

Dev app mobiles

Héloïse FOORT & Hippolyte MARTIN

Table of content

| | | | |
|--------------------------------|----|----|---------------------|
| What is accessibility ? | 01 | 04 | Focus on some norms |
| Why it is important ? | 02 | 05 | Demonstration |
| WCAG & the mobile RGAA context | 03 | 06 | Conclusion |



What is accessibility ?

• **Definition:** Accessibility is about removing barriers so that **everyone** can perceive, understand, navigate, and interact with our applications, regardless of their ability or situation.

Four pillars :

- Perceivable
- Operable
- Understandable
- Robust

Table of content

What is accessibility ? **01**

04 Focus on some norms

Why it is important ? **02**

05 Demonstration

WCAG & the mobile
RGAA context **03**

06 Conclusion

Why it is important ?



Legal obligation

- In Europe, the **European Accessibility Act (EAA)**
- In France, **RGAA**
- *The Risk*: exposes company to **fines**, **legal action**, and **reputational damage**



Ethical imperative

- Social inclusion
- **1 billion people** (15% of world population)
- In France, **1/5 users** with temporary or situational limitations
- Create a more **inclusive digital world**
- 1 small effort = gain of **autonomy**



Business value

- Wilder market reach = **larger audience**
- **Improved UX**
 - Better SEO
 - Higher quality code
 - Better usability

Table of content

| | | | |
|--------------------------------|----|----|---------------------|
| What is accessibility ? | 01 | 04 | Focus on some norms |
| Why it is important ? | 02 | 05 | Demonstration |
| WCAG & the mobile RGAA context | 03 | 06 | Conclusion |

WCAG & the mobile RGAA context

Relationship between the standards

- WCAG (Web content accessibility guidelines) is the DNA
- The **principles are universal** (eg: Alternative text or enough contrast) → they apply whether the content is on a **website** or a **native mobile app**

Implementation shift

- The RGAA document is often written with web technologies (HTML, CSS) in mind → A shift is needed
- Instead of alt text → android:contentDescription
- Instead of tabindex → layout structure and properties like android:focusable.
- Instead of ARIA Live Regions → view.setAccessibilityLiveRegion()

Table of content

What is accessibility ? **01**

04 Focus on some norms

Why it is important ? **02**

05 Demonstration

WCAG & the mobile
RGAA context **03**

06 Conclusion

Focus on some norms



Semantics and labels

- Use `android:contentDescription` to add context
- Use `android:importantForAccessibility = no` to skip element for reading helpers



Visual & interaction essentials

- Carrefull of **contrast and size**
- Use **complex gestures** instead of simple tap to avoid missclicks
- Use `view.setAccessibilityLiveRegion(View.ACCESSIBILITY_LIVE_REGION_PORTRAIT)` to tell the reader helper to focus any pop-up message (like form sent confirmation)



User control and error handling

- Use `android:accessibilityTraversalAfter` to read the error message **immediately after** failed input
- Make sure to have both **portrait** and **landscape** modes
- Make sure to **not trap** the focus

Table of content

What is accessibility ? **01**

04 Focus on some norms

Why it is important ? **02**

05 Demonstration

WCAG & the mobile
RGAA context **03**

06 Conclusion

Demonstration

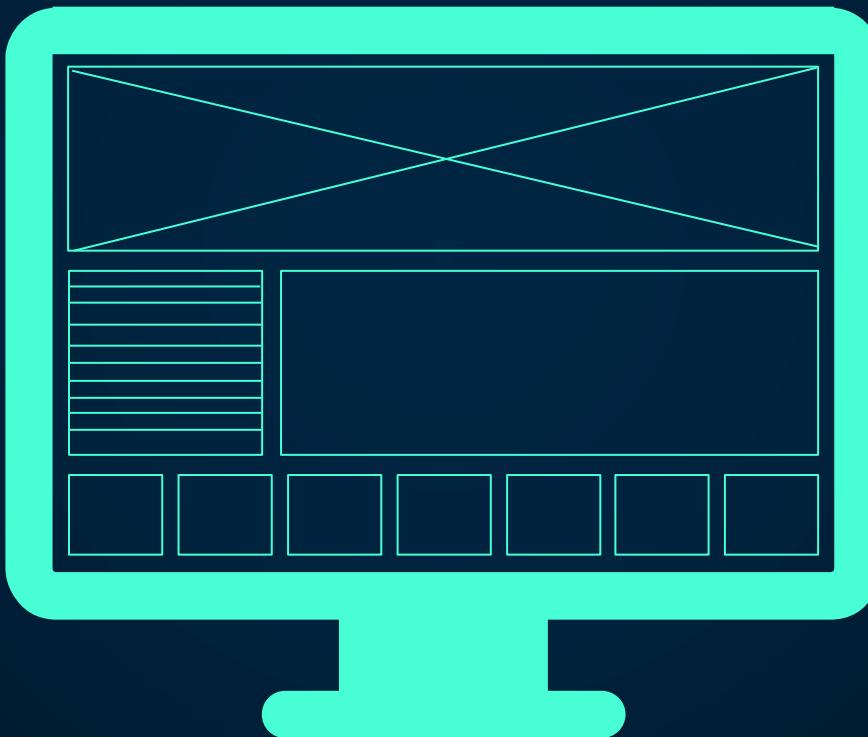


Table of content

| | | | |
|--------------------------------|----|----|---------------------|
| What is accessibility ? | 01 | 04 | Focus on some norms |
| Why it is important ? | 02 | 05 | Demonstration |
| WCAG & the mobile RGAA context | 03 | 06 | Conclusion |

Conclusion

1

Use semantics

- If TalkBack doesn't know what it is, the user can't use it.

2

Control the flow

- A broken flow leads to a lost user

3

Check the design

- If they can't see or tap it the app is unusable



Thank you !