## Description

Design and implement a simple WPF application using the provided project skeleton, that fulfills the following criteria:

* The app retrieves a list of presentations from <https://s3.amazonaws.com/prezi-desktop/other/Assesment/prezilist.json> and stores the necessary data locally, so that the presentations are available offline.
* In the starting view, the app shows the presentations using their thumbnail and title, and sorts them based on their last modified date starting with the most recent one.
* When a presentation is clicked a "detail view" about the selected item should appear showing their relevant details. Navigating back to the starting view should be possible.
* A status bar shows the application’s status

How to submit the completed task

* Please set up a cloud git repository of your choice (GitHub/Bitbucket/GitLab) and commit all your changes there along with the project skeleton we provided
* Make the repository available to us either making it public or adding the team directly (this might be only an option if you decide to use a GitHub repository)
* Please, put the outcome of Task 1-3 into “Documents” directory provided in the project skeleton
* You will be invited to a dedicated Slack channel. When you feel ready, please notify us, describing the details how we can access the repository containing the completed task

How can I get help

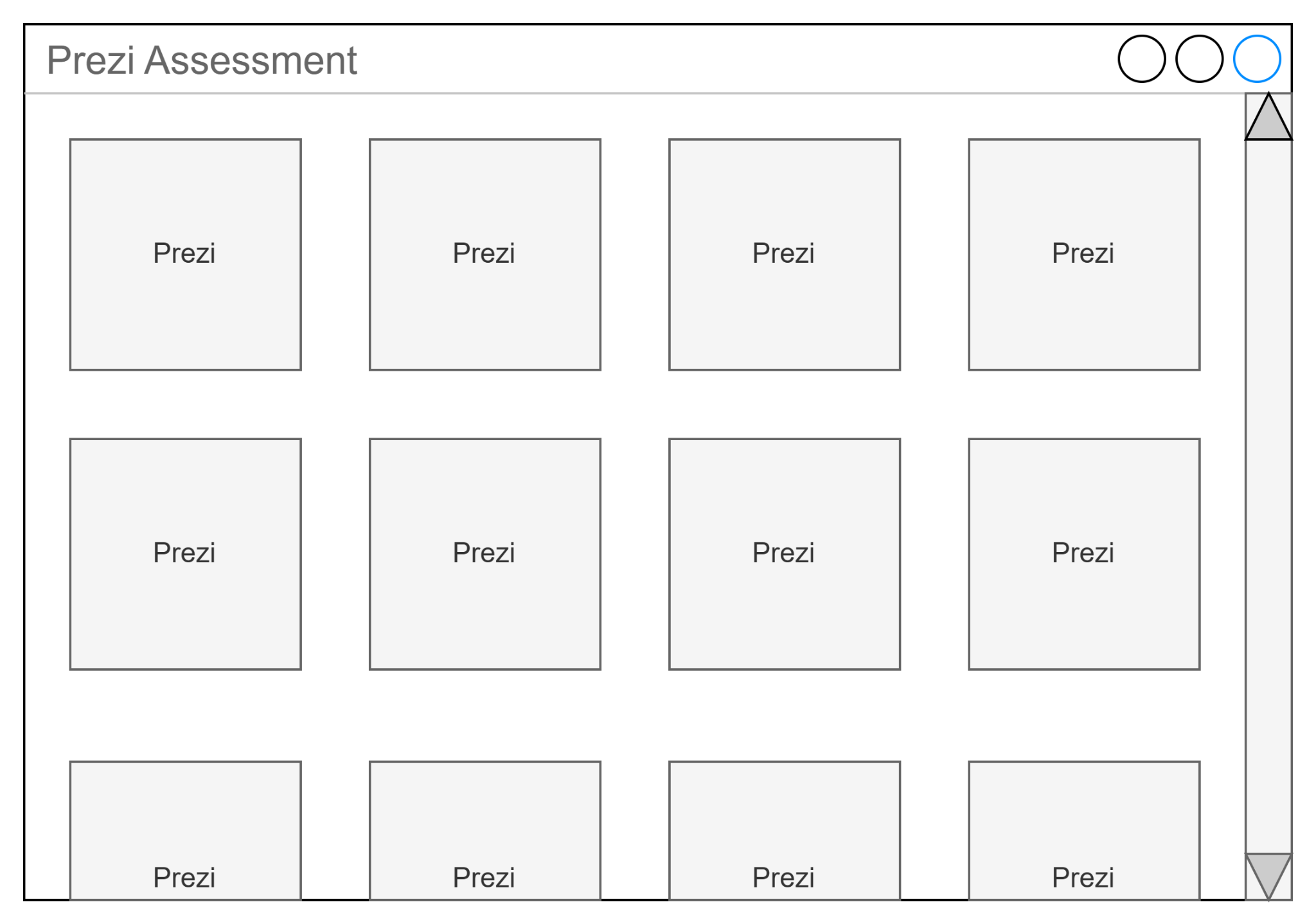
As mentioned above, you will be invited to a dedicated Slack channel. Please do not hesitate to ask us questions in case you need help or clarification about the tasks. We are happy to help

We wish you good luck and fun coding!

## Tasks

### Task 1

The layout of the starting presentations list view should be as depicted:



This view should be scrollable vertically, but horizontally the number of presentations in a row needs to adapt to the width of the view.

Please design the rough layout for the detail view as you see fit, consider what you would emphasize when looking at presentations. You can use any tool you like (e.g. hand drawn, draw.io, etc., and paste the result here).

#### Desired outcome: Detail View Layout

### Task 2

Design the architecture: think through how you would implement the requirements in the provided project skeleton. Draw the main layers and components of the implementation and the connection between them. Include the 3rd parties you may use and the specific architecture pattern. Any kind of diagram is acceptable that conveys the concepts (even a simple one of connected boxes), and you can use any tool you like (e.g. hand drawn, draw.io, Visual Studio tools etc., and paste the result here).

#### Desired outcome: Architecture Design diagram

### Task 3

Create a list of implementation tasks: break down the requirements to specific tasks, which may be converted to cards on a scrum or kanban board. Think about the steps how you would approach the implementation of the requirements.

#### Desired outcome: Task Breakdown, list of implementation tasks

Tasks Breakdown:

* Create the three main projects and have empty views in place
* Create Presentation Model
* Create Presentation Online fetcher
* Create Presentation Local fetcher
* Create Presentation Fetcher that will use Online Local and Online Fetcher
* Implement DI in Service project for registration
* Create Presentations view
* Create Presentation view
* Create Detailed View
* Create Nunits Service project
* Create Nunits App project
* Create Nunits Model project

### Task 4

Implement the requirements, while considering the following:

* What could the reasonable app statuses be that can be shown in the status bar?
* What are the error scenarios, how to handle and communicate the errors meaningfully?
* Can the solution be extended in the future? How easy would it be to extend the code?
* Can the code be tested effectively/efficiently?

#### Desired outcome: working implementation in a Git repository