

Feasibility Study: Snapfeet Marketplace

Date: 24 mag 2024

First decision: Two different apps	1
From template or from the existing code base	1
How to handle different marketplaces	2
Fork	3
Activities involved by the forking	3
Landing page	3
Design of the new landing page	3
Remove the QR reader	3
Save for later	3
Current marketplace view	3
Multiple Users	4
Timetable	6

First decision: Two different apps

We decided to keep two separate app:

1. Snapfeet
2. Snapfeet Marketplace (NAME TO BE DECIDED)

Snapfeet will stay the same: see current version in the App Store 3.1.6

This will be used in traditional B2B sales (e.g. Spinglemove, Dike, U-Power ...)

Snapfeet Marketplace will be a new app in App Store.

This will be used in new B2C business.

From template or from the existing code base

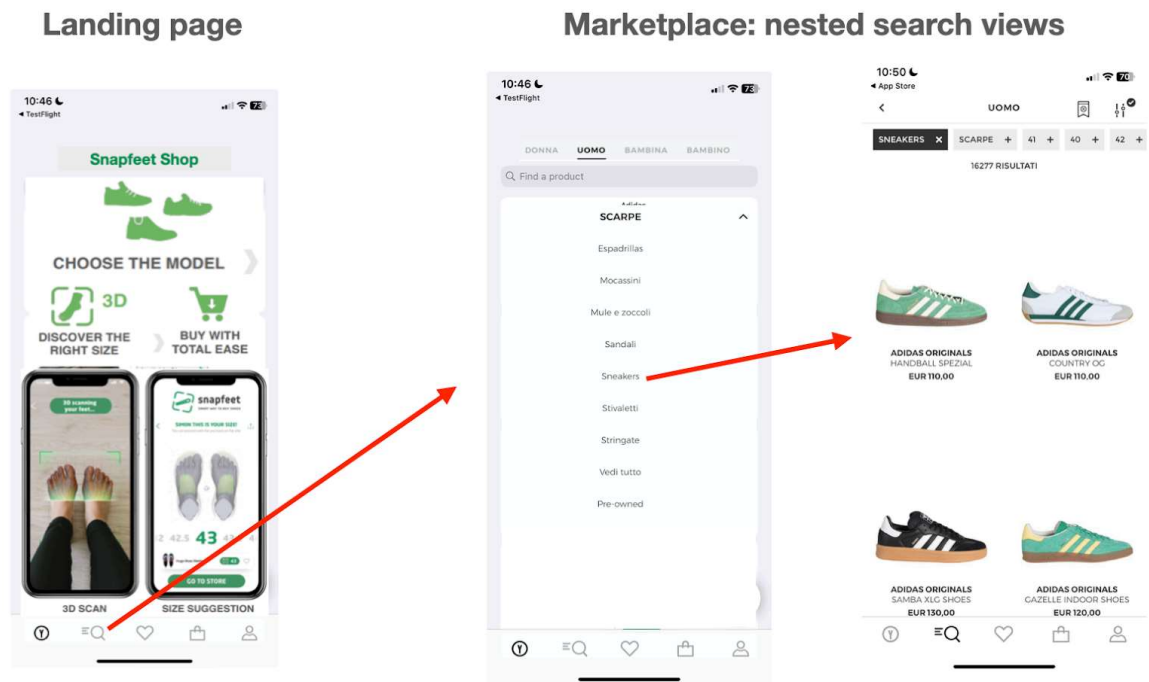
From template:

- PROs:
 - nice interface
 - new look and feel
 - Snapfeet Marketplace can look different from Snapfeet
- CONs
 - starting from a template means rewriting Snapfeet from scratch

- too high uncertainty about how to import the foot scan process: unclear is how long it will take to reconstruct the full process in the template paradigm

Therefore we decide to fork from the current Snapfeet project:

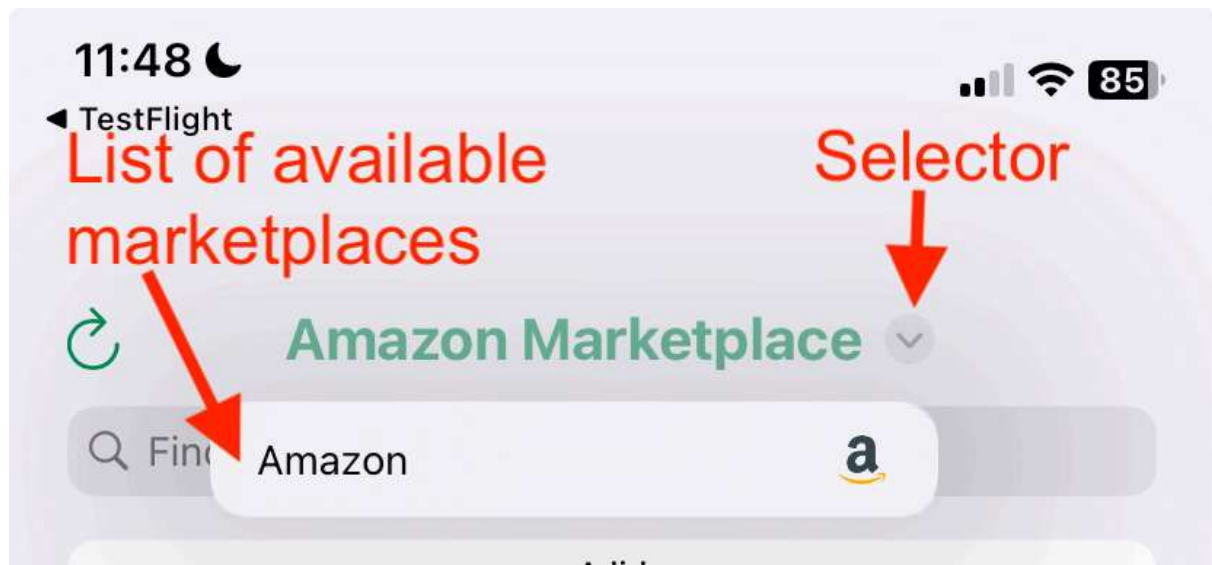
- the Snapfeet home page will become the landing page
- the Snapfeet QR code reader will be hidden
- the current marketplace will become a set of view starting from a macro filter all the way to the lists of products



How to handle different marketplaces

Nowadays we are dealing with Amazon, in the future we might affiliate to other marketplaces.

Before the user starts to look for shoes, he needs to choose what marketplace to search:



Fork

We decided to fork the Snapfeet project in two independent projects:

1. Snapfeet (the original)
2. Snapfeet Marketplace.

Activities involved by the forking

- Repository
- Backend
- Endpoints
- New XCode project
- New app in App Store (prepare graphics, texts, ...)

Landing page

First of all remove the current Home Page that displays Spinglemove shoes.
Put a static graphic for now

Design of the new landing page

We need to design the new landing page:

1. what do we want to display
2. make the graphic design
3. import it on Snapfeet Marketplace

Remove the QR reader

For now we can just hide it by removing the icon

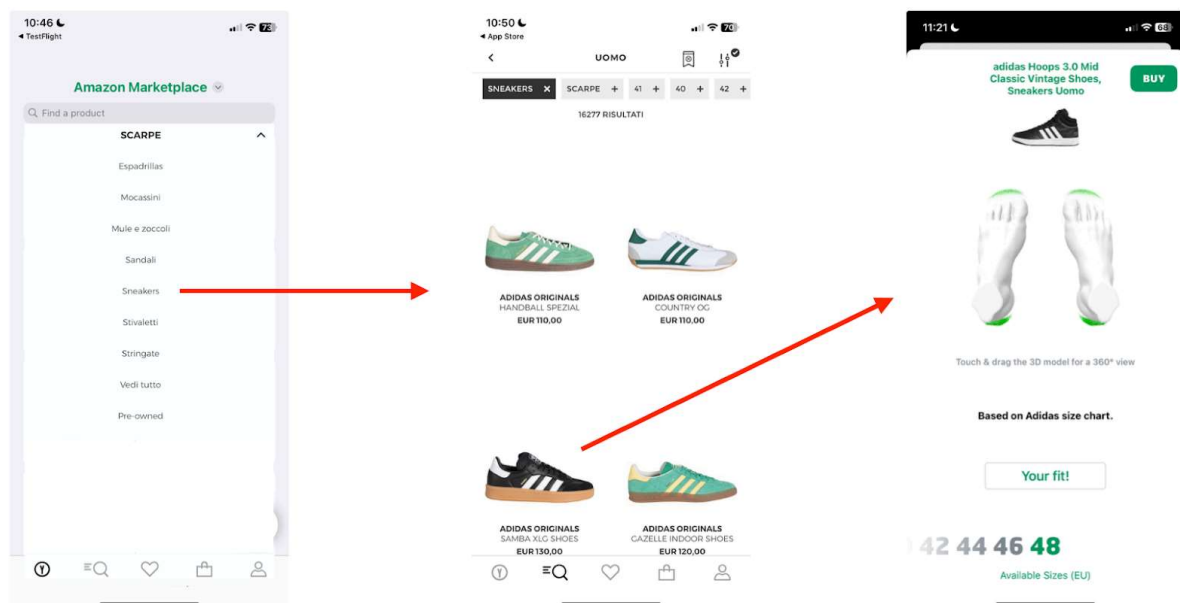
Save for later

The user might want to mark a product for later inquire: by tapping on a “heart” icon, the product will be put on a page of preference

1. product page must have the “heart” symbol
2. we need to design a “Preference” page

Current marketplace view

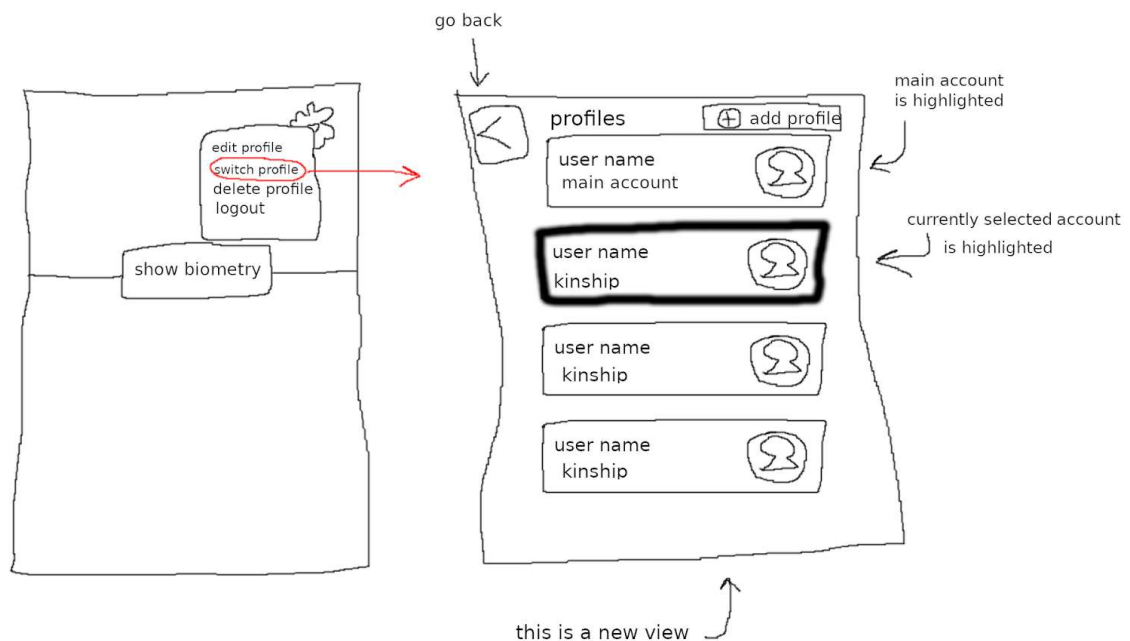
It is going to be a nested set of view



Macro list of activities:

- how to store the proper information in the backend in order to allow this kind of search from the app
- implementation of the filter engine in the app
- graphic design of the pages
- implementation of the views in the app

Multiple Users



Request:

GET <https://webservice.trya.it/testing/api/v3/profiles>

Response:

```
{
  [
    {
      "birth_date": "2024-05-27",
      "first_name": "first_name",
      "gender": "man",
      "id": "f8c86224b78541b9a720af92149fa400",
      "kinship": "self",
      "last_name": "last_name"
    },
    {
      "birth_date": "2010-01-01",
      "first_name": "first_name",
      "gender": "man",
      "id": "a707edeaca2140efa52b4112298dcb6f",
      "kinship": "son",
      "last_name": "last_name"
    }
  ]
}
```

The list of endpoint declarations is here:

```
// Trya/Managers/APIManager/APIManager.swift

protocol APIManager {
    func checkBiometryStatus(id userId: String) -> Observable<BiometryStatus>
    func createBiometry(for userId: String, with data: CreateBiometry) ->
    Single<Result<Biometry, Error>>
    ...
    func uploadBiometricScan(with scan: UploadScan) -> Single<Result<Void,
    Error>>
}
```

Timetable

DATA	ATTIVITA'	
lunedì 27 maggio 2024	Preparazione del nuovo repository	Week 1: Proto 1, con prima bozza di filtri di ricerca
	Registrazione della nuova app su TestFlight	
	Studio del filtro	
	Studio del filtro	
	Documentazione per Eugeniy: codice per interazione app multiprofilo con backend	
martedì 28 maggio 2024	Rimozione del QR reader	
	Rimozione della vecchia home	
	Creazione di una home con una immagine statica	
	TestFlight: App Testing con: nuova home, niente QR reader, pagina Market Place disponibile dalla Snapfeet Testing 3.7.0(1)	
	End-point multiutente (server)	
mercoledì 29 maggio 2024	Implementazione motore del nuovo filtro	
giovedì 30 maggio 2024	Implementazione della grafica del nuovo filtro da template	
	End-point multiutente (server)	
venerdì 31 maggio 2024	Integrazione della nuova UI nel paradigma della vecchia app	
	End-point multiutente (server)	
lunedì 3 giugno 2024	Preparazione di una nuova app su TestFlight con implementazione del filtro	Week 2: Proto 2, filtri definitivi e nuova landing page
	TestFlight: App Testing con nuova UI/UX di ricerca scarp,	

	nuovo display dei prodotti	
	Test della nuova versione di TestFligh	
	Adattamento grabber catalogo per nuovo firltro	
martedì 4 giugno 2024	Bug fix	
	studio dei feedback provenienti dai test	
	implementazione delle migliorie segnalate	
	Adattamento grabber catalogo per nuovo firltro	
mercoledì 5 giugno 2024	implementazione delle migliorie segnalate	
	implementazione landing page	
giovedì 6 giugno 2024	implementazione landing page	
	Adattamento grabber catalogo per nuovo firltro	
venerdì 7 giugno 2024	TestFlight: App Testing con feedback e miglioramenti + nuova landing page	
lunedì 10 giugno 2024	Porting del multiutente	Week 3: Proto 3 con multiutente (importazione del lavoro di Eugeny)
martedì 11 giugno 2024	Porting del multiutente	
mercoledì 12 giugno 2024	Debug miultiutente	
giovedì 13 giugno 2024	Debug miultiutente	
venerdì 14 giugno 2024	TeestFilght: prototipo app completa	

lunedì 17 giugno 2024	Bug fix	RILASCIO FINALE
	studio dei feedback provenienti dai test	
martedì 18 giugno 2024	Bug fix	
	studio dei feedback provenienti dai test	
mercoledì 19 giugno 2024	Rilascio su AppStore	