

**International School**

**Capstone Project 1**

CMU-SE 450/CMU-IS 450/CMU-CS 450

**Architecture Design**

**Version 1.1**

**Date: 25/09/2020**

**viBOTour - Smart Chatbot for Tourist 4.0**

**Submitted by**

**Sang, Nguyen Thanh**

**Thuyen, Pham Van**

**Tuan, Doan Kim**

**Loc, Phung Hoang Phu**

**Approved by Nguyen Duc Man**

# Project Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Project acronym** | viBOTour | | |
| **Project Title** | Smart Chatbot for Tourist 4.0 | | |
| **Start Date** | 24 Aug 2020 | **End Date** | 5 Dec 2020 |
| **Lead Institution** | International School, Duy Tan University | | |
| **Project Mentor** | Man, Nguyen Duc, M.Sc. | | |
| **Scrum master / Project Leader & contact details** | Sang, Nguyen Thanh  Email: thanhsang3111999.ntb@gmail.com  Tel: 0344684069  ID: 23211212363 | | |
| **Partner Organization** |  | | |
| **Project Web URL** |  | | |
| **Team members** | Name | Email | Tel |
| 2321121321 | Tuan, Doan Kim | doantuan21101999@gmail.com | 0913622928 |
| 2321124963 | Loc, Phung Hoang Phu | phunghoangphuloc@gmail.com | 0912698173 |
| 2321125061 | Thuyen, Pham Van | phamvanthuyen2802@gmail.com | 0795697332 |

# 

# Architecture Document

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Title** | Architecture Document |  |  |
| **Reporting Period** |  |  |  |
| **Author(s)** | Thuyen, Pham Van | |  |
| **Team Information** | **Name** | **Role** |  |
| Sang, Nguyen Thanh | Member |  |
| Tuan, Doan Kim | Member |  |
| Loc, Phung Hoang Phu | Member |  |
| Thuyen, Pham Van | Member |  |
| **Date** | 25/09/2020 | **Filename** | Architecture\_Design\_V1.1.docx |
| **Access** | Project and CMU Program | |  |

|  |  |  |
| --- | --- | --- |
| Document History | | |
| **Version** | **Date** | **Comments** |
| V1.0 | 04/09/2020 | Create Architecture Document |
| V1.1 | 25/09/2020 | Update 3 |

# 

# Document Approvals

The following signatures are required for approval of this document.

|  |  |  |
| --- | --- | --- |
| Msc Man, Nguyen Duc  Mentor |  | Date |
| Sang, Nguyen Thanh  Scrum master, DevTeam |  | Date |
| Tuan, Doan Kim  Product Owner, DevTeam |  | Date |
| Thuyen, Pham Van  DevTeam |  | Date |
| Loc, Phung Hoang Phu  DevTeam |  | Date |

**TABLE OF CONTENTS**

[Project Information 1](#_Toc54476938)

[Architecture Document 2](#_Toc54476939)

[Document Approvals 3](#_Toc54476940)

[1. Introduction 5](#_Toc54476941)

[1.1 Project overview 5](#_Toc54476942)

[1.2 Purpose 5](#_Toc54476943)

[2. Architecture driver 6](#_Toc54476944)

[2.1 Business constraints 6](#_Toc54476945)

[2.2 Technical constraints 6](#_Toc54476946)

[2.3 Functional requirement 6](#_Toc54476947)

[2.4 Quality attributes 6](#_Toc54476948)

[2.4.2.1 Security 7](#_Toc54476949)

[2.4.2.2 Performance 8](#_Toc54476950)

[2.4.2.3 Usability 9](#_Toc54476951)

[2.4.2.4 Correctness 11](#_Toc54476952)

[2.4.2.4 Modifiability 12](#_Toc54476953)

[3. Architecture overview 13](#_Toc54476954)

[3.1 System context 13](#_Toc54476955)

[3.2 Component and connector 15](#_Toc54476956)

[3.3 Sequence diagram 16](#_Toc54476957)

[3.4 Module view 22](#_Toc54476958)

[3.5 Allocation view 22](#_Toc54476959)

# Introduction

## Project overview

The viBOTour system is a travel system that integrates smart chat bot. This system helps tourists find suitable tours on the system. In addition, this system allows users to interact with the chat bot to receive useful information such as tour suggestions,..

## Purpose

This specification covers following:

* Brief specification of the project, high level requirement.
* Detail quality attribution.
* System context, sequence diagrams.
* Architecture presented by various view types: Component an Connect, Module view and Allocation view.
  1. **Business driver**

Business Problems:

* Tourists need assistance with details of a trip, the weather or the place.
* Users need an automated system to assist with billing, travel recommendations, weather, local food or hotel or room rental services.

Business Need:

* A chatbot can communication with tourists,
* Support for their tourists can book a tour, create a custom tour and get suggestions when they need it.
* Tourists can search, view tour details, book and pay for tours.
* Administrators can manage tour, user, promotion and tourist attractions.

# Architecture driver

## 2.1 Business constraints

* Sources: 4 people.
* Project was started on: 24/08/2020.
* Project will be ended on: 05/12/2020.
* Project will be finished in 104 days (980 hours).
* Cost: $2352.

## 2.2 Technical constraints

Technical to develop:

* Programming Language: JavaScript, Solidity.
* Frameworks / Libraries: ReactNative, ReactJS, NodeJS, Bootstrap 4.
* Database Management System: Postgresql.

Environment:

* Web browsers: Google Chrome, FireFox, Opera.
* App environment: IOS and Android
* Operation systems: Microsoft Windows 10, Ubuntu.

## 2.3 Functional requirement

References to Product Backlog specification of ProductBacklogV1.0.docx

## 2.4 Quality attributes

**2.4.1 Utility table**

There are following quality attributes that drive the design of architecture. Each quality attribute scenario is ranked with importance (I) defined by the Product Owner, and the estimated level difficulty (D). Both values are based on a scale of High (H) - Medium (M) - Low (L).

**2.4.2 Quality attributes**

#### 2.4.2.1 Security

|  |  |
| --- | --- |
| Quality attributes: When customers want to pay for the tour they choose, they are required to provide a confirmation code sent to them on the phone they provided, if within 1 minute they do not enter the correct code will ask for a new confirmation | |
| Type | Security |
| Stimulus | provide a confirmation code |
| Source of stimulus | Tourist |
| Environment | In runtime |
| Artifact stimulated | Application |
| Response | Confirmation code sent to phone |
| Response measure | If within 1 minute they do not enter the correct code will ask for a new confirmation |

|  |  |
| --- | --- |
| Quality attributes: When a tourist registers, their password will be encrypted through 3 layers of security before being saved to database | |
| Type | Security |
| Stimulus | Register |
| Source of stimulus | Tourist |
| Environment | In runtime |
| Artifact stimulated | System |
| Response | Encrypt password |
| Response measure | Encrypted through 3 layers of security |

#### 2.4.2.2 Performance

|  |  |
| --- | --- |
| Quality attributes: Tourists require a suggested tour in the chatbot while using the app. The system summarizes tour and response expected tours for them in less than 5 seconds. | |
| Type | Performance |
| Stimulus | Require a suggested tour |
| Source of stimulus | Tourists |
| Environment | In runtime |
| Artifact stimulated | Application |
| Response | Summarizes tour and response expected tour |
| Response measure | less than 5 seconds |

|  |  |
| --- | --- |
| Quality attributes: Tourists send a request payment for their tour while paying a tour. The system verifies this tour, makes a transaction and sends a result in less than 10 seconds. | |
| Type | Performance |
| Stimulus | Send a request payment |
| Source of stimulus | Tourists |
| Environment | In runtime |
| Artifact stimulated | Application |
| Response | Verify paid tour information, make transaction and response a result |
| Response measure | less than 10 seconds |

|  |  |
| --- | --- |
| Quality attributes: When a tourist chats to a chatbot, it will respond to search results by text or voice message from its existing data to give the closest answer to a traveler request within 10 seconds with stable network connection. | |
| Type | Performance |
| Stimulus | Respond the search result from tourist request |
| Source of stimulus | Tourists |
| Environment | In runtime and stable network connection |
| Artifact stimulated | Application |
| Response | A text or voice message |
| Response measure | within 5 seconds |

#### 2.4.2.3 Usability

|  |  |
| --- | --- |
| Quality attributes: Tourists access the system by signing up for a new account or logging in with their Google account with the same email. The application allows access with this email account in both ways. | |
| Type | Usability |
| Stimulus | Access the system by signing up for a new account or logging in with their Google account with the same email |
| Source of stimulus | Tourists |
| Environment | In runtime |
| Artifact stimulated | Application |
| Response | Allows access with this email account |
| Response measure | both ways |

|  |  |
| --- | --- |
| Quality attributes: Tourists book a tour in the application. When they have other devices they can also see the booked tour and pay for it instantly. | |
| Type | Usability |
| Stimulus | Book a tour and use other devices |
| Source of stimulus | Tourists |
| Environment | The system |
| Artifact stimulated | Mobile devices |
| Response | See the booked tour and pay it |
| Response measure | Instantly |

|  |  |
| --- | --- |
| Quality attributes: The staff will check the tourist ticket quickly via the QR code sent when payment is successful or the ticket information at the tourist’s email | |
| Type | Usability |
| Stimulus | Check ticket |
| Source of stimulus | The staff |
| Environment | In runtime |
| Artifact stimulated | The system |
| Response | Ticket information |
| Response measure | QR code or the ticket information at the tourist’s email |

#### 2.4.2.4 Correctness

|  |  |
| --- | --- |
| Quality attributes: Tourists tick the tour they want to pay in their mobile phone and checkout them. The system calculates the total amount and pays the correct quantity and selected tour. | |
| Type | Correctness |
| Stimulus | Tick the tour and checkout them |
| Source of stimulus | Tourists |
| Environment | The system |
| Artifact stimulated | Mobile phone |
| Response | Calculates the total amount and pays tour |
| Response measure | Correct quantity and selected tour |

|  |  |
| --- | --- |
| Quality attributes: When the tourist makes a successful payment, the system will ask the baking system to deduct the money in the account provided by the tourist equal to the amount paid by the tourist | |
| Type | Correctness |
| Stimulus | Pay successfully |
| Source of stimulus | Tourists |
| Environment | Runtime |
| Artifact stimulated | The System |
| Response | Deduct the money in the account provided |
| Response measure | Deduct the money in the account provided by the tourist equal to the amount paid by the tourist |

|  |  |
| --- | --- |
| Quality attributes: When the staff performs operations with a tour including adding, modifying, and deleting the tour, the corresponding tour information will be updated correctly on the tourist application. | |
| Type | Correctness |
| Stimulus | Performs operations with a tour |
| Source of stimulus | Staff |
| Environment | Runtime |
| Artifact stimulated | The System |
| Response | Change the information tour |
| Response measure | The corresponding tour information will be updated correctly on the tourist application. |

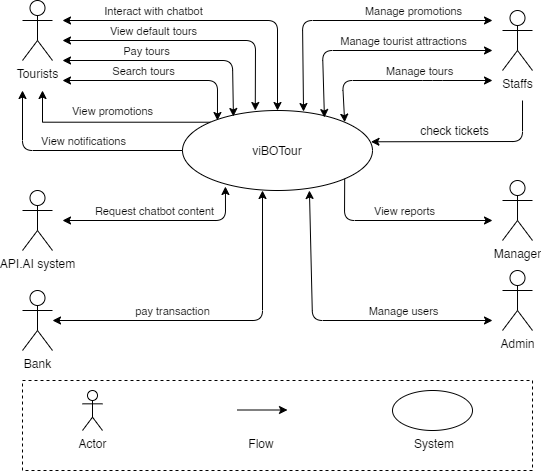
#### 2.4.2.4 Modifiability

|  |  |
| --- | --- |
| Quality attributes: A product manager wants to develop additional features and construct a travel social network in the next version. The system allows for an expansion within 4 person months of effort without affecting existing functions. | |
| Type | Modifiability |
| Stimulus | Tick the tour and checkout them |
| Source of stimulus | A product manager |
| Environment | The next version |
| Artifact stimulated | The system |
| Response | Allows for an expansion |
| Response measure | 4 person months of effort without affecting existing functions |

# Architecture overview

This section shows the diagrams which bounds our target system and describes the architecture and interaction between components

## 3.1 System context

****

**Figure 1: System Context Overview**

**Tourist:**

* Request Login/Logout to the system, System check information and response
* Request to view tour and system response list tours
* Request for payment in the vibotour system, will the system respond successfully or not?
* Request to search the tour list, the system will respond to the list of required tours
* Request to display the tours added to the cart, the system will respond to the list of tours that the user has added.

**Admin:**

* Request Login/Logout to the system, System check information and response
* Enter sign up new account, System saves account.
* Checkout the tour, System saves that payment.
* Request to list view tour, System response view tour.
* Request to view tour detail, system response tour detail

**Bank:**

* Transfer money to the another account, the system
* Response table result.
* Receive notification from system.

**Manager :**

* Request Login/Logout to the system, System check information and response
* Enter sign up new account, System saves account.
* Checkout the tour, System saves that payment.
* Request to list view tour, System response view tour.
* Request to view tour detail, system response tour detail

**Staff :**

* Request Login/Logout to the system, System check information and response
* Enter sign up new account, System saves account.
* Checkout the tour, System saves that payment.
* Request to list view tour, System response view tour.
* Request to view tour detail, system response tour detail

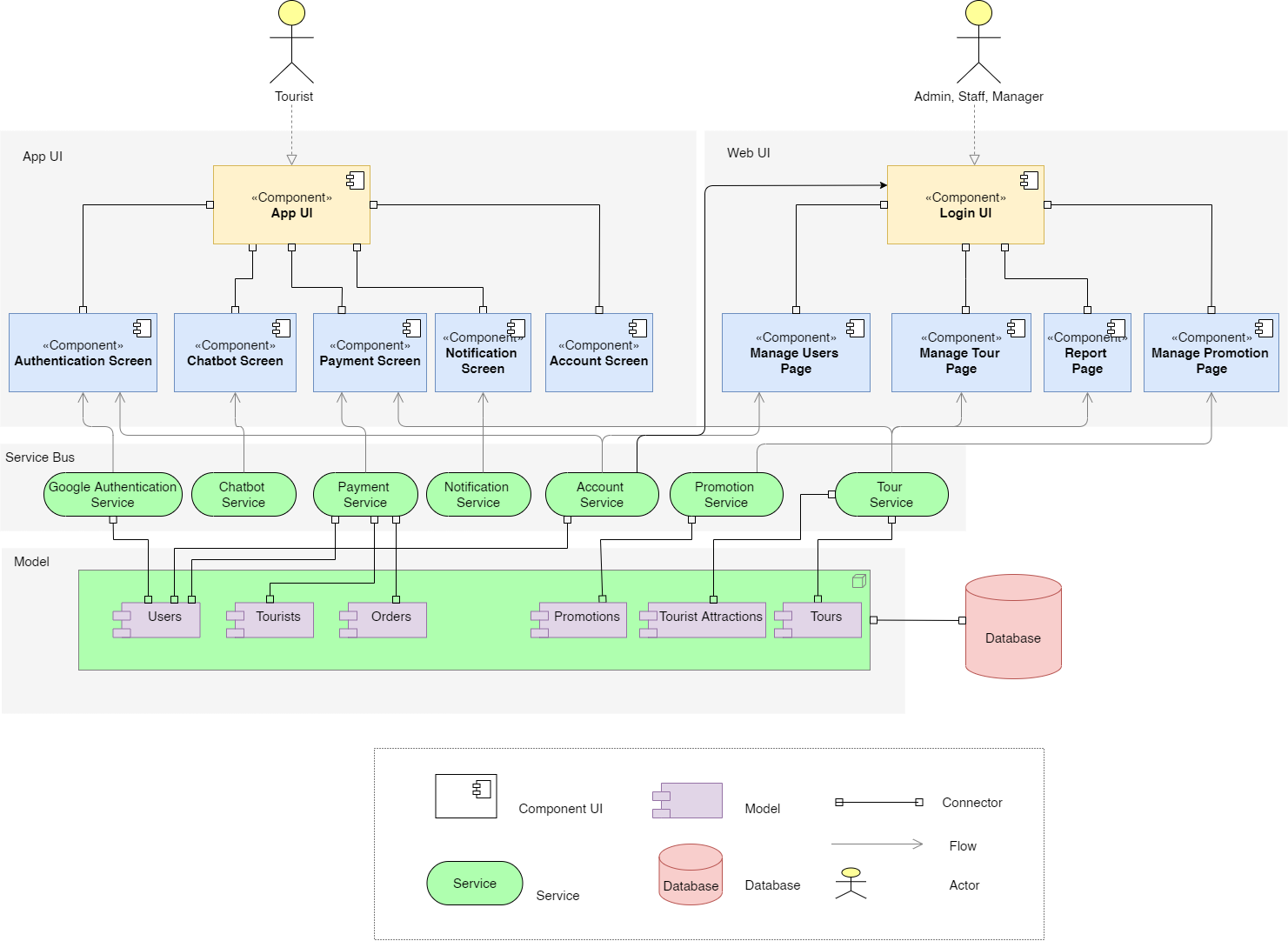
**API.AI System:**

* Request Login/Logout to the system, System check information and response
* Enter sign up new account, System saves account.
* Checkout the tour, System saves that payment.
* Request to list view tour, System response view tour.
* Request to view tour detail, system response tour detail

## 3.2 Component and connector

We mainly used a C&C view to argue and reason about architectural properties, quality attribute requirements, and functional requirements that the system must add here.

This view type partitions the system into components that have some runtime presence such as processes, objects, data stores, and connectors or that represent pathways of communication such as data flows and access to shared storage.



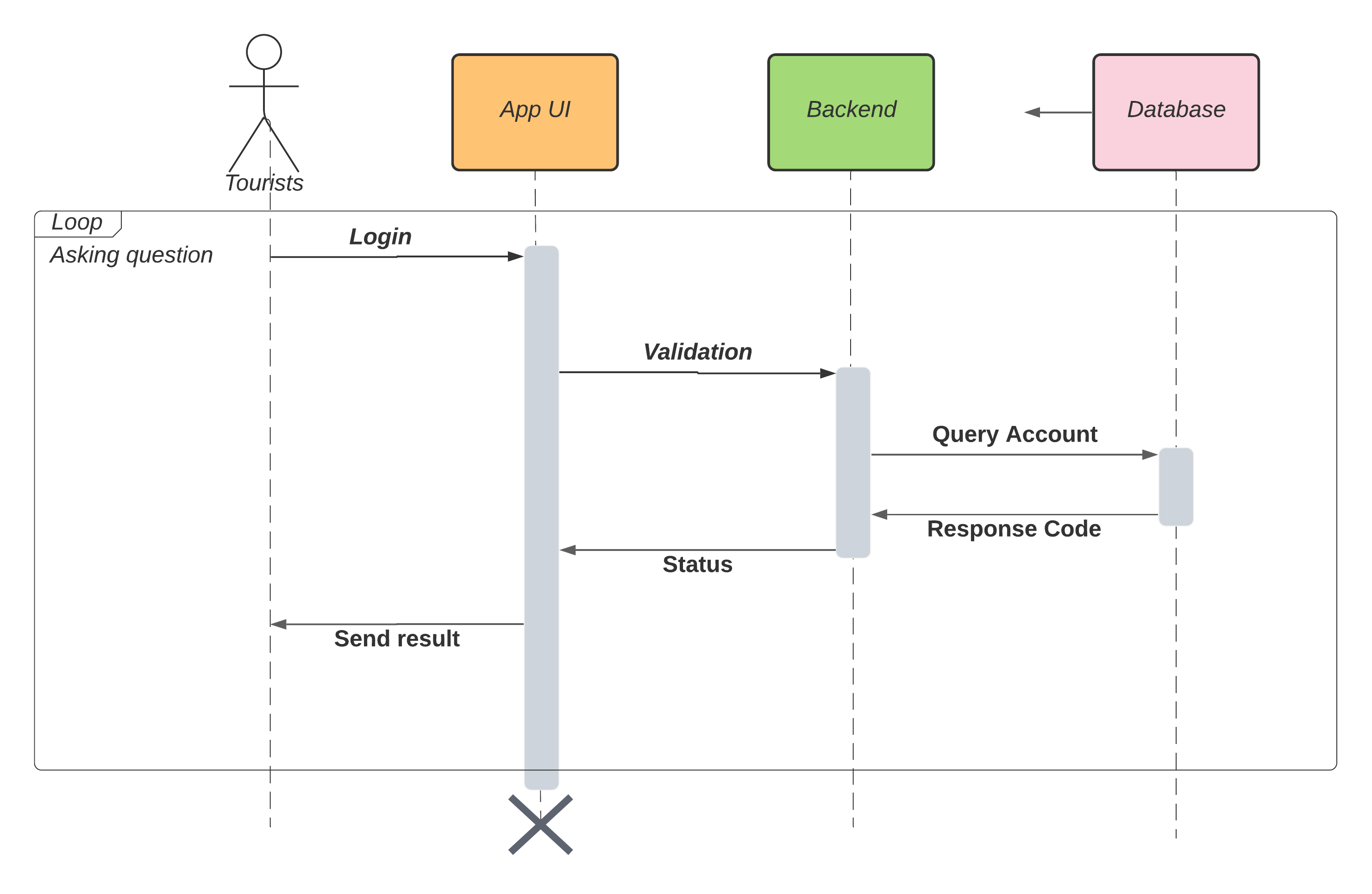
**Prose**

|  |  |
| --- | --- |
| **Element** | **Responsibilities** |
| Application | Web Client is a component that manages and implements interactive functions for users that are handled on the UI. Web Clients interact directly with users, receive requests and call APIs from Web Service to handle and return results. |
| Web service | Web Service is a component that manages and performs activities related to retrieval and storage of data such as get the students list, user authorization. It also interacts with Infura to get data, send transactions and get transaction details. |
| Google authentication service | The service provides APIs for login with Google Account and phone authentication. |
| Chatbot service | The Dialog Flow service API is used to get chat content and train the chatbot. |
| Stripe service | Stripe is a platform that supports payment of international card transactions. |
| Account service | Service API in the backend to register a new account and login to the system. |
| Book tour service | Backend API service to choose and add a tour to the shopping card. |
| Management service | Service interacts with the database and provides APIs for managing the system. |
| Database | Database is a component which contains information of users, tours, tourist attractions, payment histories. All data the system needs |
| Data Source | Data source is a source that provides data to chat bot systems. |

## 3.3 Sequence diagram

Sequence diagram is used to display the sequence of activities. Sequence diagrams show the workflow from a start point to the finish point detailing the many decision paths that exist in the progression of events contained in the activity.

**3.3.1 Login**



**Prose:**

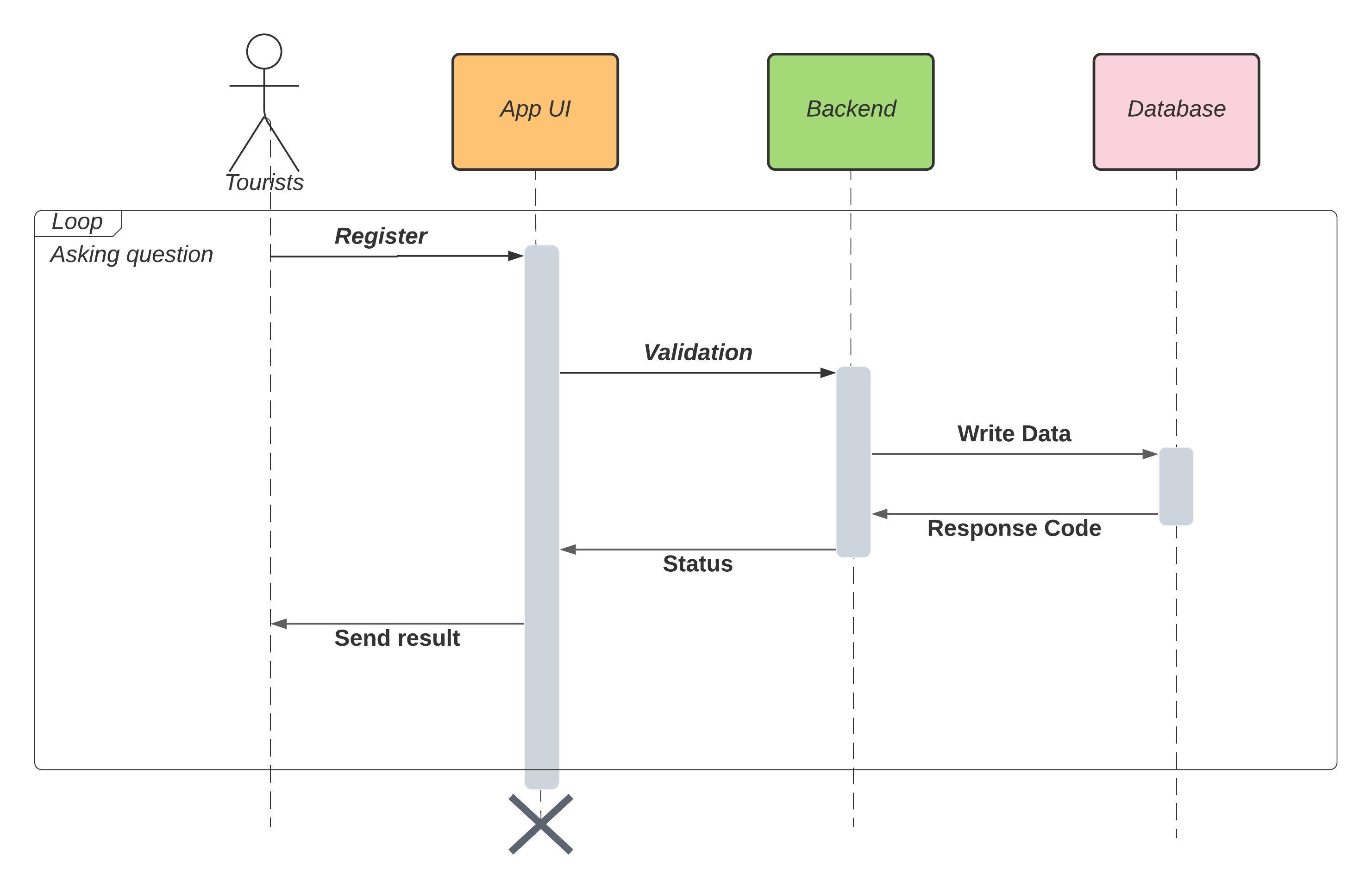
|  |  |
| --- | --- |
| **Element** | **Responsibilities** |
| Users | The person who want to login to the system |
| Client UI | The application or website, where clients interact with the system, run on platform device |
| Web Service Backend | Providing APIs to read data from databases, handle requests from clients. |
| Database | Storing data of a system such as tours, tourist attractions, users and booked tour history. |

**Description:**

In order to use the viBOTour system, users have to login at first by using username and password. Our system receives requests, authenticates and returns login status.

If login successfully, users can use all functions of the system in their permission. Otherwise, their access will be denied.

**3.3.2 Sign up**



**Prose:**

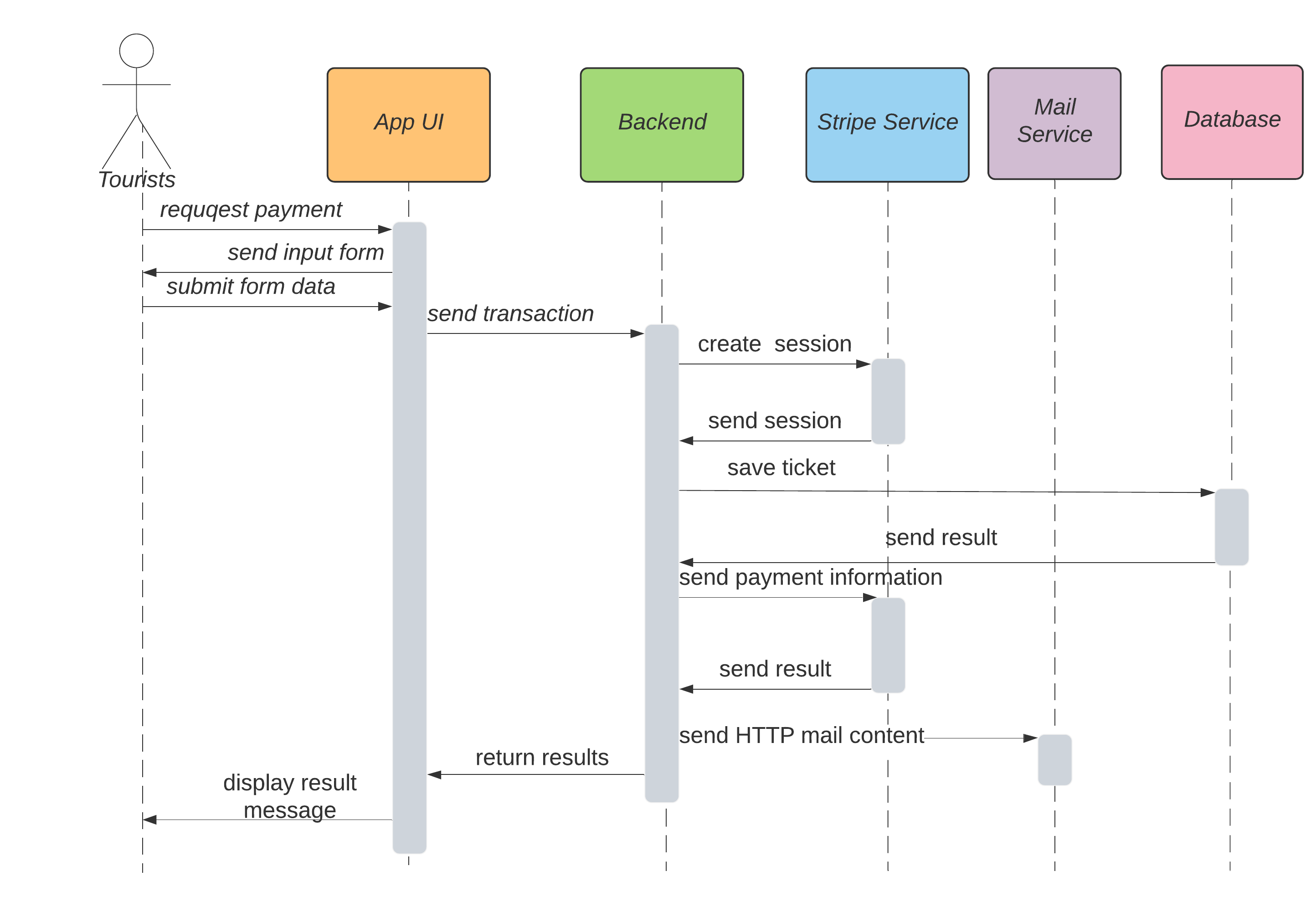
|  |  |
| --- | --- |
| **Element** | **Responsibilities** |
| Tourists | The person who wants to sign up for the system and book tour. |
| Mobile application | The application of system that run on platform device |
| Web Service Backend | Providing APIs to read data from database or write data to database, handle requests from web clients and interact with Infura nodes to send transactions. |
| Database | Storing data of a system such as tours, tourist attractions, users and booked tour history. |

**Description:**

To login to the system, tourists have to register first.

Tourists are allowed to register a new account, to do that they create a new account with email, password and fullname.

**3.3.3 Payment**



**Prose:**

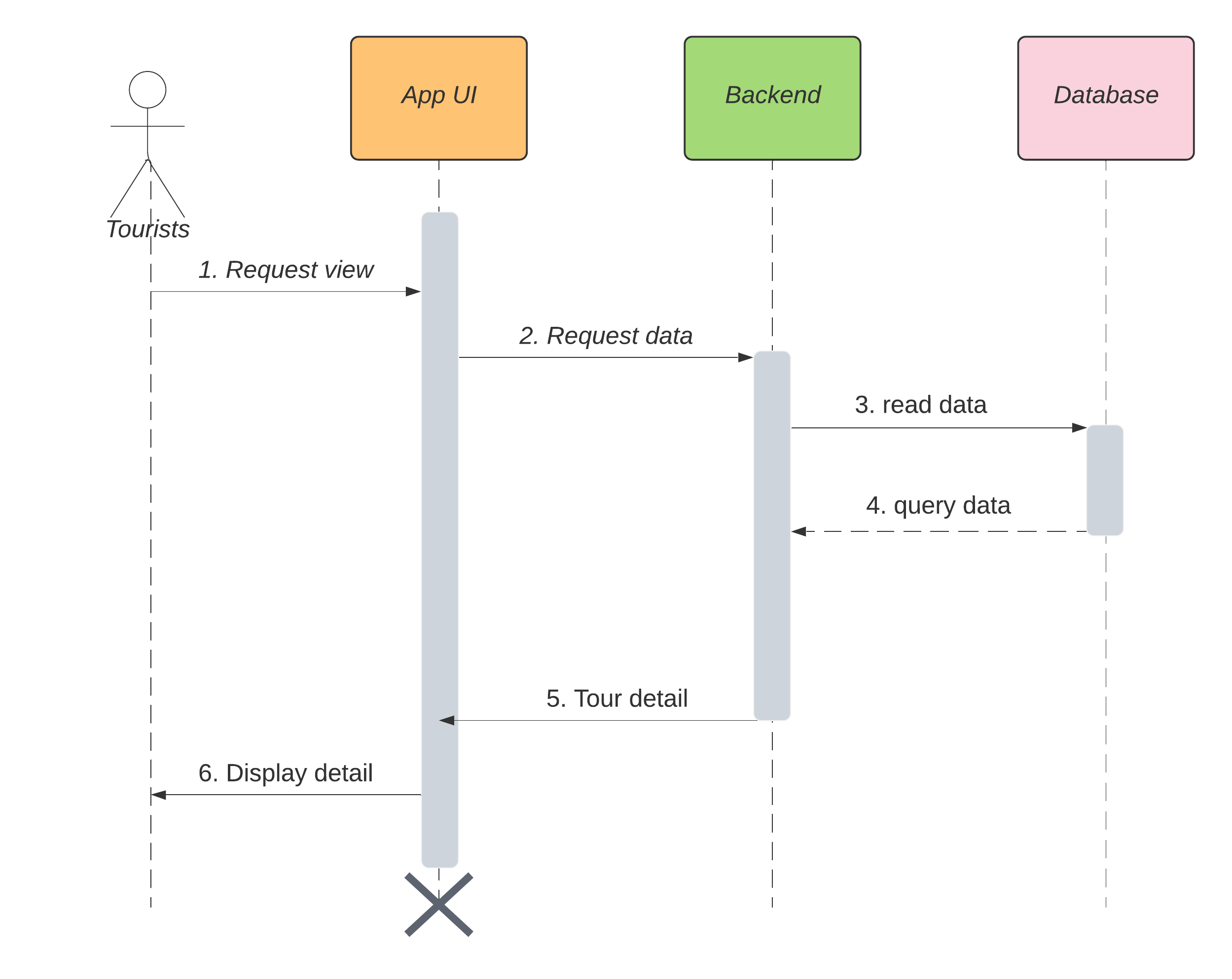
|  |  |
| --- | --- |
| **Element** | **Responsibilities** |
| Tourist | The person who want to payment for the tour their already booked |
| App UI | Payment interface on the application for smartphone |
| Backend | Handling information being provided by tourist and process transaction with the banking system through APIs |
| Stripe Service | The platform supports APIs for payment of international cards. |
| Database | Save transaction data successfully |

**Description:**

To pay, tourists have to login first.

To pay they need to choose for themselves at least one tour, fill in all contact information and confirm their phone number, and then make a payment with their bank account if successful they will receive a ticket which is sent via their email. Otherwise, they will receive an error message and the transaction is canceled

**3.3.4 View tour detail**



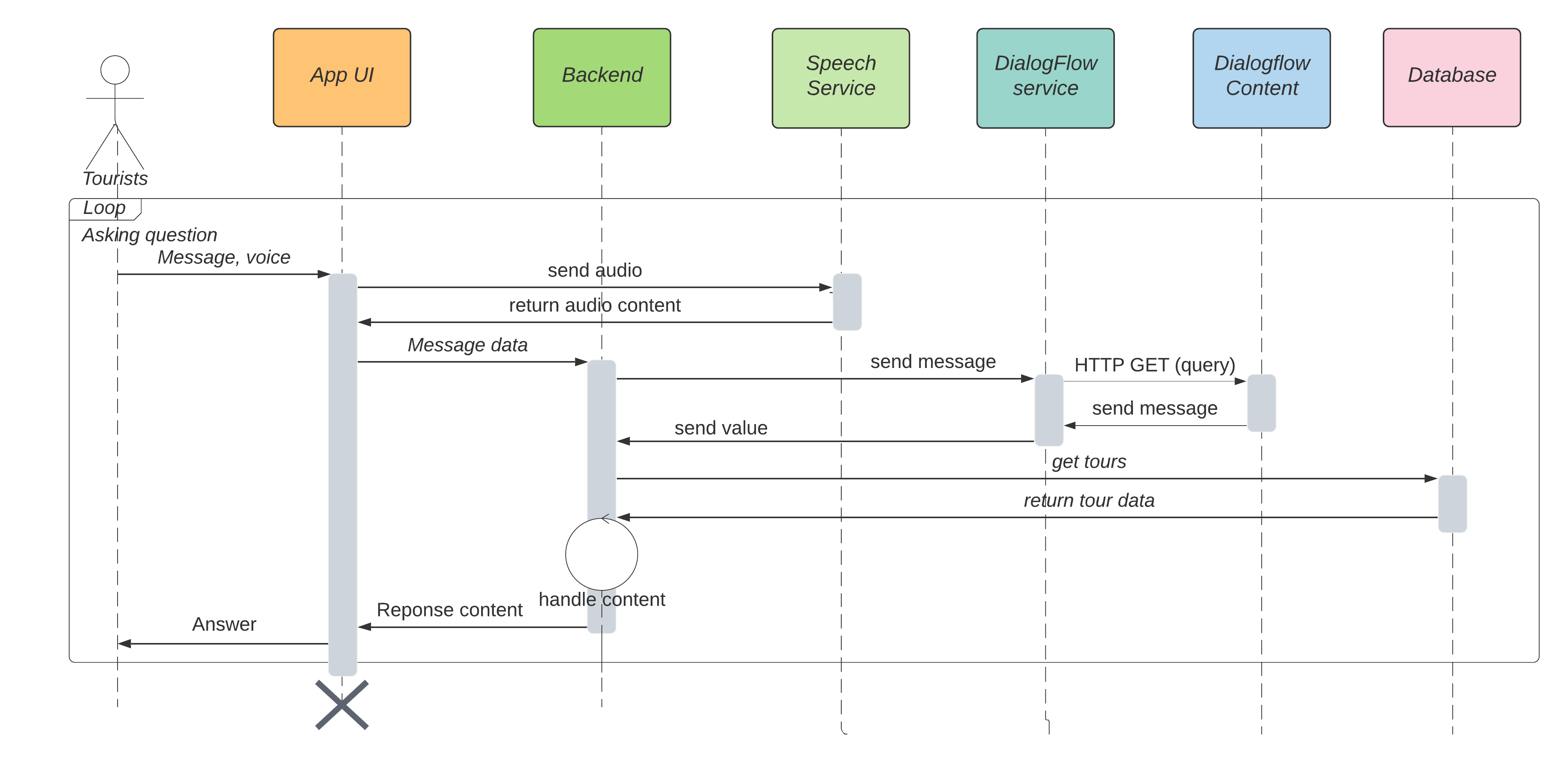
**Prose**

|  |  |
| --- | --- |
| **Element** | **Responsibilities** |
| User | The person who want to view tour detail |
| Mobile application | The application of system that run on platform device |
| Web Service | Providing APIs to read data from database or write data to database, handle requests from web clients and interact with Infura nodes to send transactions. |
| Database | Storing data of systems such as users, students, lecturers, dean and classes. |

**Description:**

Tourists are allowed to view tour details in the app, to do that they have to let the system know which tour they want to view detailed information. We service receives requests, calls API to get data and return tour information.

**3.3.5 Chat bot support**



**Prose**

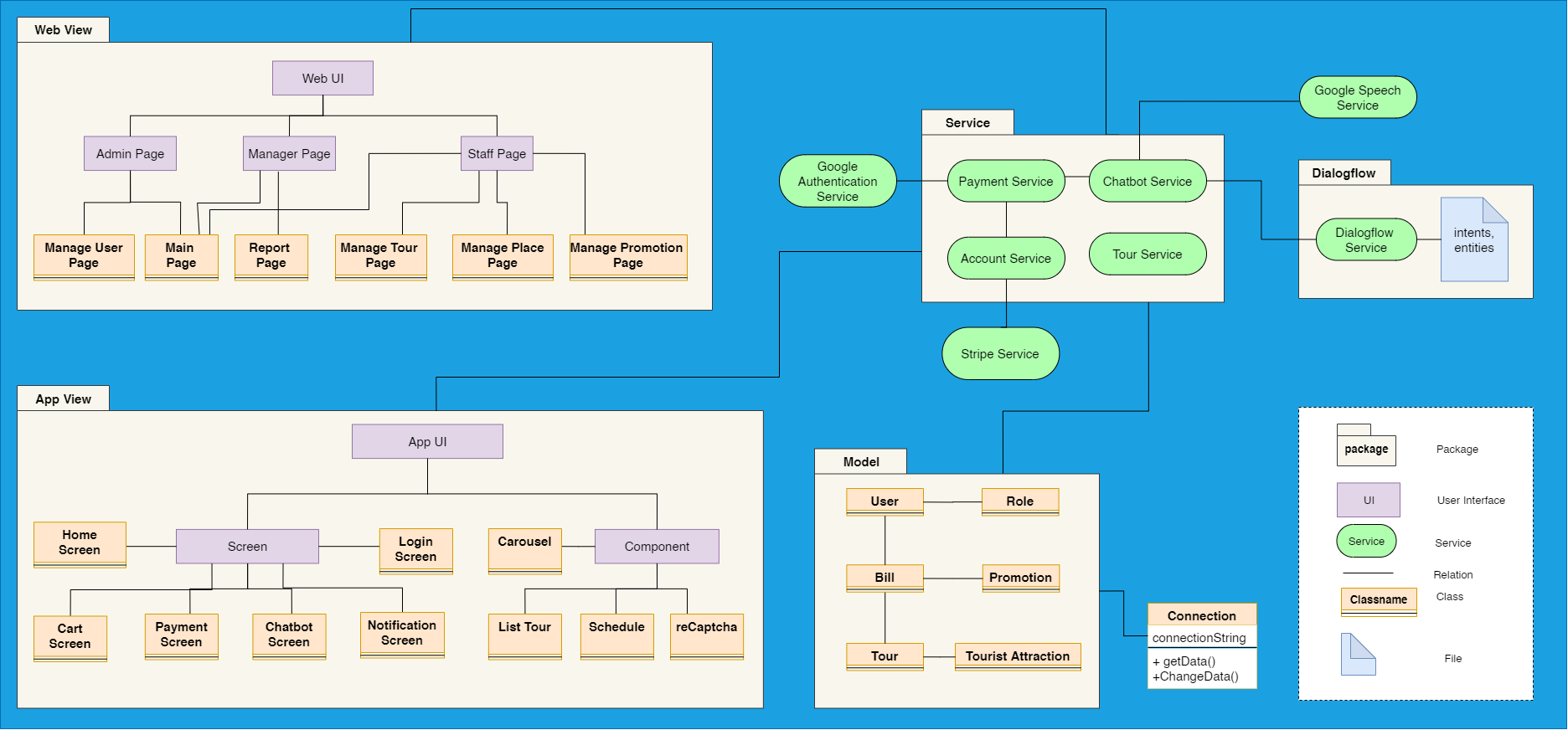
|  |  |
| --- | --- |
| **Element** | **Responsibilities** |
| Tourists | The person who wants to travel and find locations by chatting with the chatbot system. |
| Application UI | The application of system that run on platform device |
| Backend | Providing APIs to read data from database or write data to database, handle requests from web clients and interact with Infura nodes to send transactions. |
| Dialogflow Service | The chatbot service to receive data and response to the mobile application |
| Datasource | Where content chat data such as intents, entities. |

**Description:**

Tourists are allowed to interact with chatbot to receive more traveling information, input is a chat message. Application reads data and calls APIs of Web Service. Then, it requests API of Dialog Flow Service and replies messages to tourists. Tourists can receive suggestions about tourist attractions, tour schedule and book their tour.

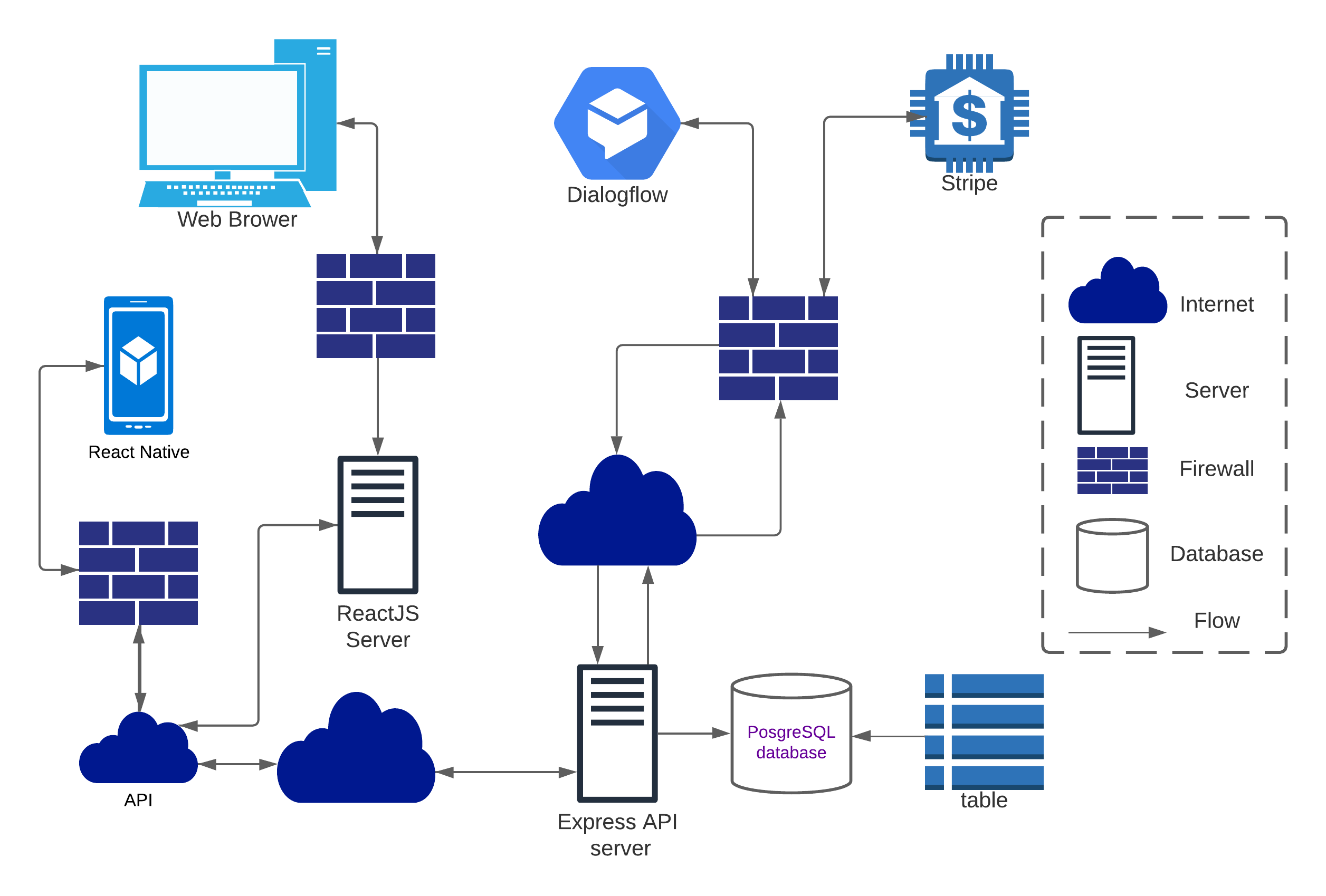
## 

## 3.4 Module view



## 3.5 Allocation view

The allocation view models the run-time architecture of a system. It shows the configuration of the hardware elements when the system is deployed.



**Prose:**