ADINT 2023/2024

Project – Part2

1 Final project general description

In the final version of the project students will implement a mobile application that allows users of an organization (for instance IST) access information about physical resources through Qr-codes.

IST students will use one application that reads QRcodes scatter in the environment (Services, restaurants, rooms, people 🤔 , …). After reading the widely available QRcodes, the applications allows the students to perform some context dependent actions.

QR code can be affixed to the following places:

- restaurants and canteens
- class rooms
- study rooms
- ~~other users can also show a QRcode in his own application.~~

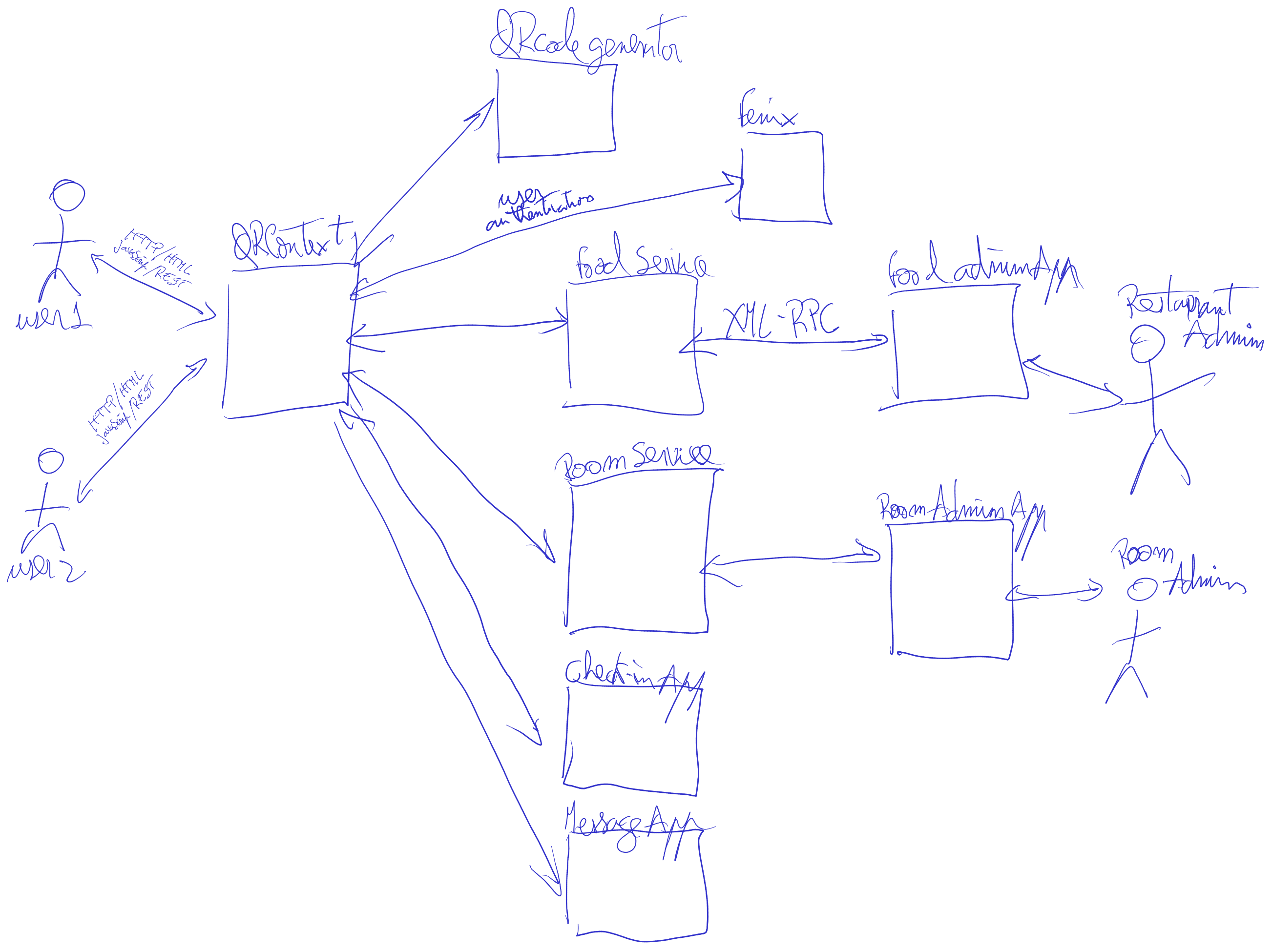
When a user with his application accesses a QRcode he is able to perform the following tasks

- Authentication
  - (F 1) login on FENIX
- restaurants and canteens
  - (F 2) see the current menu
  - (F 3) check-in in a restaurant
  - (F 4) check-out from a restaurant
  - (F 5) evaluate the meal at the restaurant that he checked-in
- Class rooms
  - (F 6) see a room schedule
  - (F 7) verify if the next class on the room is from one enrolled course
  - (F 8) check-in a class that is taking place in the room
  - (F 9) check-out a class
- study room
  - (F 10) Check-in and assign a enrolled class to a study period

    - (F 11)Check-out a study period
- other users:
    - ~~(F 10) show a personal QRCode~~
    - (F 12) Send messages to users that are on the same room
    - ~~(F 12) Send messages to users that presented their their QRCode~~

1.1 General operation

The description of the way users interact with application and executes the previous operations will be described in the second part of the project, but from the simple description of the functionalities (F 1 to F 12) it is easy to image how the mobile application will operate.



1.2 Overall architecture

The system to be implemented will be composed of various distributed components, as presented in the previous figure

The main components are:

- **QRContext –** web application that will be accessed from a smartphone to read Qr-codes and execute the user requests. This will simulate a Android/IOS mobile application. This component will only be developed in the final version of the project
- **QRCodeGenerator –** This service will generate PNG files that contains a Qr-code.
- **FENIX -** This component is external and is hosted at IST s and will provide user authentication, courses enrollment information and, on the final version of the project, rooms schedules.
- **FoodService –** this component is a web application that will store, for every registered restaurant or canteen, the menu of the day, and the user menu evaluations.
- **FoodAdminApp –** this application is a python desktop application that will allow the restaurants' administrator to create restaurants, update their menu, and see their evaluations.
- **RoomService –** this component is a web application that will store every room information and their weekly schedules.
- **RoomAdminApp –** this components is a python desktop application that will allow one administrator to insert rooms into the system and assign them a schedule.
- **Check-inApp** - this component is a web application that will store all check-in and check-out performed by the users on rooms or restaurants
- **MessageApp** - this component will store the messages that users exchange between them.

2 2nd part

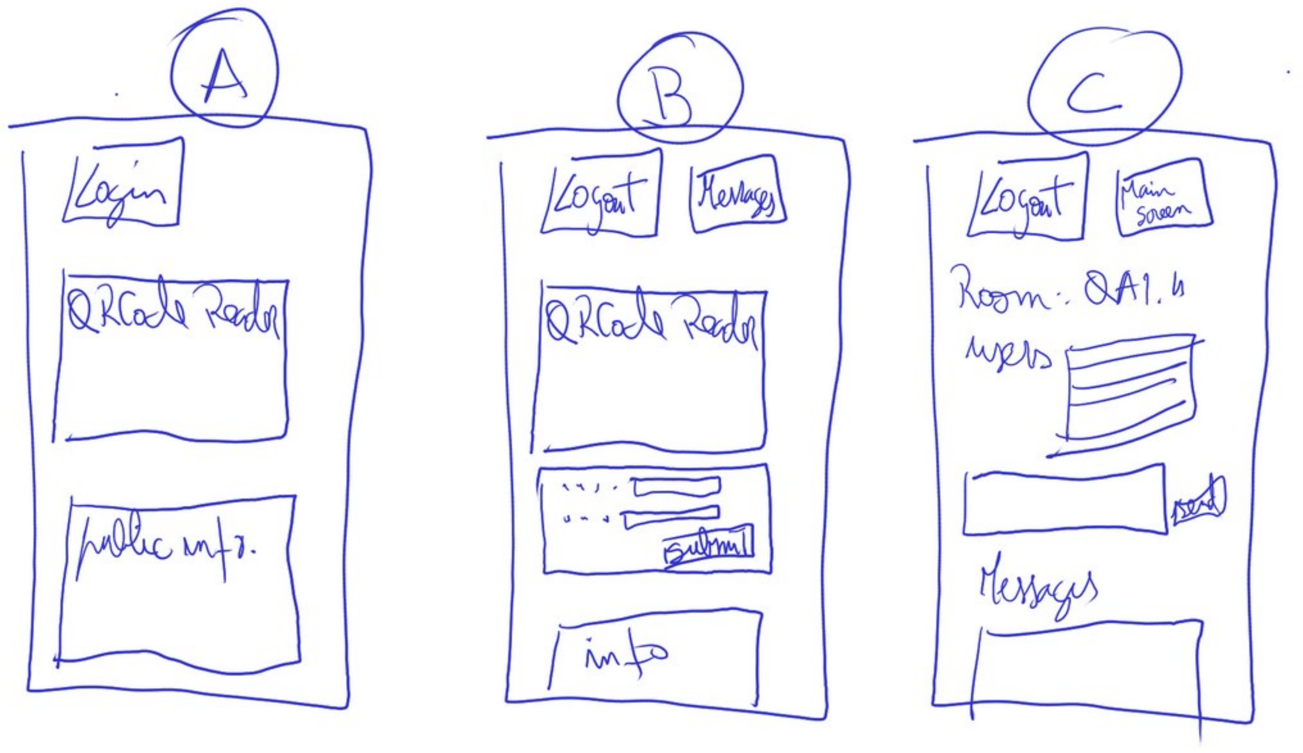
In the second part of the project students will:

- implement the QRContext server
- Link QRContext to FENIX
- Modify/extend components developed in 1st Part of the project

The user will access the system using a web based single page application, i.e. the user accesses a link on the QRContext server that provides the HTML+JavaScript for this single page application (called QRContext app. This application will make requests to the QRContext server to fetch the necessary information and perform the actions.

2.1.1 QRContext application

The QRContext application is a +simple web application, all interactions are performed in the same HTML+JavaScript page performing the minimum reload/change of pages. As such these pages should contain, besides the HTML, JavaScript that interacts with the server and changes the appearance of the application.



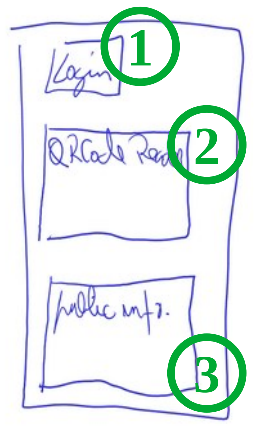
The application should have the following screens:

A) **Login screen –** this screen/page allows the user to scan Qr-codes (restaurant and rooms) to fetch public information (menu and schedule) and perform the login on FENIX

B) **Main screen** – This screen/paged allows the authenticated user to perform most of the operations (scan qr-code, check-in, check-out, ...)

C) **Message screen –** this screen/page allows a user to send a message to any other user that is in the same room

2.1.2 Login screen



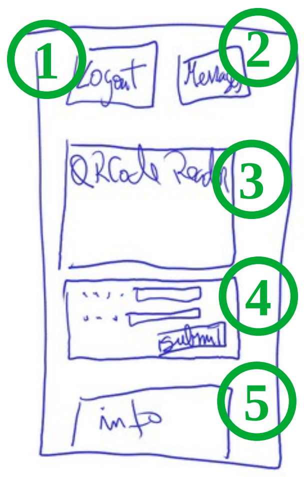
If the user is not authenticate on FENIX, the application should display the qr-code reader for the user to scan qr-codes on Rooms or restaurants (**component 2**). after the qr code read detects a qr-code, the server should be contacted to provide the correct information (schedule of the room or menu of the restaurant (that is presented in **component 3**).

If the user clicks the login button (**component 1**), the Oauth authentication process should start.

The following functionalities are implemented in this page:

- (F 1) login on FENIX
- (F 2) see the current menu
- (F 6) see a room schedule

2.1.3 Main screen



After the user authenticates on FENIX this page will be presented and the various components will allow the implementation of the following functionalities:

- (F 2) see the current menu
- (F 3) check-in in a restaurant
- (F 4) check-out from a restaurant
- (F 5) evaluate the meal at the restaurant that he checked-in
- (F 6) see a room schedule
- (F 7) verify if the next class on the room is from one enrolled course

- (F 8) check-in a class that is taking place in the room
- (F 9) check-out a class
- (F 10) Check-in and assign a enrolled class to a study period
- (F 11)Check-out a study period

For the user to logout ans send messages to others in the same room he should press the corresponding button: **component 1** for logout or **component 2** for messages.

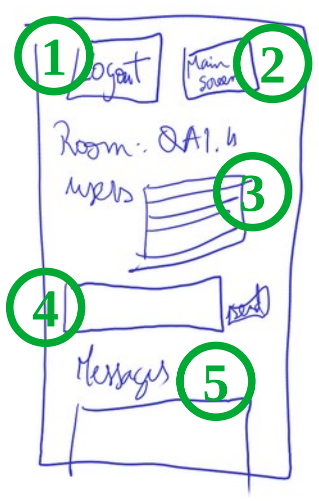
This page will show a qr-code reader (component 3) that when finding a Room or restaurante provides suitable forms for the user interaction.

The form that will allow the user to check-in/check-out, evaluate a meal is presented in **component 4**. The information about restaurant (menu) or room (schedule) is presented in **component 5**.

Components 4 and 5 should change dynamically whenever a qr-code is read or action performed (check-in/check-out or evaluation of meal)

Component 2 should only be activated when the student is in a Room.

2.1.4 Message screen



This screen will allow the user to perform the following functionality:

- (F 12) Send messages to users that are on the same room

As such it will present the following components:

- drop down list of users in the same room (**component 3**)
- form to write a message to the sleected user (**component 4)**
- list of all the received messages (**component 5**)

Components 1 and **2** allow the user to logout (**component 1**) or return to the **main Screen** (**component 2**)

2.2 QRContext Server

The server should implement 3 endpoints that provide the QRContext application HTML/JavaScript and all the endpoints for all the REST calls. This server should act as a proxy between the browser and the other components.

The only database that this server should implement is the list of users, all other information should continue to be stored in the various other distributed components.

This server should also perform all interaction with FENIX to retrieve user information (schedule and enrolled courses.

Although the JavaScript may do some validations with respect to the authorization for the execution of certain operations, the server code should also implement all those validations.

2.3 REST services

All interactions between the QRContext server and other distributed services should be done preferably using REST web services. As such students should define the endpoints (names, data transfer and error returns) closest possible to the guidelines for developing REST web-services.

2.4 Room admin App

Students should add one more option to this app so that the schedule information is retrieved from FENIX. The administrator should only provide the room id, and this component st code will contact FENIX.

3 1st part

Student can change or complement the components implemented in the 1st part in order to be able to satisfy the functionalities of the QRContext application

4 Technologies

In order to implement the requested functionalities and services, students should use Python, Flask, SQLAlchemy and FLASK-XML-RPC as already exercised in the laboratories.

With respect to the authentications of user in FENIX, students should use the provided flask server that already implements OAuth.

5 2nd Part Development guidelines

Students should implement the project in the following order:

1) Change on the FoodService/FoodAdminApp
2) Login screen
3) Main screen
4) Message screen

Associated with each screen students should implement the corresponding endpoints.

6 Final delivery

Student should submit on FENIX a zip file with the code of the various implemented components along with a simple report stating the implemented functionalities, the endpoints of the various services/components and the SQLAlchemy classes. Students should also ste in this report what was changed in the code submitted on Part 1.

Deadline: 29th October 2023 - 20h00