Requirements Analysis Document

City Tours Reservation System

Version 1

Authors	Group 2
Date of creation	05.01.2024
File name	Requirements_Analysis_Document_Reservation_System.docx
Number of pages	16

Contents

1 Introduction

This chapter gives an introduction into the project and describes what purpose it has.

This chapter serves as an introduction to the project, offering an in-depth look at the purpose, scope, and objectives. The document provides a thorough understanding of the tour reservation system and its associated project requirements, offering a comprehensive view.

1.1 Purpose of the System

Briefly describe who might use the system and for which purpose. What is the target group of the application?

The system is designed to cater to individuals like Edward, avid travelers planning to explore various cities in Europe. Users can seamlessly navigate through an extensive list of countries, cities, and tour types, empowering them to plan and book city tours based on their specific preferences. The target audience includes discerning travelers seeking a user-friendly platform to discover, explore, and reserve culturally enriching or culinary tours during their European journeys.

1.2 Scope of the System

Explain the problem your system is solving. How wide is the area you can apply your project to?

The system addresses the growing need for efficient tour reservations by providing a diverse range of countries, cities, and tour options. Users can effortlessly explore available tours without the need for a specific search option, making the application well-suited for travelers looking to plan city tours across different regions in Europe. The platform ensures a seamless user experience by offering an extensive list of cities and tour types, facilitating easy navigation and selection.

1.3 Objectives and Success Criteria of the Project

Explain the project objectives and the criteria you'd use to judge if it was successfully executed.

The primary objectives of the project are as follows:

1. **Efficient Tour Exploration and Reservation:** Users can seamlessly explore and reserve city tours, selecting specific tour types, cities, and preferred time slots. The system aims to provide a user-friendly interface that simplifies the tour discovery process.

2. **Secure Account Creation and Login Functionality:** The reservation process includes account creation and login functionalities, ensuring a secure and personalized experience for users. This feature enhances security and enables users to track and manage their reservations effectively.

3. **Comprehensive Reservation Management:** Users can manage their reservations efficiently by modifying ticket quantities, deleting bookings, and accessing their profile information. The system prioritizes user control and flexibility in handling their tour reservations.

4. **User Authentication and Account Security:** The system maintains a robust security framework through user authentication, requiring account creation for reservation-related actions. This measure ensures a secure environment for users to conduct reservation-related activities.

5. **User Account Recovery:** The inclusion of a "Forgot Password" feature enhances usability. In case of forgotten passwords, users can easily recover their accounts by answering a predefined security question, streamlining the login and recovery process.

The success of the project is contingent upon achieving these objectives, with the product owner's approval contingent on the effective execution of these features. This detailed approach aims to provide a clear and comprehensive overview of the project's scope and goals.

1.4 Definitions, Acronyms and Abbreviations

This part should prevent any wrong interpretations of the terms used in this document. For example, when we say cinema, we mean one of the cinema branches and not the whole cinema company or one of the movie projecting halls. Such explanations are helpful in designing the system correctly. In your document give just a few explanations. ***It's enough to define only a few (3 - 4) terms, just to prove that you understand what kind of content should go here.***

This section aims to prevent any misinterpretations of terms used throughout the document. By providing clear definitions, acronyms, and abbreviations, we ensure a common understanding of key concepts. While this is a simplified example, it demonstrates the importance of defining essential

terms. For our introductory document, it's sufficient to define a few (3-4) key terms relevant to the project.

Here are the definitions of the terms, acronyms, and abbreviations crucial for understanding this document:

Term / Acronym / Abbreviation	Definition
Tour	A guided experience offered in a specific city or area.
City	A geographical location where tours are available.
Reservation	A booking made by a user for a specific tour at a designated time. Each reservation records the number of tickets reserved, allowing for multiple tickets per reservation.
User	An individual registered on the system, enabling interaction with tour reservation features.
Scenario	A specific interaction or use case within the tour reservation system, illustrating various functionalities.

1.5 References

Here one should provide references to the development context (e.g. references to existing systems with a similar purpose). The project described in this document is an independent project and has no dependencies to any other project. It is part of the course IN0827 and as such is a standalone project. This means, no further projects will be built upon the here described project. The same is correct for your project.

There is no previous system, it's a greenfield engineering project.

1.6 Overview

*Give a **very short** overview for the reader, introducing the content which follows in the document.*

This document comprehensively outlines the Tour Reservation System project, covering its purpose, scope, and objectives. It delves into the current and proposed systems, details non-functional requirements, and presents visual system models. Whether examining the existing system, proposed enhancements, or navigating through user interface elements, this document serves as a concise guide to the Tour Reservation System's development and functionality.

2 Current System

This chapter is used to describe the current state of affairs. If your system replaces an existing system, you should describe the functionalities and problems of the old system. Otherwise, you should describe how the tasks supported by the new system are accomplished now (e.g. they are done manually on paper).

Your system is a toy system and we imagine that you're the first ones implementing this kind of software system. ***Imagine and describe a world where there are no other reservation systems for the problem you're trying to solve. You should also assume that there is no pre-existing system for your application.***

The system is an independent project and doesn't serve as a replacement for any other existing system.

Here is an overview of the current approach employed by City Tours for selling tickets through their cash registers.

Customers line up at the City Tour cash register to purchase tickets. The customers must come during the cash register's opening hours and ask for a ticket for a particular city tour. Payments can be made using either cash or card.

Pros	**Cons**
Quick Purchase, No Account Needed	Fixed Operating Hours
No Bank Card Required	Inability to Plan in Advance
Personal Interaction	Queueing Time
Local Support	Risk of Changes in Venue
Online Reviews	No Digital Convenience
	Difficulty in Comparisons
	Missed Online Exclusive Offers
	Dependency on Physical Location
	Paper Ticket Risks

The provided overview highlights that those with the capability to make online bookings for City Tour tickets could gain advantages from this option.

The City Tour Booking System will facilitate online ticket reservations and provide a digital way for searching and securing tickets for preferred city tours.

With the online booking system, customers can conveniently explore available city tours online, view details, and select the most suitable tour based on their preferences and schedule.

Moreover, customers can quickly create an account and initiate the booking process instantly online by choosing the number of tickets.

Customers can also review their booking history and upcoming tours on their profile page, providing a comprehensive overview for better planning. The online system also allows the modification of customers' reservations effortlessly, adapting the number of tickets to accommodate changes in their plans.

In unexpected circumstances (e.g., in the event of a delayed flight), customers are allowed to delete their reservations online easily and quickly. This ensures control over customers' bookings.

Consequently, The City Tour Booking System includes convenience, efficiency, flexibility, and user-friendly features that improve the overall experience for customers.

3 Proposed System

This chapter documents the requirements elicitation and the analysis model of the new system.

3.1 Overview

Present a functional overview of the system.

The general procedure of a user is that the user browses the available cities and the corresponding city tours that are offered. After deciding for a specific city, the user needs to select a certain type of the tour. Following, the user can make a booking if the user is logged in to their account. If the user is logged in, they can always view the made booking and can modify or delete them as desired.

3.2 Non-functional Requirements

In this section, describe user-level requirements that are not directly related to functionality.

In this section each of the non-functional requirements will be listed i.e. all the additional requirements that are not contained in the functional requirement but are still relevant and further constraints will be described.

3.2.1 Usability

The proposed system should be user-friendly and intuitive, ensuring that people of all ages and backgrounds can easily navigate and operate it. This specifically means that no complex and nested graphical components should be used.

3.2.2 Reliability

The list of all available types of city tours for a city should be displayed within 5 seconds after the user clicked on the city. The same time limit is applied to other actions, such as: making a booking, changing or deleting a booking or changing the profile data.

3.2.3 Performance

The system should handle a number of simultaneous requests from different users, providing each with a quick response for any type of actions, such as selecting a city or booking a tour. All the actions performed should have an immediate effect, that is no delay in performance.

The application should be stable when the wrong data is provided. All user inputs should be validated and the user should be notified if they entered some invalid data.

3.2.4 Supportability

The implemented software should be well structured and should have a good code style throughout the project. Existing features are easy to modify and the system can easily be extended with additional requests and features.

3.2.5 Implementation

No implementation details *should be provided in the Requirements Analysis Document, but it makes sense to give some* ***very rough*** *information telling the developers about* ***the target OS***

and platform *for the system that needs to be implemented. This is needed simply because different developers specialize for working on for example Android mobile applications and they would have to learn to develop some Windows stand-alone application. To hire the right people for the project, we need to know what their rough area of work needs to be.*

The implemented system needs to be a stand-alone Java application able to run on normal PC or laptop running Windows.

3.2.6 Interface

The interface of the application should be intuitive and easy to understand. All people interested in booking a city tour should be provided with a smooth selection and booking process.

3.2.7 Packaging

The system should be delivered to the user as a Java package.

3.2.8 Legal

No customer data should be available without logging in. No customer should be able to view or alter the data of other customers. Here the data includes both personal information and the list of bookings.

3.3 System Models

This section should contain the complete functional specification.

3.3.1 Scenarios

Write the scenarios for your application in the form of short paragraphs. Give concrete examples (a specific user does a specific action). Here are some typical scenarios that all the projects should have. ***Add more scenarios if some are missing for your application****.*

3.3.1.1 Scenario 1: View items

Edward is planning on visiting Europe for the first time and wants to know what tours are available for him to get to know the cities there. He sees a list of all the countries that have tours available to them and picks France because that is his first destination. He picks the city of Avignon sees a selection of different tour types and clicks on the culture tour. He sees that the culture tour only occurs in the mornings and he is busy in the mornings. He goes back to the list of tours in Avignon and picks the food tour.

3.3.1.2 Scenario 2: Login and Registration

Edward wants to reserve a spot on the tour but does not yet have that option because he did not log in. He has also not used the system before, so he doesn't have an account. He clicks the "Create an Account" button and gets a form to fill in. He types his first and last name, e-mail and a password that he just invented (1234). He also types in the answer to the security question "What is the name of your first pet" (max) for the case of him forgetting his

password. After he clicks the “Sign up” button in the form to confirm his entry, he gets a message saying that he should create a longer password. He thinks of a better password and clicks “Sign Up” again. This time, a success message is shown and he is already logged in.

3.3.1.3 Scenario 3: Simple Login

Edward is already planning all his trips in Europe and uses the tour reservation system very often, using his account. He was thinking of passing by Spain and he wants to reserve a tour there. He clicks on the “Log In” button in the reservation system interface and logs into the system by typing his e-mail with the corresponding password in a form. After he clicks the “Log In” button in the form, a success message is shown and he continues to browse the system, ready to book a tour when he finds an interesting tour.

3.3.1.4 Scenario 4: Reserve Item

Edward decides he wants to go on the food tour in Avignon when he gets there next week. He found that there is a food tour on Tuesday at 14:00. He is already logged in, so he can start the booking process by clicking the button “Book”. Now he can choose the amount of tickets to book. Edward chooses 7 tickets for him and his family and clicks on the “Book” button to confirm his choice. He sees a success message and he is directed to his profile page he also sees that his new reservation was successful.

3.3.1.5 Scenario 5: Show Profile

Edward wants to check if he reserved enough tickets for his family trip, so he logs in the city tour booking system and looks at his profile page. There he sees his personal information (his first and last name and email address). He also sees all his past and future bookings in a table. He can easily see his booking because all of the past bookings are greyed out.

3.3.1.6 Scenario 6: Modify Reserved Item

Edward’s sister now also plans to come on the family trip so he needs to book her a ticket as well. He goes on his profile page and selects the reservation for Avignon food tour and clicks the “Modify” button. He can now modify the amount of tickets that are booked. He changes the number of tickets from 7 to 8 and clicks the “change reservation” button. Then he sees the success message is led back to the profile page where the modification of the booking is visible.

3.3.1.7 Scenario 7: Delete Reserved Item

Edward’s plane will be delayed so he and his family won’t get there in time for the tour and he needs to delete their tour bookings for the “Avignon Food tour”. He logs into the city tour booking system and goes on his profile page and selects the reservation for the “Avignon Food tour”. He then gets an option to delete the reservation by clicking the “Delete” button. Edward clicks on the booking for “Avignon Food tour” is no longer shown in the bookings table on Edward’s profile.

3.3.1.8 Scenario 8: Log out

Edward wants to be sure his younger cousin doesn't use his city tour booking account while she's playing games on his laptop, so he logs out by clicking the "Log Out" button.

3.3.1.9 Scenario 9: Forgot Password

Edward wants to show the reservation to his uncle before they set out on the trip but he hasn't used his account in a while. He goes to the city tour booking system and clicks on the "Log In" button. He puts in his e-mail and types in his password. A message appears saying that the login failed due to an incorrect password. He goes back to the login page and clicks on the button which says, "Forgot Password". There the security question "What is the name of your first pet" is asked and he puts in his answer which he had previously set and he is taken to a page where he can insert a new password and click on save. Then he is on the main page and already logged in.

3.3.1.10 Scenario 10: Payment

After showing his uncle the booked trip, his uncle insists on paying for this tour. He goes to the "Payment" button. Here his uncle is asked to put his full name, his Iban for his bank account and he is asked to click on a button that states that his uncle accepts that the city tour booking system is allowed to book xx€ from his account for the trip. He then gets an option to pay for the reservation by clicking the "Save" button. Then he sees the success message is led back to the profile page where the modification of the payment is visible.

3.3.1.11 Scenario 11: Reviews

After the tour, Edward wants to review his family trip. He goes to the city tour booking system and clicks on the "Review" button. First, he is asked to rate his tour on a scale of 1 to 5 stars, after choosing 4.5 stars he moves on to the Feedback Section. Here he can give written feedback, so he highlights aspects Edward and his family enjoyed and mentions areas that could be improved. After he finished the review, he clicked "Submit" to send his review. Then he sees the success message is led back to the profile page where the modification of the review is visible.

3.3.2 Use Case Model

Place your use case diagrams here. We will skip the detailed textual description of the use cases here, you can do the same in your document.

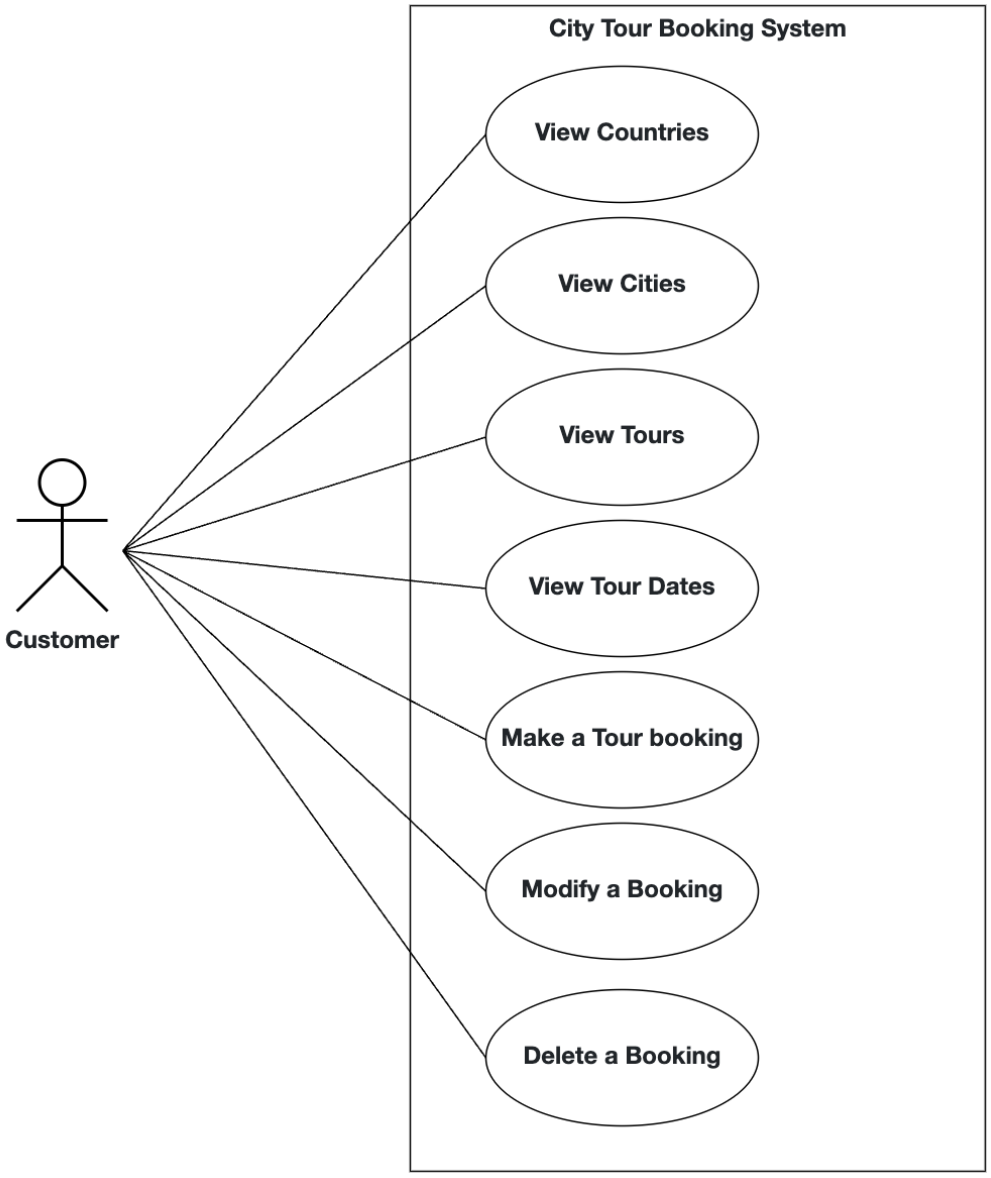

Figure 1 : Use case diagram.

3.3.3 Object Model

Place your class diagram showing the application domain here. No textual description is required.

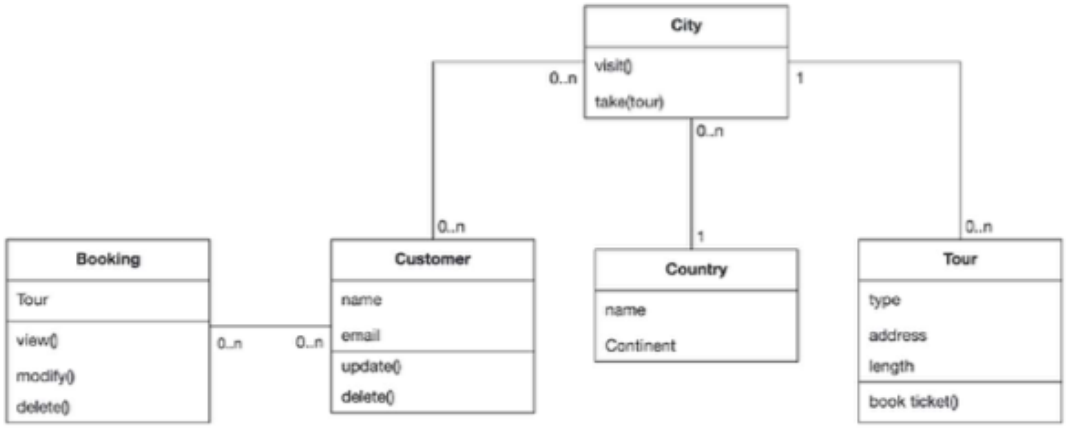

Figure 2: Class diagram from the application domain.

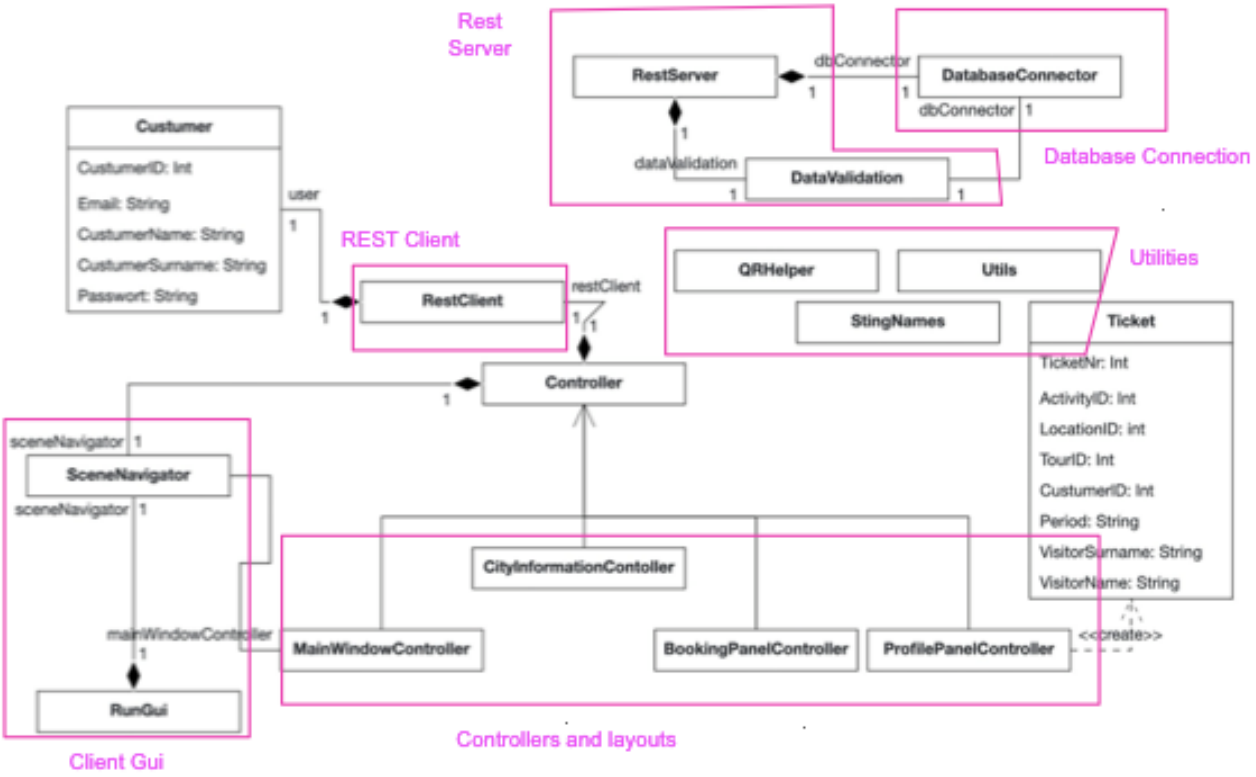

Figure 3: Class diagram from the solution domain.

3.3.4 Dynamic Model

Place your communication diagrams here. No textual description is required.

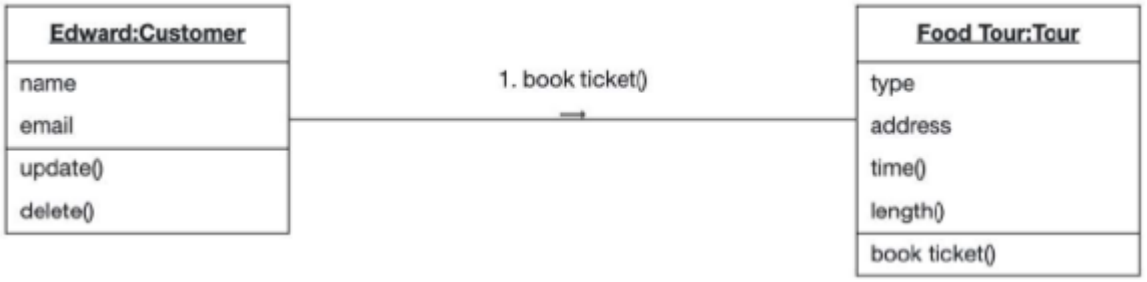

Figure 4: Communication diagram showing ticket booking.

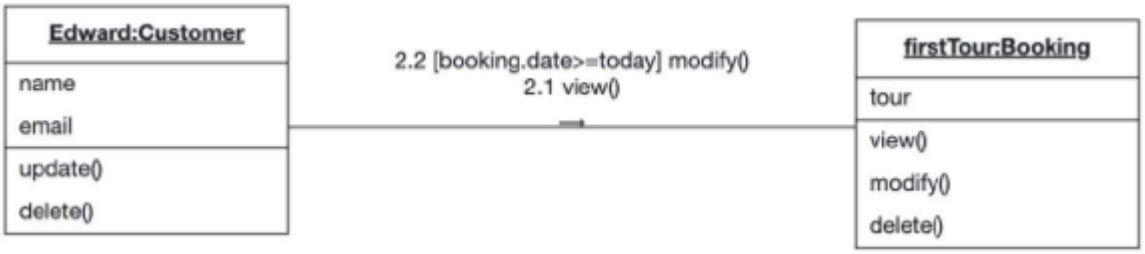

Figure 5: Communication diagram showing booking modification.

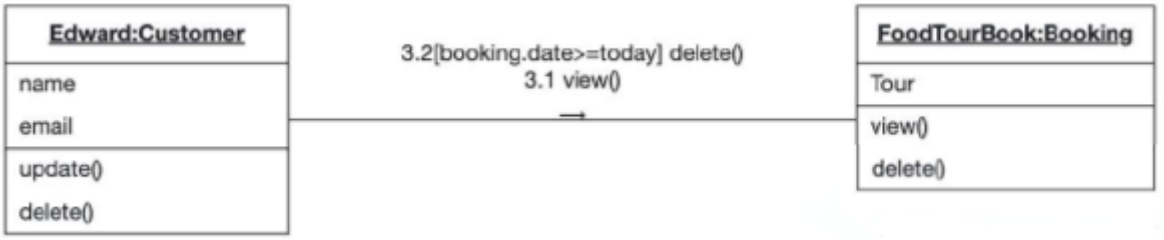

Figure 6: Communication diagram showing booking deletion.

Figure 7: Communication diagram showing a visit to Avignon City to take the tour.

3.3.5 User Interface – Navigational Paths and Screen Mock-ups

This part of the document should illustrate the user interface of the system and navigational paths represented by the sequence of screens.

You can leave this part out *from your document or leave it empty, since you are only working on a small toy project for the purpose of the course.* ***This part will not be graded, but feel free to put mock ups here if you like****, especially if you created them during your discussions with your teammates. Here some mock ups are provided such that you get a clearer picture of what kind of content should go to this section. Mockups don't need to be perfect. They should just provide an idea about what element goes where in the interface. There are many online tools for creating them, but you can also draw them in any way you like as long as they serve their purpose.*

4 Glossary

One of the obstacles between developers and users is differing terminology. To establish a clear terminology, developers identify the participating objects for each use case in a glossary.

In the Cinema Case, the glossary would mostly duplicate the chapter called "Definitions, Acronyms and Abbreviations", so we left it blank and you can leave it blank for your Requirements Analysis Document.

The Glossary section serves as a vital reference point in eliminating potential misinterpretations by establishing a common understanding of key terms used throughout the document. This ensures that developers and users share a precise and unified vocabulary, promoting clarity and consistency in communication.

Tour: A guided experience offered in a specific city or area, providing users with the opportunity to explore and engage in cultural or culinary activities.

City: A geographical location where tours are available, forming the backdrop for users to discover and reserve diverse city experiences.

Reservation: A booking made by a user for a specific tour at a specific time. Each reservation records the number of tickets reserved, allowing for multiple tickets per reservation, facilitating efficient planning for group bookings.

User: An individual registered on the system, enabling interaction with tour reservation features, creating a personalized experience for exploring and managing city tours.

Scenario: A specific interaction or use case within the tour reservation system, illustrating various functionalities. Scenarios serve as practical examples, showcasing how users can seamlessly navigate and utilize the system's features to fulfill their tour-related needs.

Within this glossary, essential terms central to the Tour Reservation System are clearly defined to avoid confusion. From the definition of "Tour" and "City" to the details of a "Reservation" and the significance of a "User," each entry aims to cultivate a shared language. Additionally, the term "Scenario" is explained, providing practical examples of user interactions within the system. This glossary acts as a fundamental for a seamless interpretation of the document, ensuring a cohesive understanding of the project's key concepts.