Ecole d'ingénieurs et d'architectes de Fribourg
Hochschule für Technik und Architektur Freiburg

EIA-FR

Cahier des charges

Behind Food

Grégory Geinoz

04/10/2021

Supervisor :
Pascal Bruegger - pascal.bruegger@hefr.ch

Client :
Samuel Fringeli - samuel.fringeli@hefr.ch

Ecole d'ingénieurs et d'architectes de Fribourg
Hochschule für Technik und Architektur Freiburg

Table of Content

1 Introduction

Behind food is a mobile application [1] used in an exhibition on sustainable development. It allows visitors to explore the hidden sides of different everyday foods through a journey of various themes related to these foods. This pathway provides access to images, videos, and texts to illustrate the characteristics of the foods concerned. These elements are updated by the team managing the exhibition, through a backend interface, and are accessible from the application through an API. In the current version of the application, a webview is loaded which displays images and videos as the user moves through the structure, but the media is not saved in the application's local storage, making it impossible to use the application offline. As the exhibition aims to operate entirely offline, it would be necessary to ensure that the data displayed is downloaded locally, with a system for updating the latest changes made on the backend by the user of the application.

2 Context of the project

The base application is a web application that uses the library zircle-ui, a simple yet clever UI with a built-in zoomable navigation through circles, displayed in an iOS application with a WebView from SwiftUI. All the content displayed, including text, images, and videos, are gotten from an API created by the client.

3 Main Goals

The main goals of the project explained during the first meeting.

3.1 Goal 1: cross-platform application

Self-explanatory title, the idea is to adapt the base iOS application to a cross-platform application. The technology that will be used will be analyzed during the analyzation phase between Cordova, Flutter or Xamarin.

3.2 Goal 2: offline compatible

The application must become offline compatible. This means that it will be capable of checking the connection of the device. If the device has no internet connection available, the app uses the local files and data initially downloaded at the first installation. If the device has an internet connection, it will check at each boot through the API if some content has been changed, hidden by a loading screen. If the content has not been changed, nothing happens, everything is up-to-date, and the user does not need to know that this check occurred, other that he/she waited during the loading screen. If some content has been changed, a pop-up will appear to let the user know that some new content can be downloaded. The user has the choice to accept this content update, if so, the application will display the loading screen again so it has time to download the new content and display everything updated, if the user declines, the actual local content will be used. As long as the application is not rebooted, this check will not occur again.

4 Secondary goals and future developments

4.1 Goal 3:

Background download, then pop-up to refresh the content when it is ready. The idea would be to use the application at any time with the actual local files. The same first check will be done, and the same pop-up will ask the user if he/she wants to download the updated/new content. The difference is that the download does not block the application.

5 Technologies

- JavaScript to adapt the script of the webapp to be offline-compatible
- xCode and SwiftUI because the base application was made for iOS
- Cordova, Flutter or Xamarin for the cross-platform adaptation
- The way to download and store the files is yet to be searched

6 Tasks

6.1 Preliminary studies:

This phase is used to find solutions and technologies to achieve the goals described above. It is also used to take charge of the base web and iOS applications for a much simpler adaptation.

6.2 Conception:

Conception of the architecture of the base application with the features added by the goals.

6.3 Implementation:

Implementation of the architecture previously created, using the technologies chosen during the analyzation phase.

6.4 Validation and tests:

System and users testing.

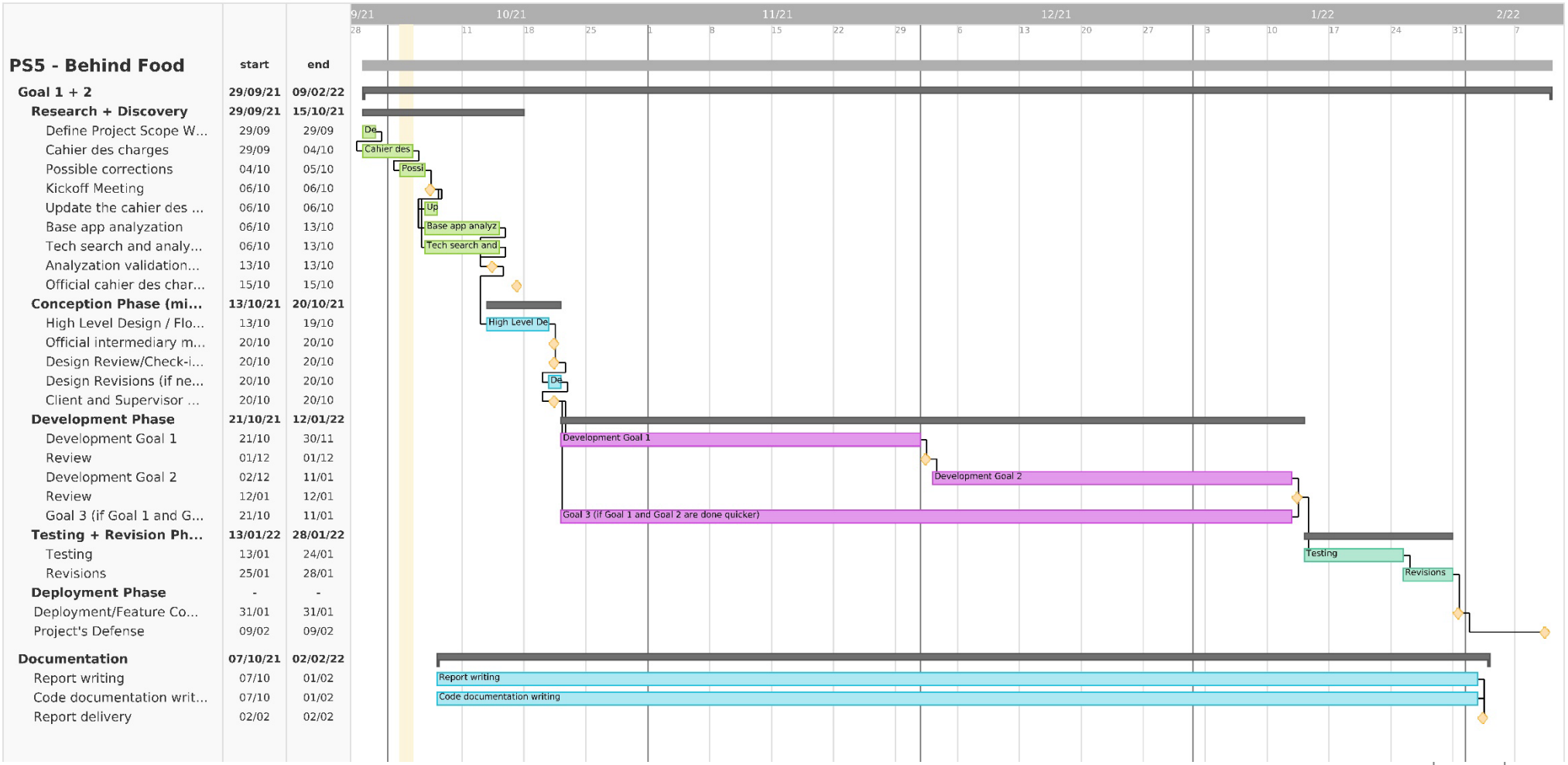
6.5 Documentation:

The report and code documentation will be written during the entire process.

7 GANTT diagram

teamgantt
Created with Free Edition

Printed: 4 October, 2021



8 References

[1] https://adelente-admin.samf.me/app