TED UNIVERSITY

CMPE 491
Senior Project
2022-2023 Fall

High Level Design Report

RESERVE IT

Authors:
Abdusselam Koç
Bahadır ÜNAL
Enes Yardım
Oğuz Öztürk

Supervisor: Dr. Emin Kuğu

Table of Contents

1 Introduction

With the beginning of the millennium, the world started to turn faster with a face paced work environment and technological development. People started to need more time to be able to meet working requirements. This business brings a need for good time management. According to most managers in big companies, time management is the most important part of the work because it helps you reduce stress and prioritize your time. Therefore, the need for a business focused time planning management application for meeting reservations increases day by day. The old meeting reservation software cannot meet the demands of the modern business. The purpose of this project is to create a contemporary product that enhances the meeting reservation ways. While finding out a new solution to the problems of finding an appointment at a suitable time in the comfort of our home, protection of private user information from malicious users is also vital for professionality and ethicality of the software. In this document we are going to provide the current project description with the additional depth needed to describe an appropriate coding model, define proposed system architecture and system models. This report can be used as a reference guide for how the modules interact at a high level and is also meant to aid in identifying conflicts before coding.

1.1 Purpose of the system

The meeting reservation applications used in the past cannot satisfy the new business requirements. The aim of this project is to develop a software that can extend the facilities of meeting reservation applications beyond the current bare situation. We believe that this project will meet strategic, operational, and administrative requirements of business. Also finding out a new solution to the problems of finding an appointment at a suitable time in the comfort of our home by using mobile solutions which are popular and convenient nowadays for users.

Turkey has a marginal socio-economic structure. Football pitches are used by Turkish men of every age. Most of these pitches are used to play football regularly. Every time they want to play football on pitch, they need to research nearby pitches first, then they look for availability of these pitches if they are not available and don't fit with the customer's schedule, this cycle must return to the beginning and people look for new pitches. Also, all these stages are made by phone call without any visual process. Every time in this cycle people need to contact another person to make arrangements. Our system is going to exist to prevent this time loss and dealing with the human relationship as a social constraint.

Turkey has hairdressers for women and men of every age. Especially in the big cities in Turkey, with the effect of the pandemic most of the hairdressers work with a reservation. They do that with a notebook and phone calls. They store the reservation hours and customer info in a notebook. This situation decreases efficiency and causes time loss. And interactivity between customer and hairdresser reduces to the minimum level. Customers must call hairdressers in every unexpected situation. This is the same process as at the beginning of 2000. There has been no progress since that time. Lastly, the individual reservation system will act like an agenda for people who need time management such as businessmen or teachers. The system will provide them to give appointments to the students or other partners for a real job meeting.

Nowadays, the world is changing, mobile apps reduce the interaction between people and most of the people like it. Even in some of the apps giving tip operation is handled online and it makes people comfortable. The ecosystem which will be developed by us is going to create great communication between buyer and seller. It is going to decrease one-to-one communication. In this way, social stress, talking to another person and asking for something, is going to be reduced.

When we look at the project from a bird's eye view, users can discover the nearby suppliers' providers, their available times, price ranges and the services they offer. They can make reservations by pre-paying some amount on the e-payment system. Also, users are able to comment and evaluate their past deals. We are planning to develop this project for hairdressers, football pitch industry and individuals which will benefit both service seller and buyer.

1.2 Design goals

The architecture, user interface design, external interface, database, process relation, and automation are the five key design components. These designs have been explained in the attachments to make them easier to understand.

1.2.1 Security

Users' passwords will be hashed in the database; plain text passwords won't be kept there. It is important to think about how securely the hardware and software are connected.

1.2.2 Maintainability

This system should require very little maintenance. After the system is set together, the only involvement needed is the first configuration. Any modifications to settings made after initial setup and any specific unusual situations when user settings or history need to be updated would be the only other user maintenance. It may be necessary to do physical maintenance on the system's components, which would cause a brief loss of data or Internet. While downtime may arise, upgrades to the hardware and software should have little impact on this project.

1.2.3 Usability

The system is based on mobile actions using required servers. Mobile applications are required to be user-friendly and simple to comprehend. The program ought to be simple enough for the average user to utilize in a brief period of time.

1.2.4 Performance

Performance is very important for this application. For everything to run smoothly for this project, the information and status will have to be able to update data on the database and refresh every required time. Pages will need to be served to users at a reasonable speed. The database server will need to keep up with all database requests and operations.

1.2.5 Availability

The System requires a high availability, due to being an online platform interacting with other users and providers. Users shall be able to reach the servers anytime. When problems do occur, the system shall continue to function because of availability characteristics. Error in one part shall not affect other parts of the system.

1.2.6 Reliability

The implementation of a redundant database server will allow the gateways to switch over automatically to the backup server in the event that the primary database server becomes unresponsive. The method for synchronizing these two databases is still being worked out. Likely contender Updates to both servers' gateways are one possible remedy.

• The redundant server is constantly checking the tables for fresh data and an archive field is being implemented.

1.2.7 Error Handling

If mistakes are made, a description of what went wrong will be shown. Anything that deviates from the typical and intended usage will be regarded as an error.

1.3 Definitions, acronyms, and abbreviations

Provider: Service supplier that can serve customers (users) specific types of labour. For instance, hairdresser, football pitch owner etc.

Provider Category: A directory containing a specific type of service providers. Users can seek various amounts of location, service and price information to find best fit for their budget.

Individual: Person that organize some sort of meeting for some group of people and while applying this procedure, person utilizes Reserve-It app features.

Client: The mobile or web application interacting with the user.

Server: The system communicating with each client and processing images sent from them.

API: Stands for “application programming interface.” APIs allow one application to communicate with another to allow for the transfer of stored data or instructions.

IDE: Software application that provides comprehensive facilities to computer programmers for software development.

Mobile Client: Stands for the user interface of our Flutter application.

Cloud Functions: It is used for responding to the user's activities and making real time updates in the application.

1.4 Overview

Reserve- It is an android, IOS, and web application, targeted for mobile phones and web usage for partners. partners will be able to reach detailed usage of the application with web and normal usages with phones. customers will only be able to use it on their phones since the web application functionality is not required for the customers.

The application aims to provide a manageable reservation system both for customers and partners. The customer will be able to make reservations from any partner in the application according to partners availability. The application consists of a mobile application that targets android and IOS operating system, a web application that targets every operating system that has a browser, a back-end server that is responsible for storing and organizing data, and ensuring everything on the client-side actually works. The backend communicates with the frontend, sending and receiving information to be displayed as a web page or mobile page. a database which is set up for easy access, management and updating information.

This document will include present all of the design aspects and define them in detail

- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project

• List and describe the non-functional attributes like:

  o Security

  o Reliability

  o Maintainability

  o Portability

  o Reusability

  o Application compatibility

  o Resource utilization

  o Serviceability

This report also contains information about engineering and ethical considerations. Thus, it will help clarify the system and workload of plan.

2 Proposed software architecture

2.1 Overview

In this section, the system's software architecture and subsystem decomposition are explained in detail. We have many design goals, and to achieve these goals, we need a software architecture that is eligible for our design goals. So, we choose to separate the project's architecture into different layers, and it is based on the communication of these layers. We choose to build our project on the communication of Client, Azure, and External API. Dividing the project into smaller parts helps us to implement our design goals more efficiently. Moreover, this section explains the hardware/software mapping of the project. In the parts you would find below, certain high-level aspects of the application or explained in detail. Some diagrams are given below to help better the understanding of our hierarchy and our design goals. The devices that will enable us to run our platform are described in the subsystem decomposition and the hardware/software mapping parts. The below parts additionally describe the boundary conditions of our application, the termination operations and possible points of failure. We have also explained the database connections, which systems components will communicate with which and how to make those connections and communications patterns more secure and safe.

2.2 Subsystem decomposition

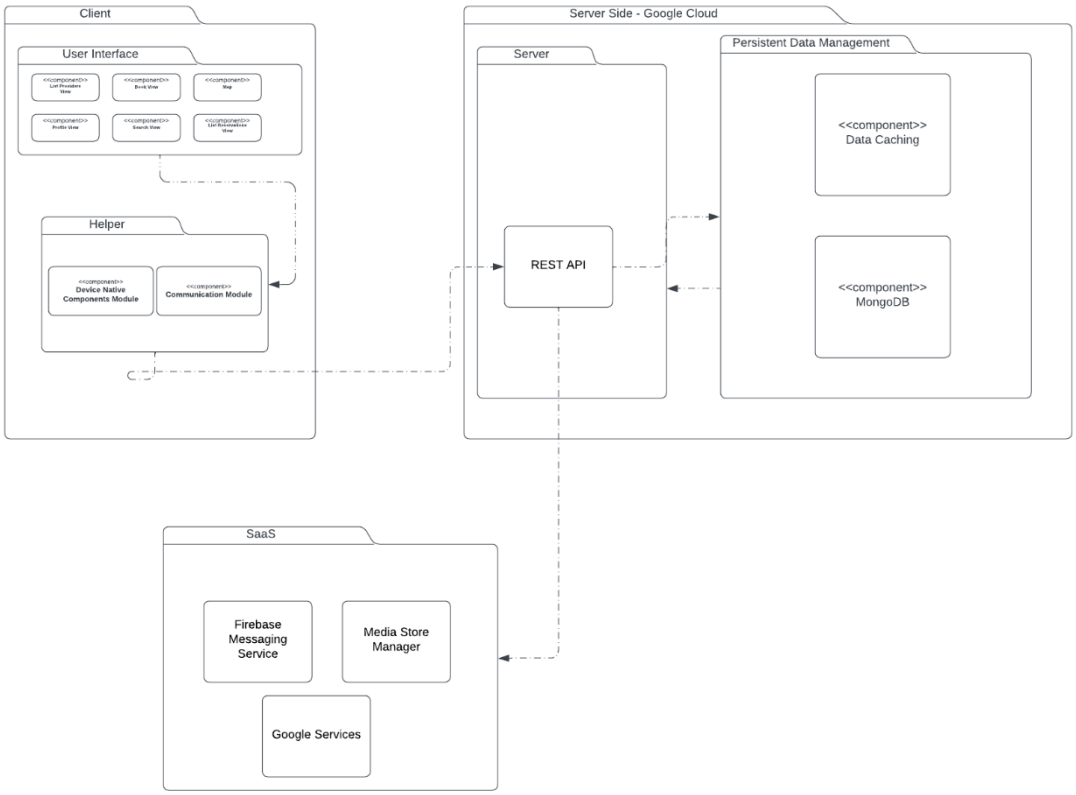

Figure 1: Subsystem Decomposition of Reserve-It

We have decided to build our project on the communication of Client, Azure and External API. The Client part consists of User Interface and Helper. User Interface controls which screen to show to the user. Helper is responsible for the communication between User Interface mobile devices and Azure. On the server side, we are using REST API as an entry point. All of our requests are responded to from the API. When there is a need to use external API, our API reaches these external API's. Also, our persistent data management is connected to API and clients cannot access it directly. It can send requests to get information from the database and API can access and return it.

2.3 Hardware/software mapping

Reserve-It is an application that runs on both an Android and IOS device by using Flutter code base. We use Flutter in our implementation to benefit from Flutter's cross-platform abilities. Our project's subsystem will be realized as software and a limited amount of hardware which belongs to persistent data management in user's mobile devices. Software subsystems that are relatable with Flutter application deal with authentication, notification services, location determination, etc. And Web platform part deals with data entrance for provider data's will be implemented by JavaScript (TypeScript), Flutter, MongoDB, and their several frameworks.

The client and the server communicate through a REST (Representational State Transfer) API. HTTP requests will be made by the client depending on user actions and the server will return with an appropriate response to constitute a communication. The client can be run on mobile devices as well as some limited sort of data can be available through a web client for providers. Both clients can configure already made reservations and make new reservations. Simply, the client and server nodes communicate via the HTTP protocol as seen in the figure below. The details of nodes can be found in subsystems.

2.4 Persistent data management

Persistent data includes both the data taken by the user and the information kept within the server. When a user is done making changes and booking a new reservation, the reservation details are saved remotely. Since we need to provide persistent login in our app, we need to preserve some data of users in their mobile devices. For providers "Tax Numbers" information, which is crucial and needs to be protected, we use secure hashing and storage options such as KeyChain and

AES. The profile pictures and images about providers are saved as jpeg/png images. They stay on the server as long as their total size does not exceed the limit provided by the server.

The application session and persistent user data will be stored using JSON type. Model parameters can be stored based on JSON or encrypted data frames.

The server also keeps information about the users, providers and their reservation details in a MongoDB database. This information are used for services such as authentication, authorization and notification purposes. Server-side triggers and some possible cloud functions are also kept within the server side and used when a module that uses them is invoked.

To store reservations, user and provider details such as location, profile etc. authentication data and other data related to users; a database system is used. As our database system, we have chosen NoSQL, which is a small, fast, self-contained, low-cost scalable database methodology.

2.5 Access control and security

In the Reserve-It Application, users must create an account to reach inside of the application. Every user shall provide their email, name, surname, date of birth and number. Although they must determine a unique password that must be at least six digits and one of letters must be capital letter. The providers need a different process to register as admins. In this process the providers will ask for a partner and redirect to an internet page or in app solution which will ask about the name, surname of the owner of the provider, phone number, location, tax number, business name and valid account number.

For email verification, generated OTP code will be sent to the user for authentication. Generally, Authentication and Authorization are high priority features which help the Reservation System to provide a secure and more accurate

authentication system to prevent hackings and viruses in the system. System also provides “confidentiality”.

Importantly, in Authorization of users, JSON Web Token or JWT as its commonly called, which is an internet standard (RFC 7519) for securely transmitting trusted data between each party in a compact manner. The tokens contain claims that are encoded as JSON objects and digitally signed using public key/private key pairs. JWTs can be encrypted and/or signed. While the claims in the encrypted tokens are kept secret from outside parties, the claims in the signed tokens are checked for accuracy. JWT token is divided into 3 parts named header, payload, and signature.

The system preserves user passwords in a secure manner using a hashing/signing algorithm like “Bcrypt”, so the passwords are not preserved as plaintext. REST API endpoints deliberately require authentication and authorization. The token is obtained by request and client related API endpoints by authorization token.

2.6 Global software control

Reserve-It built on client server architecture. The server handles the requests and uses the database when needed. Reserve-It also uses an external API for creating more detailed user profiles. This external API is also accessed on the server. Reserve-Itis used NoSQL for all the required information. This database is based on the cloud service just like the server itself. For other storage needs like location files or profile photos, FCM is used. This project tends to have lots of users. Therefore, we need to ensure that our system can handle huge traffic. At that point JavaScript is helping us to work asynchronously. However, for the process that needs big computational needs, we are using message queues and worker machines. This method will ensure a balanced workload on our servers.

Azure's auto scaling service can also help to maintain balanced workload and keep the traffic sustainable.

2.7 Boundary conditions

2.7.1 Initialization

The users of the Reserve-It application must install the application from either via the Play Store or the App Store. The providers also can reach the application with a browser such as Mozilla Firefox, Google chrome or Microsoft Edge. A stable internet condition is required as all the information the app will display will be transmitted from the internet via http protocols. GPS location data is required to use and interact with certain functionalities of the app. The user must register to the application in order to use the app.

After clicking the website, the user may see a clear interface. The Providers have to login the application in order to reach the system. All the initializations should be initialized without failure. Moreover, the user should give permissions to the applications for initializing the required adjustments.

2.7.2 Termination

The app can be closed at any time without the need of any additional action, however for security of use, the user will not remain logged in to the application. In authorization, if the user enters their information wrongly, then the system will prompt an error to the user. The users have to login the application after closing the application. All the actions users take while using the app are simultaneously transmitted to the backend server and the database, so exiting the app at any time does not result in loss of data provided there is a stable internet connection.

2.7.3 Failure

The lack of a stable internet connection is the largest failure point of the application as any action taken by the user with the lack of an internet connection will be lost and could not be accessible or visible to any user. Other than the lack of an internet connection, the app shouldn’t fail overall but certain functionalities may be unavailable to the user. To give an example, if the connection fails between the server and the user's mobile device or computer, their actions could not be logged on to the app, or the loss of connection to the gps data might result in incomplete or non-present route data. Another possible point-of-failure might be the unavailability of third party-API’s. These availabilities will not crash the app but might result in the unavailability of certain features.

3 Subsystem services

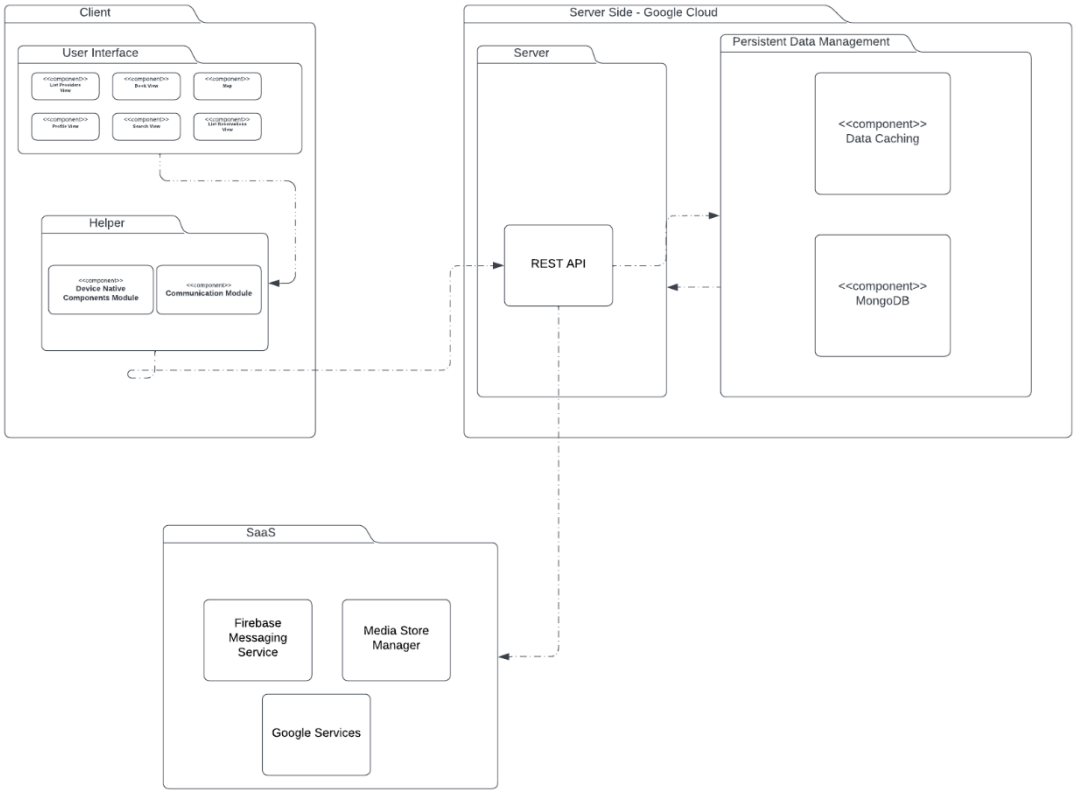

Figure 2: Subsystem Decomposition of Reserve-It

3.1 Client

The first package of the Reserve-It is Client. It contains 2 sub-packages: User Interface and Helper.

User Interface

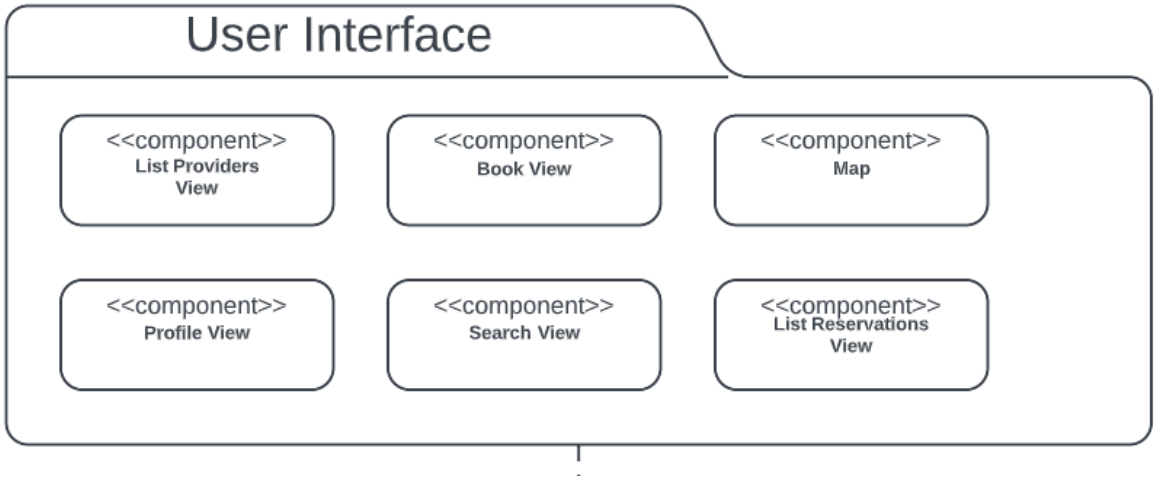

Figure 3: The hierarchy of modules inside the client system.

User Interface is the first sub-package of the Client.

User Interface component contains the screens and other required components for the User Interface for users. Its contents are as follows:

● Map View: This view contains the Map where Users can see nearby providers.

● Book View: This view contains the Booking operations.

● Profile View: This view contains the user profile.

● List providers View: This view contains the list of providers that provide services.

● Search View: This view is responsible for the search operations that are made by users.

● List Reservation View: This view contains the list of reservations according to filters.

Helper:

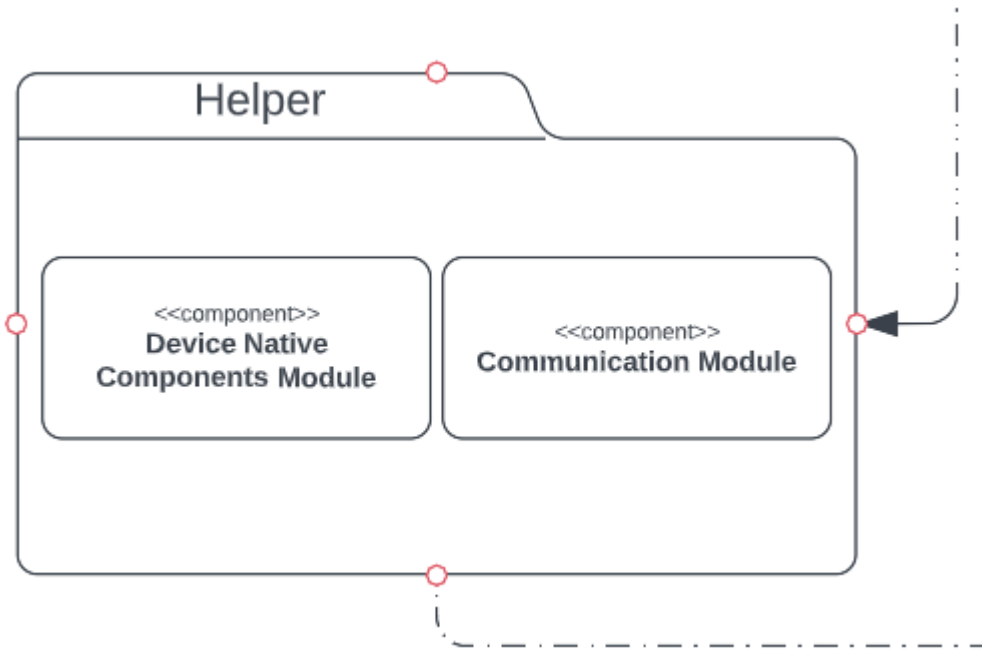

Figure 4: Helper Module of Reserve-It

Helper is the second sub-package of the Client. It contains two components: Device Native Components Module and Communication Module.

● Device Native Components Module: This component is responsible for the usage of devices native components such as camera, storage and Calendar.

● Communication Module: This component is responsible for the retrieval, transmitting and manipulation of user data. It contains classes for parsing JSON data, reaching endpoints and API communication rules.

Server-Side:

The second package of the Reserve-It is Server-Side. It contains 2 sub-packages: Server and Persistent Data Management.

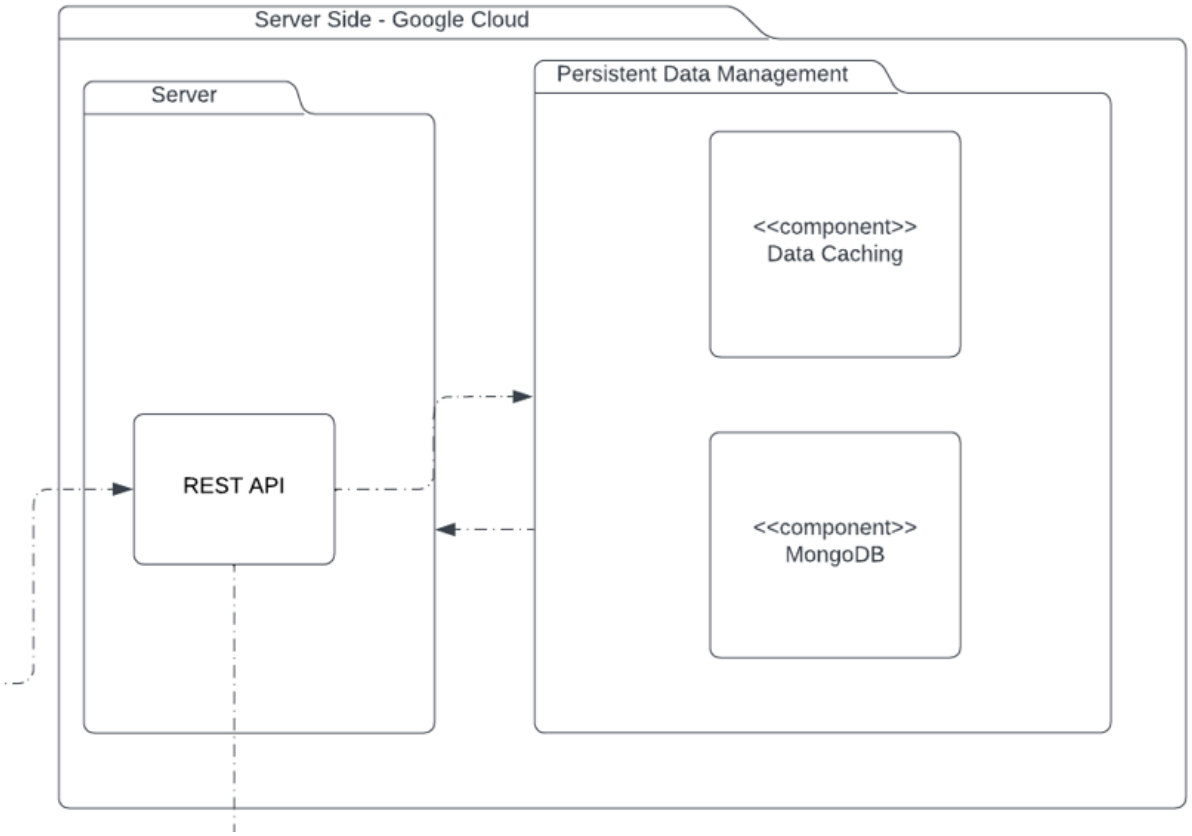

Figure 5: AWS Sub-Package of Reserve-It

● REST API: This component is used the transmit data to the application through various endpoints.

● Data Caching: This container is a persistent storage manager that is being used to store local data in an encrypted format.

● MongoDB: A database that is being used to store user and post information.

SaaS

The third package of the Reserve-It is External API. It contains two components: Firebase Messaging service, Media Store manager.

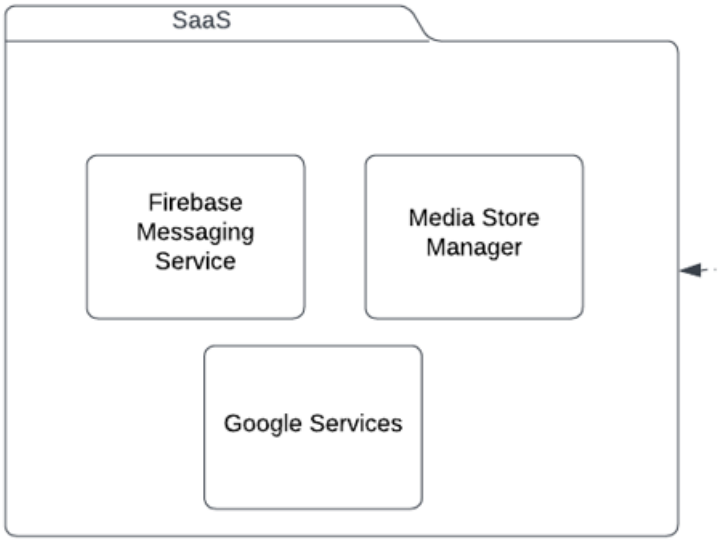

Figure 6: External API Sub-package of Reserve-It

● Firebase Messaging service: This service is a cloud messaging service that has been provided by google, It is used as messaging service at the application.

● Media Store manager: This service is for storing user’s and providers image-based data in the format of JPEG or PNG.

● Google Services: this service is used for many services that are related to google such as Google maps and google authentication.

4 Glossary

The System should be integrable in terms of database variations. A company who demands to use HRMS in a different database, the system should rapidly accommodate the change of database requirements in order to ensure maintainability of our software.

Application Programming Interface (API): is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for representational state transfer and was created by computer scientist Roy Fielding.

Extensible Markup Language (XML): A markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable in accordance with the World Wide Web Consortium (W3C) specification.

React-JS: React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “components”.

Flutter: Flutter is Google's open-source technology for creating mobile, desktop, and web apps with a single codebase. Unlike other popular solutions, Flutter is not a framework or library; it's a complete SDK – software development kit

Firebase Cloud Messaging: Firebase Cloud Messaging (FCM) provides a reliable and battery-efficient connection between your server and devices that allows you to deliver and receive messages and notifications on iOS, Android, and the web at no cost.

Hypertext Markup Language (HTML): The standard markup language for creating web pages and web applications in accordance with ISO/IEC 15445 and W3C HTML5.

CSS: CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language.

JavaScript: JavaScript is a scripting language that enables you to dynamically update content, control multimedia, animate images, and pretty much everything else.

Hypertext Transfer Protocol Secure (HTTPS): A communications protocol for secure communication over a computer network using Hypertext Transfer Protocol (HTTP) within a connection encrypted by Transport Layer Security (TLS), or its predecessor, Secure Sockets Layer (SSL).

JavaScript Object Notation (JSON): An open-standard file format that uses human readable text to transmit data objects consisting of attribute-value pairs and array data types.

References

[1]”Reliability, Availability and Serviceability” https://en.wikipedia.org/wiki/Reliability,_availability_and_serviceability#:~:text=Availability%20means%20the%20probability%20that,hours%20of%20downtime%20per%20year.

[Accessed Dec 21, 2022].

[2] ” JSON Web Token Introduction”.

https://jwt.io/introduction

[Accessed Dec 21, 2022].

[3] ” Introduction to Node.js”.

https://nodejs.dev/learn

[Accessed Dec 21, 2022].