

Connecting the
best of you to
the best of us



Social DistAPP

With the restrictions imposed by governments due to the COVID-19 situation, we want people to find a way to keep the social distance and handle customers/visitors flow in places where people should wait in a queue in order to get in.

Our company wants to launch a system which **allows people to plan and manage their visits** to any kind of places (gyms, shops, supermarkets, theme parks, museums, etc..) and eventually pre-order or book stuff they would like to buy or try (for example, in dress shops).

The aim is to **allow the system to pick the best and more suitable** date/time to visit the place in order to maintain a constant flow of customers, while **avoiding queues** out of the physical place and **overcrowding**, we might want to allow the user to select a range of dates and/or times s/he would like to visit the place.

The system should be accessed by different kind of users:

The **Owner** must access (and register to) a website and enter the following information:

- Store name, location and description
- Items list and pricing
- Maximum number of people allowed in the shop
- Maximum/average time allowed for a person to stay in the shop

The **User** must access (and register to) a mobile app and be able to:

- View available/nearby stores
- Reserve a time slot or join a queue to visit the store
- Purchase items and reserve a time slot to pick them up
- Specify what he wants to do (for example if s/he wants to try a t-shirt, or use a specific machine at the gym, etc..)



The **Application System** must:

- Send an email containing the summary of the purchased items and a confirm of the reserved time slot
- Use an optimization algorithm or machine learning to manage the various reserved time slots in order to avoid crowding, respecting the constraints provided by each user.
- Generate a QRCode. If the store has his own lockers the user will be able to pick his
- order up by scanning the associate locker QRCode through the mobile app.

Requirements:

- Define the system architecture in detail (languages, frameworks, platforms, minimal data model, an estimation of customers flow)
- Build the application or website's frontend using the favorite framework (Angular.js, Javascript, Vue.js, React.js)
- Build the application's mobile app using the favourite native/hybrid frameworks and languages (swift, kotlin, flutter, nativescript, etc....)
- Build the application or website's backend with the favorite languages hosting it on a PaaS

It's not required to implement all of the features described above but it's mandatory to implement at least one feature for each actor (Owner, User, Application System).

If you are alone you can alternatively try to implement all feature for just one actor.

It is allowed to work within a group of maximum 3 people.

Final Presentation

The student(s) is supposed to **provide a powerpoint or video presentation (embedded in the powerpoint)** containing:

- An explanation of the **architectural choices** made, an explanation of the **reasons** behind the choices and an **estimate of the volumes** of API calls which the backend should be able to withstand.
- A **demo** of the implemented part(s) of the system.