
**CS491 SENIOR DESIGN PROJECT**

**HIGH-LEVEL DESIGN REPORT**

**MedTour**


| | |
|---|---|
| **Asaf Kağan Bezgin** | **21402006** |
| **Hüseyin Taşkesen** | **21402271** |
| **Akant Atılgan** | **21502406** |
| **Skerd Xhafa** | **21503353** |

**Table of Contents**

## 1. Introduction

Medical tourism is rapidly developing in Turkey and apparently it will become a much more popular sector in the future. There are many patients all over the world visiting the country in order to have operations. To be more specific; in 2017 seven hundred thousand patients are warmly welcomed. In 2008, Turkey hosted seventy-five thousand visitors and the numbers increased tenfold and the future figures expected to hit more than a million according to ISTUSAD.

The reason behind the high demand lies under economic reasons and the quality of the operations. “The cost of angiography is $47,000 in the U.S., $13,000 in Singapore, $11,000 in India and $10,000 in Thailand, while $5,000 in Turkey,” [1].

This project is aiming to become the main platform of medical tourism in Turkey by providing a web-based service. First mission is to embody transportation, accommodation and surgical information and provide the options to the users. Our system will consist of two parts: clinic side and patient side. In patient side, there will be many alternatives of aforementioned choices. Clinics and hospitals are going to be listed according to the surgery chosen by customer and the doctors are going to be listed with their profiles including their background and previous operations which will help the user make an accurate selection. Than the user is going to have the option to choose where to stay. Many hotels are going to be listed as well and the final step will be the transportation. The existing flights are going to be provided between the specified dates. By using MedTour, a patient could prepare a complete medical trip package to Turkey without getting distracted from different sources.

MedTour is also very advantageous for the clinics to find the patients all around the world easily and secure. We will be selective about working with high-quality clinics and experienced doctors to give better experience to our users. In clinic side, doctors will be able to create their profile and start waiting for their customers.

Furthermore, the system is going to offer certain privileges to the customers in order to increase our sales. This feature will make our system appealing to patients. We will also provide discounts in local shops, restaurants etc. where MedTour has a partnership. We believe that these privileges will take us one step further from other medical tourism companies, because we will use the advantage of being a local company.

### 1.1. Purpose of The System

Today, there are several websites which are specifically for finding a clinic and arranging an appointment with the desired doctor. However, one of the most commonly used platform called *WhatClinic* does not provide accomodation nor transportation. Meaning that you can only find yourself a doctor from the system. In addition to that, since it is a worldwide platform, the amount of doctors and clinics are limited and prices are very high in Turkey since it is calculated on a different currency. *Medigo* is an another example which has the same properties.

The purpose *MedTour* system is to make a difference compared to the existing examples. The main difference is going to be targeting only Turkey for people all around the world. The second is providing accomodation and transportation options to customers and giving the users the chance to get a full package of medical treatment tour from our system.

At the end of implementation, the expectations from *MedTour* is that it is going to be the medical tourism monopoly in Turkey.

### 1.2. Design Goals and Trade-Offs

By analyzing the design goals and trade-offs of our system, we define critical aspects of our project. Then by these definitions, we can prioritize and improve some aspects of the project in order to make the product relatable with the expectations.

### 1.2.1. Design Goals

Following design goals are going to help articulate the problem our product is going to solve. A clear direction, purpose and intent is going to be followed.

#### 1.2.1.1. Usability

The system is about medical treatment which is a very sensitive issue for everyone. Therefore, the system should be user-friendly by all means. Since it is about healthcare, a large customer base is targeted. In order to maintain the customer base, the system must have a user-friendly interface.

#### 1.2.1.2. Privacy

Medical informations are very private and sensitive information commonly agreed all around the globe.Therefore, our system is going to store information of our customers very carefully and it is going to be strictly forbidden to share customer information with any of the enterprises. It is also very important to work with the Republic of Turkey Ministry of Health.

#### 1.2.1.3. Security

Besides the medical informations of our customers, *MedTour* system is going to ask for payment information at the end of the procedures followed through our system. Therefore, necessary security must be provided in order to keep our users' payment information safe.

#### 1.2.1.4. Performance

As it is explained above, healthcare is a very sensitive issue. Therefore, the system must have perform a good performance during the procedures followed by our customers through their *MedTour* experience. A low-performance system might cause losing customer base which is unacceptable from our end.

#### 1.2.1.5. Reliability

Reliability is the most important design goal for the *MedTour* system. A non-reliable platform means immediately losing the customer base. To gain

customer base and maintain it, our system must be reliable at all services we are providing to our customers. From sign-up to payment, every single process must give the confidence to the customer.

#### 1.2.1.6. Availability

It is very important for us to provide a 24/7 service to our customers. *MedTour* customer base consists of people all around the globe. Therefore, time differences are going to occur which requires attention. Keeping in mind that our system is about health care, in order to provide a reliable service to our customers, *MedTour* should always be available to everyone anytime.

### 1.2.2. Trade-Offs

Currently, there is only one trade-off which should be mentioned about *MedTour*.

#### 1.2.2.1. Portability vs Reliability

*MedTour* team decided not to develop a system rather than a web-based application. We believe that medical treatment systems are serious applications and mobile applications are not proper platforms to take vital decisions such as deciding having a surgery or any kind of medical treatment. Therefore, as a team, we are going to move forward with only a web-based platform.

In the future, a companion application might be developed with respect to the feedbacks of the users and our observations on our system.

## 1.3. Definitions, Keywords, Acronyms, Abbreviations

ISTUSAD - Istanbul Health Tourism Association

### 1.4. Overview

*MedTour* is a web-based application which provides a wide range of medical treatments, clinics which are suitable for the desired treatment, doctors who are experienced in the domain of the desired treatment and accommodation & transportation information between the dates specified. The system creates a tour package located in Turkey and help people all around the world to recruit. With a user-friendly interface, *MedTour* is going to become health tourism monopoly, following many more design goals. In order to achieve the overall goal, suitable architectural structure is going to be used. The choices are going to be made in order to fulfill the design goals specified before the implementation.

## 2. Current System Architecture

Currently there are two similar systems exist compared to *MedTour*. However, certain differences strikes as we compare those existing systems with the MedTour.

### 2.1. WhatClinic


*Figure 1*

*Figure 1* shows the landing page of the *WhatClinic*. As it is mentioned, the targeted customer could be from anywhere and he/she has multiple country and treatment options to proceed. However, there is no transportation or accommodation service provided by the system.

### 2.2. Medigo

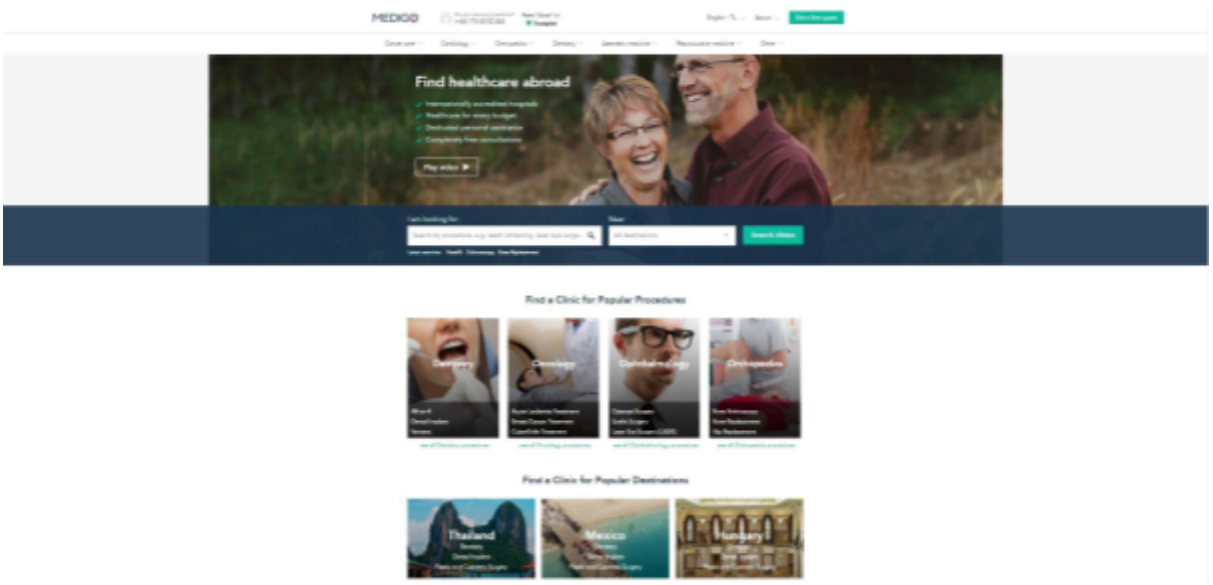



*Figure 2*

*Figure 2* shows the landing page of *Medigo*. It is also very similar to what clinic. It has a slightly better user interface. The idea of the system is same which is letting customers to chose the medical treatment and the desired country. Again transportation and accommodation is not provided.

## 3. Proposed Software Architecture

### 3.1. Overview

MedTour software architecture will take advantage of many features that will be broken down and explained in detail to fully show how the system aims to work. One of them is System Decomposition, that will be used to present the system's

structures and all its composing subsystems. This decomposition is explained and illustrated in detail by the diagrams, as shown in section 3.2. After that hardware and software mapping will be explained, to show how the system will allocate resources and how it will work in different hardware settings. Persistent Data Management, will introduce the main data the system will use including databases that will store all data and objects that will be used to implement the system. In Access Control and Security, the boundaries of all users, what can and cannot happen and all the security that will be used shall be explained. Global Software Control clarifies the general flow of the system. At the end of the section, the Boundary Conditions will be explained containing Initialization, Termination and Failure Conditions of our MedTour system.

### 3.2. Subsystem Decomposition

MedTour will be built on 3-tier architectural design which separate the subsystems to allow updating them with low coupling. Therefore, the tiers are presentation, logic and data. Below is a diagram showing the complete high level structure of MedTour (Figure I).

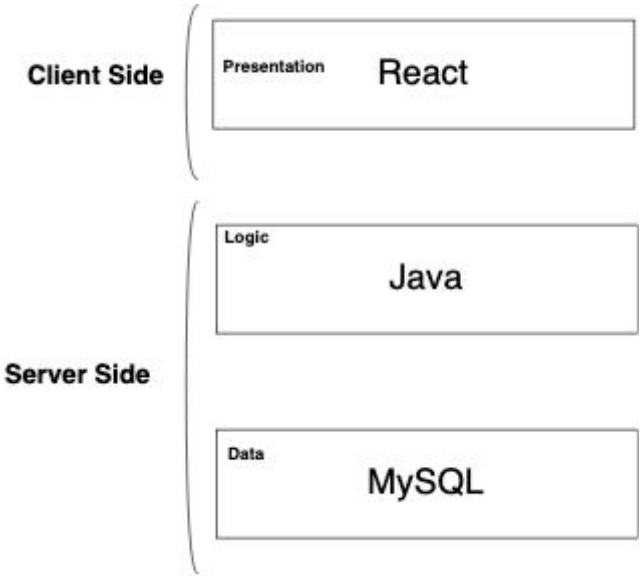



**Figure 1.** Architecture of the system

The presentation layer is in the client-side, whereas logic and data layers are in the server-side. Presentation layer is listening the user actions on front-end and responsible for making REST calls by sending HTTP commands to the related endpoint of the server.

Logic tier is handling the manipulation and the delivery of the data to the client-side, named Presentation in our subsystem design. Controllers and services in Logic tier are responsible for handling the logic behind the REST calls: including filtering and aggregating data. In the end the processed data is passed to the presentation layer which communicates to the server using logic layer. REST API module contains HTTP endpoints which will receive client-side requests. Data tier is responsible for communicating with database and retrieving data required for logic tier to process and return back to the presentation tier. Wrapper package will be responsible to abstract out the database operations which cover every functionality that is needed by Logic subsystem. Using this 3-Tier system architecture, we believe that our system will be more secure, maintainable, flexible, and extendable. Below is the the subsystem decomposition diagram of our system(Figure 2):

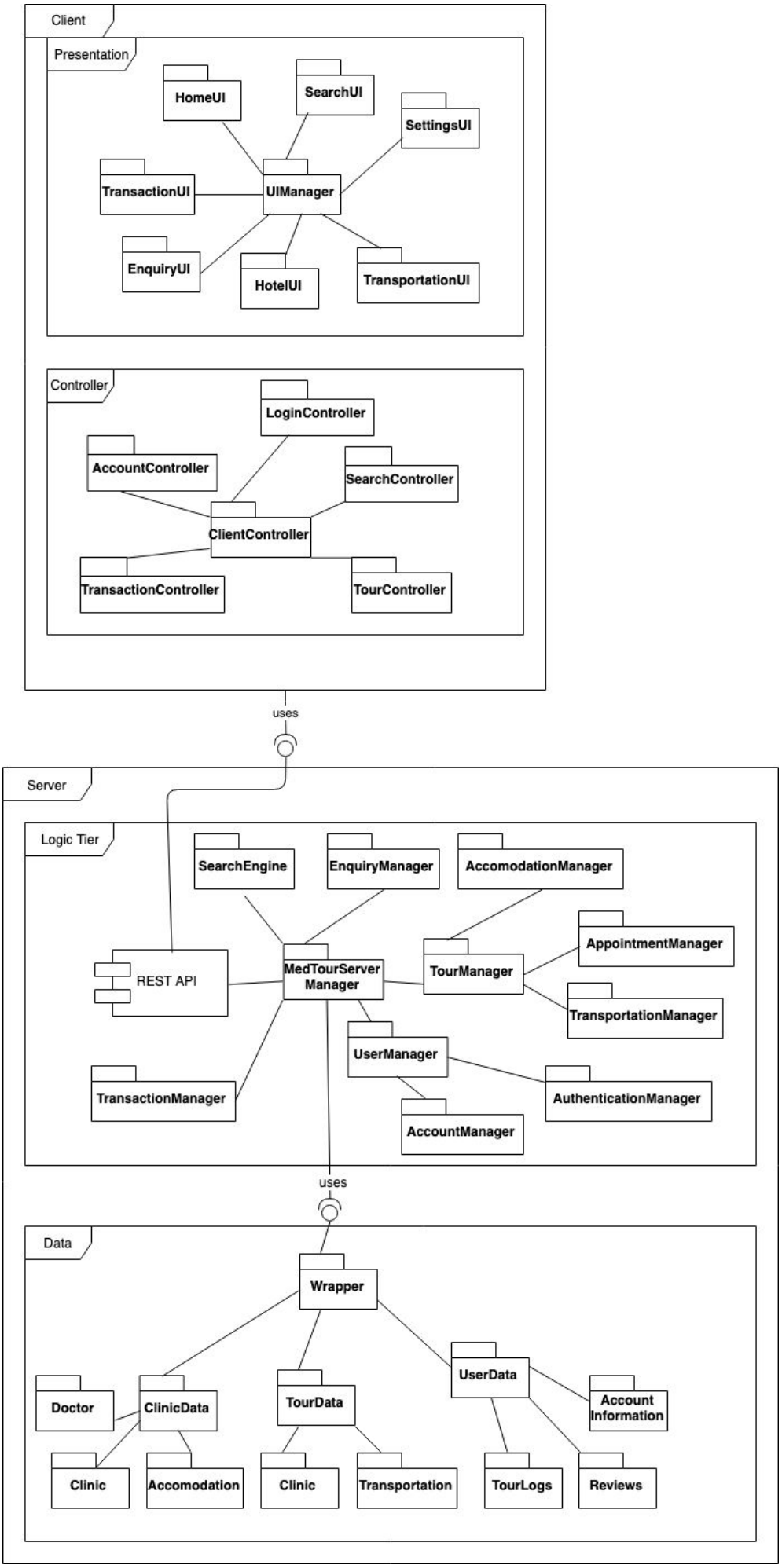
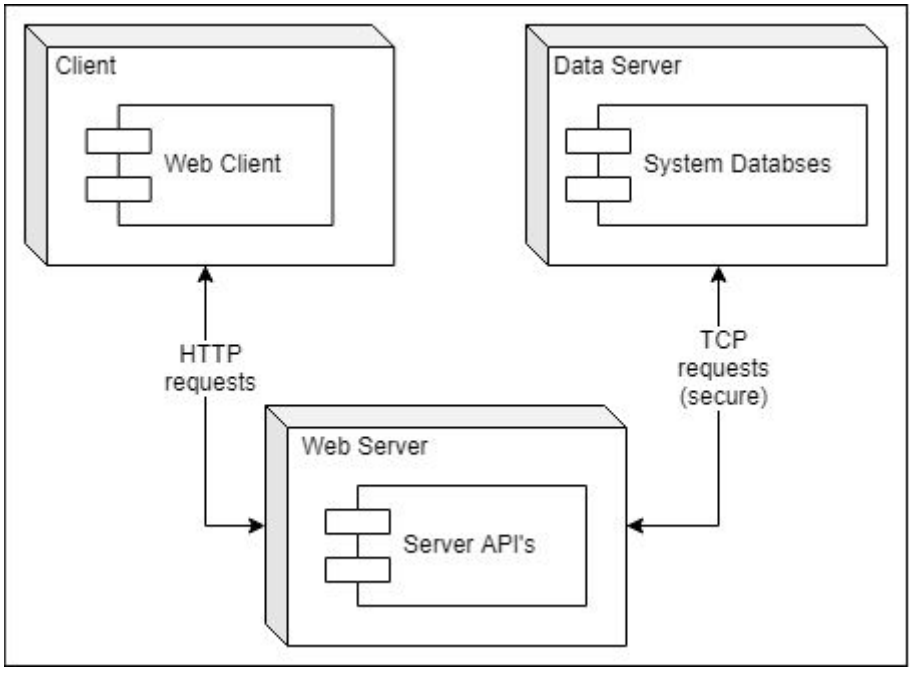

Client
Presentation
HomeUI
SearchUI
SettingsUI
TransactionUI
UIManager
EnquiryUI
HotelUI
TransportationUI
Controller
LoginController
AccountController
SearchController
ClientController
TransactionController
TourController
uses
Server
Logic Tier
SearchEngine
EnquiryManager
AccomodationManager
REST API
MedTourServer Manager
TourManager
AppointmentManager
TransportationManager
UserManager
TransactionManager
AuthenticationManager
AccountManager
uses
Data
Wrapper
Doctor
ClinicData
TourData
UserData
Account Information
Clinic
Accomodation
Clinic
Transportation
TourLogs
Reviews

**Figure 2.** Subsystem decomposition of client-server architecture

### 3.3. Hardware/Software Mapping




The system will have one client type, namely a Web Client. This will allow any user to easily access our system without any prior installations. As shown in the picture above, The client will request data from the Data Server that will keep every information needed by the system. By using several API's, the client will communicate with the server in a synchronous way achieved by utilizing HTTP requests and asynchronously by using Async Event Handlers. This will allow the user to have a seemingly uninterrupted experience. All the fetched information will be filtered on server-side as it will have the responsibility for all data related actions. As these data will be processed on server, the memory impact on the client will be minimal.

### 3.4. Persistent Data Management

In our system, having a stable and consistent database is a must. Since the system will hold a very large collection of data, they should all be persistent throughout the lifespan of our system. Information such as usernames, passwords

(encrypted and secured), phone numbers, emails and many more will be saved in our databases for each user. As the user uses our system, unique data will start to populate their account. Things such as the number of visits made, the clinics visited, services requested, reservations made will be some of the specific information saved for each user. Other information, crucial for the our system like Clinics, their services, their doctors, contact information and more will be saved in our databases to provide the basic functionalities of MedTour. All this information should be consistent and persistent to ensure a stable and trustworthy system. To achieve this, all data will be handled very carefully and frequent backups will be made to ensure a stable database even in the worst case.

### 3.5. Access Control and Security

Every user in MedTour system will be able to have full control over his/her personal information. Users will be able to fully edit these data according to their own preferences. Sensitive information such as passwords, will be firstly encrypted than saved in our database to ensure maximum security. Seeing these credentials even by MedTour staff will be strictly forbidden. MedTour will require some basic information upon account creation. A username, a password and a valid email address will be required in order to register in our system. The email should be valid and it can be associated to only one account. This, to limit the number of accounts one user can create in our system. This email will be used for all enquiries, one of them being the process of resetting a forgotten password. User can also sign up and use the system by logging in using Google Accounts or Facebook. In such a case, permission to use the personal information will be requested upon registration.

### 3.6. Global Software Control

The main controller of our system will be the server. Our application will heavily rely on server-side actions, for primarily two reasons. One being the increased security and the second being the flexibility and scalability that it offers. The server will provide results based on user request, and by using the proper API these data will promptly be displayed to the user. One of the cases may be when the

user uploads photos to our system. In order to check for inappropriate photos, Google Vision API will be used.Google Maps API will be used for all location related requests, and Google Firebase API will be used to handle all data, and maybe two step authentication for a better and modern security solution. Combining these and all other server-side events, the appropriate information will be displayed to the user upon all requests.

### 3.7. Boundary Conditions

As all systems, MedTour has its own boundary conditions. Starting the client, terminating the client, and handling all failures. These boundaries will be explained in detail on the next subsections.

#### 3.7.1 Initialization

The main way to initialize our application will be by using any browser such as: Google Chrome, FireFox, Safari etc. Users that are already registered or register will be able to unlock every feature of our system. The users should log in to fully utilize MedTour. If login fails, the user will be redirected to re-enter the login information, otherwise the welcoming home-page will be displayed. As it is an online system, an active internet connection is required.

#### 3.7.2 Termination

A logout button will be present for all users wishing to log out of the system. Without pressing this button, the user will remain logged in as cookies will be used to save accounts and keep users logged in, to not request login every time. Other way to terminate, should be to delete all cache and cookies of our site. This will reset everything and log-out any logged in users.

#### 3.7.3 Failures

The system will not respond/work if there is no active internet connection. Since data should be passed back and forth to the server, without such connection the system will not be able to work at all. If there is no connection, only preloaded page snapshots will be visible.

## 4. Subsystem Services

In this section, we will discuss our services and how they take place in our system. As mentioned at section 3.2, we use client-server architecture to build our application; presentation, logic, data.

### 4.1 Server-side Services

#### 4.1.1 Logic Tier Subsystem

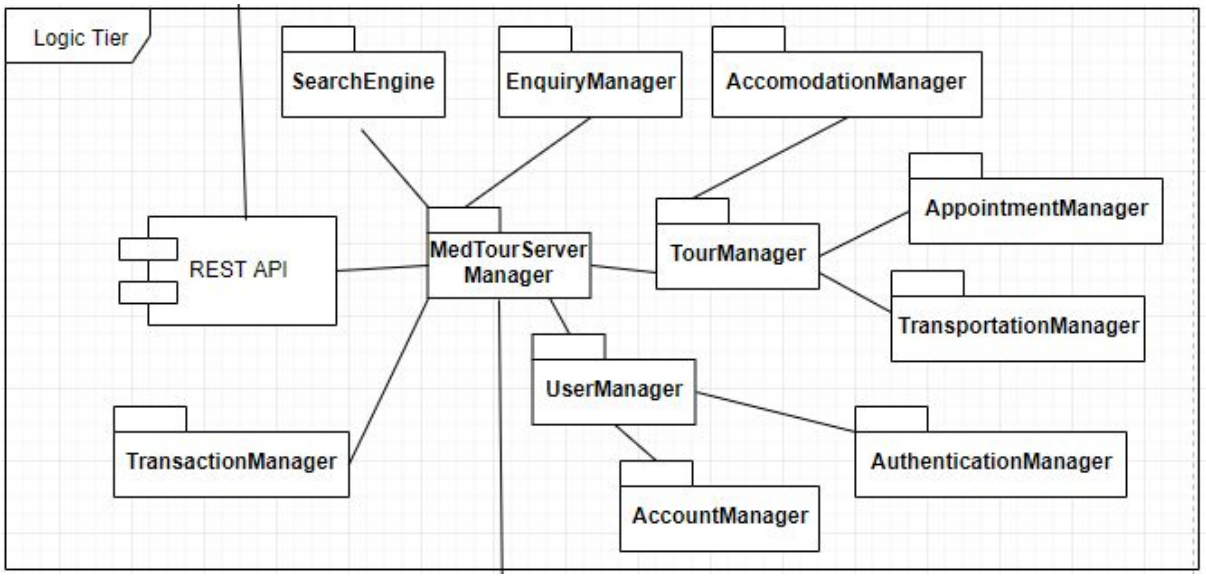



**Figure 3.** Logic Tier Subsystem

Controllers and services exist in Logic tier are responsible for handling the logic behind the REST calls. Main objective of this subsystem is to manage the API endpoints that has been requested from the presentation layer. This tier takes responsibility to provide the connection between the presentation and the data layer. To achieve this, there are some core components that has been used by logic tier subsystem: REST API and “MedTourServerManager”(Server Manager). API endpoints will be requested by POST, GET and PUT methods in JSON formatted payloads.

Server Manager will be responsible to handle the requests that come from the patients; such as listing the dentist clinics in Ankara. In that case, client side will be have to send an HTTP requests with treatment type specified as dentist and city

name Ankara; so that server logic tier will request a query from the database which contain only filtered results.

Logic tier also acts as a secondary security point for MedTour server system; any GET/PUT/POST request arriving to Logic Tier via. REST API is going through a validity check.

### 4.1.2 Data Tier Subsystem

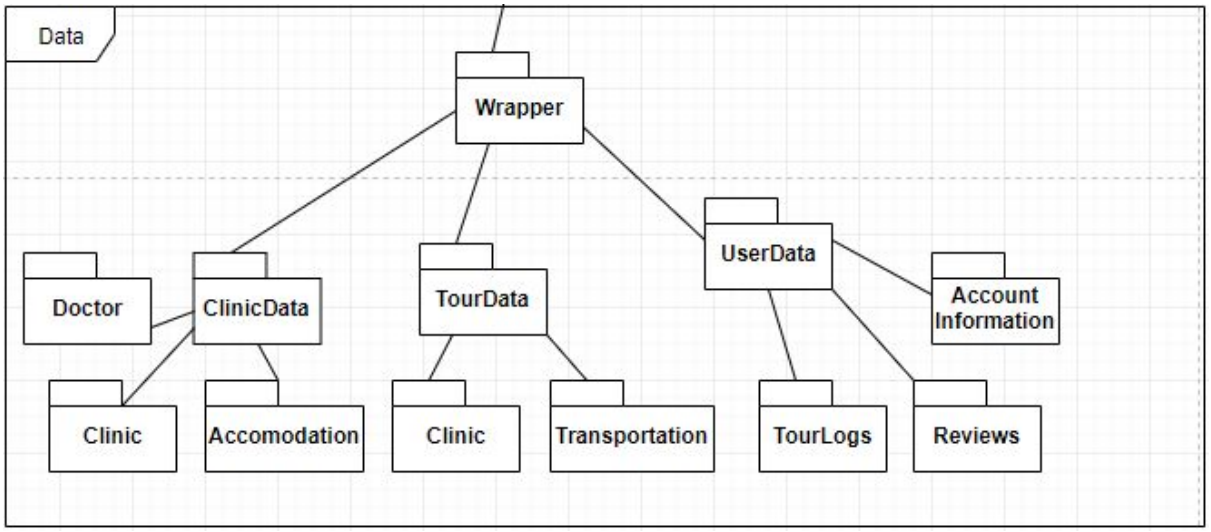



**Figure 4.** Data Tier Subsystem

Data Tier Subsystem is responsible for data management system that provides access to application data. The data tier services are protected from direct access by the Presentation Tier components residing within a secure network. Every interaction in data tier must occur through Logic Tier. Wrapper component in data tier responsible for abstracting out the database operations which cover every functionality that is needed by Logic subsystem.

## 4.2 Client-side Services

### 4.2.1 Presentation Tier Subsystem

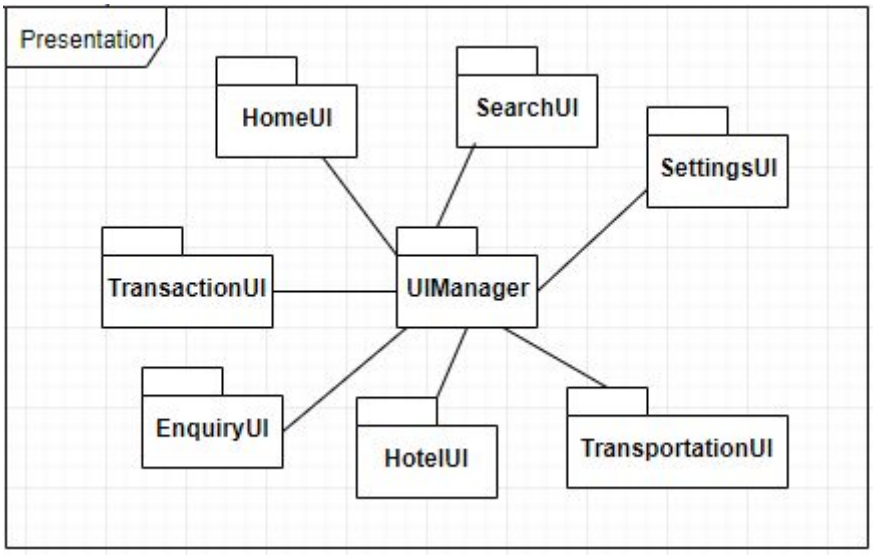



**Figure 5.** Presentation Tier Subsystem

Presentation Tier Subsystem is responsible for providing user interface and listening events that needs to be send to Logic Tier Subsystem. Meaning, presentation tier is the front end subsystem and it consists of the user interface. The user interface is accessible through a web-browser or web application and it displays content and information to an end user. The content displayed enables user to interact with the Logic Tier within a secure and intuitive manner. Being specific, this layer will be built on web technologies such as HTML5, CSS, JavaScript framework React.js.

## 5. Conclusion

Technical informations about *MedTour* is explained in the report. With respect to the high-level design of a software project, specific design goals are provided which are very important to deliver a complete software fulfilling the requirements at the end of implementation. Software architecture is the main important aspect of this report since it is the most detailed and explanatory part about *MedTour*. Subsystem decomposition, hardware/software mapping, persistent data management, access control & security, global software control and boundary conditions are the main aspects of architectural details of *MedTour*. The architectural styles chosen are going to be followed through the implementation process. Therefore, well-calculated decisions have been taken. Subsystem services are also provided to extend the architectural decisions.

## 6. Glossary

**HTML:**Hypertext Markup Language, a standardized system for tagging text files to achieve font, colour, graphic, and hyperlink effects on World Wide Web pages.[3]

**CSS:**Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.[5]

**ReactJS:**React is a tool for building UI components.[4]

**RESTAPI:** A REST API defines a set of functions which developers can perform requests and receive responses via HTTP protocol such as GET and POST.[6]

**JSON**: JSON stands for JavaScript Object Notation. JSON is a lightweight format for storing and transporting data. JSON is often used when data is sent from a server to a web page. JSON is "self-describing" and easy to understand[7]

**HTTP:** HTTP stands for Hyper Text Transfer Protocol.WWW is about communication between web clients and servers. Communication between client computers and web servers is done by sending HTTP Requests and receiving HTTP Responses.[8]

**MYSQL:**MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.[9]

## 7. References


[1] "700,000 medical tourists visited Turkey in 2017 - Latest News," *Hürriyet Daily News*. [Online]. Available: http://www.hurriyetdailynews.com/700-000-medical-tourists-visited-turkey-in-2017-134942.

[2] "About Us - Company,"*WhatClinic.* [Online]. Available: https://www.whatclinic.com/about/

[3] "HTML". [Online]. Available: https://en.wikipedia.org/wiki/HTML

[4] "What is React?". [Online]. Available: https://www.w3schools.com/whatis/whatis_react.asp

[5] "Cascading Style Sheets". [Online]. Available: https://en.wikipedia.org/wiki/Cascading_Style_Sheets

[6] “Do you know what a RESTAPI is?”. [Online] Available: https://www.sitepoint.com/developers-rest-api/

[7] “What is JSON?”. [Online] Available: https://www.w3schools.com/whatis/whatis_json.asp

[8] “What is HTTP?”. [Online] Available: https://www.w3schools.com/whatis/whatis_http.asp

[9] “MySQL”. [Online] Available: https://en.wikipedia.org/wiki/MySQL