

and human resource to maintain the server. However, after the application is on the market for a period of time, the demand for the application may grow and overwhelm the capacity of the static server, potentially causing jamming or crashing of the server. As such, the traditional static server architecture inevitably mismatches supply and demand, providing either too much or not enough computational resources.

SUMMARY

[0004] The embodiments herein disclose chatbot check-in platforms implemented with serverless computing architecture and application management methods. The platform analyzes a chatbot request of a user to find a particular service and presents options to the user for such service. The serverless architecture responds to demand dynamically. The serverless architecture is automatically scalable to match the change of demand, providing no mismatch between demand and supply of computational resources.

[0005] This disclosure uses a salon application that uses a chatbot check-in platform, as an example of an application management. It is noted that the computational architecture disclosed is not limited to a salon application. Other software applications such as finding a car, a house, a restaurant, a nail shop, a movie, a hotel, a service, or the like can all be similarly managed by the chatbot platform.

[0005a] In a first aspect, the present disclosure provides a chatbox check-in platform, comprising: a salon application associated with a messenger chatbot, wherein a user submits a request for an appointment for a salon service via input on a user interactive device associated with the messenger chatbot; a salon services API in communication with the salon application, having a first booking engine including available appointments and a location service for finding a salon nearby the user, the salon service API matching the request to an available appointment at a location nearby the user, sending the available appointment to the user via the user interactive device, and providing confirmation to the salon application for confirmation in response to an acceptance by the user via the messenger

chatbot; a serverless backend computing system having a second booking engine, wherein upon receiving confirmation by the user, the salon services API submits a booking request to the second booking engine for booking the appointment with the salon; a gateway API between the salon services API and the serverless backend computing system for throttling an amount of data and rate of data the serverless backend computing system can receive from the salon services API in order to maintain computing efficiency on the serverless backend computing system; a dynamic database configured to store appointments using unstructured data to store cache configuration files, wherein when the gateway API receives a request, the gateway API is configured to access a cache configuration file from the dynamic database specific for the request; and a database interface in communication with the serverless backend computing system for storing the appointment in the dynamic database, and upon storing the appointment sending an update to the first booking engine removing the available appointment; wherein the first booking engine contains updated available appointments such that it can respond to requests by users for services without communicating with the serverless backend computing system until the user confirms the appointment to maintain efficiency on the serverless backend computing system.

[0006] A chatbot check-in platform includes a salon application associated with a messenger chatbot, wherein a user requests via voice an appointment for a salon service. The platform includes a location service in communication with the salon application for finding a salon nearby a user; a salon services API in communication with the salon application for finding a requested salon service and time for appointment for the salon; and a database interface responding to a request for making an appointment with the salon.

[0007] In a second aspect, the present disclosure provides a method of using a chatbot check-in platform, the method including instructions, when executed by a processor, that cause the processor to execute actions comprising: receiving, by a first processor of a serverless backend system comprising a gateway API, a request received via input on a user interactive device associated with a messenger chatbot for booking an appointment for a service provided by a salon, wherein the first processor is a processor of a device, the device includes

machine readable memory accessible by the first processor; wherein when the gateway API receives the request, the gateway API is configured to access a cache configuration file from a dynamic database specific for the request, wherein the dynamic database is configured to store appointments using unstructured data to store cache configuration files; finding a salon and service, by the first processor in response to the request using a salon service API running on the serverless backend system; providing, by the first processor, a suggested booking for the salon and service via the associated messenger chatbot; and upon receiving, by the first processor, a confirmation from the user via the chatbot, sending a request to a second processor to book the salon and service through the gateway API that controls the amount of data and rate of data that can be send to the second processor to maintain efficiency; and booking the salon and service by the second processor, storing the booking in the dynamic database, and sending an update of available appointments to the first processor; wherein the first processor can respond to the request without communicating with the second processor until the confirmation is received.

[0008] The present invention relates generally to a chatbot check-in platform implemented with serverless computing architecture. The platform dynamically responds to service requests without maintaining a fixed amount of computational resources. The present specification further discloses application management using the platform implemented with serverless computing architecture. The platform's serverless database architecture has the benefit of being scalable to match the change of demand. A database center that operates this platform can dynamically allocate computational resources to any client/application whenever there is a need. The computational resources operated by the database center can be efficiently allocated to meet the demands. An application developer who uses the services provided by the database center, does not have to worry about the large amount of hardware investments. There is no need for the application developer to provision a piece of hardware. There is no need for the application developer to increase or decrease the amount of physical hardware to meet the change of the demand. The platform, serverless database is event driven.

[0009] The foregoing has outlined rather broadly the features and technical advantages of the present invention in order that the detailed description of the invention that follows may be better understood. Additional features and advantages of the invention will be described hereinafter that

form the subject of the claims of the invention. It should be appreciated by those skilled in the art that the concepts and specific embodiments disclosed may be readily utilized as a basis for modifying or designing other structures for carrying out the same purposes of the present invention. It should also be realized by those skilled in the art that such equivalent constructions do not depart from the spirit and scope of the invention as set forth in the appended claims. The novel features that are believed to be characteristic of the invention, both as to its organization and method of operation, together with further objects and advantages will be better understood from the following description when considered in connection with the accompanying figures. It is to be expressly understood, however, that each of the figures is provided for the purpose of

illustration and description only and is not intended as a definition of the limits of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] For a more complete understanding of the disclosed systems and methods, reference is now made to the following descriptions taken in conjunction with the accompanying drawings.

[0011] FIG. 1 shows a block diagram of a chatbot check-in platform including a salon application according to one embodiment of the disclosure.

[0012] FIG. 2 shows a method of a salon application that uses a chatbot check-in platform and serverless architecture as a backend structure according to one embodiment of the disclosure.

[0013] FIG. 3 shows a method of a chatbot check-in platform that uses a serverless architecture to handle a request from a salon application according to one embodiment of the disclosure.

[0014] FIG. 4 shows a method of using a chatbot to check-in with a salon that uses a serverless backend architecture, according to one example embodiment of the disclosure.

[0015] FIG. 5 shows a method of using a chatbot to cancel a check-in with a salon that uses a serverless backend architecture, according to one example embodiment of the disclosure.

[0016] FIG. 6 is a block diagram illustrating a computing system according to one embodiment of the disclosure.

DETAILED DESCRIPTION

[0017] In general, the present disclosure relates to a messenger chatbot check-in platform for checking for salon services through a salon application. The check-in platform analyzes the intent of a user to determine a user's desire for some service and then presents options to the user for choices based on the intent or request. For example, a user could type the word "haircut" into a messenger chatbot, for example Facebook Messenger. The chatbot then interfaces with a salon

CLAIMS

What is claimed is:

1. A chatbot check-in platform, comprising:

a salon application associated with a messenger chatbot, wherein a user submits a request for an appointment for a salon service via input on a user interactive device associated with the messenger chatbot;

a salon services API in communication with the salon application, having a first booking engine including available appointments and a location service for finding a salon nearby the user, the salon service API matching the request to an available appointment at a location nearby the user, sending the available appointment to the user via the user interactive device, and providing confirmation to the salon application for confirmation in response to an acceptance by the user via the messenger chatbot;

a serverless backend computing system having a second booking engine, wherein upon receiving confirmation by the user, the salon services API submits a booking request to the second booking engine for booking the appointment with the salon;

a gateway API between the salon services API and the serverless backend computing system for throttling an amount of data and rate of data the serverless backend computing system can receive from the salon services API in order to maintain computing efficiency on the serverless backend computing system;

a dynamic database configured to store appointments using unstructured data to store cache configuration files, wherein when the gateway API receives a request, the gateway API is configured to access a cache configuration file from the dynamic database specific for the request; and

a database interface in communication with the serverless backend computing system for storing the appointment in the dynamic database, and upon storing the appointment sending an update to the first booking engine removing the available appointment;

wherein the first booking engine contains updated available appointments such that it can respond to requests by users for services without communicating with the serverless backend computing system until the user confirms the appointment to maintain efficiency on the serverless backend computing system.

2. A chatbot check-in platform according to claim 1, further comprising a pay service in communication with the salon application.
3. A chatbot check-in platform according to claim 1, further comprising a demand generation platform in communication with the salon application.
4. A chatbot check-in platform according to claim 1, wherein the second booking engine provides updated available appointments to the first booking engine at regular intervals.
5. A chatbot check-in platform according to claim 4, wherein the salon application resides on a user device and the user device receives the updated available appointments from the second booking engine at regular intervals without requesting the updated available appointments.
6. A chatbot check-in platform according to claim 1, wherein the salon application executes on a serverless backend architecture.
7. A method of using a chatbot check-in platform, the method including instructions, when executed by a processor, cause the processor to execute actions comprising:

receiving, by a first processor of a serverless backend system comprising a gateway API, a request received via input on a user interactive device associated with a messenger chatbot for booking an appointment for a service provided by a salon, wherein the first processor is a processor of a device, the device includes machine readable memory accessible by the first processor;

wherein when the gateway API receives the request, the gateway API is configured to access a cache configuration file from a dynamic database specific for the request, wherein the dynamic database is configured to store appointments using unstructured data to store cache configuration files;

finding a salon and service, by the first processor in response to the request using a salon service API running on the serverless backend system;

providing, by the first processor, a suggested booking for the salon and service via the associated messenger chatbot;

upon receiving, by the first processor, a confirmation from the user via the messenger chatbot, sending a request to a second processor to book the salon and service through the gateway API that controls the amount of data and rate of data that can be sent to the second processor to maintain efficiency; and

booking the salon and service by the second processor, storing the booking in the dynamic database, and sending an update of available appointments to the first processor;

wherein the first processor can respond to the request without communicating with the second processor until the confirmation is received.

8. A method according to claim 7, further comprising after providing the suggested booking, determining by the first processor a user response and if the user response included the confirmation from the user, booking the salon service.
9. A method according to claim 8, wherein determining a user response includes if the user response included a request for a different location, then determining by the first processor alternative nearby salons for the booking.
10. A method according to claim 9, wherein determining a user response includes if the user response included a request for a different time, then determining by the first processor alternative times for the booking.

11. A method according to claim 7, further comprising determining by the first processor if the user has booked through the system before and if the user has booked through the system before, using prior booking information at least in part to find a salon and service.
12. A method according to claim 11, wherein determining if the user has booked before includes if the user has not booked through the system before, prompting the user via the messenger chatbot for the user's current location.
13. A method according to claim 11, wherein determining if the user has booked through the system before includes using a user identification to determine if the user has booked through the system before.