Android Print Spooler

Write an Android application that acts like a Print Spooler with option to queue and print text, images and PDF files.

Functional Requirements

Queue list screen

On this screen the application user must be able to see all print items in the Print Spoler queue for a selected printer. Each item should have:

- file name
- file type
- current status in the print spoler (`Waiting`, `In progress`, `Done`)

The queue list must contains a dropdown with available printers, each with a name and brand model and if it is online or offline. When changing the selected printer the queue list should be updated showing the right print jobs.

The queue screen must provide a way the user to add new items (images, PDF files or text) to the selected printer queue. **Right after an item was added to the queue and the queue is empty the print job can start.**

Optional

- Implement a way of canceling and removing items from the queue
- The list items to have progress bar with the print progress

Print spoler service

Implement an Android Service with the following functionality:

1. The service should have a list of available printers. Each printer must have name, a brand or model and status indicating if it is online or offline.
2. The service should manage print spoler queue and accept print jobs.
3. The service should be able also to:
    1. Notify the `Queue list screen` about the jobs statuses
    2. Report the available printers to the `Queue list screen`

When a job from the queue is ready to be printed, simulate the print action. The simulated print action takes time equal:

Predefined print prepration time of **4 seconds** + `file size in bytes / 100 in ms`. When text is going to be printed, it takes time equal: preparation time + `text length / 100 s`

> Example 1: If file is 1MB and assume that's 1_000_000 bytes
> `preptime + 1_000_000 / 100 = 10_000 ms (4 + 10 seconds)`
> The print job should take 14 seconds

> Example 2: If text with length 200 is being print
> `preptime + 200 / 100 = 4 + 2 seconds`
> The print job should take 6 seconds

Optional

1. Extend the service to report progress of the print jobs to interested parties
2. Ability to cancel jobs **already in progress**

Solution requirements

- This needs to be a completely working Android application, supporting API24 and above
- The implementation must meet high quality programming practices regarding architecture, code design, code structure and formatting, unit tests
- While you don’t need to focus on the visual design of the app, make it in such a way, that is clear and straightforward to use.
- Compose cannot be used, use XML layouts instead.
- All other Jetpack libraries except Compose are allowed.
- For the evaluation of the solution, the code quality will be the most important factor.
- If something is not specified in the requirements, use your own judgement on how to implement it.
- As a final solution, please send us a link to a public git repo with the codebase.
- Provide us with clear instructions in English how to build, test and run the app.