# ProOffice interview

Contents

[ProOffice interview 1](#_Toc62222205)

[Candidate assignment 1](#_Toc62222206)

[Requirements 1](#_Toc62222207)

[Project title 1](#_Toc62222208)

[Project description 1](#_Toc62222209)

[Acceptance criteria 2](#_Toc62222210)

[Activity diagram 3](#_Toc62222211)

[Class diagram 3](#_Toc62222212)

[UI Mockups 4](#_Toc62222213)

[Directions 4](#_Toc62222214)

## Candidate assignment

The purpose of this assignment is to evaluate following candidates’ skills:

* Understanding requirements
* Designing a solution
* Writing clean and easily readable code

This assignment requires the candidate to use:

* ASP.NET Core
* C#
* Any ORM (NHibernate or Entity Framework)
* Any modern frontend technology (Angular, Vue, React)

## Requirements

### Project title

Simple booking system

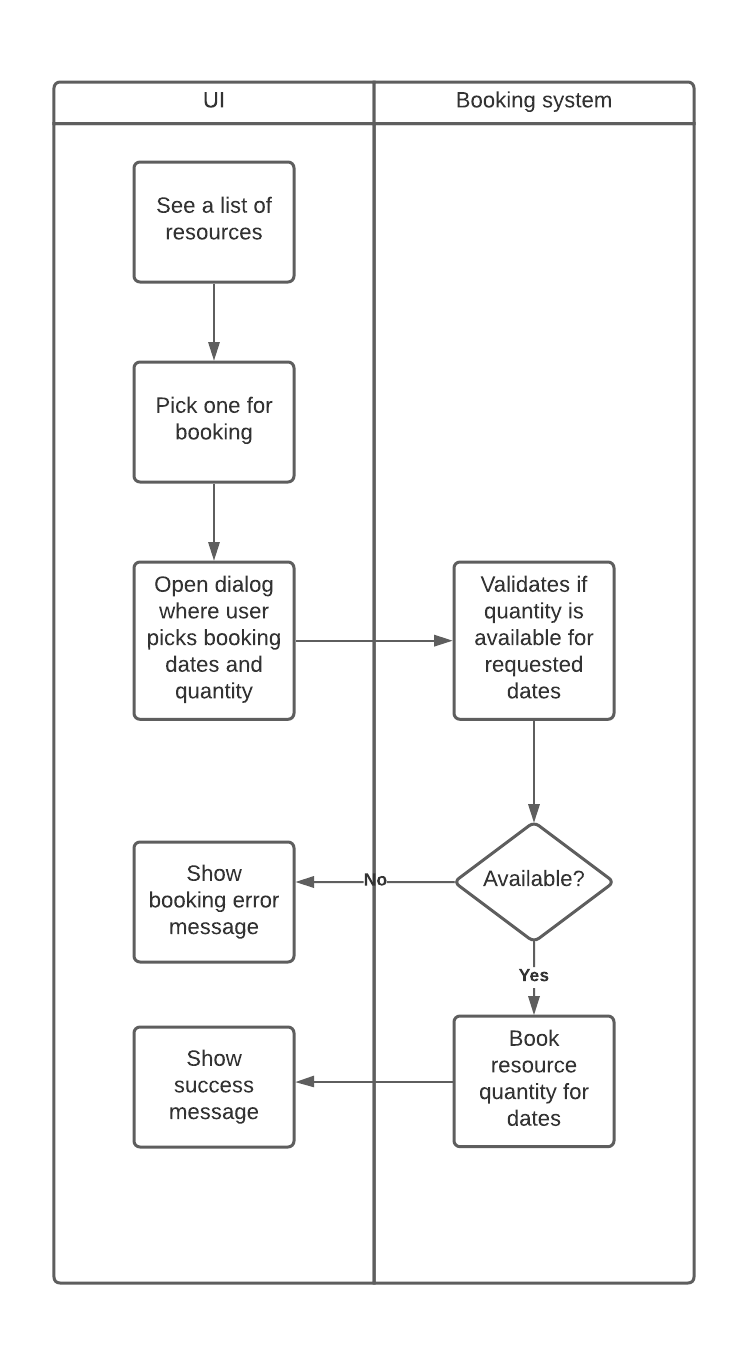
### Project description

The purpose of this system is to provide the user with simple way to book a resource for chosen time-period. The system has a list of resources that have name and total quantity (representing what is the quantity in stock for this resource). The user is presented with a list of resources on the UI. He can pick any of them and initiate a booking process.

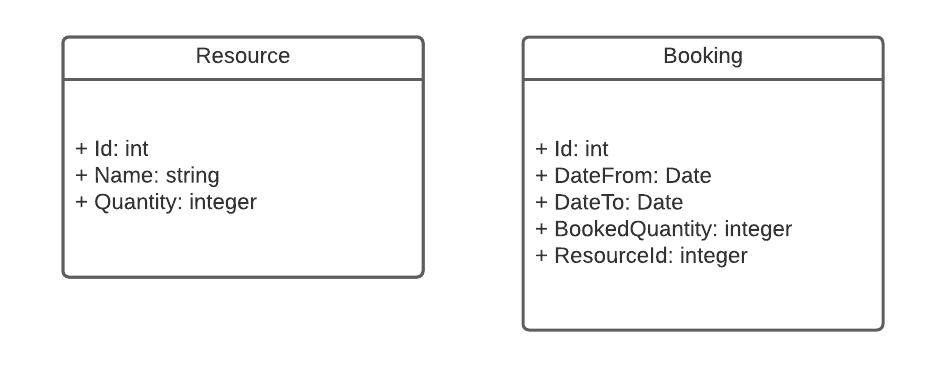
### Acceptance criteria

* User can book resources.
* Resources can be booked only if the requested quantity is available for the requested period.
* If the user tries to book a resource that is not available, he is presented with error message.
* If the user tries to book a resource that is available, after successful booking the user is presented with success message.
* The part of the validation that checks for conflicts between existing booking time ranges and requested booking time range should be designed as reusable in other places.
* At the end of the booking process, an email should be sent to [admin@admin.com](mailto:admin@admin.com). For this task email implementation should ONLY write to console “*EMAIL SENT TO* [*admin@admin.com*](mailto:admin@admin.com) *FOR CREATED BOOKING WITH ID {Id}*”. For this task there is no need for real email sending and it is only mocked with console write.

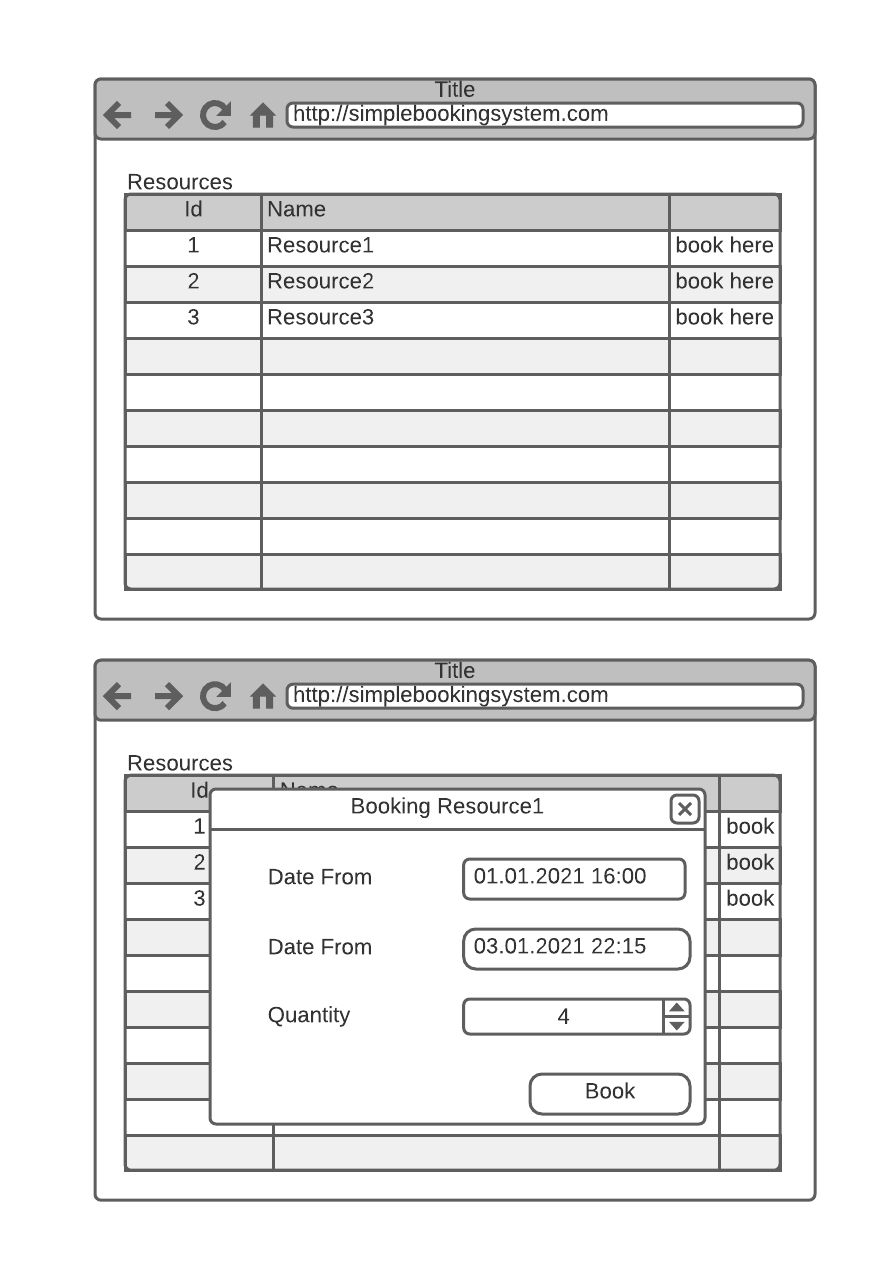
### Activity diagram



### Class diagram



### UI Mockups



## Directions

1. Use the UI framework that you are most familiar with. For a quick start use any bootstrapping script or public repo that will help you start up coding without losing time for setting up infrastructure.
   1. Example for React would be react-scripts
   2. Example for Angular would be Angular CLI
2. For simplicity use SQLite database
3. Do not spend too much time on UI looks but make it usable.
4. Use backend architecture that you are most familiar with.
5. Before starting to do any coding, plan and ask questions.
6. Be mindful of the task deadline.