Software Design Description for Razzaz Tours

Ammar Ahmed, Kareem Yasser, Mohamed Tarek, Mostafa Khaled
Supervised by: Dr. Essam Eliwa, Eng. Nada Ayman

June 19, 2021

Table 1: Document version history

Version	Date	Reason for Change
1.0	16-Jan-2021	SDD first version's description are defined.

GitHub: https://github.com/kareem337/razzaz

Contents

Abstract

Razzaz Tours is a firm that assists individuals with booking services, promotes domestic tourism in Egypt, and recommends the finest spots to visit during their vacations, such as Sharm El-Sheik and Gouna. In addition, help visitors in deciding on their tours. All reservations, such as those for a tour and museum, were gathered by Razzaz Tours using just our online application. Aside from the bespoke service, the application relies heavily on it. Instead of taking chances and attempting to create a bug-free product, we planned to use an incremental development method. A web application is proposed as a solution to aid users in organising leisure and tourism itineraries. These can assist them in having a fantastic experience in Egypt while still allowing them to use our web application.

1 Introduction

1.1 Purpose

The goal of this SDD is to outline the design document's architecture. SDD has a simple yet effective structure. The goal is to outline how the system project will function. SDD's architecture is created via a variety of tools. In order to function effectively, architecture must be efficient and dependable. Tourists are unable to see all of the attractions during their vacation due to a lack of up-to-date information and guidance services. The system's main features include a tour of key sites in Egypt and assistance in locating the location of a chosen destination via a web application. The main aim of this project is to assist travelers in travelling independently and taking full benefit of their stay without missing any of Egypt's iconic landmarks. The goal of this app is to describe the project's software requirements. This software generates a tour plan in Egypt and is used by developers to validate their work. This paper also covers aspects of visitor requirements, as well as a description of the scope. The other major goal is to promote our country to the rest of the world as a tourist destination so that Egypt may generate money to support its people. Egypt can proudly compete with other developed countries with the aid of this App.

1.2 Scope

This software design description (SDD) explains the Razzaz Tours system and communicates the system's core design viewpoints to key design stakeholders. From a client-server perspective, the basic architecture is a web server. Applying the MVC principle and the basic pages will be written in HTML, JS, CSS and PHP.

1.3 Overview

This paper was created to assist with design and comprehension of a Web application for a tourism agency. Hopefully, these documents will support with the system's growth and give a well-structured plan. In this document we will break down the many aspects of our tourism web application. We'll start by looking at different design viewpoints
(Context,Composition,logical,Algorithm,etc) in order to better comprehend the following topics, which will be about the systems data design and human interface design. We'll break the system down into subsystems, then examine different types of data description and management, user interface and different types of users and their functionalities. Finally, we provide some supporting Appendices that could help in more understanding of the software design document.

1.4 Intended audience

This document is targeted (but not limited) to:

- Client: The owner of the problem who provides a clear idea of the software and hardware requirements to the developers be engaged with.
- Developers: Project developers have an advantage of quickly understanding the methodology enabled and personalizing the problem. It is assumed that the reader has a technical background in software design and development.
- Users: Tourists who are interested in visiting any tourism place in Egypt.

1.5 Reference Material

SRS : https://www.overleaf.com/project/6084cdcb11aca6f6d2901d7e

Software Design Document for USS: https://arxiv.org/ftp/arxiv/papers/1005/1005.0169.pdf

1.6 Definitions and Acronyms

Term	Definition
Structured Query Language (SQL)	Structured Query Language (SQL) is a programming language that is typically used in relational database or data stream management systems. It was developed by IBM in the early 1970s and is now an official standard recognized by the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO)
Hypertext Preprocessor (PHP)	PHP is a general-purpose scripting language especially suited to web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994. The PHP reference implementation is now produced by The PHP Group.
Model, View, and Controller (MVC)	MVC separates an application into three components - Model, View, and Controller. Model represents the shape of the data. A class in C sharp is used to describe a model. Model objects store data retrieved from the database. View is a user interface. View display model data to the user and also enables them to modify them. The controller handles the user request. Typically, the user uses the view and raises an HTTP request, which will be handled by the controller.

2 System Overview

2.1 System Scope

Razzaz Tours web application provides to the tourists:

- helps tourists to book their trips and museums easily.
- information about tour packages.
- some social media links.
- Tourists will have the ability to communicate with their guide to fill their needs by chatting and replying to their reviews or by the hotline.
- chatting with admin or employee for any enquiries.
- facility to modify and delete tourist's data as well as client data.

2.2 System objectives

- To provide best trips services to the customer and travel agents.
- To promote responsible and interesting tourism so that people can enjoy their holiday at their favourite places.
- To provide an efficient way to connect with various events.
- To provide a dependability way for booking various trips from one time in a fixable and user-friendly way.

2.3 System Timeline

Table 2: Project name time plan

Id	Task	Start Date	Number of Days	Team Member
1	Home Page	25/5/2021	8	Kareem Yasser,Ammar Ahmed
2	Trip Page	28/5/2021	10	Kareem Yasser,Mostafa Khaled
3	Booking Page	28/5/2021	11	Mostafa,Kareem,Ammar,Mohamed
4	Edit Profile	01/6/2021	6	Mohamed Tarek,Ammar Ahmed
5	About US	04/6/2021	8	Mohamed Tarek,Ammar Ahmed
6	Contact us	10/6/2021	5	Mostafa Khaled,Kareem Yasser
8	Sign in	01/5/2021	6	Mostafa Khaled,Mohamed Tarek
9	Sign up	01/6/2021	6	Mostafa Khaled,Mohamed Tarek
10	Admin Pages	28/05/2021	15	Mohamed Tarek,Ammar Ahmed
11	Data Base Design	25/05/2021	20	Mostafa,Kareem,Ammar,Mohamed

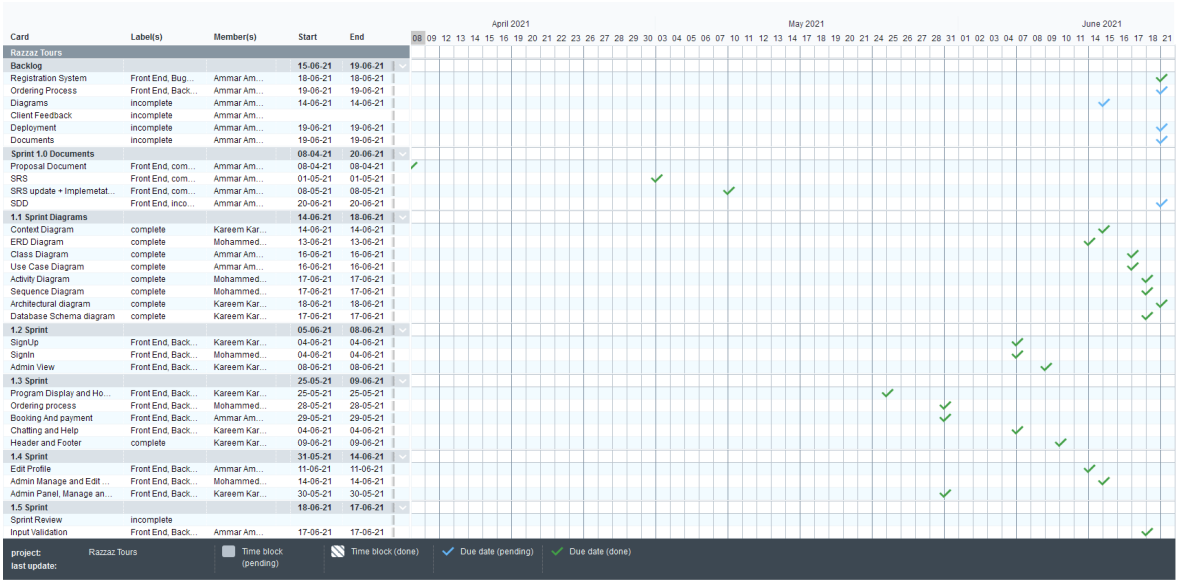


Figure 1: Trello Gant Chart

3 Design viewpoints

3.1 Context viewpoint

Razzaz tours offers different services for different users. In the user perspective. He can reserve a trip or a ticket for a museum. He also can edit his profile and contact admins for any requests. In the admin perspectives, He can edit the users data, trips, museums and reply to users enquiries.

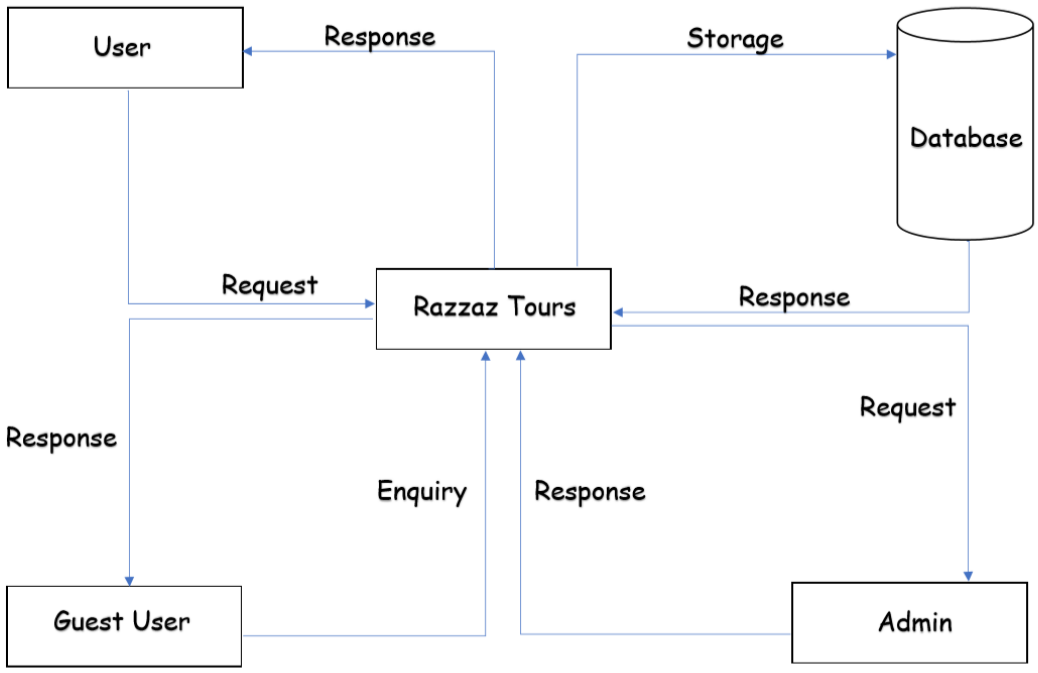


Figure 2: Context Diagram

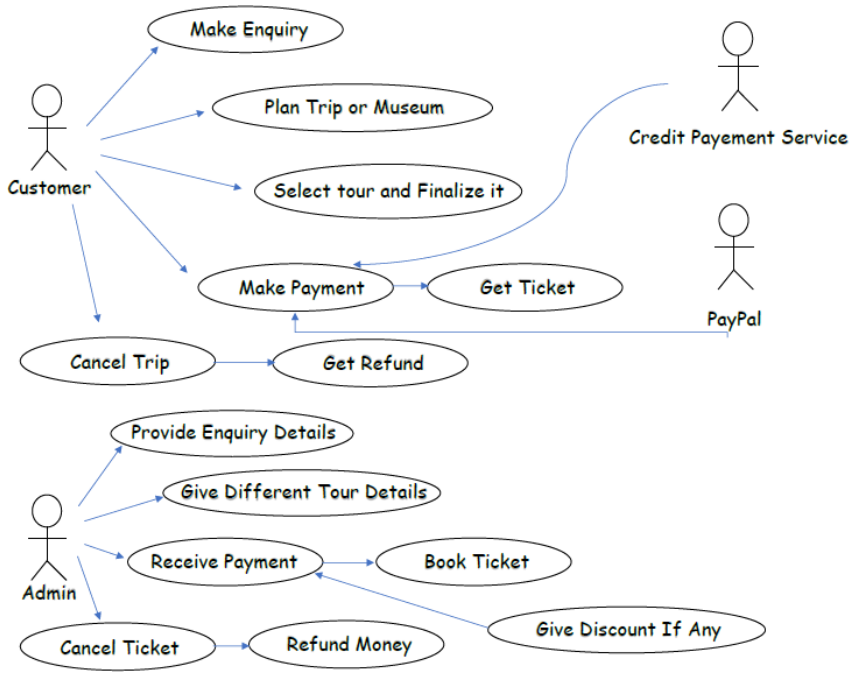


Figure 3: use case diagram

3.2 Composition viewpoint

Razzaz Tours development team used the concept of MVC design pattern in implementing the system which consists of three layers. The view layer depicts and shows the user interface while also visualising data from the model. Tourists may look at the various tours and museums and book the one they choose. The model layer is responsible for connecting databases and transporting data. The controller layer serves as a link between the model and the view layer.

3.2.1 Design Rationale

We used MVC design pattern to avoid the the modification effect on the entire model. As using the MVC framework the modification doesn't affect the entire model. For any web application, the user interface tends to change more frequently than even the business rules of the .net development company. It is obvious that you make frequent changes in your web application like changing colors, fonts, screen layouts, and adding new device support for mobile phones or tablets. Furthermore, because the Model portion is independent of the Views section, adding a new type of view is simple under the MVC approach. As a result, any modifications to the Model will have no impact on the overall design. This will take less time to change.

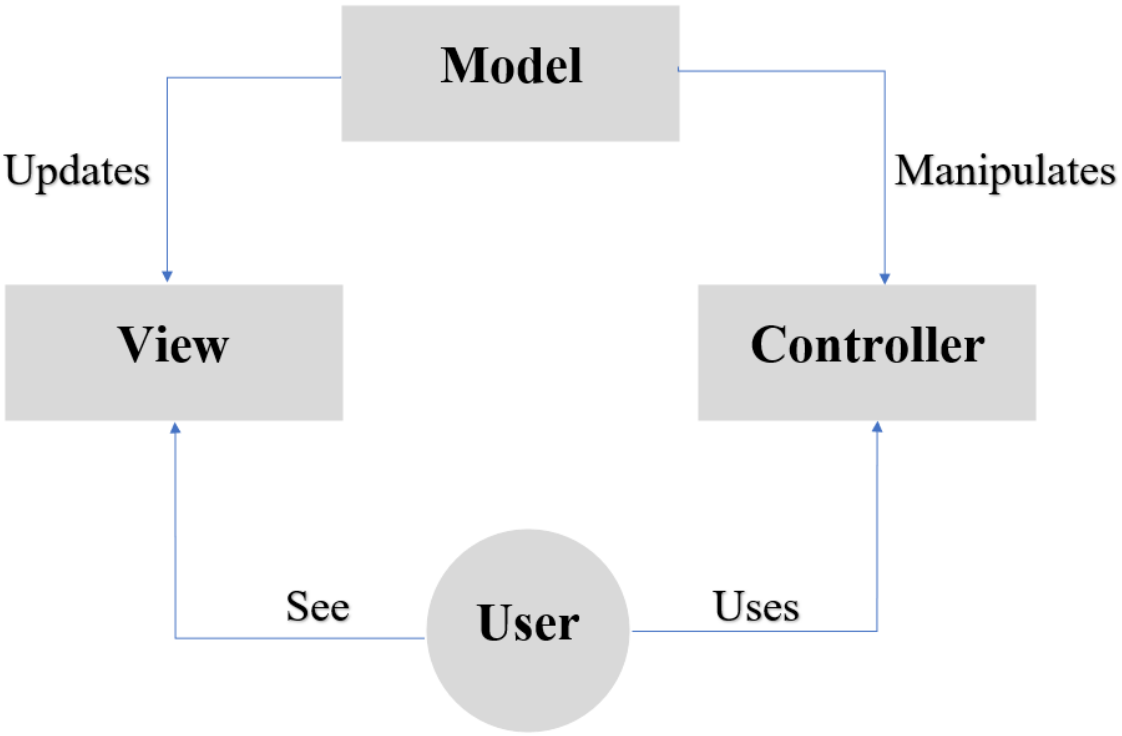


Figure 4: MVC structure

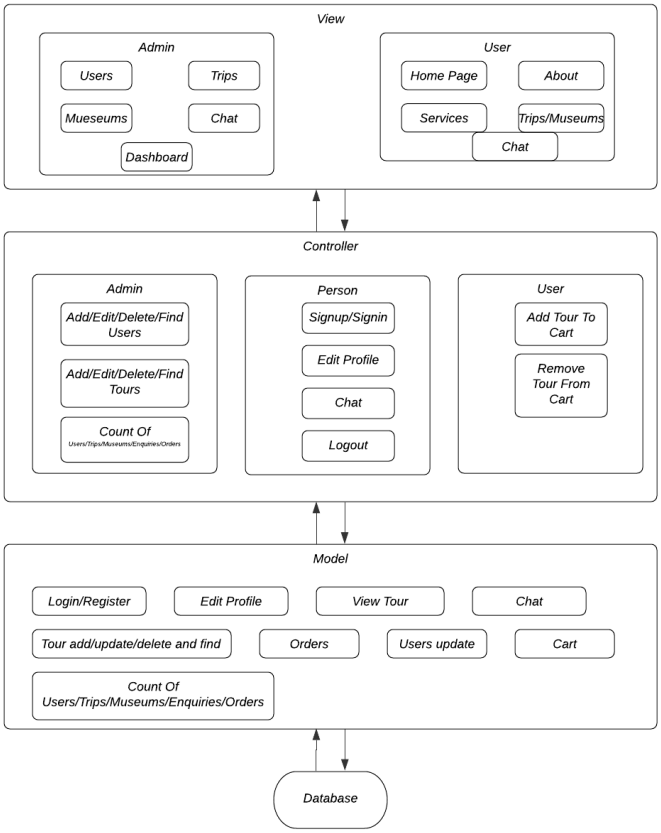


Figure 5: Architectural Design Diagram

3.3 Logical viewpoint

The purpose of the Logical viewpoint is to elaborate existing and designed types and their implementations as classes and interfaces with their structural static relationships.

Design concerns: The Logical viewpoint is used to address the development and reuse of adequate abstractions and their implementations. For any implementation platform, a set of types is readily available for the domain abstractions of interest in a design subject, and a number of new types is to be designed, some of which may be considered for reuse. The main concern is the proper choice of abstractions and their expression in terms of existing types.

In this section you can:

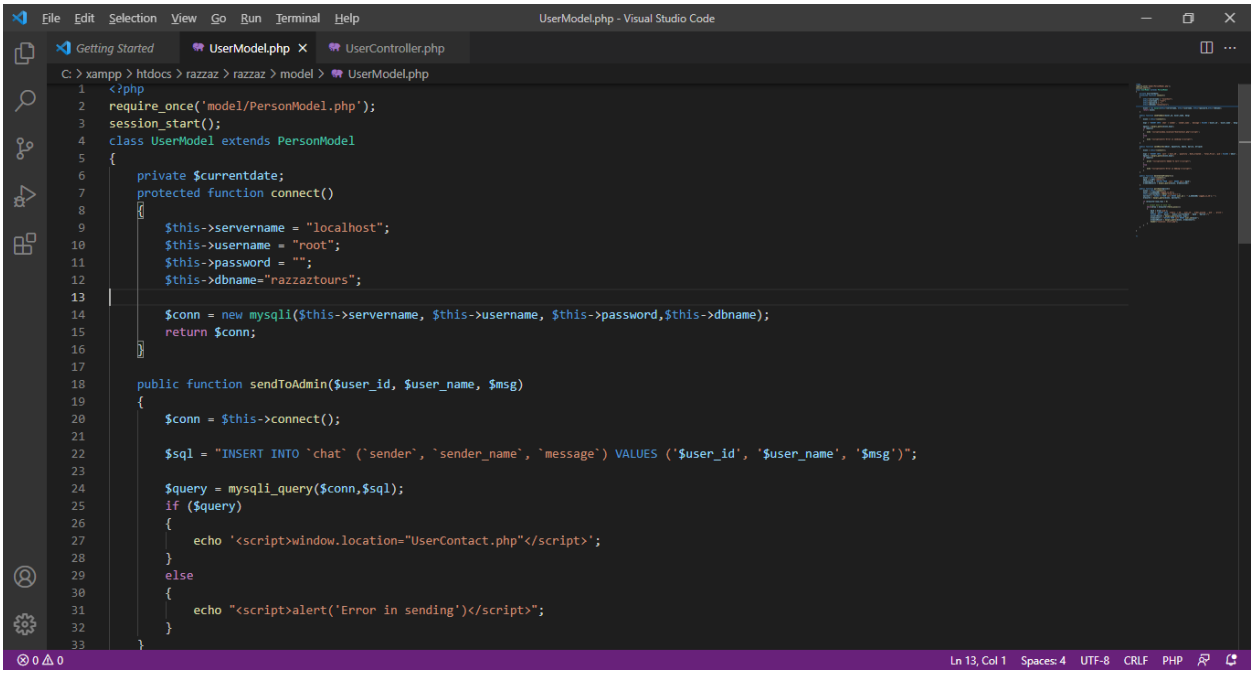
- Provide a UML class diagram to Illustrate static structure (classes, interfaces, and their relationships).
- Provide a table to describe each of your classes.

Table 3: ClassName

Abstract or Concrete:	xxxx
Superclasses	xxxx
Subclasses	xxxx
Purpose	xxxx
Collaborations	xxxx
Attributes	xxxx
Operations	xxxx

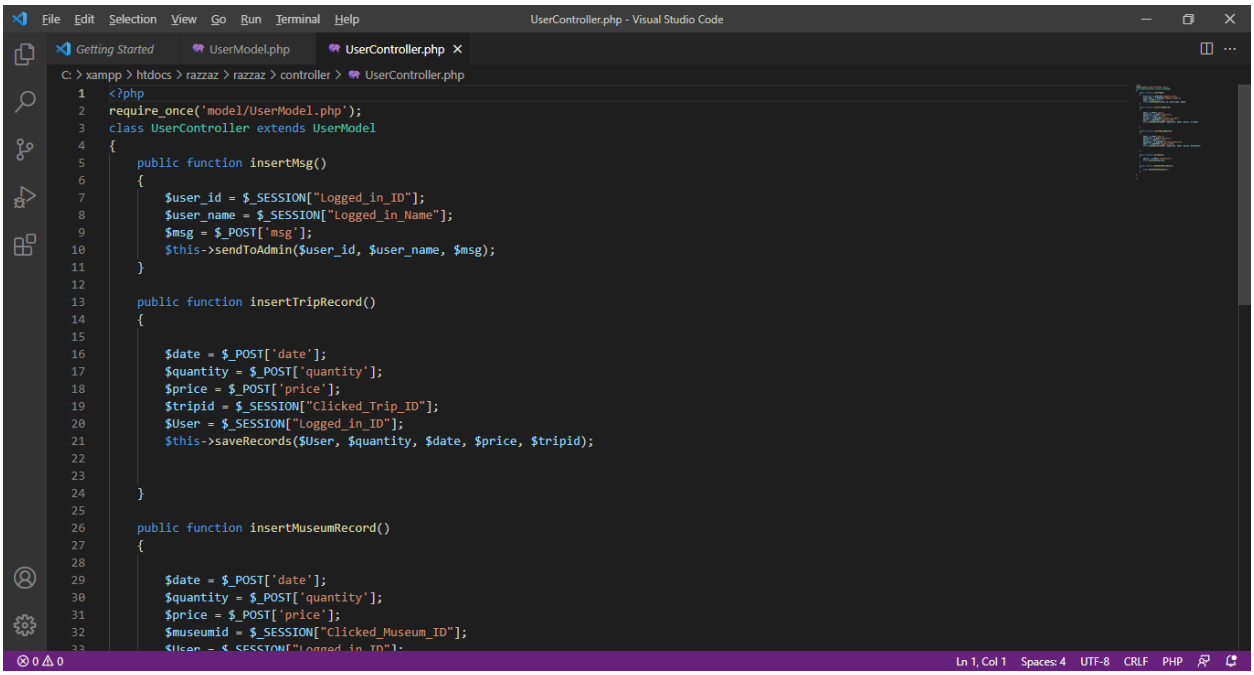
3.4 Algorithm viewpoint

Here is how we implemented the system. We have mentioned that we used MVC concept. The image below shows The user model and a function that a user uses to send message to an admin.



Model

The image below shows The user controller and how the function passes the data to the model to insert it into the database.



Controller

3.5 Interaction viewpoint

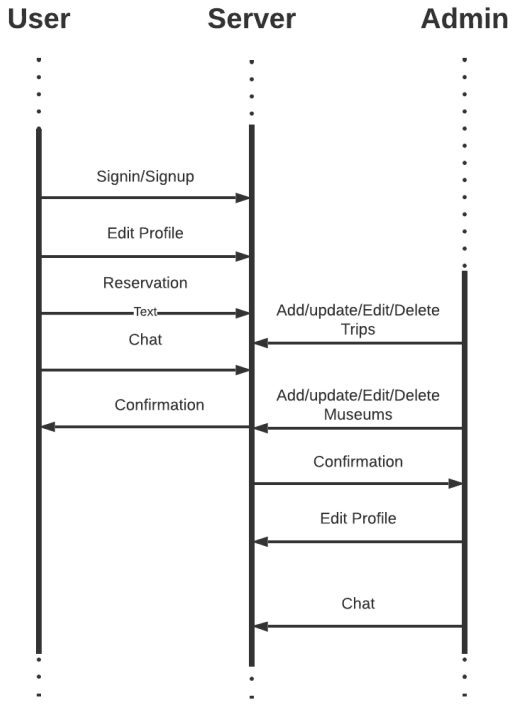


Figure 6: Sequence diagram

3.6 Interface viewpoint

The Interface viewpoint provides programmers, and testers the means to know how to correctly use the provided services. This viewpoint consists of a set of interface specifications for each entity. NOTE: User interfaces are addressed separately in section 5.

4 Data Design

4.1 Data Description

The original format of the data in Razzaz Tours web application is paper based and excel files. The data will be collected through web page forms such as: registration form, edit profile form, booking either trip or museum and contact us form. The main entities of the database are about to be from 10 to 20 columns. While the number of rows may be initially starting from 500. The expected number of users of the system may be 200. The entity keys constructed as follows, each user has his own id which is automatically generated in the database while registering same as products and chat. The system contains data in form of date and time such as in orders and chat tables. The format of date and time are dd/mm/yy and hh/mm/ss respectively. The used language for the database is SQL and the most important queries that will be used are: INSERT, SELECT, DELETE AND UPDATE.

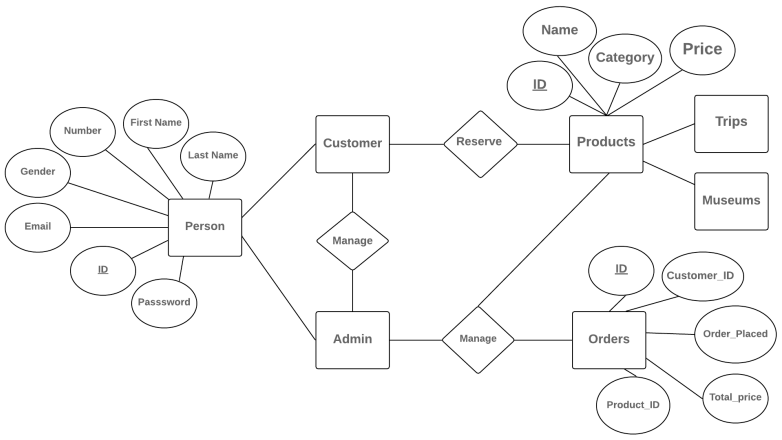


Figure 7: Entity Relationship diagram

4.2 Database design description

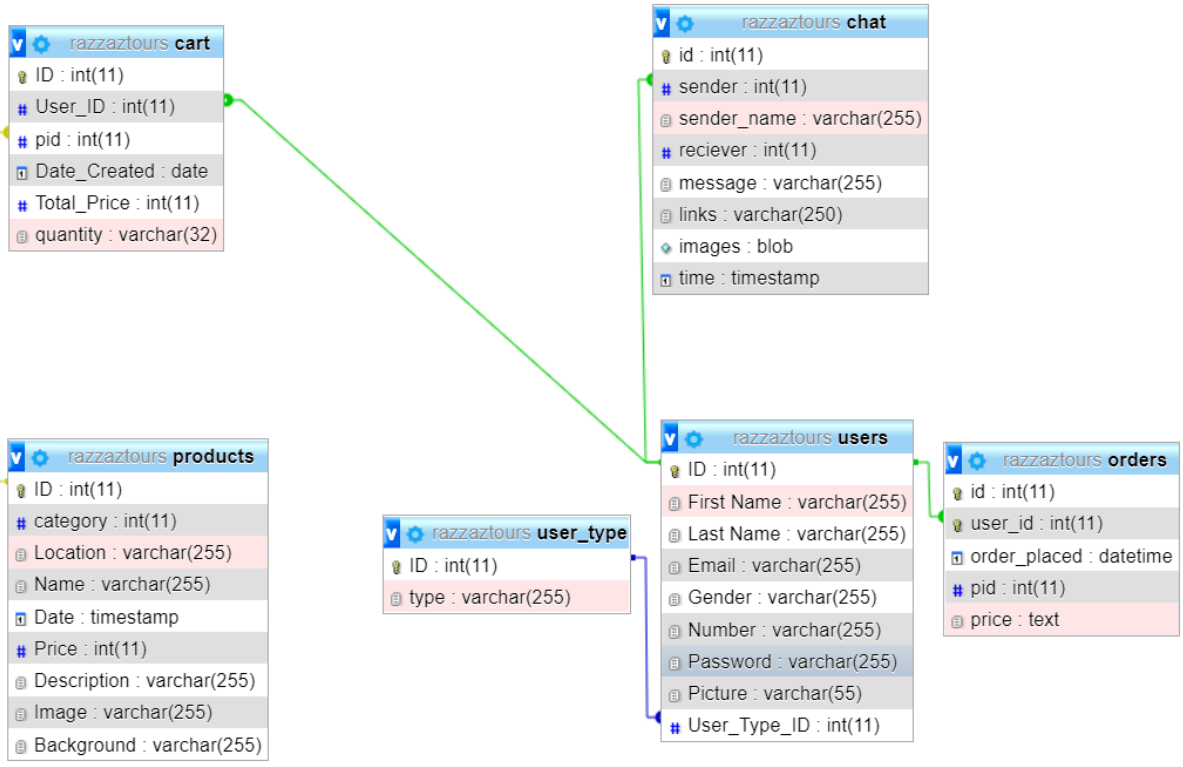


Figure 8: database schema diagram

5 Human Interface Design

5.1 User Interface

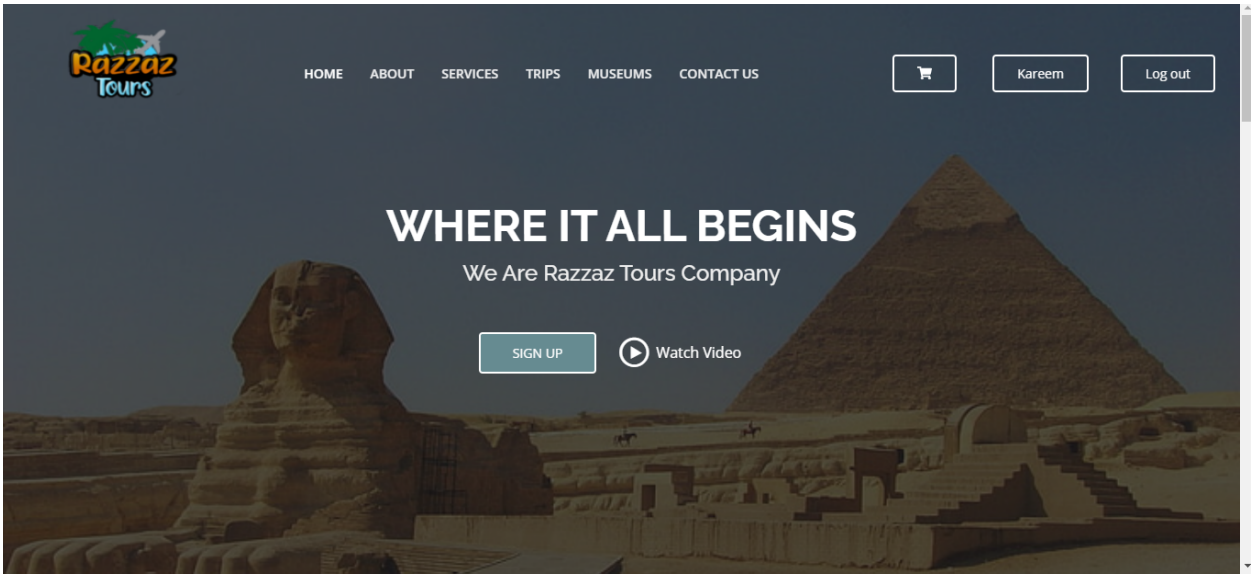
Razzaz tours web application was implemented for two types of users, Normal user, like the tourist, and admins to control the web application. According to this, the web application will have two different perspectives, user side and admin side.

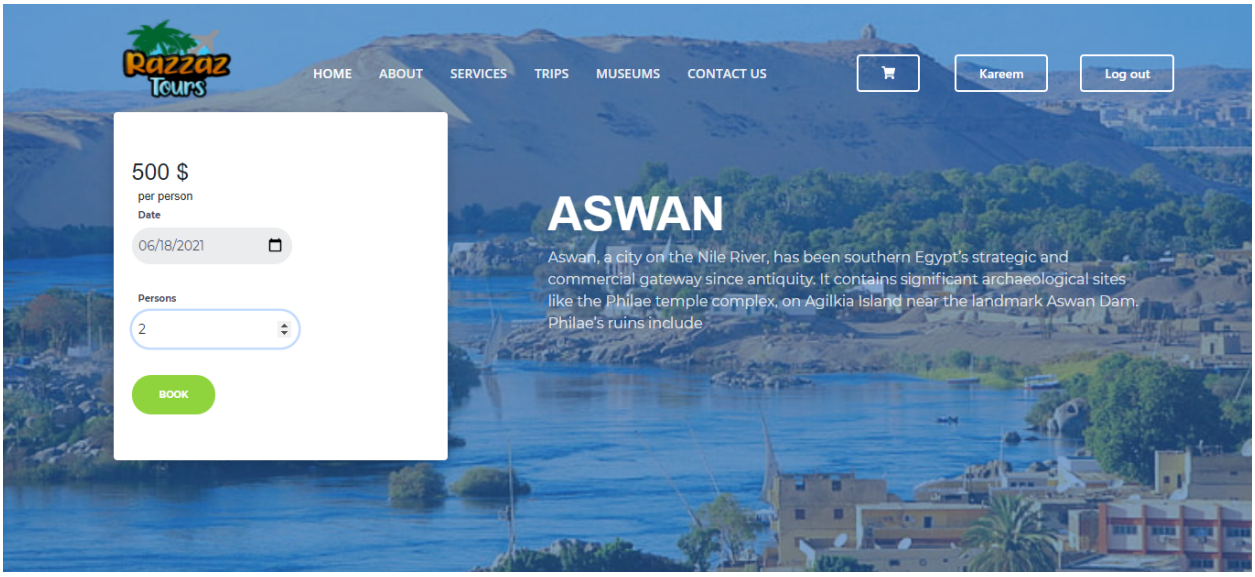
Before any registration, he firstly enters the home page by default. What will appear to him is a background photo of the pyramids of Giza. By scrolling down, the user will see some records of the company, a video and description about tourism in Egypt. After then, He will see the mission, plan, vision and the services the company will offer.

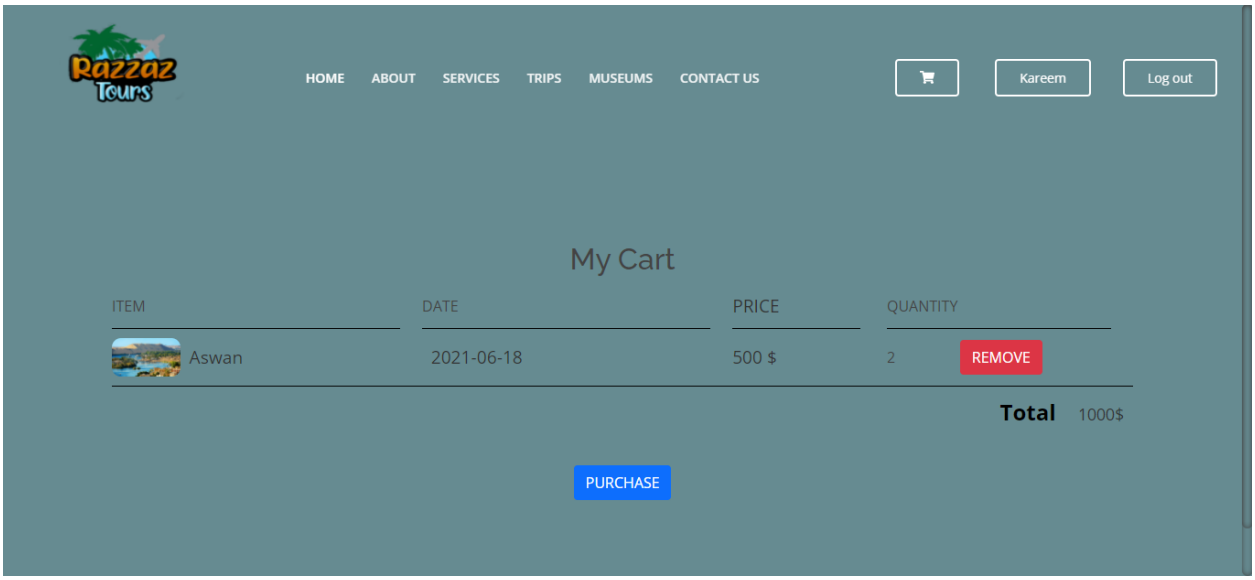
Functionalities of normal users: To make any action like booking or contacting an admin, he should sign up to create an account if this is a first time visit to the web application. Besides, if he should login. The user will have different functionalities which the web application will offer. Firstly, to order a trip or a museum, he will click trips or museums from navigation bar. After then he will choose his desired destination. Then he will choose a date and how many tickets he wants. Finally, he will go to the cart to purchase his order. Secondly, he can contact an admin if he faced any problem by just clicking the button contact us. Thirdly, he can edit his profile by clicking on his name in the navigation bar. He can change his personal details, update his profile picture or simply delete his account. Finally, he can log out by clicking on log out in the navigation bar.

Functionalities of admins: When the admin log in, dashboard page will be displayed. He will have five different functionalities to make. Firstly, he can update or delete any user he will choose. He can search for a specific user to make any of the mentioned actions on him. Secondly, he can add or delete a trip or museum if he wants just by clicking on it is button in the dashboard or the menu bar. Thirdly he can reply to any users who contacted the admin any time by clicking on enquiries. Finally, he can log out by clicking on the icon log out in the menu bar.

5.2 Screen Images







6 Requirements Matrix

Table 4: Requirements Ratrix

Req. ID	Req Desc	Class	Test Cases ID	Status
R01	validate inputs	person	TC01	Developed
R02	add product	admin	TC02	Developed
R03	edit product	admin	TC03	Developed
R04	delete product	admin	TC04	Developed
R05	find product	admin and user	TC05	Developed
R06	book product	user	TC06	Developed
R07	payment	user	TC07	In progress
R08	chat	admin and user	TC08	Developed
R09	reviews and rate	user	TC09	In progress

7 APPENDICES

Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.

7.1 Github

Add screenshots from Github repository showing your project.

Trello: https://trello.com/invite/b/BIVRWXLn/c97e2af121fec0e9e826e8e056b5b079/razzaz-tours