Bilkent University

Department of Computer Engineering

Senior Design Project

whotello: time to interact with your hotel

High Level Design Report

Alina Zhumasheva, Bayram Muradov, Imran Hajiyev

Supervisor: Uğur Doğrusöz
Jury Members: H. Altay Güvenir, İbrahim Körpeoğlu

High Level Design Report
May 17, 2019

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

1 Introduction

1.1 Purpose of the System

The number of hotel users around the globe is quite large, surpassing the benchmark of 15.5 million rooms available for active usage scattered around the globe [1]. However, despite the fact that hotel service exists from 705, even nowadays using hotel services sometimes becomes challenging for both the guests and the administrators [2]. Reserving restaurants, chaise longues, requesting the cleaning or food, learning answers to frequently asked questions and even managing the room temperature for guests; managing the personnel, user preferences, food habits, overall hotel statistics for administrators may present itself a bit difficult due to the human factor in all of the processes and all of them may be optimized for the both groups benefits.

Our system is designed in a way to tackle and solve all of those problems, making the guest experience at the hotel as effortless and pleasant as possible and at the same time easing and simplifying the managing process for the administrators, giving the latter undesired benefits such as specified statistics and data for the further improvement of the hotel service - and all of this is automated, removing the human factor.

1.2 Design Goals

1.2.1. Usability

Since the main purpose of the project whotello is to make the hotel industry much easier and simpler by helping both Guest and Administration side to do their main tasks in a simplified and comfortable way, it should be easy to use. In order to achieve this, we are planning to design an intuitive user interface and comfortable user experience for both Guest's Mobile Application and Administration's Website. Users should not be mislead by any component of the UI and they should be able to perform the tasks that they want without thinking too much.

1.2.2. Reliability

Since we are proposing data analysis to hotel administration which will result in hotel-running suggestions, and expect them to follow our suggestions - they should be reliable. Which hotel facilities are the most popular among guests, meals in the restaurants and even employees of hotels - are some of the criteria, that suggestions will be given about - and that we are planning to make reliable. We will collect the data and design our algorithms accordingly to create a reliable system.

1.2.3. Security and Privacy

Since data privacy is an important concern, we need to keep both our users secure. We will make sure that payment systems we use are secure and reliable. For the Guest side, all the data generated by their mobile application, will be shared only with the agreement from the guest side. For administration side, we will encode/encrypt their data.

1.3 Definitions, Acronyms, and Abbreviations

- API: Application Programming Interface
- iOS and Android: Mobile operating systems
- HDM: Hotel Database Management

1.4 Overview

We are planning to make whotello a digital ecosystem for the hotel industry. While doing this project, we need to consider the needs of guest of the hotels, travellers and hotel managers, and design the system accordingly. In this report, the high level design of the project is discussed.

2 Current Software Architecture

Nowadays there are a lot of mobile applications that help guests with the hotel of their choice. Those applications serve almost any simple goal a guest would need in their hotel - from ordering any kind of service and FAQ to opening a room lock with their phone. However, most numbers of such applications are hotel itself sponsored applications, that earn their active users some points that later can be redeemed for sales: *Marriott Bonvoy* of Marriott International [3]*, Hyatt Hotels* of Hyatt Corporation [4]*, Hilton Honors* of Hilton [5] - to name but a few. There are also some services for the administration part of the hotel as well, such as *eZee Absolute* [6], which can auto-generate your possible income, detailed business insights and lots more.

However, there is only one software currently that includes both sides of the hotel business, that is both the guest and administration sides, and it is the *ALICE* software. This system is similar to our proposed system in a number of aspects, however, does not tackle the problems that ours does. Here are the short description and features of the *ALICE* software.

2.1. ALICE Platform

Alice platform is a complex system that aims at two main aspects: guest satisfaction and operational excellence. This serves the reason for the following features/products of the platform:

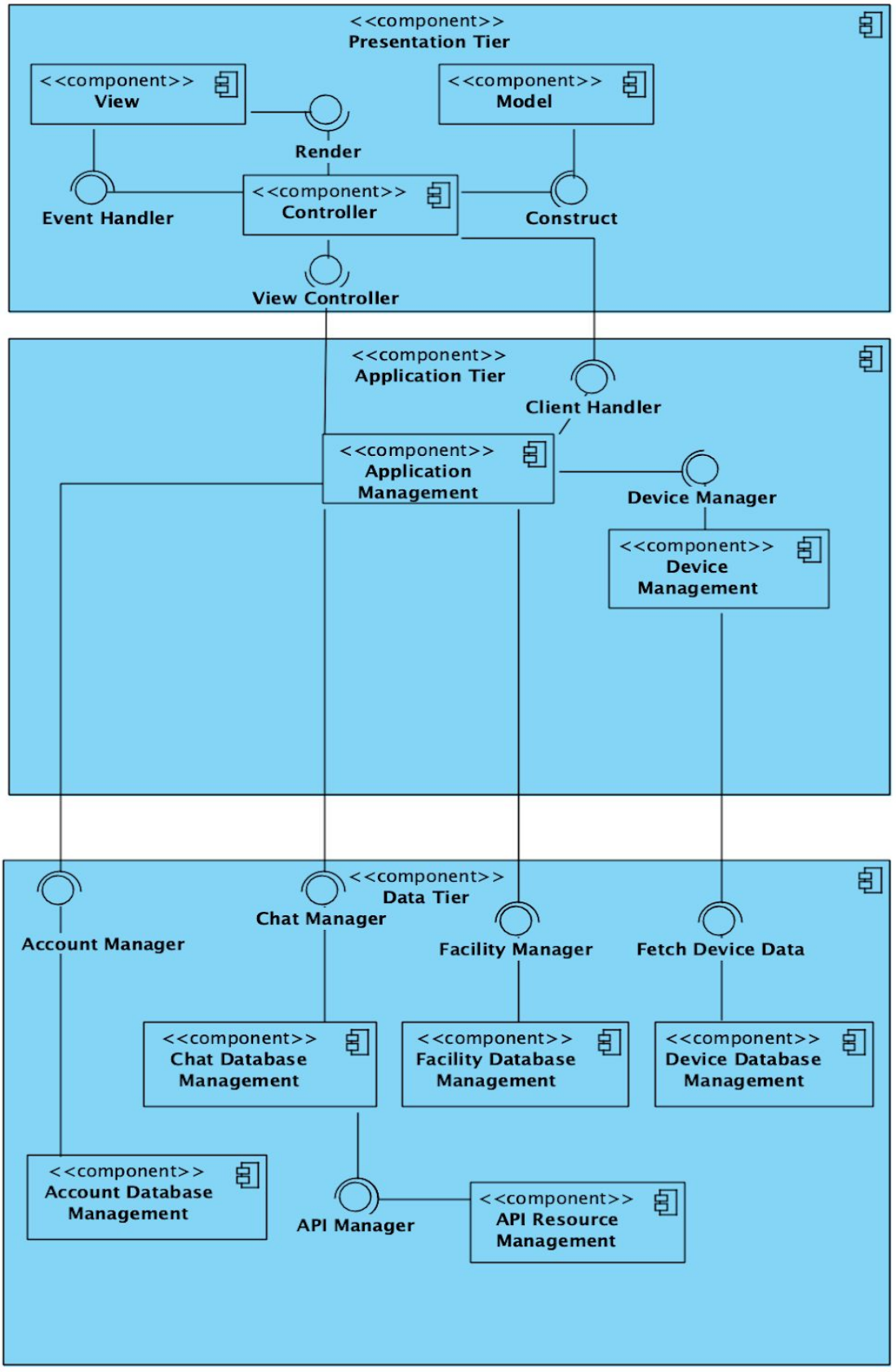
- *Concierge* - provides guests with reservation details, tips, and makes the reservation for them across all the hotel services. Gives guests quick information on places to visit, nearby shops, restaurants, bus and taxi stops outside the hotel.
- *Guest Messaging* - helps with FAQ and other questions or requests that guests may have. Instead of talking to the personnel in real life, messages them.
- *Guest App & Website* - includes both of the features inside a mobile app, which makes it easier to interact with the hotel, without any additional technology to carry around.
- *Logbooks* - tracks every information about the guest's items during his hotel stay: personal items, souvenirs, shopping bags, laptops and etc. Package management and lost and found functionality included.
- *Service Delivery* - optimizes the work for the personnel to minimize the time for completion of their tasks, in order to serve as many guests as possible, so that no guest stays unserviced.
- Preventative Maintenance - maintain and track information on all the equipment inside the hotel, like statistics on what is used more frequently.

3 Proposed Software Architecture

3.1 Overview

In the following sections proposed architecture of the software that aims to create interactive environment for communication of hotel and its guests will be discussed.
Firstly, subsystem structure of our system will be discussed in "Subsystem Decomposition" section. Secondly, details of the tiers will be discussed in "Hardware Software Mapping" section in detail.
Furthermore, persistent data management, boundary conditions of user's access and issues related with the security of the system are also described in the related sections. General flow of the data and control of the software are discussed afterwards. Lastly, discussion about initialization, termination and failure conditions of the system can be found in the final section.

3.2 Subsystem Decomposition



The system is decomposed into 3-tier architecture since the application is client-server based and large number of operations are database intensive. Decomposed system includes 3 different tiers called **Presentation Tier**, **Application Tier** and **Data Tier**. Minimization of coupling and maximization of coherence between mentioned components is the main objective.

Components of the system in detail:

- **Presentation Tier:** Is going to run on client side and display information according to actions of user and responses from Application Tier. Tier itself is decomposed into 3 parts:
  - ❏ *View*- User interface for front-end of the system.
  - ❏ *Model*- Models the data to be viewed.
  - ❏ Controller- renders View via data it gets from Model, handles user inputs.

- **Application Tier:** Is going to run on server side and contain the logic related to the system. Inputs from the user will be processed in this layer by using Client Handler. Furthermore, Application tier creates interaction between Presentation and Data Tiers since there is no direct connection between them. Tier is decomposed into the following parts:
  - ❏ *Application Management*- Handles requests and generates response accordingly. Only component that has permission to change data in Data Tier.
  - ❏ *Device Management*- fetches the devices of the facility from the database.

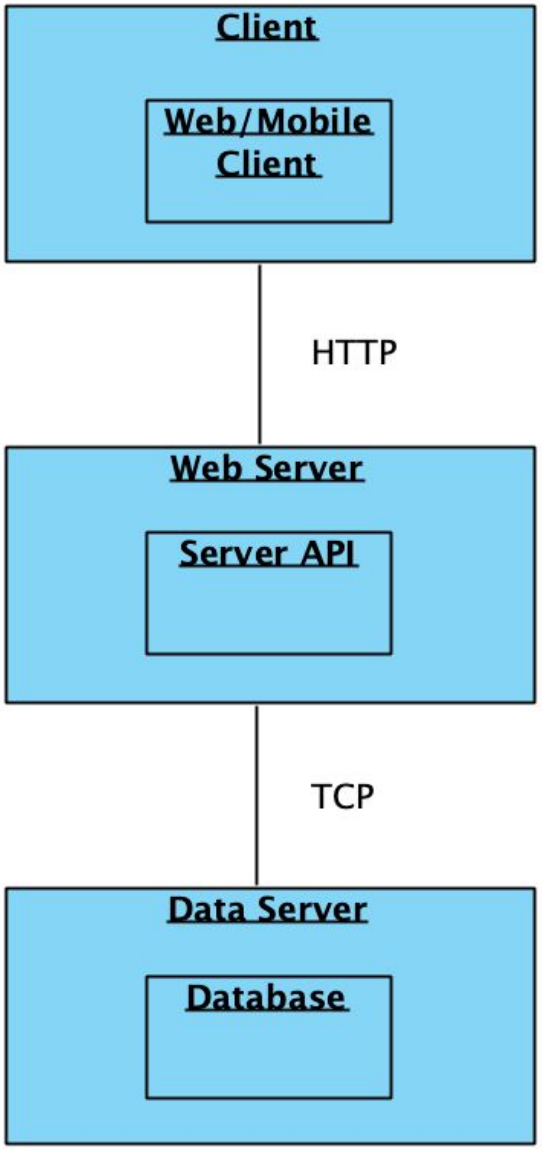
- **Data Tier:** Is going to be deployed on server side. Providing secure and reliable access to the database is the main functionality of this layer.
  Tier is decomposed into following components:
  - ❏ *Account Database Management*- Manages the User database.
  - ❏ *Chat Database Management*- Manages the Chat database.
  - ❏ *Facility Database Management*- Manages the Facility database.
  - ❏ Device Database Management- Manages the Device database.
  - ❏ *API Resource Management*- Makes use of several APIs to provide functionality of chatbot.

3.3 Hardware / Software Mapping

Tiers of the system will create client and server software.

- **Server software** will include logic for data and application tiers to respond requests made by clients. It is going to run on application server.
- **Data Tier logic** will handle the retrieval of data from databases.

Database Tier will be developed by using MySQL. Django will be used to create the application tier.



3.4 Persistent Data Management

User related information including email, password, preferred room settings and etc. will be stored in the system by their permission. Frequently asked questions by a particular user will be analyzed by the system and be cached in order to minimize the number of API calls.

3.5 Access Control and Security

No user information will be stored or shared with third parties without permission. Through testing and evaluation will be done on the system to ensure the security.

3.6 Global Software Control

Each server will follow event-driven control in order to avoid race conditions which can happen when users request same data simultaneously.

3.7 Boundary Conditions

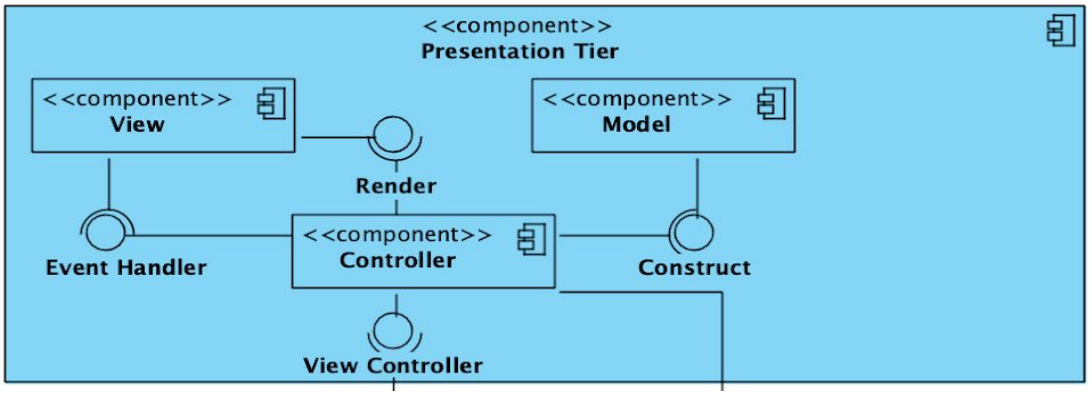
- **Initialization**
  Depending on user type, users may choose to use web or mobile implementation. Both types of users including guest and administrator need to sign up in the system. Application requires internet connection to become fully functional.

- **Termination**
  Both types of applications (web and mobile) can be terminated at any time by the appropriate type of users.

- **Failure**
  In case of any failure that may include: wrong data input by user, slow internet connection, API related issues and etc. system will halt and display appropriate error messages.

4 Subsystem Services

4.1 Presentation Tier Subsystem



Presentation Tier Subsystem is responsible for providing user interface and listening events that needs to be sent to *Application Tier Subsystem*. *Presentation tier* is the front end subsystem and it consists of the user interface. The user interface is accessible through a web-browser and mobile application and it displays content and information to an end user. The displayed content enables user to interact with the *Application Tier* within a secure and intuitive manner. This layer will be built on web technologies such as HTML5, CSS, JavaScript framework React.js. It has 3 main components:

1. Model Component
2. View Component
3. Controller Component

Model Component provides view, control and construction of queries that needs to be send to *Application tier*. The classes in this component are separated from view and controller components in order to reduce the coupling in presentation tier.

View Component provides interface for our view/pages. It contains classes such as MainView, InfoView, ReservationsView, etc. These classes basically construct the user interface of the project. Additionally, the classes of view component consists of listeners which are basically listens user inputs, when they receive an input, they send a notification to Controller by transmitting the event to Event Handler. Controller decides what to with the event which is listened.

Controller Component is basically the manager of the *Presentation tier*. It contains the classes which are responsible for controlling operations that needs to be done in views/models and also handling events that are coming

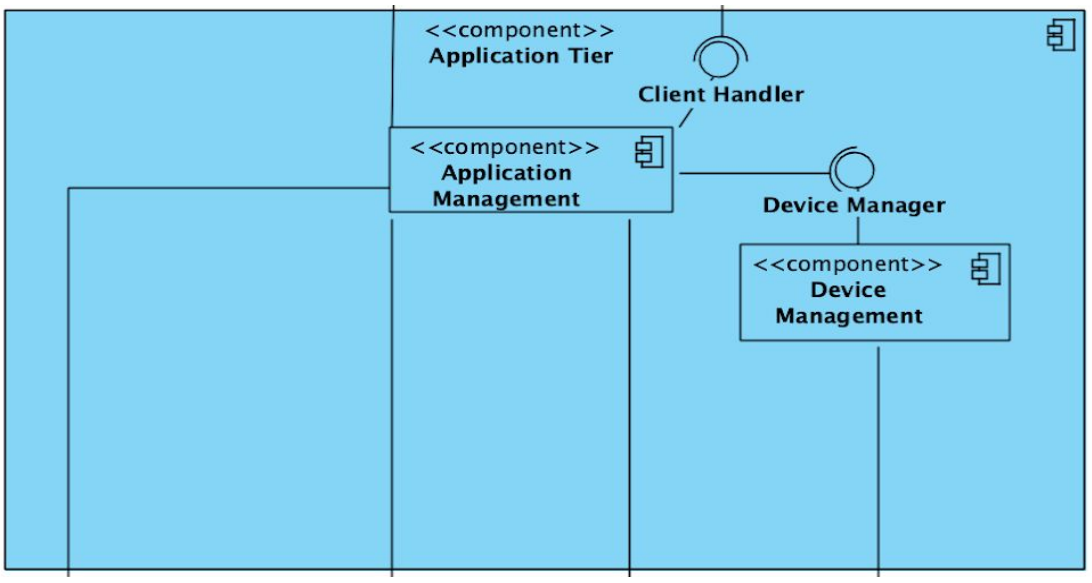
from view components. Controller component acts has a Façade class called ViewController. Because all operations between Application tier and Presentation tier are done through ViewController class. Moreover, in general, the controller has these main responsibilities:

● Modifying views according to events happening in *Presentation Tier* or events coming from *Application Tier*.

● Creates a new query model via Construct method.

● Sending clients side requests coming from views/queries to *Application Tier.*

4.2 Application Tier Subsystem

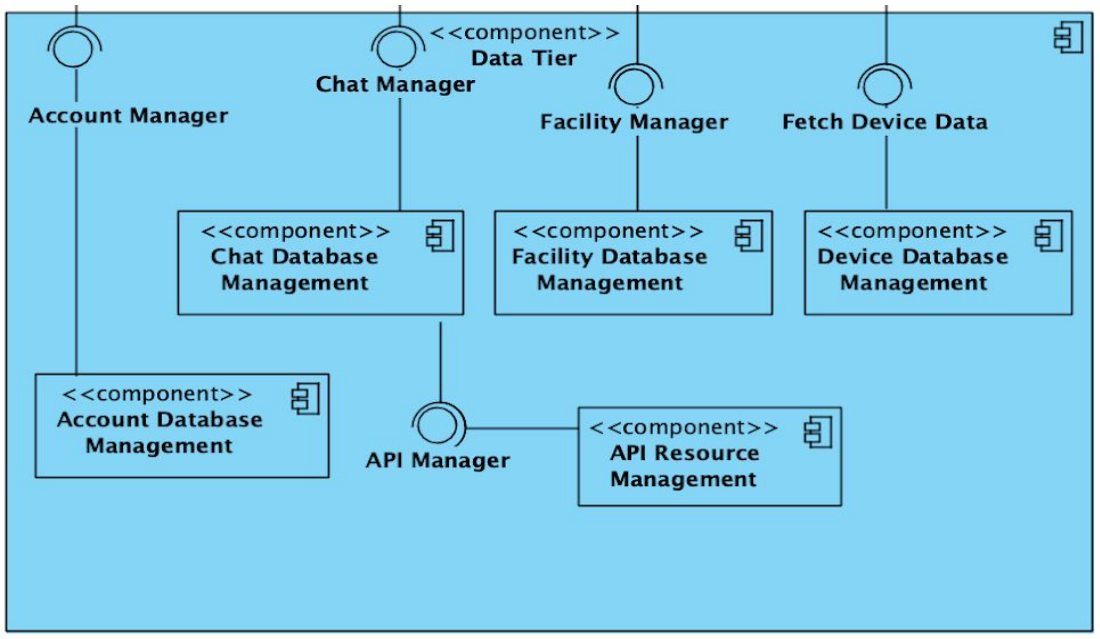


Application Tier Subsystem is responsible for the main functionality of our system. DJANGO will be used in implementation of this layer. It is going to run on server side and contain the logic related to the system. It receives requests from *Presentation tier* and constructs response data according to them. Inputs from the user will be processed in this layer by using Client Handler. This subsystem is also the only connection between *Presentation and Data tiers*. *Application tier* is further divided into two components.

● **Application Management** is responsible for handling requests from client side. It has a Façade class called ClientHandler. This class includes main operations that the subsystem can do. Only component that has permission to change data in Data Tier.

● **Device Management**- fetches the devices of the facility from the database.

4.3 Data Tier Subsystem



Data Tier Subsystem is going to be deployed on server side. It's main functionality is providing secure and reliable access to the database. MySQL will be used in implementation of this layer. *Data Tier* is decomposed into five main components:

1. **Account Database Management** which is responsible for managing user database.
2. **Chat Database Management** will manage the Chat database.
3. **Facility Database Management** will manages the Facility database.
4. **Device Database Management** will be responsible for the Device database.
5. **API Resource Management** makes use of several APIs to provide functionality of chatbot.

5 Glossary

Server is the system where access management to the service happens.

Client is the system where user's interaction with the service happens.

Hotel Database Management is responsible for managing data related to the hotel.

6 References

1. Rebecca Lake, "*Hotel Industry Statistics*", 26-Apr-2016. [Online]. Retrieved from: https://www.creditdonkey.com/hotel-industry-statistics.html [Accessed: 24-Feb-2019].
2. Guinness World Records, "*Oldest Hotel*", 29-Oct-2011. [Online]. Retrieved from: http://www.guinnessworldrecords.com/world-records/oldest-hotel/ [Accessed: 24-Feb-2019].
3. Marriott International, "*Marriott Bonvoy*" [Online] Retrieved from: https://play.google.com/store/apps/details?id=com.marriott.mrt [Accessed: 24-March-2019].
4. Hyatt Corporation, "*Hyatt Hotels*" [Online] Retrieved from: https://play.google.com/store/apps/details?id=com.Hyatt.hyt Accessed: 24-March-2019].
5. Hilton, "*Hilton Honors*" [Online] Retrieved from: https://play.google.com/store/apps/details?id=com.hilton.android.hhonors [Accessed: 24-March-2019].
6. eZee Absolute, "*eZee Absolute*" [Online] Retrieved from: https://www.ezeeabsolute.com/ [Accessed: 24-March-2019].