Web API Integration Project Design - Integration of Booking Application API into C# Application

Table of Contents

List of Figure

List of Tables

1 – Introduction

In this report we will cover the design and planning phases of integrating a console based Web API for a userfriendly platform to enable users explore and book their rooms using the RapidAPI Booking.com API. The features to be supplied in this project will include filtering based on location, price according to budget, filtration according to customer reviews, advise on flow of traffic, transit, or adjacent landmarks/attractions.

API integrations are the backbones of modern programs that allow developers enhance functionality by integrating more and more data and services from third-party sources. In this context, the Booking.com API is the backbone for our site, which will deliver real-time hotel data and augment the user experience with seamless access to transit alternatives available and all nearby attractions.

It will explain the technical and user requirements of the project, the technologies to be utilised, the high-level architecture of our program, comprehensive development with dates. Finally, we do a conclusion followed by some references.

2 – API Selection

This section will focus on our pick of Booking.com for our project. The application picked for this project is named Booking Web API, and the API utilised will be from RapidAPI Booking.com. This will offer us with access to a great quantity of hotel data as well as additional services such as filtering by location, price, user reviews, proposing transit choices, and neighbouring attractions.

2.1 – Chosen API

The API on which this work is constructed is RapidAPI Booking.com API, a REST application program interface. With this API, the developer may have easy access to and integrate a large range of travel-related data, including hotel listings, their ratings, and booking options.

2.2 – Description

Accomplished by the hotel data given, Booking.com API totally supplies real-time accessibility to data with regard to properties, price, and availability related to different types of accommodation, client evaluations, and photographs. Moreover, it offers rapid searching and booking of hotels. Apart from offering information on the region of stay, the API may also recommend other landmarks and attractions in its neighbourhood, therefore expanding user experience even further than just booking.

2.3 – Justification

We choose the Booking.com API because of the depth of the capabilities, totally corresponding to the objective of the project. Eventually, the key concept was documentation and convenience of use, delivering a guarantee that its services would be implemented into our platform effortlessly. Furthermore, the thorough filtering options and data on transportation supplied in this API make it perfect for developing a fully integrated booking application. Other APIs did not give same depth in terms of travel-related data and simple integration into booking.

3 – Use Case, User and Technical Requirements

In this section, we will explore our project's use case and give two real-world example of how our API can be utilised. Furthermore, we will also explore the primary user and technical needs. We designed at least four well-defined user criteria that fulfil the expectations of passengers that apply the site. In addition, we will develop four technical criteria to verify that the application fulfils performance, security, and compatibility requirements. These parameters are crucial for establishing a workable and user-friendly booking platform that connects with the Booking.com API.

3.1 – Use Case 1: Personal Accommodation Search

Accommodations may be shown to users, filtered by choice, using the Booking.com API on any travel platform. Once connected into the console platform, customers may search for hotels by region, price range, and customer reviews. For example, searching inside the budget range in London and with the highest customer reviews. The platform would therefore deliver the most relevant results to the user. This technology increases the user experience by personalising hotel selections, allowing users to make more informed booking decisions.

3.2 – Use Case 2: Trip Planning with transportation suggestions

Booking.com's hotel searching API might perhaps be beneficial in a trip planning application, aiding users not only in selecting a decent spot to stay but also in planning how to get there. By putting such an API into the program, it may display local transportation possibilities such as airports, rail stations, and bus terminals depending on the location of the hotel. An example would be the airport or train station closest to any hotel that one is renting in Paris, allowing him to plan and go with ease rather than spending so much time planning travels or dealing with various booking sites. The API will emerge with information on how clients may simply build otherwise hectic and complicated trip plans while keeping simplicity, resulting in a single solution for both booking and transportation requirements.

3.3 – User Requirements

The following user criteria have been established to provide a pleasant and easy experience for travellers on the platform:

Requirement ID	Requirement	Description	Priority
1	**Search Feature**	The user should be able to search for the sort of housing they desire based on crucial filtering characteristics such as location, price range, and customer reviews.	1
2	**Transport advise**	Depending on the location of the hosted flat, customers should be offered transportation information, such as the closest airport or train station, to help them schedule their journey.	4
3	**Accommodation Details**	Users must be offered thorough information about each	2

		accommodation so that they may make sensible decisions based on facilities, user ratings, and photographs.	
4	**Booking Confirmation**	It is envisioned that clients will be able to book their selected lodging directly via the website, gaining confirmation with all relevant data.	3

Table 1 - User Requirements

3.4 – Technical Requirements

The technological characteristics for this project guarantee that the platform is durable, safe, and highly efficient:

Requirement ID	Requirement	Description	Priority
1	**API Integration:**	The application must connect the Booking.com API to get real-time hotel data while also offering smooth communication between the front and back end.	1
2	**Performance**	The web application should be capable of executing many API calls simultaneously. Performance and quickness should not be disregarded since this assures that clients have minimum delays while finishing a search or making a booking.	2
3	**Security**	To safeguard sensitive information while booking, the system should incorporate secure data transfer mechanisms such as HTTPS, as well as reliable user authentication.	3

4	**Compatibility**	The platform should be interoperable with a multitude of devices and browsers, offering a consistent user experience on PCs, tablets, and mobile devices.	5
5	**Error Handling**	The application should gracefully handle any issues linked to API calls or data fetching by giving out user-friendly error messages.	4

Table 2 - Technical Requirements

4 – Technology/Tools to Be Used

This section describes the core technologies and tools we will need to develop our Console based Booking Web API Application. These include the programming language, framework, integrated development environment (IDE), and testing tools essential to implement the Booking.com API.

4.1 – Selected Technologies

The following technologies and techniques were employed to properly implement the project for a Console based Booking Web API Application:

S Number	Tool/Technology	Purpose
1	C# (Programming Language)	C# will be the principal programming language applied to construct the application.
2	.NET Framework (Framework)	The application's backbone will be built on an appropriate .NET framework utilising a console application which will provide a command line interface for user interaction.
3	Microsoft Visual Studio Installer (IDE)	MS VS Studio Installer will offer a development environment for writing, testing, and debugging the applications.
4	RapidAPI Booking.com API (Web API)	This API will supply the key lodging data from Booking.com to this application, including filters and booking options.

5	JSON (Data Format)	Since JSON, also known as JavaScript Object Notation, is lightweight, efficient, and humanreadable, it should be utilised to transfer data between API and Application.
6	Postman (API Testing Tool)	This tool is used to verify that API endpoints accept correct requests and answers.
7	Database (Notepad)	This notepad will store all of the user data alongside data about accommodation, flights and transport.
8	Command Line Interface (CMD) (User Interface)	The Console app acts as a command line interface (CMD) for
		text input. It makes it easy to reserve facilities, get housing and transportation information, and change and manage plans. The Console app acts as a command line interface (CMD) for text input. It makes it easy to reserve facilities, get housing and transportation information, and change and manage plans.

Table 3 - Selected Technologies

4.2 – Justification

Each of these technologies was selected because it is relevant and effective in achieving the project's aims.

S Number	**Tool/Technology**	**Justification**
1	C# (Programming Language)	As one of the most popular and resourceful programming languages, it is a wonderful match for the project. It supports API integration, object-oriented programming, and easy communication with other services like the Booking.com API.
2	.NET Framework (Framework)	Fast and scalable C# application development has a tidy and expressive base accessible from the NET Framework Console Application. For command-line applications like this one, which rely on ensured rapid user interactions, it is great since it offers a comprehensive set of libraries and support.

3	Microsoft Visual Studio Installer (IDE)	MS VS Installer is one of the most feature-rich IDEs, automating the development process with an integrated code authoring, debugging, and testing environment. Support will be expanded to C# and.NET, making it ideal for this project.
4	RapidAPI Booking.com API (Web API)	The RapidAPI Booking.com API was picked because it helps to access key travel information, such as a list of hotel options with ratings and the ability to book them. Its well-documented RESTful design exhibits ease of integration and data retrieval dependability.
5	JSON (Data Format)	JSON is excellent for data transport since it is simple and efficient. It offers the best structure for API answers, permitting smooth communication between the Booking.com API and the application.
6	Postman (API Testing Tool)	Postman is a standard tool used in industry for API testing. This would be great for the project team to analyse API endpoints and confirm that requests and replies execute as planned before adding them into the application.
7	Database (Notepad)	Notepad was chosen as this allowed us to store, retrieve and manipulate data. This would be great for our project because it will store vital customer information such as flight details and train details.
8	Command Line Interface (CMD) (User Interface)	CMD delivers rapid and lightweight interaction with the application, providing simplicity and clear access to factors such as housing and transit possibilities. This minimises complexity and allows most computers to utilise this program without the demand for a graphical interface.

Table 4 - Justification

5 – High-Level Architecture Diagram

In this part, we will discuss the high-level architecture for the Booking Web API application. This diagram will substantially describe how the primary system components, such as the frontend, backend, and database layers, interact with one another, as well as how it interfaces with the application via the Booking.com API utilising RapidAPI. This is an architecture overview that incorporates security and scalability to deliver a userfriendly experience.

5.1 – Architectural Overview

The high-level architectural diagram depicts how the Booking Web API application's core components work together to meet system demands on a structural level. Together with the data flow among the front-end, backend, and database, the figure depicts how they will interface with the Booking.com API. It prioritises security layers and outside services like transportation booking and payment processing. The high-level architectural diagram depicts how the Booking Web API application's core components work together to meet system demands on a structural level. Together with the data flow among the front-end, back-end, and database, the figure depicts how they will interface with the Booking.com API. It prioritises security layers and outside services like transportation booking and payment processing.

5.2 – Architectural Diagram

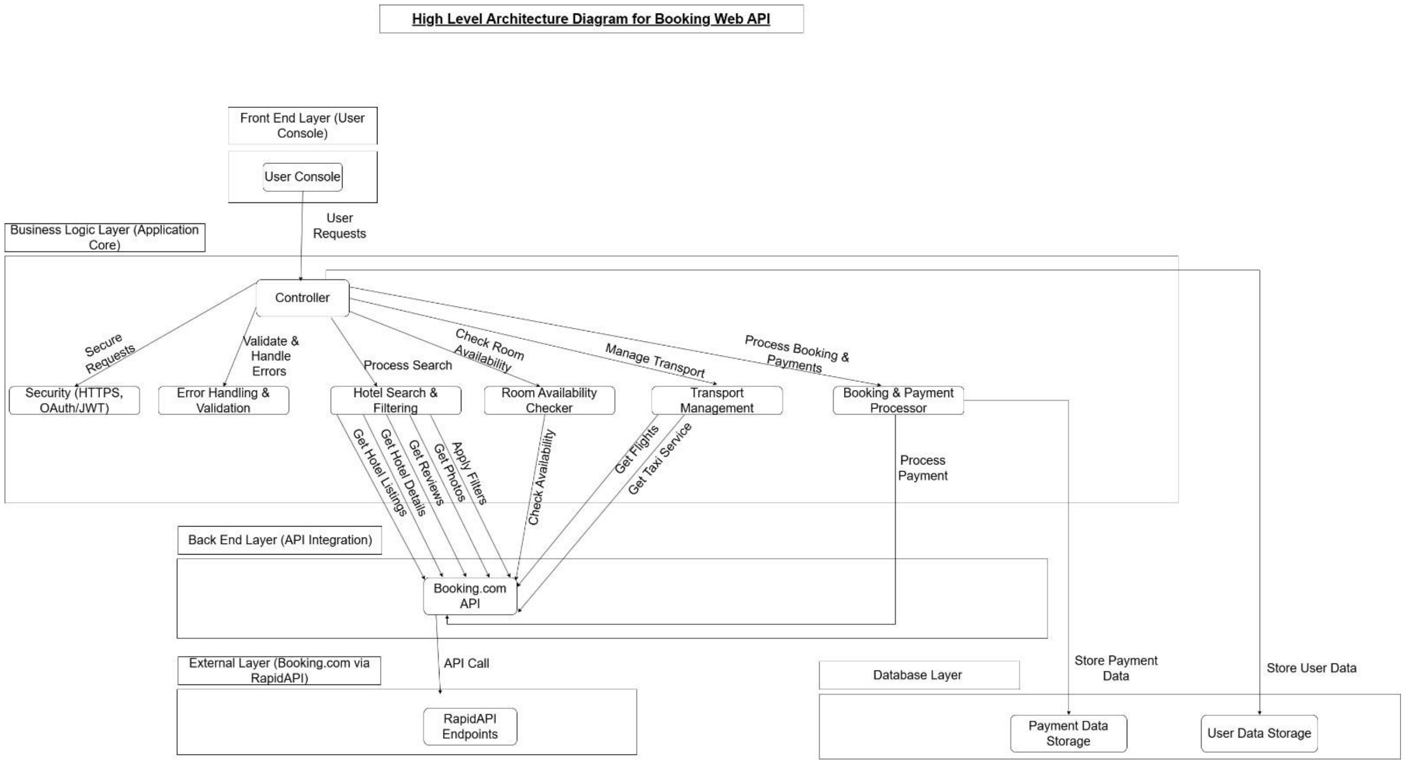

Figure 1- High Level Architectural Diagram

Figure 1 depicts the architectural design, which incorporates labelled drawings of the system's major components. We utilised draw.io to make this high-level architectural diagram. To be more clear, it displays the front-end layer (a user console), a back-end logic controller with numerous associated activities, and the database layer, which keeps information about users and payments. Furthermore, it depicts how the Booking.com API will be integrated into the system using RapidAPI; arrows indicate how the flow between components will be handled. It also covers the interface to third-party services like transportation management and payment processing.

5.3 – Explanation of Components

In this section, we will detail the components we picked and why.

- **Frontend Layer (User Console):** This layer allows users to interact with the system. This layer allows the user to search for hotels, see room availability, and book accommodations. User queries begin on the front end and are sent to the back end for further processing.

- **Business Logic Layer (Application Core):** This layer processes the user's requests. This specific layer includes:
  - **Controller:** It oversees the flow of data between the frontend and backend. It is in charge of processing the user's input, managing any search-related queries, and triggering validation. ○ **Security:** HTTPS, OAuth, and JWT - these protocols safeguard a user's interactions with an application, reliably verifying his or her credentials and securing sensitive data throughout a transaction.
  - **Error Handling & Validation:** This module manages any errors that emerge during data processing and preserves data integrity using multiple validation techniques.
  - **Hotel Search & Filtering:** This module handles the search request, receives hotel listings from the Booking.com API, and filters by price and location.
  - **Room Availability Checker:** It refreshes room availability in real time using data from the Booking.com API.
  - **Transportation Management:** This is used to handle user requests for transportation services such as aeroplanes, trains, and buses. Fetch information about third-party transportation services.

- **Backend Layer - API interface:** ○ **RapidAPI interface with Booking.com API**: This is to acquire hotel data, availability, and reviews. This API will enable booking of a room using this platform, which will be tunnelled for safe processing.
  - **External Layer (Third-Party API Calls):** This layer manages interactions with third-party services including transportation providers and payment gateways.

- **Database Layer:**
  - **User Data Storage:** This layer maintains user information such as personal data, booking history, and preferences.
  - **Payment Data Storage:** It stores and protects payment-related information in order to deliver real-time, safe, and effective payment transactions.

The architecture described here provides scalability and security, with simple integration with third-party services for booking hotels, transportation, and other services in a user-friendly fashion.

6 – Development Plan with Timelines

In this section, we will explore the development strategy for the Booking Web API application, consisting precisely specified, allocated, and expected activities with deadlines.

6.1 – Development Activities

The full action list for the project's successful completion is presented here, beginning with setup and continuing through testing and deployment.

Activity	Description	Start Date	End Date
Project Setup	Set up development environment, tools, and version control.	31/10/2024	01/11/2024
API Research and Testing	Use RapidAPI to access the Booking.com API and begin testing the endpoints.	01/11/2024	01/11/2024
Front-End Development	User interface development, including search, filters, and a booking system.	02/11/2024	03/11/2024
Back-End Development (Business Logic)	Implement business logic to insure the execution of hotel search and filtration.	03/11/2024	04/11/2024
API Integration	Integrate the Booking.com API with third-party services: Transport.	04/11/2024	05/11/2024
Database Setup and Integration	Configure user and payment data storage.	05/11/2024	07/11/2024
Security Implementation	Implement security protocols (HTTPS, OAuth, and JWT).	07/11/2024	08/11/2024
Testing and Debugging	Perform unit tests, integration tests, and issue remedies.	08/11/2024	09/11/2024
Final Deployment	Deploy the completed application and test all features.	09/11/2024	10/11/2024

Table 5 - Development Activities

8.2 – Responsibilities

To insure a well-coordinated operation, each team member would be allocated the tasks listed below.

Activity	Responsible Person
Project Setup	Shyam
API Research and Testing	Gavin
Front-End Development	Gavin
Back-End Development (Business Logic)	Shyam
API Integration	Both
Database Setup and Integration	Both
Security Implementation	Both
Testing and Debugging	Both
Final Deployment	Both

Table 6 - Responsibilities

8.3 – Timeline

Overall, the project is anticipated to be done within 10 days. The following are major milestones and deadlines:

Milestones	Completion Date
Project Setup	01/11/2024
API Research and Testing	01/11/2024
Front-End Development	03/11/2024
Back-End Development (Business Logic)	04/11/2024
API Integration	05/11/2024
Database Setup and Integration	07/11/2024
Security Implementation	08/11/2024
Testing and Debugging	09/11/2024
Final Deployment	10/11/2024

Table 7 - Timeline

9 – Conclusion

This report has covered the fundamentals of our project, which includes constructing the Booking Web API application based on the Booking.com API. Some noteworthy examples include: a profusion of information about various residences, numerous filtering options, and exhaustive documentation. User and technological needs were rigorously stated, assuring that our site would match the expectations of users while looking for and reserving a place to stay, as well as presenting some relevant transit possibilities.

The following technologies will be used: C#,.NET Framework, Postman, and RapidAPI. These technologies were picked following careful assessment of their fit with our project objectives, as well as their capacity to link the third-party service securely, quickly, and scalable. The high-level overview architecture demonstrates how different system components interface with several other services, as well as the Booking.com API, to offer seamless and safe experiences.

The complete development plan, with clearly described activities and deadlines, assures that the project is delivered correctly and on schedule. We worked on various crucial areas of the application's development, such as API integration, security implementation, and testing.

Finally, in terms of future concerns, there may be issues such as guaranteeing security for sensitive user data, API rate constraints, and enhancing the application's performance under significant user traffic. We will be regularly evaluating and upgrading our system to clearly tackle these problems while ensuring that the platform stays scalable and efficient as we go forward into the next stages of development.

10 – References

Aggelos K (2023). *Microservices Architecture: A Comprehensive Overview | Cloud Native Daily*. [online] Medium. Available at: https://medium.com/cloud-native-daily/microservices-architecture-a-comprehensive-overview-dc52a024bded [Accessed 21 Oct. 2024].

API4AI (2024). *Rapid API Hub: The Step-by-Step Guide to Subscribing and Starting with an API*. [online] Medium. Available at: https://medium.com/@API4AI/rapid-api-hub-the-step-by-step-guide-to-subscribing-andstarting-with-an-api-ea5387269dfe [Accessed 21 Oct. 2024].

AWS (2019). *What are Microservices?* [online] Amazon Web Services, Inc. Available at: https://aws.amazon.com/microservices/.

Bello, G. (2024). *Best Practices for API Error Handling*. [online] Postman Blog. Available at: https://blog.postman.com/best-practices-for-api-error-handling/.

Booking.com. (2024). *Using the API - Booking.com Developers API*. [online] Available at: https://legacy.developers.booking.com/api/commercial/index.html?page_url=how-to-book [Accessed 21 Oct. 2024].

Cochrane, R. (2022). *API Security: Best Practices for a Changing Attack Surface*. [online] Rapid7. Available at: https://www.rapid7.com/blog/post/2022/06/27/api-security-best-practices-for-a-changing-attack-surface/ [Accessed 21 Oct. 2024].

Demir, D. (2024). *RapidAPI Review: What is RapidAPI and How to Use it*. [online] Apidog Blog. Available at: https://apidog.com/blog/what-is-rapidapi-and-how-to-use-it/ [Accessed 21 Oct. 2024].

Eren Yılmaz (2023). *Understanding OAuth 2.0 and OpenID Connect - SDTR - Medium*. [online] Medium. Available at: https://medium.com/software-development-turkey/understanding-oauth-2-0-and-openidconnect-777eb1fc27f [Accessed 21 Oct. 2024].

Gordon, W.J. and Rudin, R.S. (2022). Why APIs? Anticipated value, barriers, and opportunities for standards-based application programming interfaces in healthcare: perspectives of US thought leaders. *JAMIA Open*, 5(2). doi:https://doi.org/10.1093/jamiaopen/ooac023.

https://techcommunity.microsoft.com/t5/user/viewprofilepage/user-id/164688 (2024). *Best Practices for API Error Handling: A Comprehensive Guide*. [online] TECHCOMMUNITY.MICROSOFT.COM. Available at: https://techcommunity.microsoft.com/t5/apps-on-azure/best-practices-for-api-error-handling-acomprehensive-guide/td-p/4088121 [Accessed 21 Oct. 2024].

Korly, J. (2019). *RESTful Web Services Tutorial with Example in Just 10 Minutes*. [online] Medium. Available at: https://medium.com/@johnkorly/restful-web-services-tutorial-with-example-in-just-10-minutes-a6ef4620ccb8 [Accessed 21 Oct. 2024].

Microsoft (2022). *A tour of C# - Overview*. [online] learn.microsoft.com. Available at:

https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/.

Mootaz Alhalak (2024). *In today’s interconnected digital world, the term API is ubiquitous, powering the seamless integration of various software applications. APIs, or Application Programming Interfaces, are the unsung heroes behind the scenes, enabling different systems to communicate with each other.* [online] Linkedin.com. Available at: https://www.linkedin.com/pulse/understanding-apis-what-how-work-mootazalhalak-1vpfe [Accessed 21 Oct. 2024].

Patel, R. (2023). *Understanding APIs: What They Are and How They Work*. [online] Medium. Available at: https://rutikkpatel.medium.com/understanding-apis-what-they-are-and-how-they-work-4bc1ed6dfbc0 [Accessed 21 Oct. 2024].

Rick-Anderson (2022). *Build RESTful APIs with ASP.NET Web API - ASP.NET 4.x*. [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/aspnet/web-api/overview/older-versions/build-restful-apis-with-aspnet-web-api.

Roy, J. (2022). *How To Secure Your APIs And Protect Sensitive Data ?* [online] Medium. Available at: https://medium.com/@ItsRoy69/how-to-secure-your-apis-and-protect-sensitive-data-246e633d3a16 [Accessed 21 Oct. 2024].

StackHawk (2024). *Understanding APIs: What are they and what do they do?* [online] StackHawk. Available at: https://www.stackhawk.com/blog/what-is-an-api-a-beginners-guide-to-application-programming-interfaces/.

Staff Reporter (2024). *The Role of APIs in Modern Software Development*. [online] News, Sport, Information, Property, Business in Spain. Available at: https://theleader.info/2024/05/27/the-role-of-apis-in-modernsoftware-development/ [Accessed 21 Oct. 2024].

Starreveld, A. (2023). *Understanding OAuth2*. [online] The Web Application Security Hub. Available at: https://medium.com/web-security/understanding-oauth2-a50f29f0fbf7.

W3C (2019). *Standards - W3C* . [online] W3.org. Available at: https://www.w3.org/standards/.