

Assignment #3, part 1

- Navrhněte 3 testy pro vaši aplikaci.
- Popište, jak budou vypadat jednotlivé testovací případy.
- Navrhněte klasifikaci chyb.
- Připravte si formuláře pro zaznamenání a vyhodnocení testů (popis a statistika)

Tests

#1: YT download invalidity test

There are a lot of things that can go wrong with downloading a sound from YouTube. Luckily, the View and the Viewmodel of the YouTubeDownloadDialog are separated, allowing for easy testing by accessing members of the ViewModel.

Possible testing cases:

- empty path
- path without user access (e.g. /root)
- invalid file extension (e.g. .txt)
- empty YouTube URL
- URL that does not point to YouTube (e.g. google.com)
- invalid YouTube URL
- valid YouTube URL pointing to private/deleted video

Example of one testing case:

```
create a YouTubeDownloadDialogViewModel
fill the input fields accordingly
simulate the Download button being clicked by manually firing the DownloadCommand
check that an exception was raised and caught
check that the user was shown that something went wrong
```

#2: Show favourites precondition check

To test whether Show favourites checkbox (not to be confused with the sound-specific IsFavourite checkbox) is cosmetic only and does not change underlying data.

Possible testing cases:

When toggling to show favourites only,

- there are no sounds at all
- there are multiple sounds, but none are marked as favourite
- there are multiple sounds, but only some are marked as favourite
- there are multiple sounds and all are marked as favourite

Example of one testing case:

```
create a MainWindowViewModel
fill the AllSounds member with input sounds accordingly
toggle Show Favourites Only on
```

```
check that all sounds in VisibleSounds member are marked as Favourite
toggle Show Favourites Only off
check that VisibleSounds contain the same sounds as just after the filling
check that AllSounds and VisibleSounds contain the same sounds
```

#3: EditDialog check

To assert that the edit dialog changes are properly propagated after it finishes.

Possible testing cases:

- no changes were made, no changes are therefore propagated
- changes were made, but the Cancel button was pressed
- only the path was changed
- only the keybind was changed
- both the path and the keybind was changed

Example of one testing case:

```
create a MainWindowViewModel
create a EditDialogViewModel with an initial input sound
simulate the appropriate changes to the sound
confirm/cancel the dialog by manually firing the commands linked to the buttons
check if the result matches
check if the AllSounds member contains the sound with the newly edited values
```

Error classification

| Type | Severity | Description | Example that was encountered during development |
|-----------------------------|----------|--|---|
| UI visual | Low | A UI element does not render appropriately. | A missed comma caused two items to be rendered in the same cell of a grid layout. |
| UI functional | High | Some UI element does not fire its function properly. | A leftover line from copy-pasting Buttons caused two functions to be called after clicking a Button. |
| DB-related | Extreme | Error from the DBMS. | Forgetting to run a migration causes DB-related functions to throw unexpected exceptions. |
| Malformed input | High | Related to user input. | Forgetting to check user input in any of the input form dialogs. |
| Dependency-related | High | Something is wrong with dependencies. | Forgetting to install external runtime dependencies (as per installation guide) causes unforeseen runtime errors. |
| Front-backend inconsistency | High | Misalignment between frontend and backend. | After updating the backend model, forgetting to update the frontend using the new backend. |

Test results

Using any of the test runners that C# provides (e.g. Xunit, Nunit, ...), test results with desired success counts are already provided that way:

