## 

## User experience guidelines

| **Document details** | **Version** | **Author(s)** |
| --- | --- | --- |
| Draft version | 0.1 | Prithivi, Narendra |
|  |  |  |

## 

### How do people with visual impairment use any application

* Understand the user interface using an accessibility feature known as ‘talkback’ (varies as per OEM and OS type)
* The user interface elements are read from top to bottom

### Development

* **Label all UI elements**: Particularly, the elements related to the main flow. This will allow screen readers to announce what is on the screen for users who are blind or have low vision.
* **Provide alternative text for images**: This will allow users who are blind or have low vision to understand what is in images.
* Make sure the app can be used with assistive technologies. This includes screen readers, speech-to-text software, and other assistive technologies.

### Implementation

### Accessibility tags - for low vision and visual impairment

#### Generic rules

1. Ensure all relevant UI elements have the following,
   1. **Tags**: Name of the UI element
   2. **Type**:
      1. User action related UI elements to be tagged as ‘buttons’
      2. Text based UI elements to be tagged as ‘editable fields’
   3. **Status**: Indicate selection/deselection based on user action
   4. **Description** (optional): Include description for custom UI elements
2. Add ‘**focus**’ in each screen, to order the UI elements based on their relative importance
3. Skip UI elements which are ‘nice to have’ and unrelated to the ride

#### Essential tags for mobility domain (mandatory)

Following UI elements should have the accessibility tags at BAP

1. Selection of source
2. Selection of destination
3. Estimated fare
4. Type of ride (Auto/cab/bus/metro etc.)
5. Quantity of tickets
6. Booking/Confirmation of ride
7. Payment related UI elements
8. Booking/Driver details
9. Modification of booking/Cancel/update pickup,drop,add a stop
10. Ride status
    1. Ride Booked
    2. Ride started
    3. Ride ended
    4. Ride canceled

#### Functions to implement accessibility tags

Android

1. setContentDescription <https://developer.android.com/reference/android/view/View#setContentDescription(java.lang.CharSequence)>
2. setImportantForAccessibility

<https://developer.android.com/reference/android/view/View#setImportantForAccessibility(int)>

1. accessibilityFocusable

<https://developer.android.com/reference/android/view/View#setScreenReaderFocusable(boolean)>

iOS

1. self.accessibilityLabel

<https://developer.apple.com/documentation/swiftui/view/accessibilitylabel(_:)-9ek2h>

1. self.isAccessibilityElement

<https://developer.apple.com/documentation/objectivec/nsobject/1615141-isaccessibilityelement>

1. self.accessibilityElementsHidden

<https://developer.apple.com/documentation/objectivec/nsobject/1615080-accessibilityelementshidden>

### 

### Testing

* **Test the app among users with disabilities**: This is the best way to ensure that the app is accessible to everyone.
* Use readily available accessibility testing tools.

Here are some additional tips:

* **Provide captions for audio and video content**: This will allow users who are deaf or hard of hearing to access the content.
* **Provide transcripts for audio and video content**: This will allow users who are deaf or hard of hearing, or who have difficulty understanding spoken language, to access the content.
* **Provide documentation and support for users with disabilities**: This can help users to learn how to use the app and to get help if they need it.
* [Optional]: Allow users to customize the appearance of the app. This can help users to make the app more accessible to their individual needs.

## 

## Mobile accessibility checklist

To test the mobile app for accessibility

General

* Does the app have a title?
* Are headings used to create structure?
* Are bullet and numbered lists used?
* Are tables provided with headings and summaries?
* Is the color contrast sufficient?
* Are links clear and meaningful?
* Is the language of the app identified?
* Are images of text avoided?

Touch targets

* Are touch targets large enough?
* Are touch targets spaced well apart?
* Is feedback provided for all user actions?

Keyboard navigation

* Can all elements of the app be navigated to using a keyboard?
* Is the focus order logical and consistent?
* Is feedback provided for all user actions?

Screen readers

* Are all UI elements labeled?
* Is alternative text provided for images?
* Can the app be used with a screen reader?

Audio and video content

* Are captions provided for audio and video content?
* Are transcripts provided for audio and video content?

Customization

* Can users customize the appearance of the app?

Documentation and support

* Is documentation and support provided for users with disabilities?

This checklist can be used to test the app for accessibility manually, or alternatively, accessibility testing tools can be used to help identify and fix the accessibility issues.