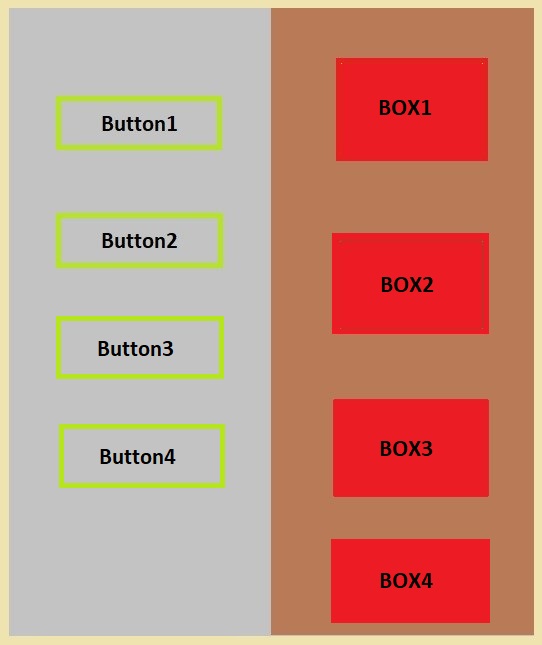
**First exercise**

Arrange the items as show in picture. Pressing the button changes color of the box item. The colors can be selected to your like.

4 points for the layout of the elements.

3 points for the method which change the color of the box.

3 points for first method modification, which return the original color of the box when the button was pressed again.



## Second exercise

Healthy Food Recipes. Possibility to choose recipes according to certain categories (soups, desserts, etc.). All recipe data is stored in JSON format. Each recipe category is stored in different program windows (React [NAVIGATOR](https://reactnavigation.org/)). There must be a way to leave a comment below the recipe. Comments are stored with [APIs AsyncStorage](https://react-native-community.github.io/async-storage/).

1. Use Navigator to navigate between different program windows. (6 points)
2. Create object array for save data about recipes. (2 points)
3. Create insert comments function. (3 points)
4. Create get comments function. (2 points)
5. Create delete comment function. (2 points)

**Third exercise**

Create ads application for computer hardware (ability to choose other topic). Using Redux to manage the state. The application must have the following functions:

* Post an ad. (3 points)
* Update an ad. (3 points)
* Delete an ad. (3 points)
* Show ads. (2 points)
* Use animated components. (2 points)

All ads are stored in array, which is initiated when the application is started. The array can be with data.

Additional (optional) application requirements:

* The data is stored in SQLITE database. Don't need to use initial array when using the database. (5 points)
* Created Login form. (3 points)
* If you select the "Remember me" function during the SIGN IN phase, when you finish working with the application and turn off it, the next time you start application you will be logged in to the same user account. (2 points)
* Each user can see all ads or own ads, it's meaning use filter function. (2 points)

**Fourth exercise**

Create an application for storing products in a database. The barcode or QR code is scanned using the application.  The information received after scanning code is storing in Firebase Realtime database.

* Use Navigator to navigate between different program windows. (4 points).
* Using barcode or QR code library for scanning product. Using camera. (6 points).
* Post data in the Firebase Realtime database. (2 points)
* Update data. (2 points)
* Delete data. (2 points)
* Show data. (2 points)
* Use animated components. (2 points)

## Final Project

Create a system consisting of the following subsystems:  
1. Create Mobile Application. (2 points a few CRUD methods, 1 points for used Api's i.e gyro sensors or camera)  
2. Create Website (or other mobile app that communicates with each other). (2 points)  
3. Connected database (can be any NoSQL database for example: firebase realtime database). (2 points)  
4. Project report. The source code is stored in GitHub or GitLab. (1.5 points for project report, 0.5 points for GitHub or GitLab).  
5. Communication between the mobile application and the website can be done through WebSockets or WebServices (Firebase). (1 point)

You choose the system theme yourself. For example: a consumer takes an order in a cafe using a smart device. At the same time, the chef in the kitchen sees the order being taken and can present the order execution stage, which the consumer sees.

The mark will depend on the subsystems created:  
5 to 6 if steps 1, 3 and 4 from the subsystems are executed and the mobile application is created with React Native or Flutter.  
7-8 if steps 1-4 from subsystems are executed and the mobile application is created with React Native or Flutter.  
9 - 10 if all steps from the subsystems are completed and the mobile application is created with React Native or Flutter.

Project report structure:  
Title page  
Report content  
Introduction (1 page)  
Practice assignment formulation: Functional, non-functional requirements.  
Practice assignment analysis: Analysis on initial data (input). Database logical design with description, if database is used. Analysis on functional requirements. Use case diagrams, scenarios.  
Software: Explanation of development tools selection; List of main software files, its purpose and relation to other files. Database physical design (if there is any) and class diagram with detailed description of each element.  
User guide: Software installation guide, including initial setup information; Description and identification of steps necessary to take in order to fulfil all functional requirements.