers om ook zijn/haar wishlist op te nemen in hun overzicht (aka. "te joinen"). Ook een koper kan een request sturen naar de ontvanger met de vraag om toegang te krijgen tot zijn/haar wishlist. De ontvanger kan deze dan accepteren of weigeren om zo al dan niet toegang te verlenen tot zijn/haar wishlist.

De gebruiker (koper) kan een overzicht krijgen van alle wishlist die hij heeft opgenomen. Hij kan ook kiezen voor een overzicht, gesorteerd op datum, van de aankomende events. Zo kan de ontvanger, op het moment dat hij een wens toevoegt aan zijn wishlist, kiezen voor een deadline (bvb. 25 december).

Een wens heeft o.a. een titel en een omschrijving. Om misverstanden te vermijden, kan de gebruiker (ontvanger) er ook voor kiezen om een foto toe te voegen aan zijn/haar wens. Uiteraard zijn er nog andere interessante eigenschappen van een Wens die je kan bijhouden. Een gebruiker kan meer dan één wishlist hebben. Dit is bvb. zinvol als je niet al jouw vrienden wil toegang geven tot al jouw wensen. Er wordt toegang verleend op het niveau van wishlists.

Een wens is ook van een bepaalde categorie. Een categorie kan bvb. zijn "Muziek en films" of "Keuken". Bij het bekijken van een wishlist zijn de wensen dan ook gegroepeerd per categorie.

Het moet ook mogelijk zijn om een wishlist te exporteren naar een pdf- of Word-document om mee te nemen als je gaat shoppen.

Het moet mogelijk zijn om als koper te zien bij een wishlist van de ontvanger wie er nog deelneemt aan deze wishlist. De koper kan ervoor kiezen om een wens uit de wishlist als "gekocht" aan te vinken. De andere deelnemers (andere kopers) zien dan dat deze wens als is ingenomen door iemand anders. Zo worden dubbele cadeaus vermeden. In zo'n wishlist kan je naast de deelnemers ook, in een handig overzicht, zien hoeveel cadeaus (en wat) elke deelnemer al heeft "gekocht" in deze lijst.

Gebruiksvriendelijkheid, complexiteit van de app, code, UI, onderhoudbaarheid, (basis) functionaliteit, presentatie, compatibiliteit, portabiliteit, schaalbaarheid, ... worden allemaal mee in rekening gebracht. Van de app wordt verwacht zowel op phone als desktop gebruiksvriendelijk te zijn. Zorg dus voor beide doeleinden een *gepaste UX*. Tot slot wordt er verwacht dat de app uiteraard bruikbaar als een logische wishlist app en wordt er verwacht dat je aan de hand van deze opdracht jouw kennis over dit opleidingsonderdeel voldoende bewijst.

De opdracht is bewust relatief vaag gehouden om ruimte te laten voor eigen inbreng en creativiteit. Leuke extraatjes en originele ideeën/aanvullen zijn uiteraard meer dan welkom.

Technisch

Clients

Meerdere devices moeten ondersteund worden: de device families mobile én desktop (laptop) zijn verplicht. Andere device families (IoT, Xbox, HoloLens, ...) mag uiteraard.

De app moet worden geschreven in XAML en de code-behind in C# volgens het Universal Windows Platform (UWP). De data in de app is uiteraard live data die moet kunnen worden aangepast via de app! Het is dus niet toegelaten om alles hard te coderen. Deze data wordt aangeleverd door de back-end. De app communiceert met de back-end via REST API calls (GET, POST, PUT, DELETE).

Back-end

De back-end zal een REST Service (WEB API); die bij voorkeur in Azure draait met een databank die is opgeslagen in de Azure cloud. Visual Studio Online kan je helpen met o.a. de communicatie met Azure vanuit VS, version control, collaboratie, backlog tracking, etc. De les over Services kan je hiermee helpen. Tot dan kan je werken met hard-coded objecten. Azure is zeker en vast optioneel. De enige technische vereiste van de werkende REST Service is, geschreven zijn in ASP.NET (Core).

Uitbreidingen

Uiteraard zijn er voor uitbreidingen en creativiteit bonuspunten te verdienen. Het is dus niet de bedoeling jouw app te beperken tot de vereisten maar verder te gaan dan dit. Een basisfunctionaliteit is echter een must. Deze opdracht leent zich werkelijk tot vele uitbreidingen en creativiteit...

Voorbeelden van uitbreidingen zijn:

- Push notification wanneer geabonneerd op wishlist met deadlines en deadline nadert terwijl je nog niets hebt "gekocht"
- De gebruikerservaring uitbreiden naar de pols van de gebruiker door communicatie met de smartwatch/activity tracker Microsoft Band te doen. Zo kan je bvb. enkele dagen voor een event/deadline een reminder sturen.
- Inbouwen van reminders
- Social media integration (bvb. Facebook, ...)
- Speech recognition integration
- Cortana: Add "DVD The Hunger Games to Wishlist 'NYE' "
- Events toevoegen aan de kalender van de gebruiker
- ...

Evaluatie

Algemeen

Deze opdracht telt mee voor 100% van het eindtotaal voor dit opleidingsonderdeel (niet-periodegebonden evaluatie). Fraude of bedrog resulteert onherroepelijk in 0/20 voor deze opdracht waarbij de procedure conform het (F)OER zal gevolgd worden.

Je wordt op verschillende zaken gequoteerd. Dit zijn o.a. een aantal specifieke dingen naar de gebruikers toe: gebruiksgemak, aantrekkelijkheid, functionaliteit, flexibiliteit, etc. Maar ook naar code toe: oa. uitbreidbaarheid, onderhoudbaarheid, verschillende technologieën, complexiteit van de code, compatibiliteit, schaalbaarheid, aantal ondersteunde devices, etc.

De score "voldoende" behalen 'kan' op voorwaarde dat je met jullie project bewijzen over de nodige kwaliteiten en capaciteiten te beschikken om UWP apps correct te schrijven en dit volgens de geziene technieken. De app dient dus o.a. voldoende complex te zijn. Een app waarvan de basisfunctionaliteit aanwezig is, maar "slecht" is geprogrammeerd, kan ook in een onvoldoende resulteren.

! Groepswerk: je werkt in teams van drie en verdeelt het werk uiteraard evenredig over de teamleden.

Inleveren

Jouw project dien je in te leveren als een .zip file. Slechts *één iemand* van het team/groep moet opleveren via de opdrachten module Chamilo. Dit is uiteraard wel een gedeelde verantwoordelijkheid. De zip heeft je als naam: *GroepX_naam1_voornaam1_naam2_voornaam2_naam3_voornaam3.zip*. Omdat de zip anders te groot wordt, hoeven de packages niet mee gekopieerd worden. Deze worden nl. automatisch gerestored via NuGet (test dit zelf eerst eens!). De **deadline** voor opladen is 'voor' zondag 17 december 2017 om 23u59 (onder voorbehoud van wijzigingen door verschuiving van de lessen). Zie Chamilo voor laatste richtlijnen mbt. de deadline!

Presenteren

Presenteren gebeurt in week 13 tijdens de reguliere contacturen. Het presentatierooster wordt in de loop van het semester online geplaatst. Gelieve dit te respecteren. Meer details over de inhoud en praktische organisatie van de presentatie volgen nog via aankondigingen later tijdens het semester.

Tweede zittijd (3^{de} examenperiode)

Wordt aangevuld na de eerste zittijd bij een volgende update van dit document (v.2.0).

Laat je creativiteit de vrije loop. Verbaas en verbluf!
Veel succes!