Description of the situation: the company is going to develop a complete analogue of **booking.com**

**The task:**

**1. List 5-10 key areas of testing that it makes sense to automate on this project.**

1 – Multi-environment   
Do for testing on various panels

format of: languages, currency, spelling, date

Graphical user interface, application

Description automatically generated Text

Description automatically generated  
Graphical user interface, text, application, email

Description automatically generated

Calendar

Description automatically generated Calendar

Description automatically generated

2 – Combinatory techniques

Do to check all possible combinations in the input parameter field  
Graphical user interface, text, application, email

Description automatically generated Calendar

Description automatically generated Graphical user interface, text, application

Description automatically generated

so that our product does not contain errors and can handle various combinations

3 – Performance testing   
To do to test the resilience and reliability of our site under a specific workload. For example, how fast the system will work under a certain load (in the form of the number of users) for a certain period, the absence of restarts under such loads

Such checks are shared to 1) understand that the site can withstand the "normal" load, 2) at the time of the rush hour or longer discount periods, the site worked correctly

4 – Basic security testing   
using special tools, it shows the threats that are on the site.

On our website, this is important because personal information may be leaked. On the site, when registering, the user fills in the fields with the name and surname, telephone number, mail and payment data. The first thing which is important do not have a customer data in the URL after the registration/adding payment

5 – Smoke test for large/small systems  
checking the main functionality of the site. For booking.comyou need to automate user capability checking: **-** creating a new user **-** authorization of an existing **-** booking apartments/cars

6 – Applications without human interface   
Do to check databases. The site stores a lot of information (personal data, payment method data, booked hotels/cars, saved places)  
Also we can check an API calls.

In the booking we can automate test for checking interaction with the payment provider

7 – Long routing operations  
Operations which are time consuming and repeated should be automated with tests.   
In our case we can create a specific test for user registration. We can check the different sets of data for approving that we have all validation

8 – Standard functionality  
For the booking.com we can create a test for checking a possibility for paying your booking.

**2. What are the 3-5 most significant testing automation risks in such a project? For each risk, indicate: the nature of the risk, the probability of the corresponding situation (from 1 to 10 points), the severity of the risk (from 1 to 10 points) and mitigation actions.**

1 – underestimating the potential of staff in certain test scenarios  
when we overestimate the strength of a team member and he does not complete (on time) this task

the probability of the corresponding situation - 4  
the severity of the risk - 5  
mitigation actions - if during the execution of the task it is observed that a person does it for too long, you can connect other team members to solve the problem

and build on experience in the future

2 – Applaying a new tools   
when we add new tools to improve the product (improve performance)

the probability of the corresponding situation - 4  
the severity of the risk - 7  
mitigation actions - before adding a new technology/tool, we should create a separate task for checking how we can apply it for the further estimation

3 – Changing an existing API routes  
when our back-end side or third party services will change the routes for an existing API

the probability of the corresponding situation - 2  
the severity of the risk - 6   
mitigation actions - you must agree on the transition from old routes to new ones and migrate them by small parts

4 – changed requirements  
the customer can change his requirements after they have been agreed on the project

the probability of the corresponding situation - 9  
the severity of the risk - 8  
mitigation actions - it is necessary to agree that if there are a large number of changes, then for this it is necessary to create a new task and hold discussions on it

5 – Adding a new team (team member)   
new team (team member) without specific experience in the domain area / technique/tool

the probability of the corresponding situation - 3  
the severity of the risk - 5  
mitigation actions - conduct onboarding sessions

**3. What are some (at least 3-5) tools that it makes sense to use for automated testing in such a project.**

1 – we can use [Selenium](https://www.selenium.dev/) tool for regression testing and testing a web-app  
2 – we can use [LambdaTest](https://www.lambdatest.com/) tool for record real-time browser compatibility testing  
3 - we can use [Eggplant](https://www.lambdatest.com/) tool is used for performance, stress testing, and load. On the other hand, the functional testing tool is precisely focused on what the name indicates

**4. Estimate project manual testing effort using the decomposition technique and the 3-point technique.**

decomposition technique:  
1. Registration module – 75 test cases (7,5 workdays)  
2. Booking:  
2.1. Hotels Booking – 200 test cases (20 workdays)  
2.2. Car Rent – 200 test cases (20 workdays)  
3. Payment module – 100 test cases (10 workdays)  
4. Localization module – 60 test cases (6 workdays)  
5. Searching module – 75 test cases (7,5 workdays)  
6. Account module – 125 test cases (12,5 workdays)  
TOTAL 83,5 workdays (= around 3 months = around 6 sprints (2 weeks in 1 sprint))

the 3-point technique

E = (75+84\*4+95)/6 = 84,3 workdays (= around 3 months = around 6 sprints (2 weeks in 1 sprint))  
SD = (95-75)/6 = 4,2 workdays

Purpose that a manual tester performs 10 test cases per a day.   
Then having a decomposition into modules (Registration, Booking, Payment, Localization, Searching, Account) we can suggest how much time we will spend on each of them.   
For example, we have added 400 test cases to the Bookings module and we will spend 40 working days in total.

Total days spent for testing the **decomposition technique** = 83.5 workdays   
(= around 3 months = around 6 sprints (2 weeks in 1 sprint))

according to **the 3-point technique**, we assume that   
in the worst case it will be spent - 95 working days

at best - 75 working days

most likely - 84 working days

Estimation for this technique = 84.3 workdays (= around 3 months = around 6 sprints (2 weeks in 1 sprint)) with a standard deviation of 4.2 workdays