US 20240045780A1

(19) **United States**

(12) **Patent Application Publication**

(10) **Pub. No.: US 2024/0045780 A1**

(43) **Pub. Date: Feb. 8, 2024**

AZIZ et al.

(54) **DIGITAL ASSISTANT USING ROBOTIC PROCESS AUTOMATION**

(71) Applicant: **UiPath, Inc.**, New York, NY (US)

(72) Inventors: **Imran AZIZ**, Seattle, WA (US); **Tuan NGUYEN**, Anaheim, CA (US); **Justin LECLAIR**, Del Valle, TX (US); **Chibichakaravarthi VIKRAMATHITHAN**, Sammamish, WA (US); **Umesh Amin**, Redmond, WA (US)

(73) Assignee: **UiPath, Inc.**, New York, NY (US)

(21) Appl. No.: **18/480,710**

(22) Filed: **Oct. 4, 2023**

Related U.S. Application Data

(63) Continuation-in-part of application No. 17/489,995, filed on Sep. 30, 2021.

Publication Classification

(51) **Int. Cl.**

G06F 11/30 (2006.01)

G06F 9/455 (2006.01)

(52) **U.S. Cl.**

CPC ***G06F 11/302*** (2013.01); ***G06F 9/45512*** (2013.01)

(57) **ABSTRACT**

A method or system for executing one or more digital assistant tasks using robotic processing automation (RPA) includes populating, by a next generation digital assistant application, a drop down menu comprising of a series of options when a user performs an action in a software application. The method or system also includes populating, by the next generation digital assistant application, a side panel comprising a task to be performed by the next generation digital assistant application and a series of options for the user to select from. The method or system further includes updating, by the next generation digital assistant application, the side panel with additional options in response to a selected option from the series of options. The additional options are recommended actions for the user to select from in addition to the task performed by the next generation digital assistant application.

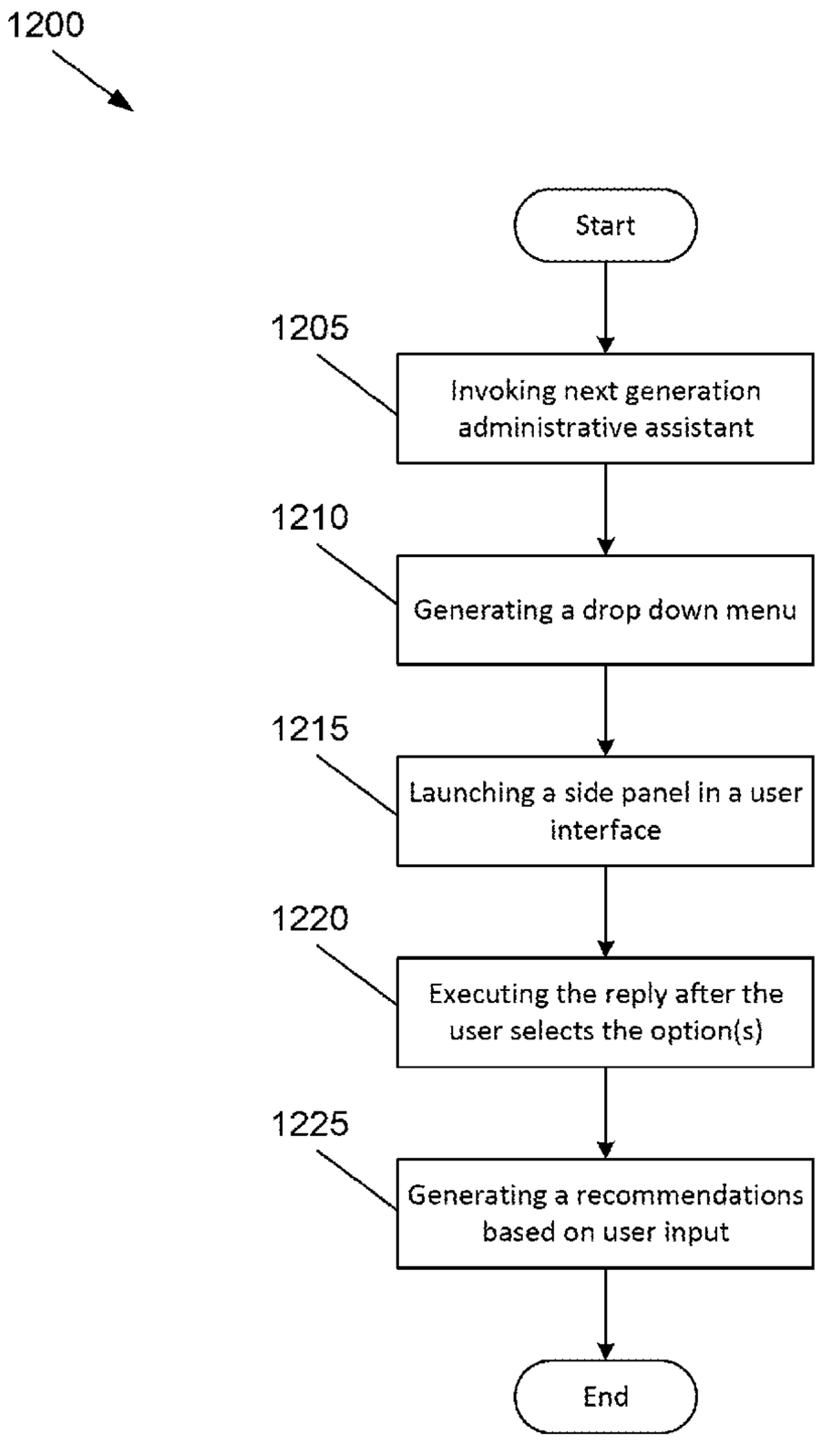


FIG. 1

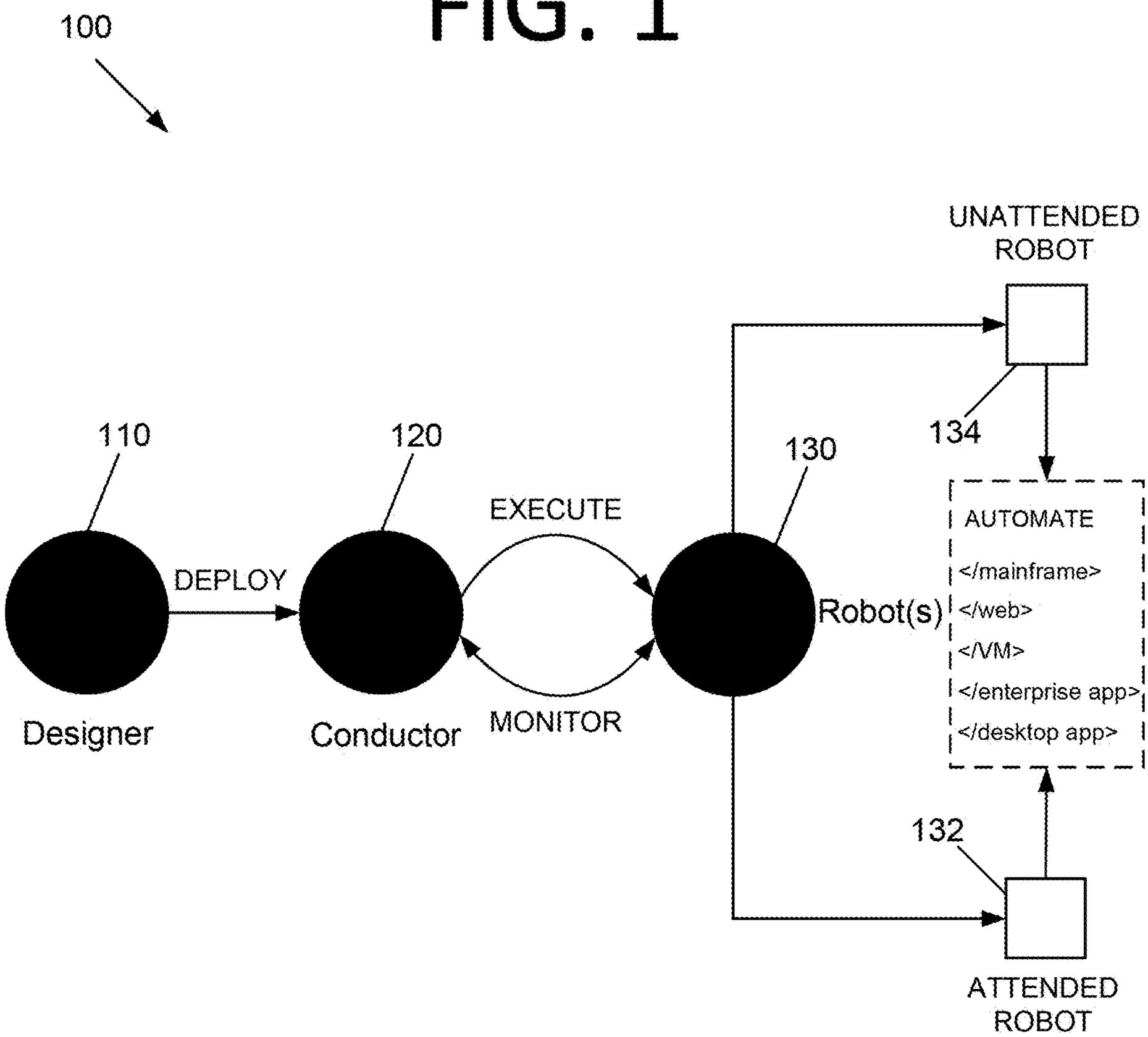

100
UNATTENDED ROBOT
110
120
130
134
EXECUTE
DEPLOY
Robot(s)
AUTOMATE
</mainframe>
</web>
</VM>
</enterprise app>
</desktop app>
Designer
Conductor
MONITOR
132
ATTENDED ROBOT

FIG. 2

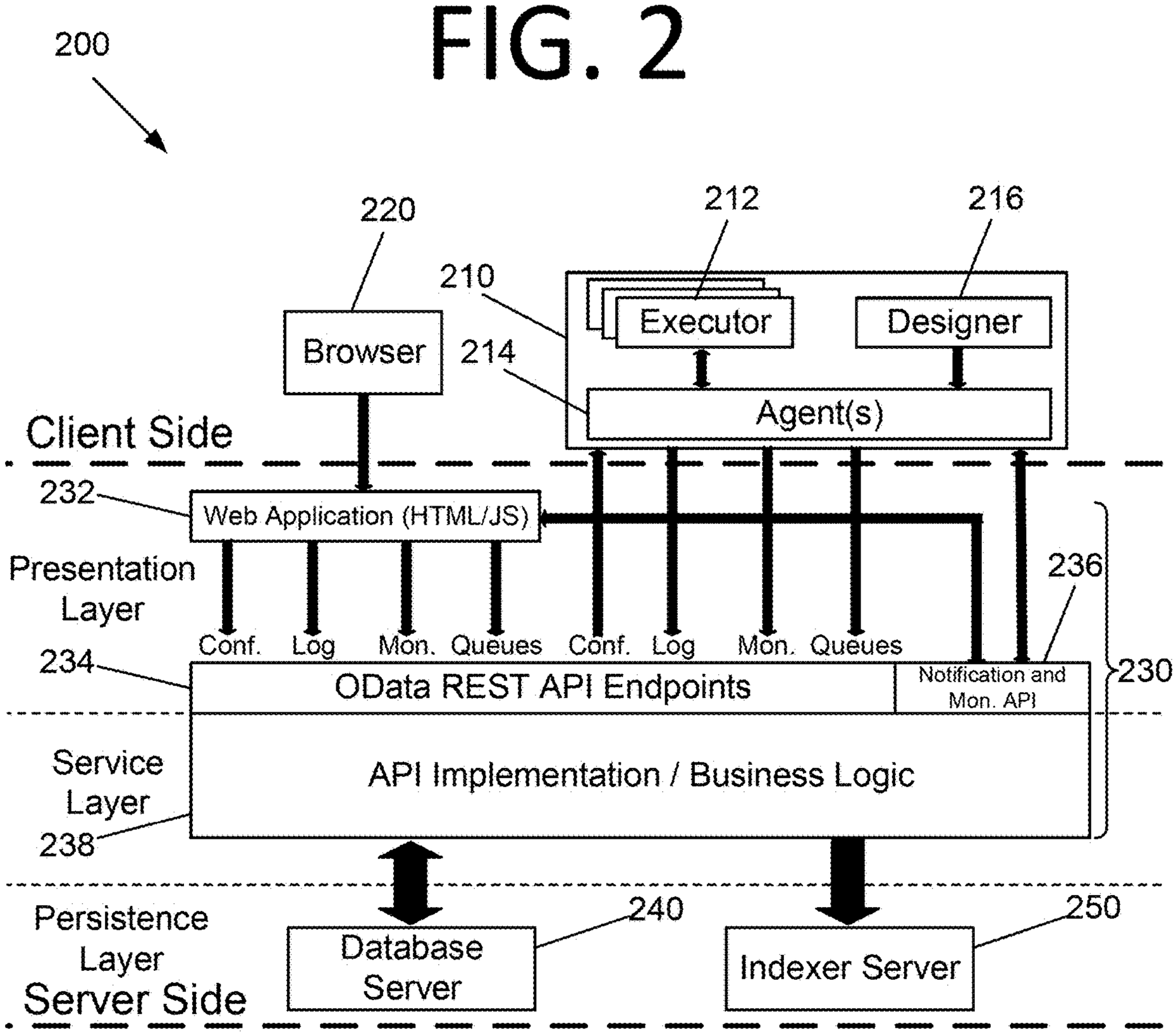

200
220
212
216
210
Executor
Designer
Browser
214
Agent(s)
Client Side
232
Web Application (HTML/JS)
Presentation Layer
236
Conf.
Log
Mon.
Queues
Conf.
Log
Mon.
Queues
234
OData REST API Endpoints
Notification and Mon. API
230
Service Layer
API Implementation / Business Logic
238
Persistence Layer
240
250
Database Server
Indexer Server
Server Side

FIG. 3

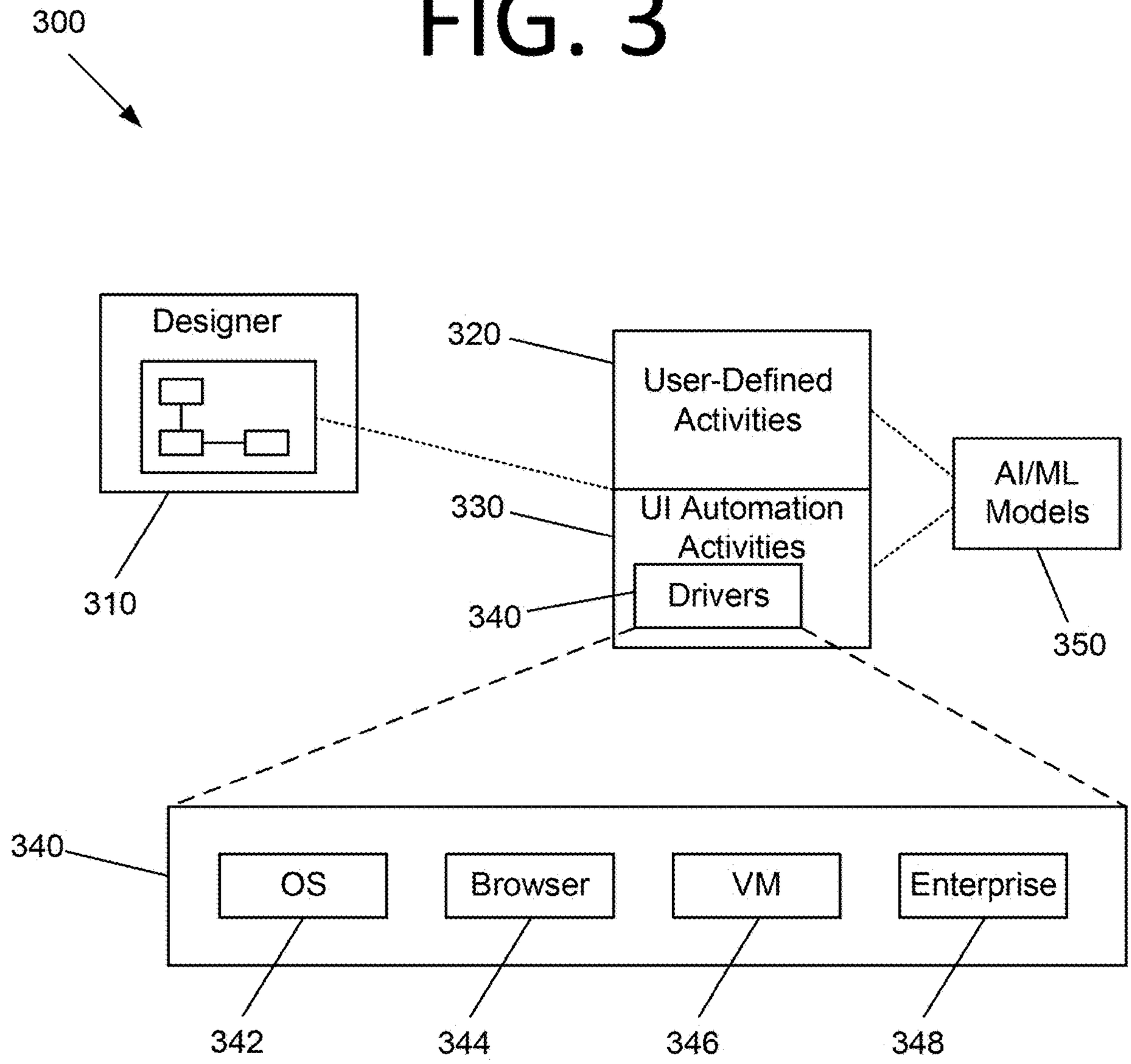

300
Designer
310
320
User-Defined Activities
330
UI Automation Activities
Drivers
340
AI/ML Models
350
340
OS
Browser
VM
Enterprise
342
344
346
348

FIG. 4

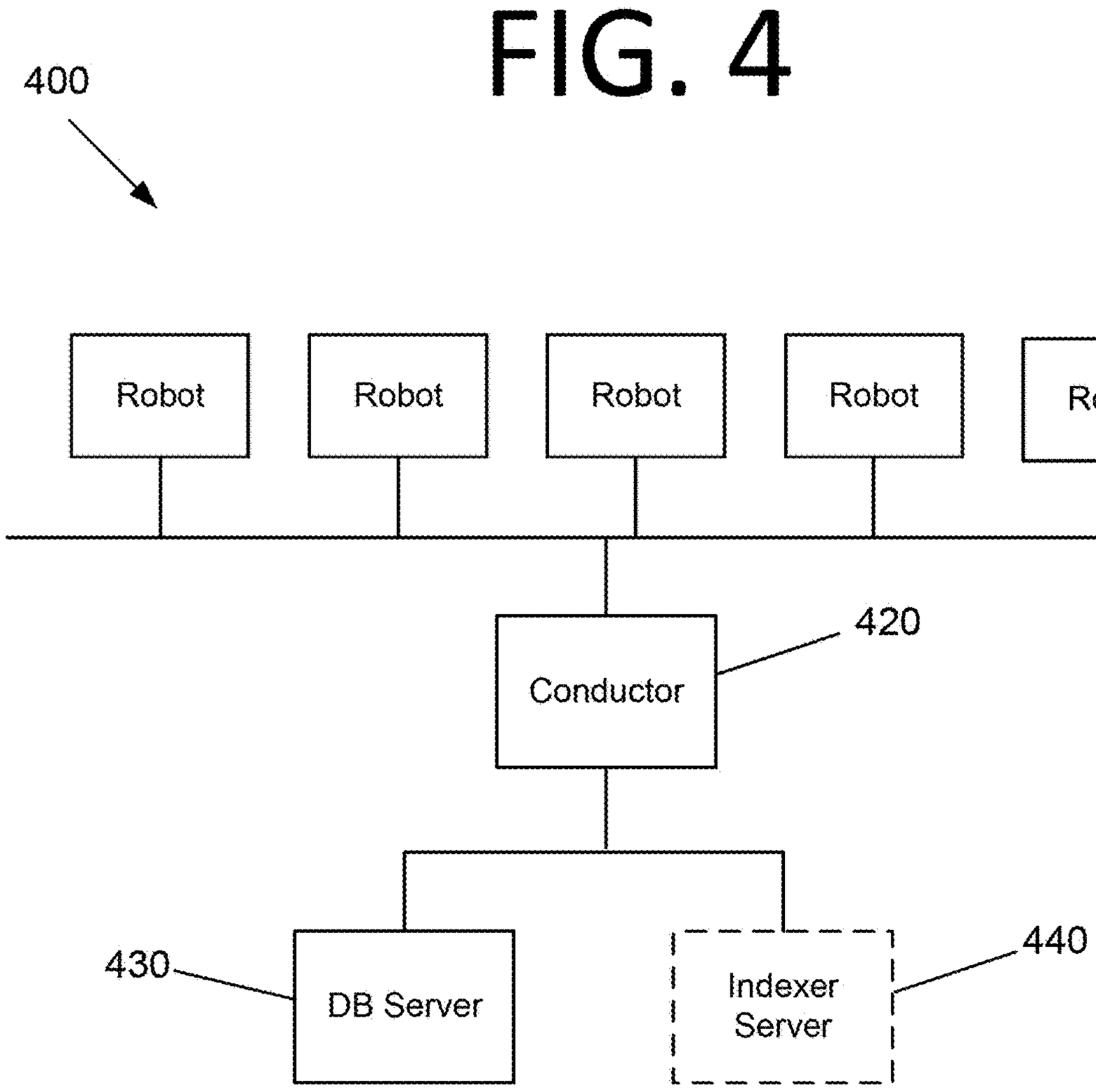

400
410
Robot
Robot
Robot
Robot
Robot
420
Conductor
430
DB Server
440
Indexer Server

FIG. 5

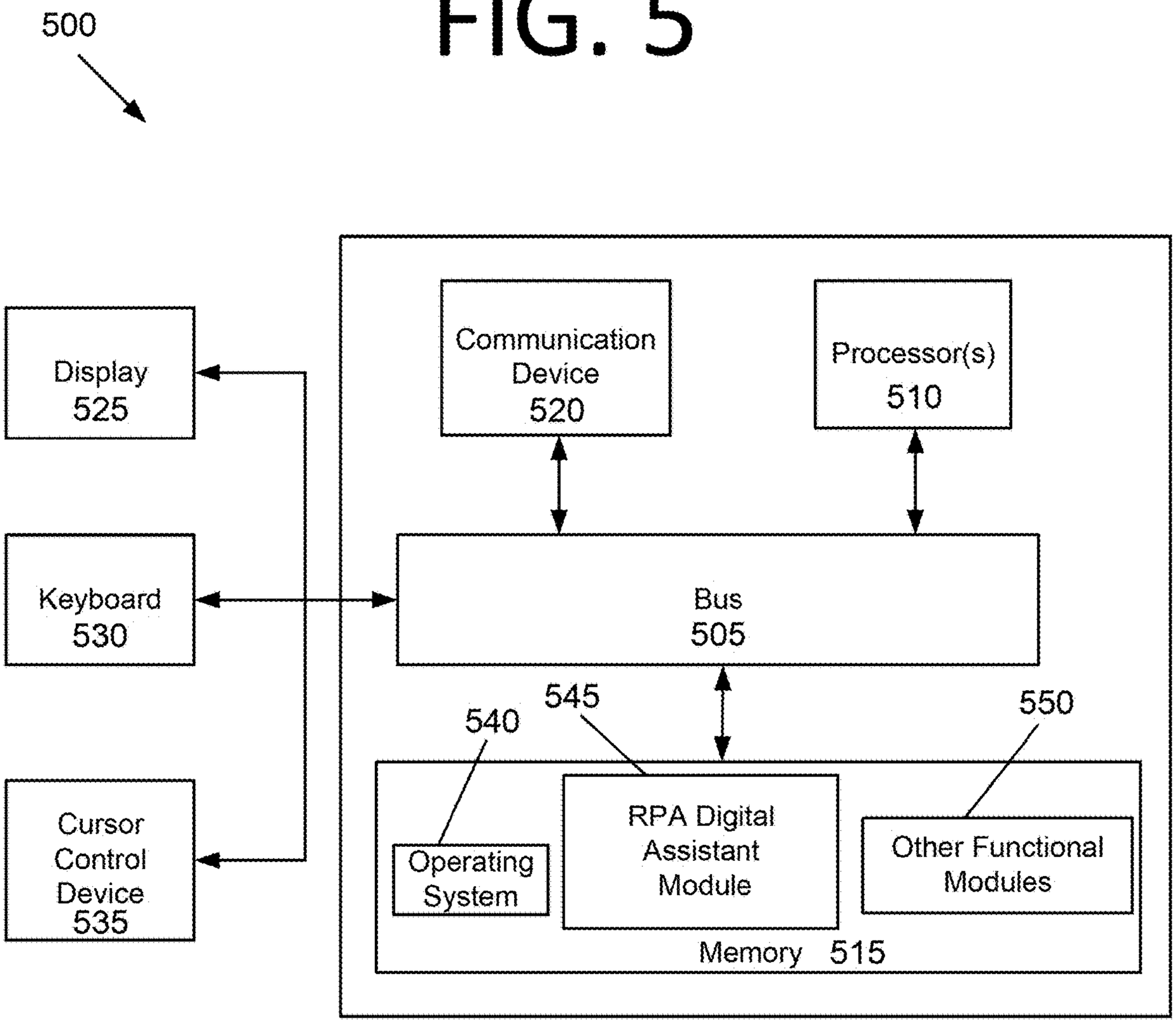
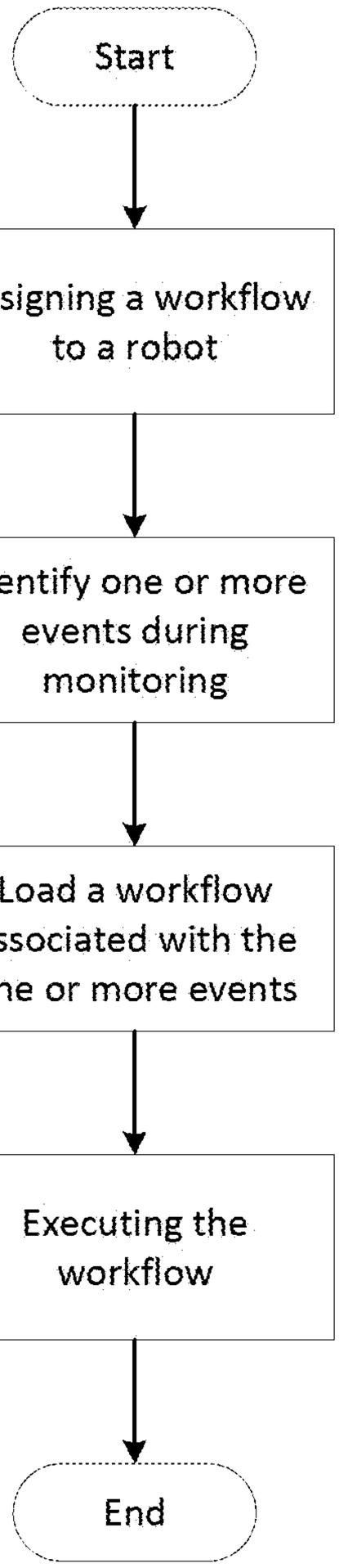

500
Display
525
Communication
Device
520
Processor(s)
510
Keyboard
530
Bus
505
545
540
550
Cursor
Control
Device
535
Operating
System
RPA Digital
Assistant
Module
Other Functional
Modules
Memory 515

FIG. 6


600


Start
Assigning a workflow to a robot
Identify one or more events during monitoring
Load a workflow associated with the one or more events
Executing the workflow
End

700

Fig. 7

Email From UiPath™ Robot:

Hey Customer,

Here are times that would work for the team:

1. 10:30-11:00am 7/12/21

2. 3:30-3:00pm 7/12/21

3. 4:00-4:30pm 7/12/21

Please respond to this email with the corresponding number that works for you, comma delimited. For example simply respond with: 1,2 and I'll take it from there!

From,
Your friendly neighborhood UiPath Robot

Fig. 8

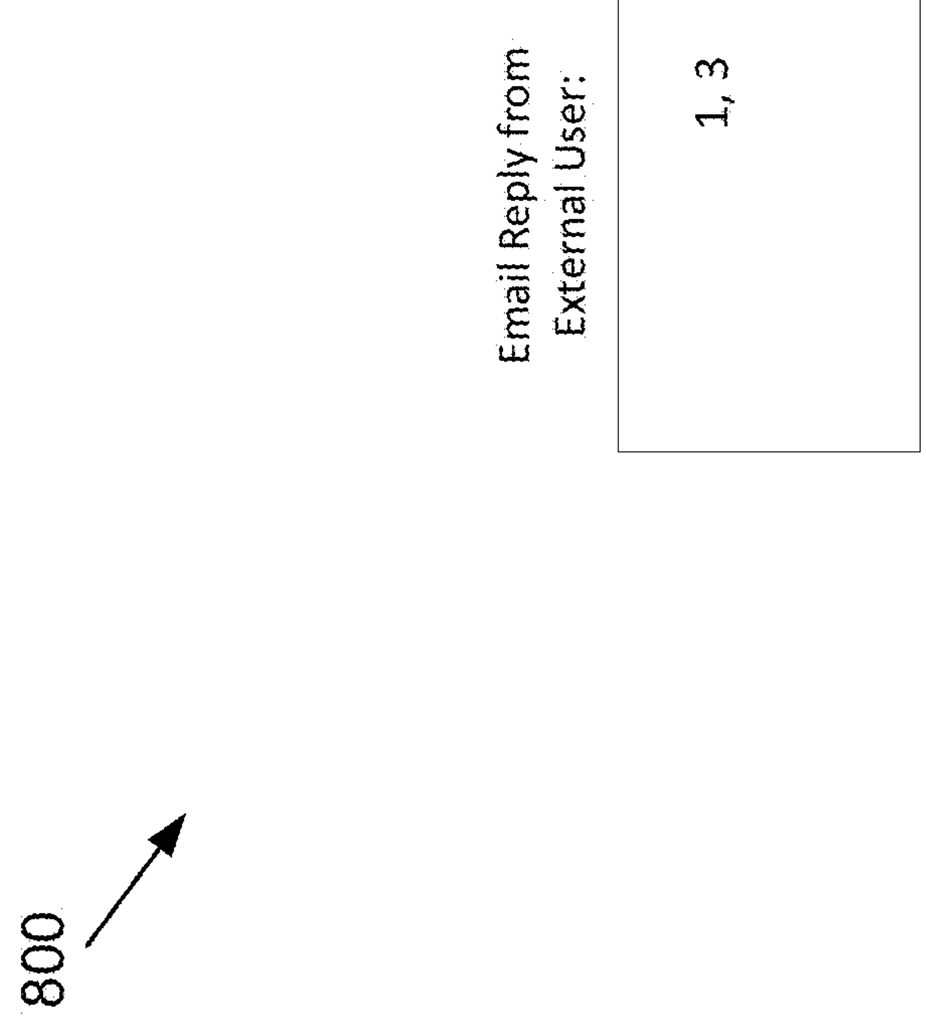
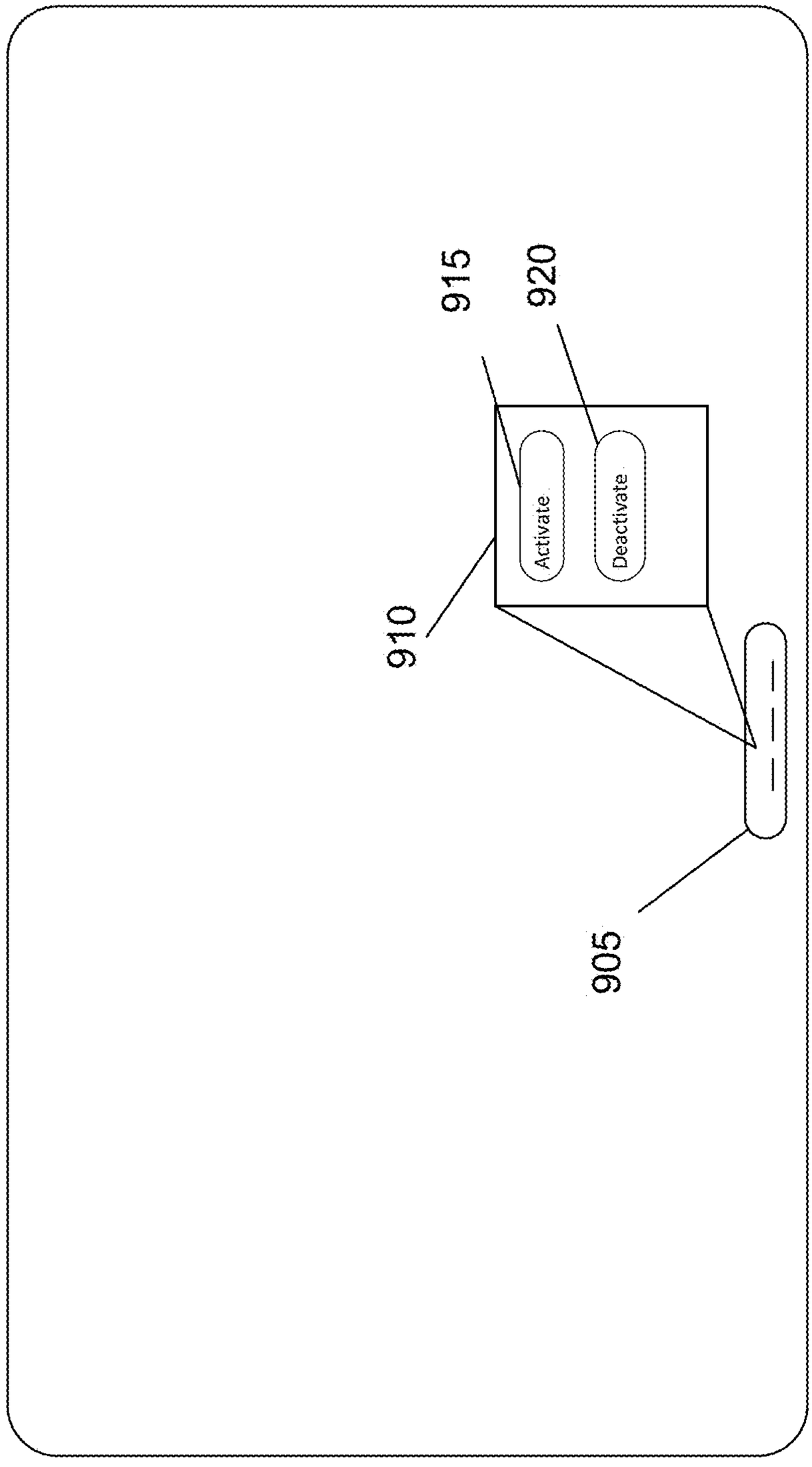


Fig. 9



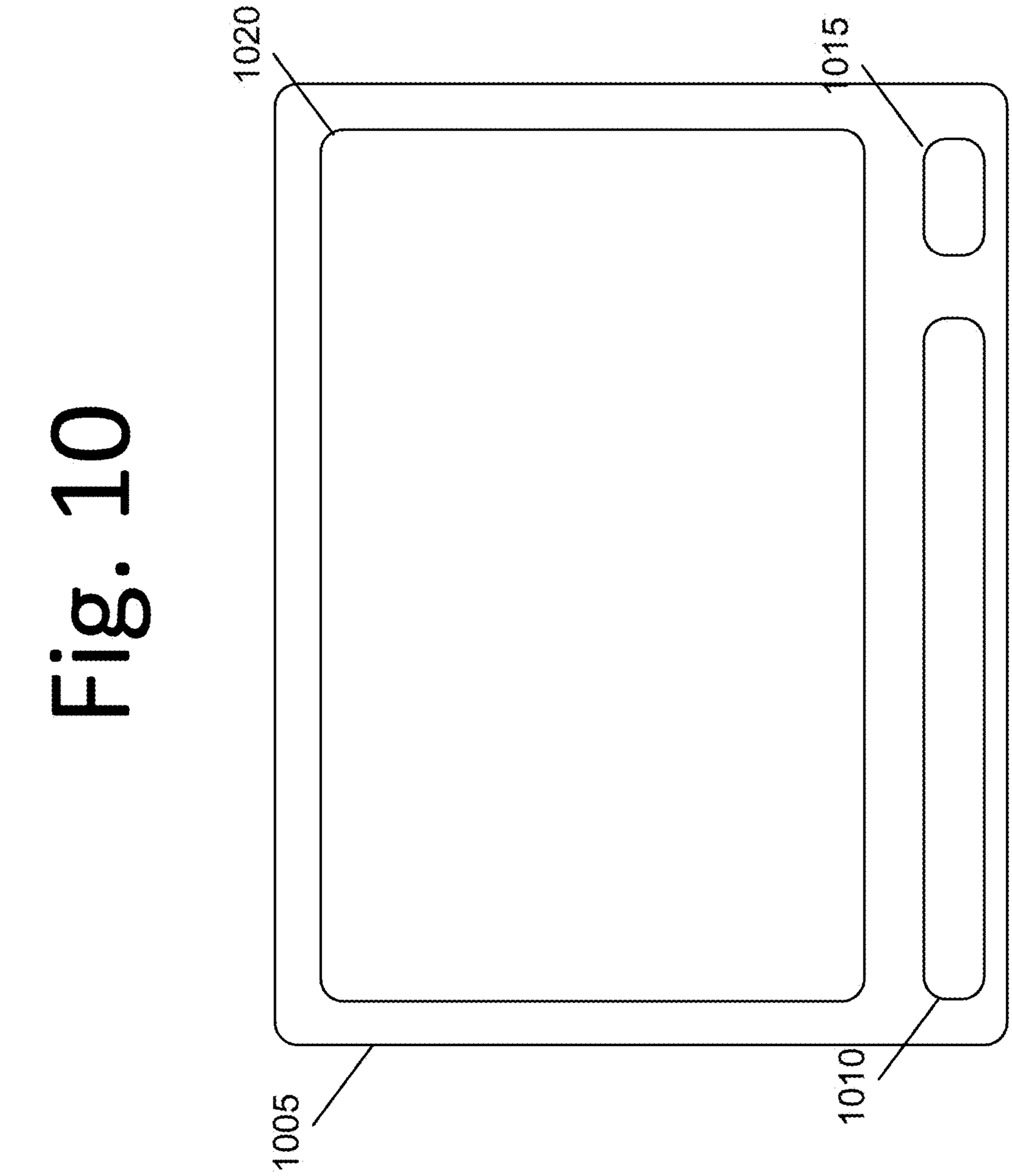


Fig. 10

1100A

Fig. 11A

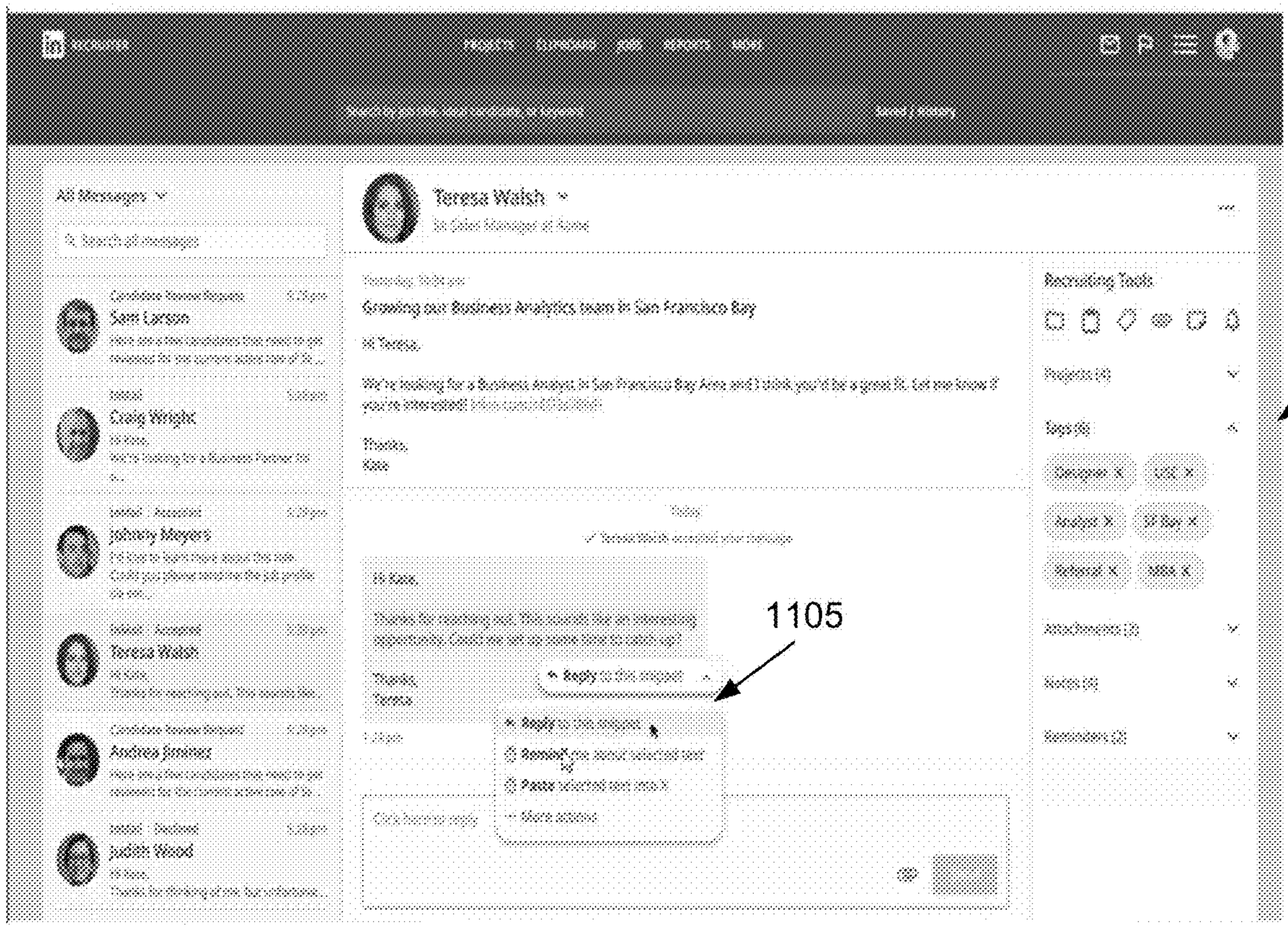


Fig. 11B

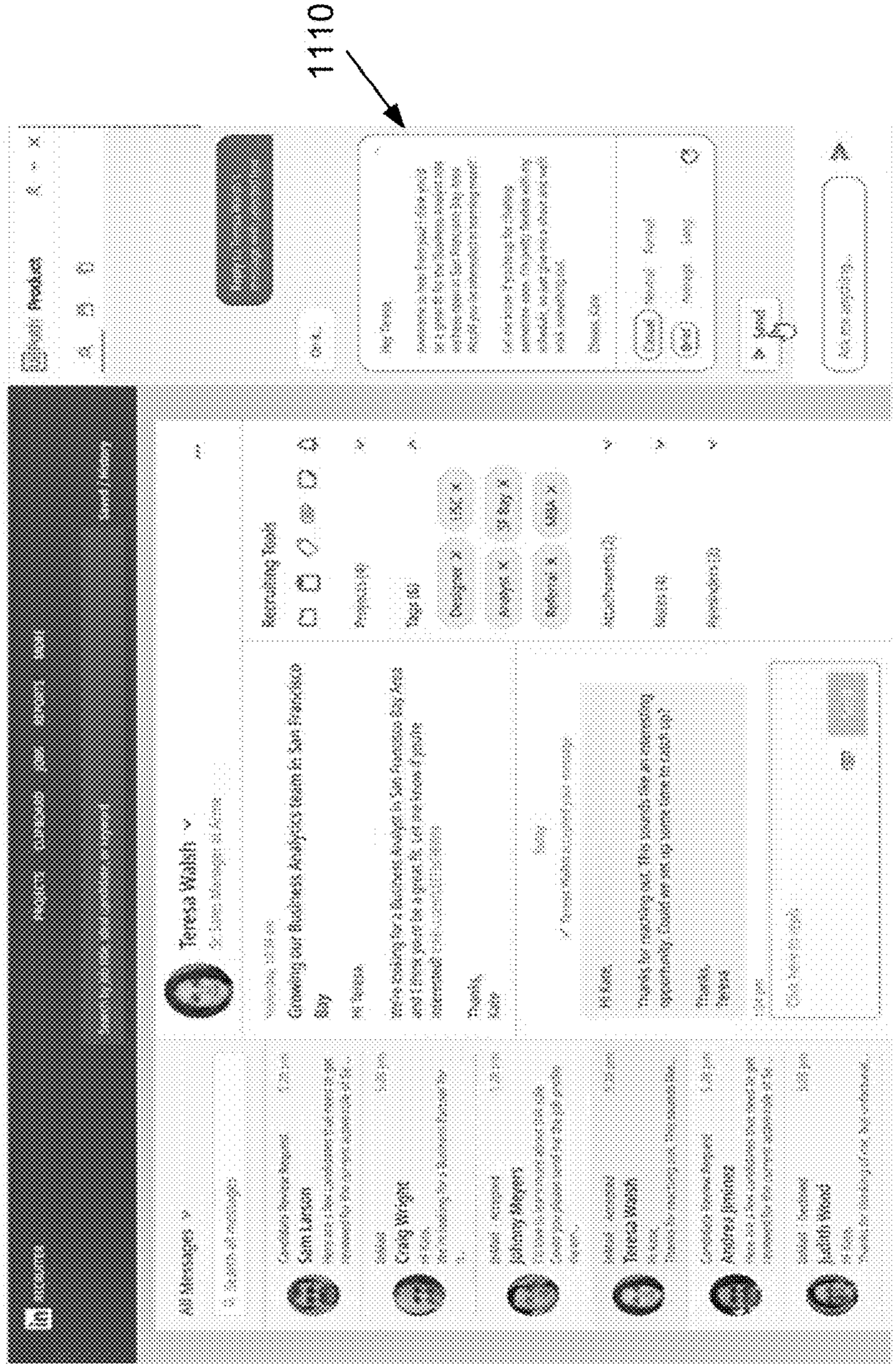


Fig. 11C

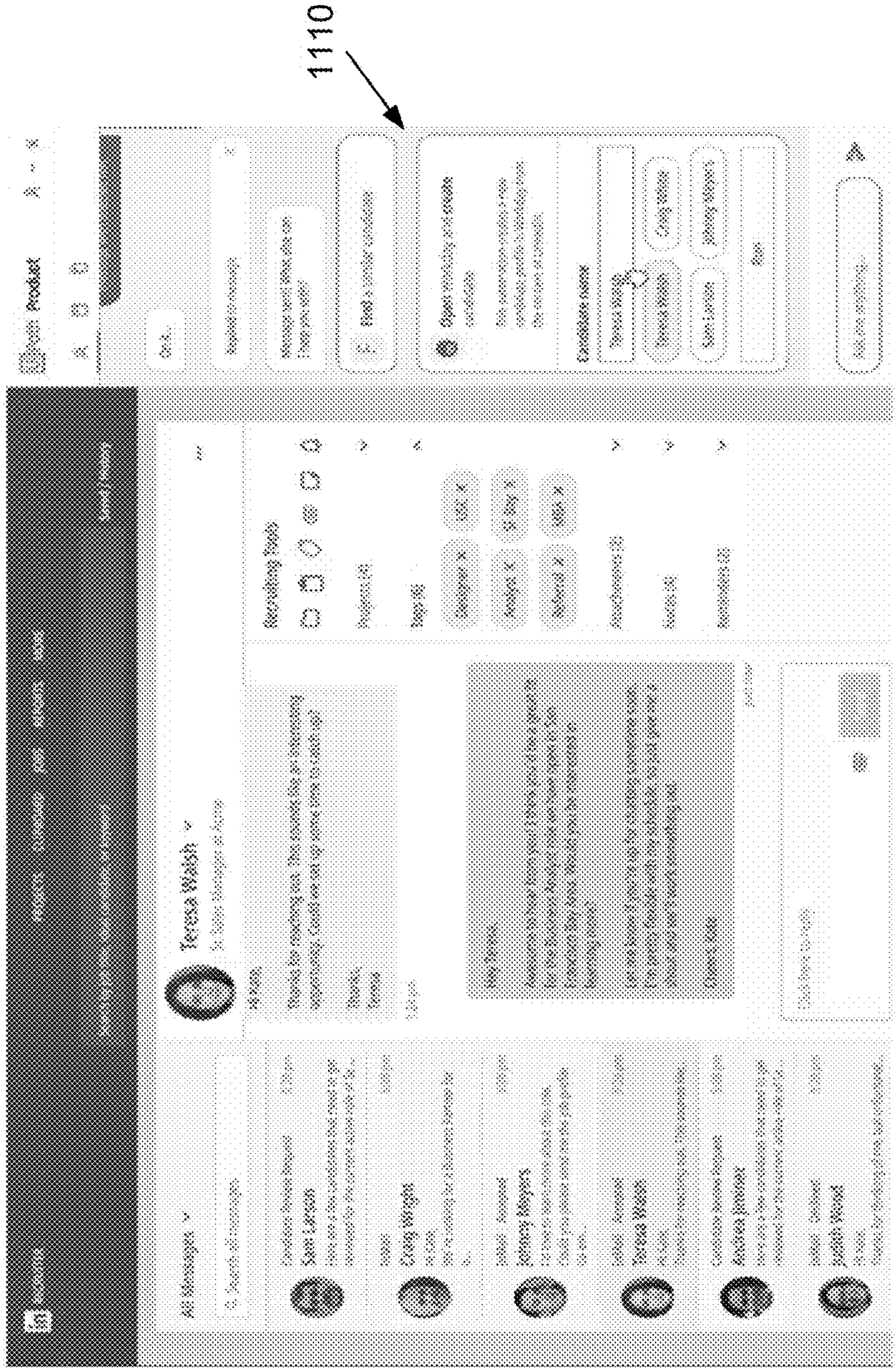


Fig. 11D

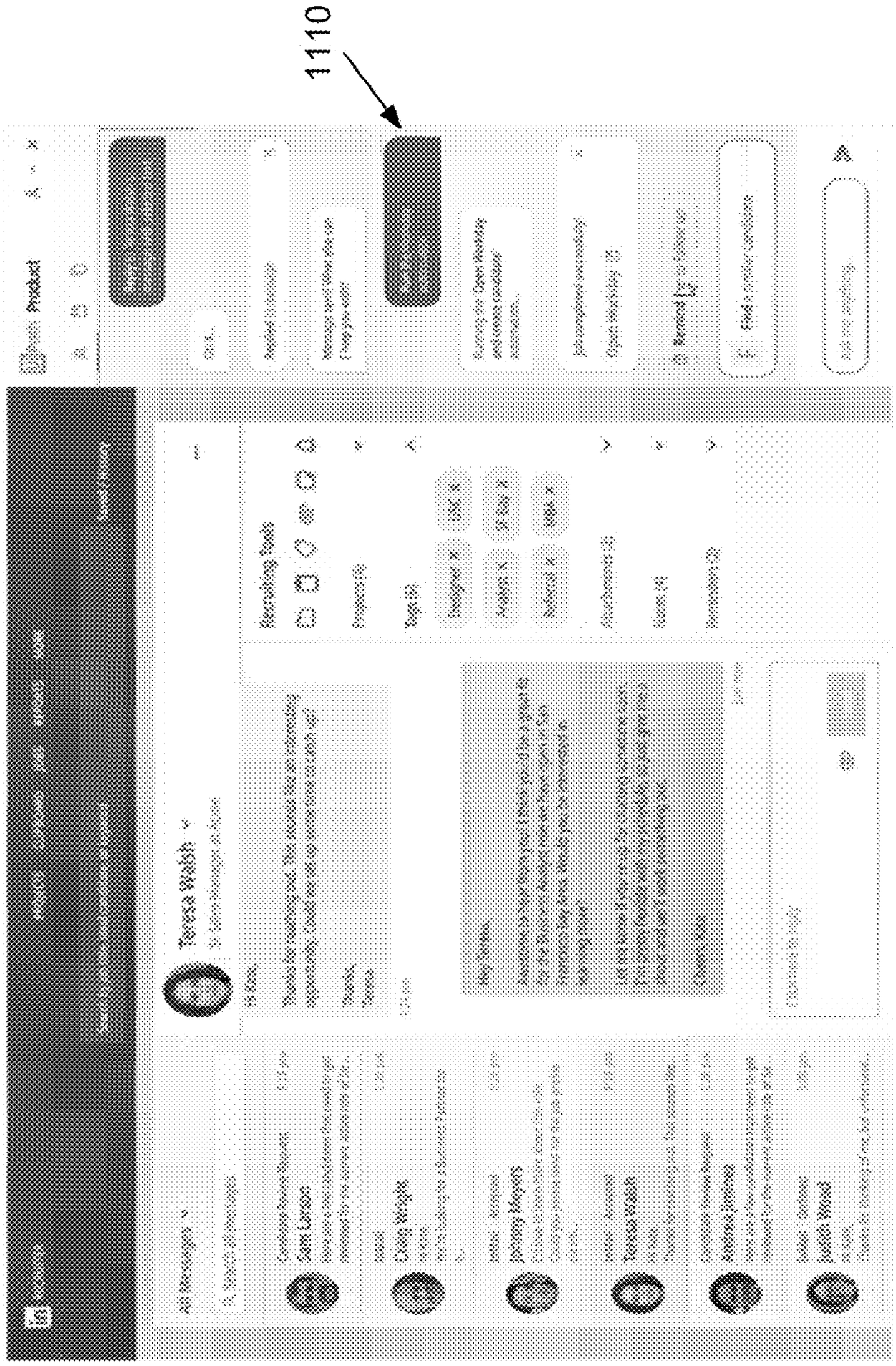
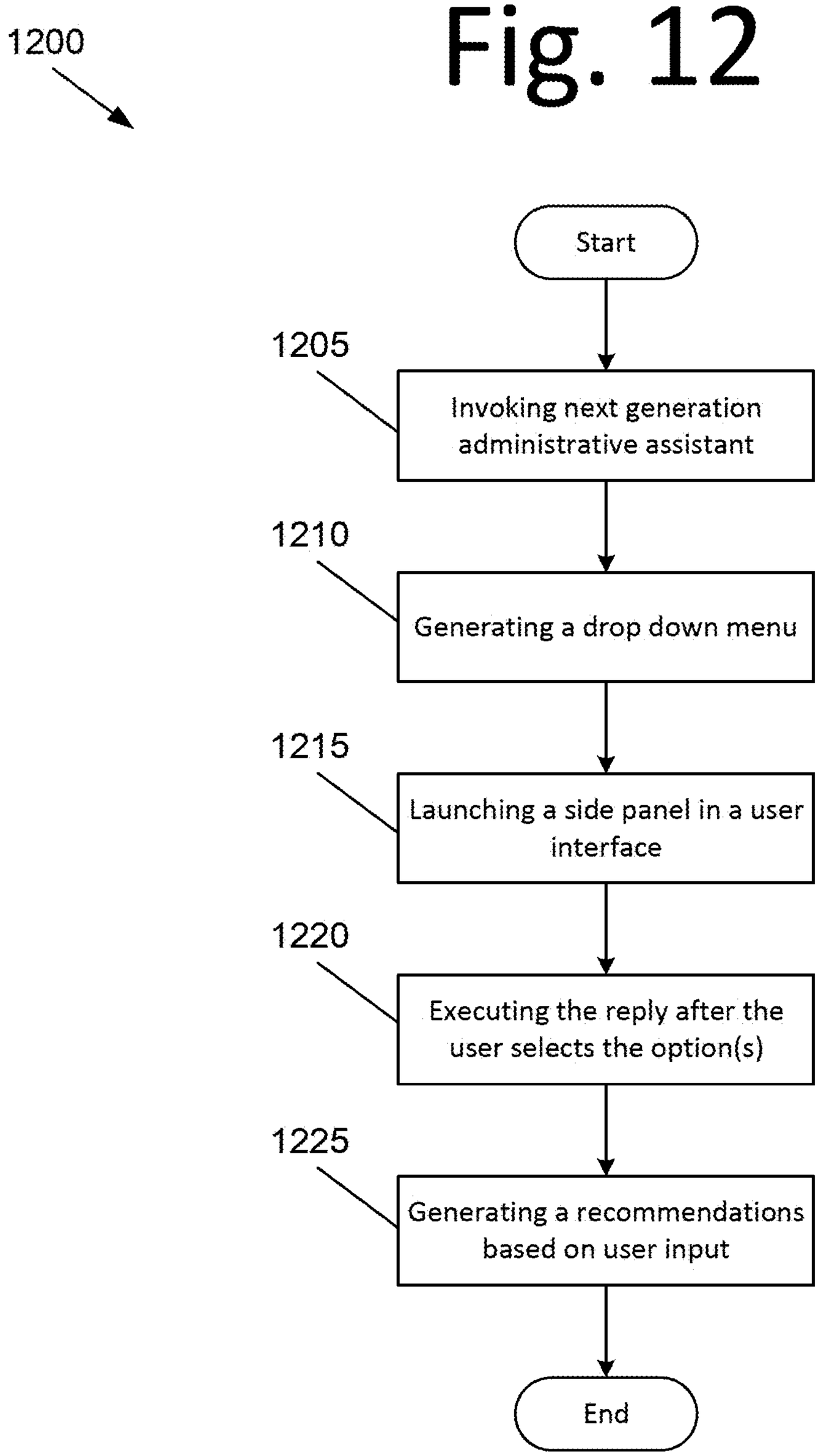


Fig. 12


1200
Start
1205
Invoking next generation administrative assistant
1210
Generating a drop down menu
1215
Launching a side panel in a user interface
1220
Executing the reply after the user selects the option(s)
1225
Generating a recommendations based on user input
End

1300

Fig. 13

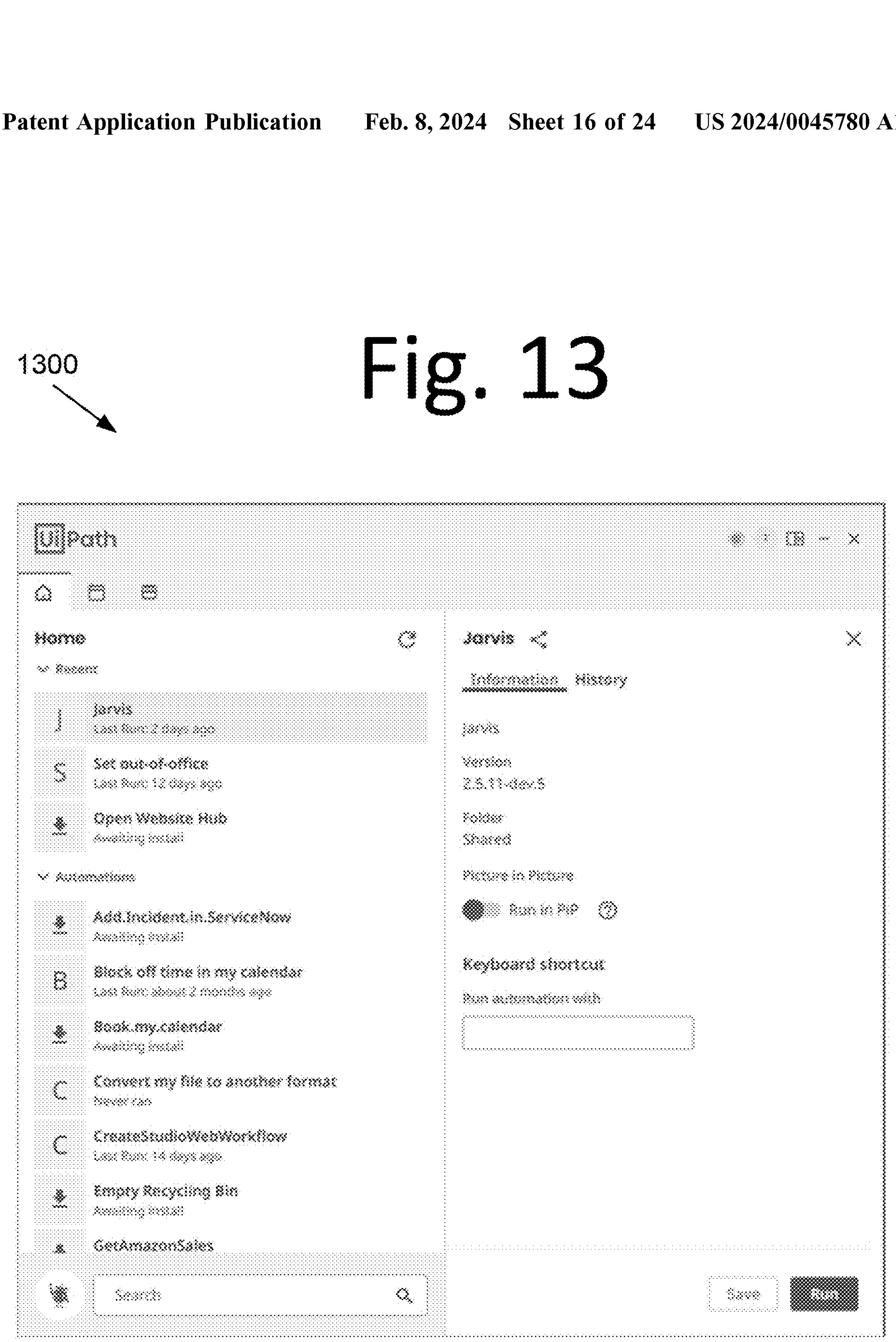
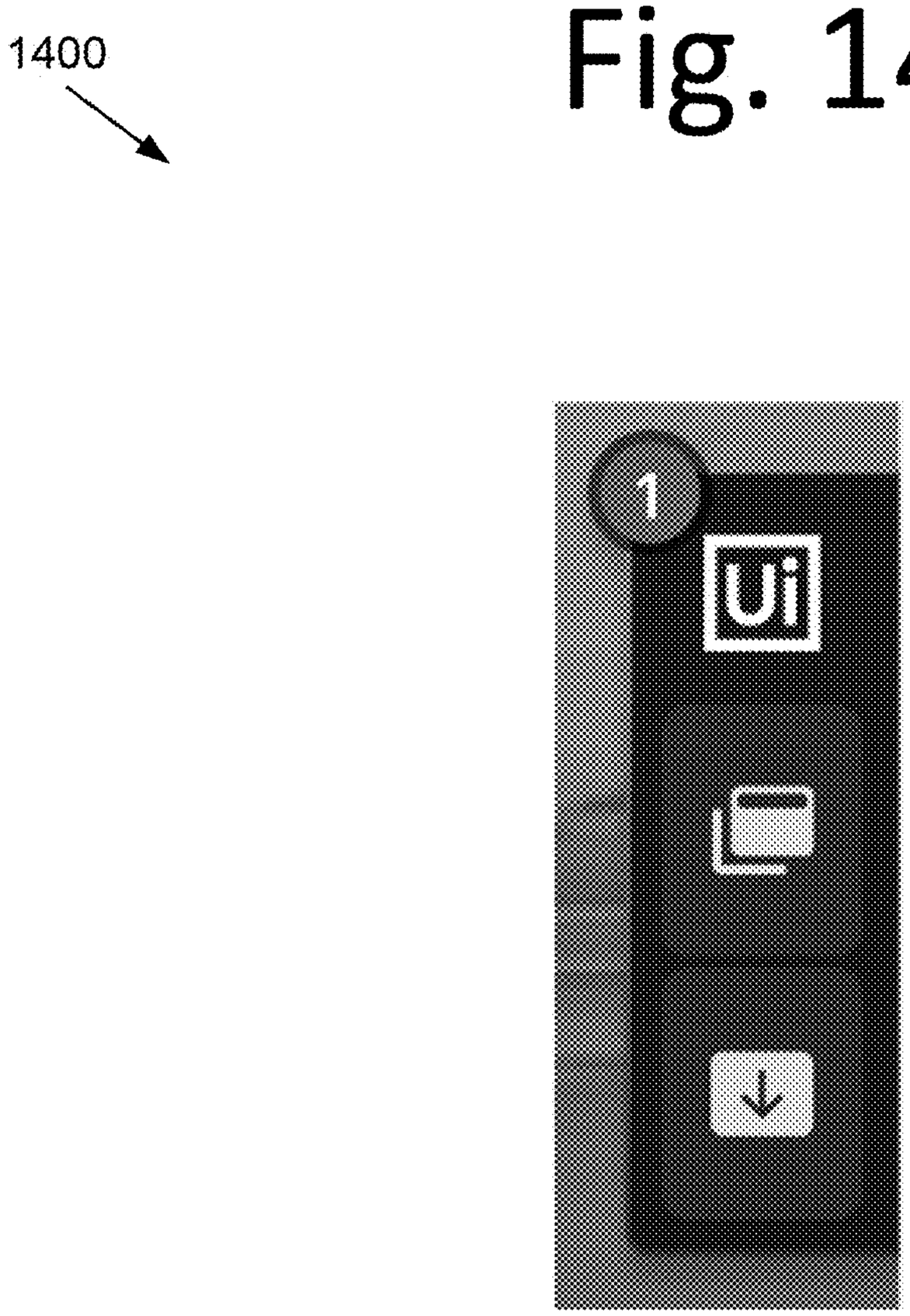

UiPath
Home
Recent
Jarvis
Last Run 2 days ago
Set out-of-office
Last Run 12 days ago
Open Website Hub
Awaiting install
Automations
Add.Incident.in.ServiceNow
Awaiting install
Block off time in my calendar
Last Run about 2 months ago
Book.my.calendar
Awaiting install
Convert my file to another format
Never ran
CreateStudioWebWorkflow
Last Run 14 days ago
Empty Recycling Bin
Awaiting install
GetAmazonSales
Search
Jarvis
Information
History
Jarvis
Version
2.5.11-dev.5
Folder
Shared
Picture in Picture
Run in PiP
Keyboard shortcut
Run automation with
Save
Run

Fig. 14



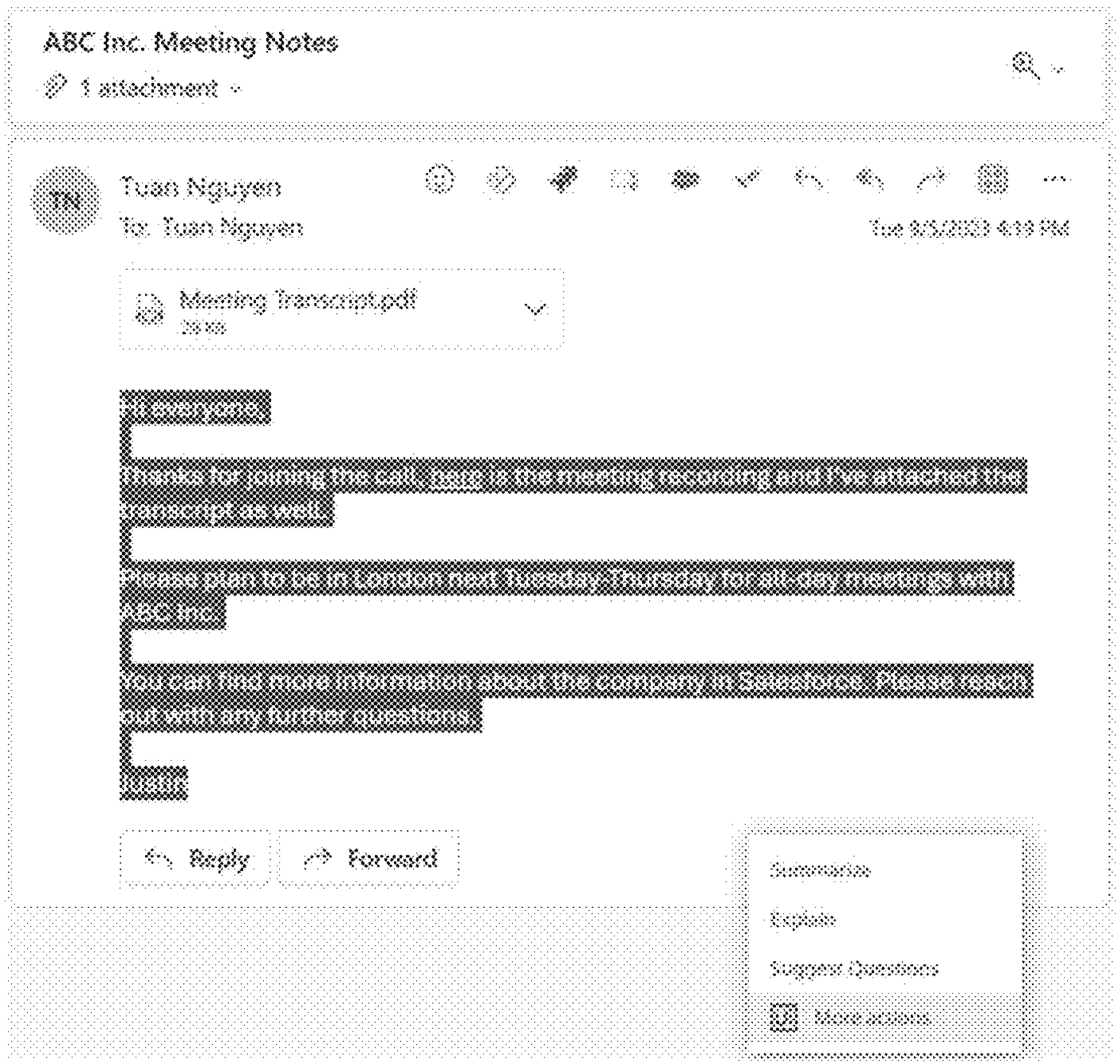
1500

Fig. 15


Java
UiPath
Welcome to UiPath's Jarvis!
Your AI-powered assistant to automating your everyday work tasks.
Find scenarios
"What kind of things can you do?"
Checking email
"Do I have any new work emails in my inbox? Can you give me a summary of them and suggest ways you could assist me?"
Help with scheduling
"Can you summarize my calendar for tomorrow? Do I have any meetings I need to prepare for?"
Travel help
"I need to fly to Toronto - can you create me a travel checklist?"
Image and Document Summarization
Paste in a screenshot of a slack conversation - I can summarize it and suggest actions
Candidate Qualifications
Drag in the PDF of a Resume and I can help you evaluate a candidate
Terms | Policy
What do you want to do?

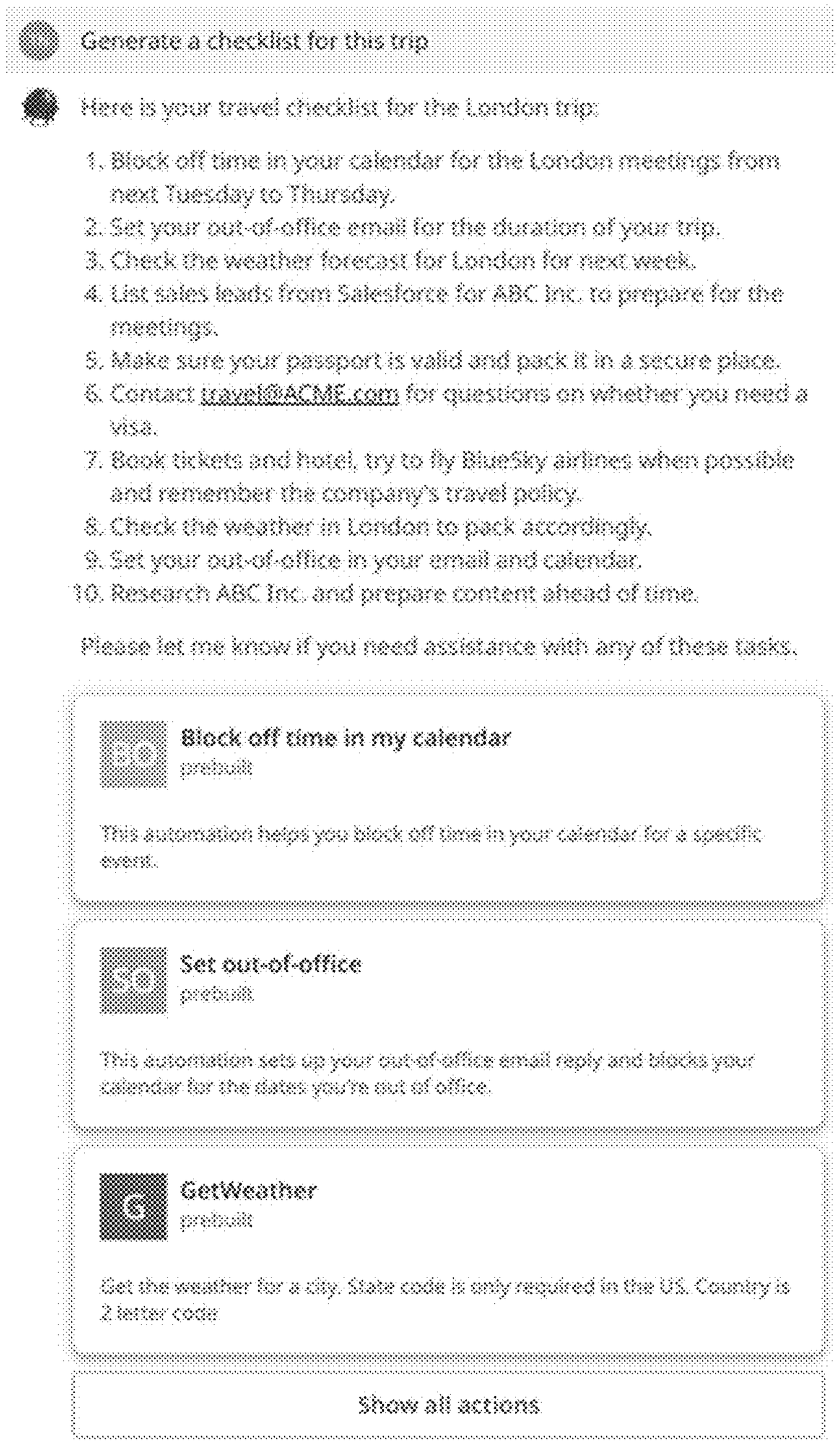
1600

Fig. 16


ABC Inc. Meeting Notes
1 attachment
TN
Tuan Nguyen
To: Tuan Nguyen
Tue 9/5/2023 4:19 PM
Meeting Transcript.pdf
Hi everyone,
Thanks for joining the call. Here is the meeting recording and I've attached the transcript as well.
Please plan to be in London next Tuesday-Thursday for all-day meetings with ABC Inc.
You can find more information about the company in Salesforce. Please reach out with any further questions.
Tuan
Reply
Forward
Summarize
Explain
Suggest Questions
Ask Copilot

1700

Fig. 17


Generate a checklist for this trip
Here is your travel checklist for the London trip:
1. Block off time in your calendar for the London meetings from next Tuesday to Thursday.
2. Set your out-of-office email for the duration of your trip.
3. Check the weather forecast for London for next week.
4. List sales leads from Salesforce for ABC Inc. to prepare for the meetings.
5. Make sure your passport is valid and pack it in a secure place.
6. Contact travel@ACME.com for questions on whether you need a visa.
7. Book tickets and hotel, try to fly BlueSky airlines when possible and remember the company's travel policy.
8. Check the weather in London to pack accordingly.
9. Set your out-of-office in your email and calendar.
10. Research ABC Inc. and prepare content ahead of time.
Please let me know if you need assistance with any of these tasks.
Block off time in my calendar
prebuilt
This automation helps you block off time in your calendar for a specific event.
Set out-of-office
prebuilt
This automation sets up your out-of-office email reply and blocks your calendar for the dates you're out of office.
GetWeather
prebuilt
Get the weather for a city. State code is only required in the US. Country is 2 letter code
Show all actions

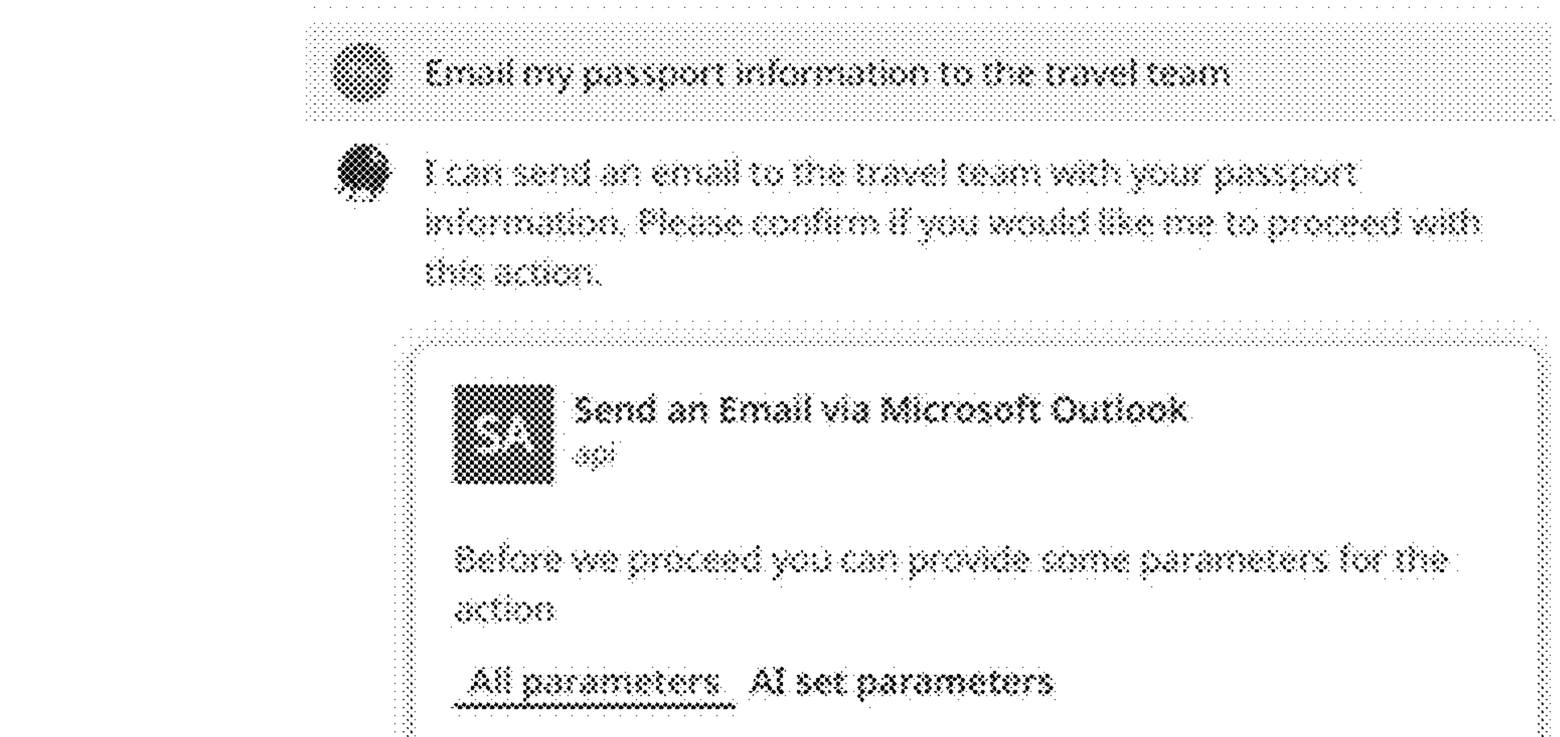
1800

Fig. 18


Analyzed file: passport.jpg type :image/jpeg
PAS / PASSPORT / PASSEPORT
An tAontas Eorpach
European Union
IP IRL
NÍ BHROIN
AOIBHEANN
EIREANNACH/IRISH 09 NOL/DEC 1975
F BAILE ÁTHA CLIATH/DUBLIN
27 AIB/APR 2015 26 AIB/APR 2020
Is my passport valid?
Based on the information provided in the passport image, your passport expired on April 26, 2020. Therefore, your passport is not valid for travel. You will need to renew your passport before your trip to London.

1900

Fig. 19


Email my passport information to the travel team
I can send an email to the travel team with your passport information. Please confirm if you would like me to proceed with this action.
Send an Email via Microsoft Outlook
api
Before we proceed you can provide some parameters for the action
All parameters
AI set parameters
Recipient
travel@ACME.com
To
Subject
Passport Information for Upcoming London Trip
Subject
Body
Dear Travel Team,
Here is my passport information for the upcoming trip to London:
Email Body
Cancel
Continue

2000

Fig. 20

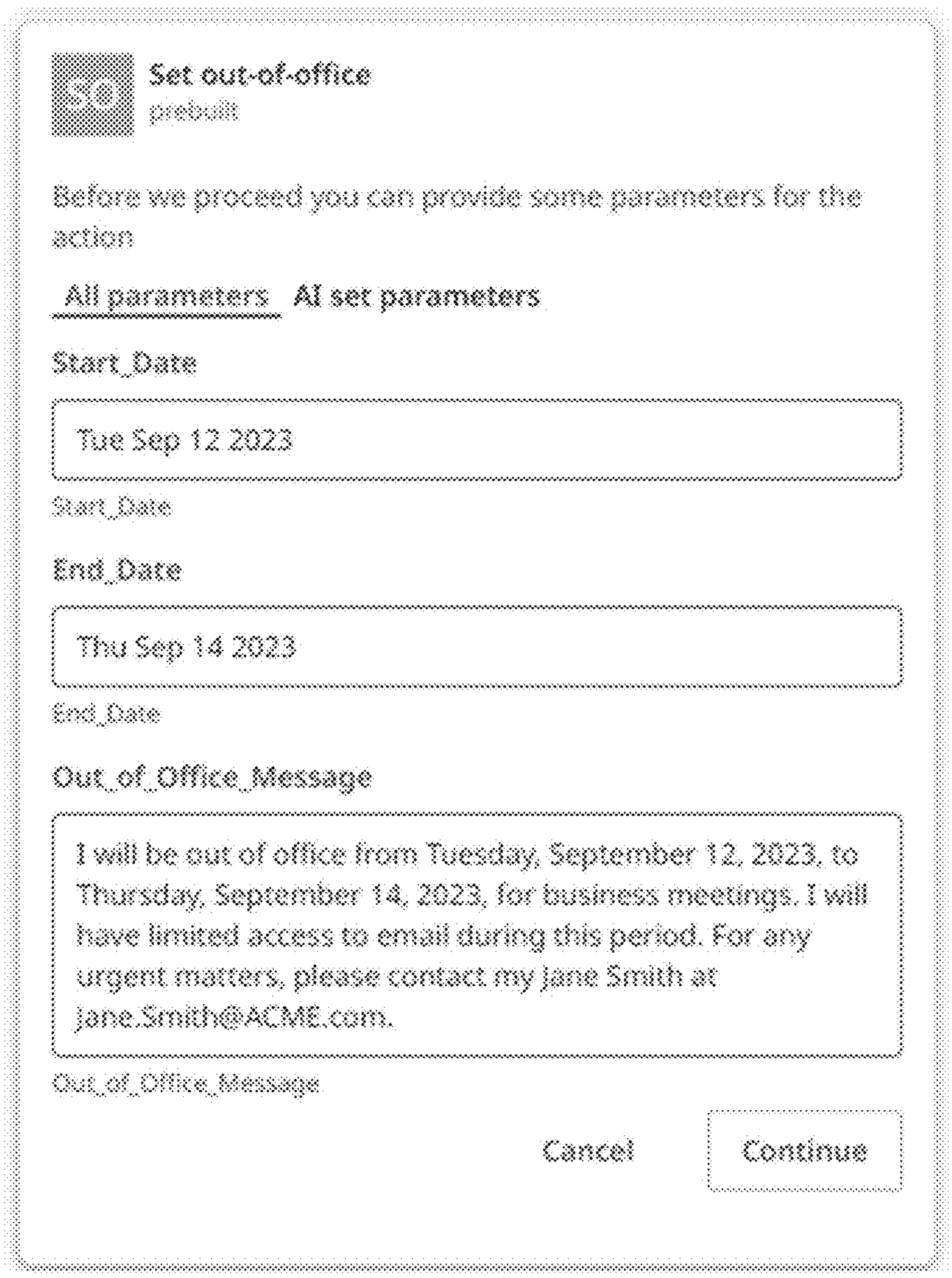
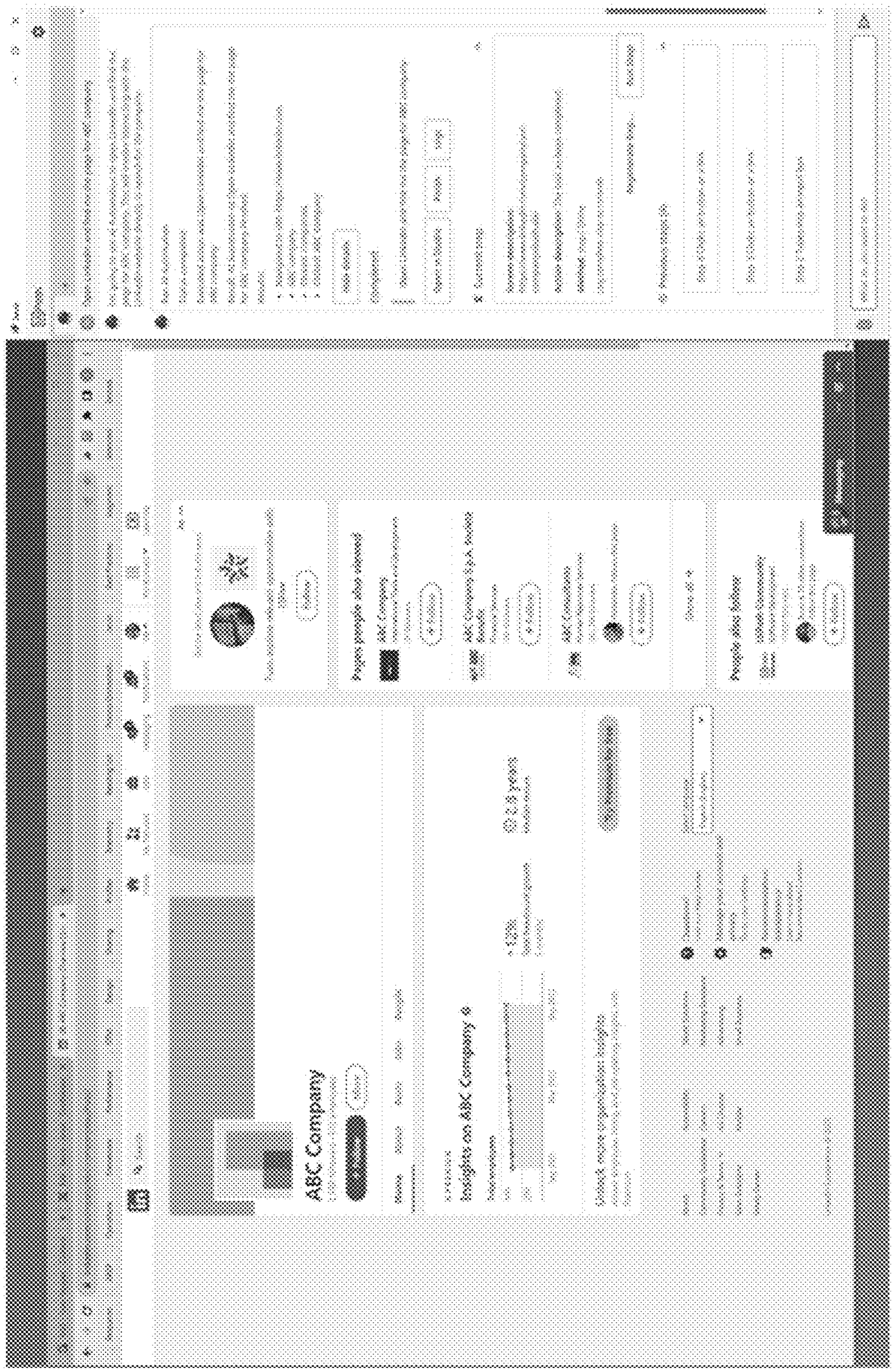

Set out-of-office
prebuilt
Before we proceed you can provide some parameters for the action
All parameters
AI set parameters
Start_Date
Tue Sep 12 2023
Start_Date
End_Date
Thu Sep 14 2023
End_Date
Out_of_Office_Message
I will be out of office from Tuesday, September 12, 2023, to Thursday, September 14, 2023, for business meetings. I will have limited access to email during this period. For any urgent matters, please contact my jane Smith at jane.Smith@ACME.com.
Out_of_Office_Message
Cancel
Continue

Fig. 21



DIGITAL ASSISTANT USING ROBOTIC PROCESS AUTOMATION

CROSS-REFERENCE OF RELATED APPLICATIONS

[0001] This application is a continuation-in-part of, and claims priority to, U.S. application Ser. No. 17/489,995, which was filed on Sep. 30, 2021. The subject matter of U.S. application Ser. No. 17/489,995 is incorporated in its entirety in this application.

FIELD

[0002] The present invention generally relates to robotic process automation (RPA), and more specifically, to a digital assistant using RPA.

BACKGROUND

[0003] Fully automated personal digital assistants do not currently exist. For example, digital assistants, such as Siri™ and Alexa™, are activated only when called upon by stating "Hello Siri™, please schedule a meeting for . . . ". These digital systems do not, however, operate in the background searching or waiting for triggers (or keywords) that would initiate workflows to execute one or more tasks without human involvement.

[0004] Accordingly, an improved method for providing a digital assistant using RPA may be beneficial.

SUMMARY

[0005] Certain embodiments of the present invention may provide solutions to the problems and needs in the art that have not yet been fully identified, appreciated, or solved by current personal digital assistant technologies. For example, some embodiments of the present invention pertain to fully automated personal digital assistant using RPA.

[0006] In an embodiment, a computer-implemented method for executing one or more digital assistant tasks using RPA includes activating a robot to monitor for one or more triggers while a user of a computing device is performing one or more tasks on the computing device, and identifying, by the robot, one of the one or more triggers for execution of the one or more digital assistant tasks. The method also includes loading, by the robot, a workflow associated with the identified one of the one or more triggers to complete one of the one or more digital assistant tasks. In another embodiment, the method includes creating, by the robot, a new workflow associated with the identified one of the one or more triggers based on bidirectional communication between the robot and a user of a computing device.

[0007] In another embodiment, a computer-implemented method for executing one or more digital assistant tasks using RPA includes populating, by a next generation digital assistant application, a drop down menu comprising a series of options when a user performs an action in a software application. The method includes populating, by the next generation digital assistant application, a side panel comprising a task to be performed by the next generation digital assistant application and a series of options for the user to select from. The method further includes updating, by the next generation digital assistant application, the side panel with additional options in response to a selected option from the series of options. The additional options are recommended actions for the user to select from in addition to the task performed by the next generation digital assistant application.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] In order that the advantages of certain embodiments of the invention will be readily understood, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments that are illustrated in the appended drawings. While it should be understood that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings, in which:

[0009] FIG. **1** is an architectural diagram illustrating an RPA system, according to an embodiment of the present invention.

[0010] FIG. **2** is an architectural diagram illustrating a deployed RPA system, according to an embodiment of the present invention.

[0011] FIG. **3** is an architectural diagram illustrating the relationship between a designer, activities, and drivers, according to an embodiment of the present invention.

[0012] FIG. **4** is an architectural diagram illustrating an RPA system, according to an embodiment of the present invention.

[0013] FIG. **5** is an architectural diagram illustrating a computing system configured to perform digital assistant using RPA, according to an embodiment of the present invention.

[0014] FIG. **6** is a flow diagram illustrating a method for executing one or more tasks using robotic processing automation (RPA), according to an embodiment of the present invention.

[0015] FIG. **7** is a graphical user interface (GUI) illustrating an email containing possible dates and times for the user to select, according to an embodiment of the present invention.

[0016] FIG. **8** is a GUI illustrating an email reply from an external user, according to an embodiment of the present invention.

[0017] FIG. **9** is a GUI illustrating a toolbar embedded with a JavaScript™ for monitoring one or more triggers, according to an embodiment of the present invention.

[0018] FIG. **10** is a GUI illustrating a popup window (or chat box), according to an embodiment of the present invention.

[0019] FIGS. **11**A-D is a GUI **1100**A-D illustrating a next generation digital assistant, according to an embodiment of the present invention.

[0020] FIG. **12** is a flow diagram illustrating method **1200** for invoking a next generation administrative assistant and executing a series of actions, according to an embodiment of the present invention.

[0021] FIG. **13** is a GUI illustrating setup and deployment of next generation digital assistant, according to an embodiment of the present invention.

[0022] FIG. **14** is a GUI illustrating launching next generation digital assistant when minimized, according to an embodiment of the present invention.

[0023] FIG. **15** is a GUI illustrating a first run start up experience, according to an embodiment of the present invention.

[0024] FIG. **16** is a GUI illustrating in-context interactions, according to an embodiment of the present invention.

[0025] FIG. **17** is a GUI illustrating a smart skill being executed by next generation digital assistant, according to an embodiment of the present invention.

[0026] FIG. **18** is a GUI illustrating a user uploading a context (e.g., a data file) using the side panel of FIGS. **11**A-D, according to an embodiment of the present invention.

[0027] FIG. **19** is a GUI illustrating an application programming interface (API) automation, according to an embodiment of the present invention.

[0028] FIG. **20** is a GUI illustrating prebuilt automation, according to an embodiment of the present invention.

[0029] FIG. **21** is a GUI illustrating AI automation, according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0030] Some embodiments pertain to a digital assistant using RPA. In an embodiment, the digital assistant may perform using artificial intelligence (AI) technology scheduling, expense report processing, time tracking, electronic messaging on behalf of the user, etc.

[0031] In some embodiment, a system includes a plurality of computing devices, where each computing system has a robot that acts as a digital assistant. This robot may receive instructions from a user of the computing device and perform the task requested by the user or may listen for triggers prior to performing the task associated with the trigger. The robot may be embedded within a toolbar of an operating system of each computing device, in some embodiments.

[0032] In embodiments where the robot is embedded into one or more applications or within the operating system, the robot may listen for triggers. Some examples of triggers are described below.

[0033] Calendaring

[0034] In an embodiment, when a user submits a request to schedule a meeting with another user of another computing device, the robot opens the calendar and communicates with the other computing device to setup and confirm the meeting. For example, the robot sends a calendar request in an email format to the email address of the user on the same or separate email systems. The request includes one or more meeting dates and times available for both the user and the other user. For example, the user may input a date and time for the meeting and identify the other user whom the meeting should be scheduled with.

[0035] In another example, the robot communicates directly with another robot of the other user by transmitting a calendar request. The other robot, upon receipt, may display a message to the other user in order for the other user to accept, deny or change the meeting date/time. Depending on the other user's response, the other robot may transmit a reply message, either accepting, denying, or changing the calendar request. Alternatively, the other robot may peruse the other user's schedule to confirm the meeting, deny the meeting and/or suggest an alternative date/time for the meeting.

[0036] In yet another example, the user requests the robot to schedule a meeting with the other user without specifying a date and time. In this example, the user's robot reviews the user's calendar prior to scheduling the meeting. Additionally, the user's robot may access a database comprising of previous meetings with the other user to identify a preferred date and time for the meeting. Upon identifying of the meeting, the user's robot communicates with the other user's robot to schedule the meeting, i.e., sends a calendar invite for a proposed date and time.

[0037] Expense Reports

[0038] In an embodiment, the user instructs the robot to generate an expense report associated with a previous trip. For example, the user uploads one or more receipts to the computing system, and in response, the robot scans the content of the uploaded receipts. During the scanning process, the robot parses out the date of the receipt, the description of the goods/services, and the associated price. The robot may open a expense report template and may input the parsed data to generate the expense report.

[0039] Flight Scheduling

[0040] In another embodiment, the user may instruct the robot to schedule a flight to a particular destination. In this embodiment, the robot, in response to the request, opens the user's calendar, travel policies (e.g., company travel rules), activity or enterprise graph (e.g., company's enterprise rule based system), to name a few. Based on the user's calendar, the robot will search for flight availability to the particular destination. The robot, upon identifying the flight, may display a message to the user to either accept or deny the proposed flight. In an further embodiment, the robot may provide a list of flight dates and times for the user to select. Depending on the user's response, the robot may schedule the flight for the destination. This embodiment is helpful when the user has a meeting with another user at the destination and needs to fly to the destination to attend the meeting. This embodiment may be executed after the calendaring invite.

[0041] Communications

[0042] In certain embodiments, the robot may be configured to perform additional tasks, such as reviewing communications, and based on keywords, notify the user that a response is required. In another embodiment, the robot may sort email communications in order of importance to the user. The metrics may determine the order of importance, which are preset by the user or are based on historical data compiled based on the user's actions.

[0043] Although some embodiments require input from the user in order for next generation digital assistant to generate an AI response, other embodiments may require very little input from the user. For example, the robot may capture data from previous inputs by the user, and may store this data in a machine learning (ML) model. The robot, by using the ML model, may continuously learn and improve when performing a given task.

[0044] In certain embodiments, when the robot searches a database storing the existing workflows, and is unable to find the workflow for a specific trigger or for the set of instructions received from the user, the robot creates a new workflow for the trigger or for the set of instructions received from the user. The new workflow is created using AI (e.g., generative AI, artificial narrow intelligence, artificial superintelligence, etc.). For example, the robot may look at existing workflows and combine one or more of the existing workflows to create a new workflow. In one example, one or more steps may be retrieved from a first workflow and one

or more steps may be retrieved from at least one additional workflow to create the new workflow. In some additional embodiments, the robot may receive a set of instructions from the user containing the steps. These steps may be used to create the workflow.

[0045] FIG. **1** is an architectural diagram illustrating an RPA system **100**, according to an embodiment of the present invention. RPA system **100** includes a designer **110** that allows a developer to design and implement workflows. Designer **110** may provide a solution for application integration, as well as automating third-party applications, administrative Information Technology (IT) tasks, and business IT processes. Designer **110** may facilitate development of an automation project, which is a graphical representation of a business process. Simply put, designer **110** facilitates the development and deployment of workflows and robots.

[0046] The automation project enables automation of rule-based processes by giving the developer control of the execution order and the relationship between a custom set of steps developed in a workflow, defined herein as "activities." One commercial example of an embodiment of designer **110** is UiPath Studio™. Each activity may include an action, such as clicking a button, reading a file, writing to a log panel, etc. In some embodiments, workflows may be nested or embedded.

[0047] Some types of workflows may include, but are not limited to, sequences, flowcharts, Finite State Machines (FSMs), and/or global exception handlers. Sequences may be particularly suitable for linear processes, enabling flow from one activity to another without cluttering a workflow. Flowcharts may be particularly suitable to more complex business logic, enabling integration of decisions and connection of activities in a more diverse manner through multiple branching logic operators. FSMs may be particularly suitable for large workflows. FSMs may use a finite number of states in their execution, which are triggered by a condition (i.e., transition) or an activity. Global exception handlers may be particularly suitable for determining workflow behavior when encountering an execution error and for debugging processes.

[0048] Once a workflow is developed in designer **110**, execution of business processes is orchestrated by conductor **120**, which orchestrates one or more robots **130** that execute the workflows developed in designer **110**. One commercial example of an embodiment of conductor **120** is UiPath Orchestrator™. Conductor **120** facilitates management of the creation, monitoring, and deployment of resources in an environment. Conductor **120** may act as an integration point with third-party solutions and applications.

[0049] Conductor **120** may manage a fleet of robots **130**, connecting and executing robots **130** from a centralized point. Types of robots **130** that may be managed include, but are not limited to, attended robots **132**, unattended robots **134**, development robots (similar to unattended robots **134**, but used for development and testing purposes), and non-production robots (similar to attended robots **132**, but used for development and testing purposes). Attended robots **132** are triggered by user events and operate alongside a human on the same computing system. Attended robots **132** may be used with conductor **120** for a centralized process deployment and logging medium. Attended robots **132** may help the human user accomplish various tasks, and may be triggered by user events. In some embodiments, processes cannot be started from conductor **120** on this type of robot and/or they cannot run under a locked screen. In certain embodiments, attended robots **132** can only be started from a robot tray or from a command prompt. Attended robots **132** should run under human supervision in some embodiments.

[0050] Unattended robots **134** run unattended in virtual environments (or bare-bones hardware) and can automate many processes. Unattended robots **134** may be responsible for remote execution, monitoring, scheduling, and providing support for work queues. Debugging for all robot types may be run in designer **110** in some embodiments. Both attended and unattended robots may automate various systems and applications including, but not limited to, mainframes, web applications, VMs, enterprise applications (e.g., those produced by SAP®, SalesForce®, Oracle®, etc.), and computing system applications (e.g., desktop and laptop applications, mobile device applications, wearable computer applications, cloud-based applications, etc.).

[0051] Conductor **120** may have various capabilities including, but not limited to, provisioning, deployment, configuration, queueing, monitoring, logging, and/or providing interconnectivity. Provisioning may include creating and maintenance of connections between robots **130** and conductor **120** (e.g., a web application). Deployment may include assuring the correct delivery of package versions to assigned robots **130** for execution. Configuration may include maintenance and delivery of robot environments and process configurations. Queueing may include providing management of queues and queue items. Monitoring may include keeping track of robot identification data and maintaining user permissions. Logging may include storing and indexing logs to a database (e.g., an SQL database) and/or another storage mechanism (e.g., ElasticSearch®, which provides the ability to store and quickly query large datasets). Conductor **120** may provide interconnectivity by acting as the centralized point of communication for third-party solutions and/or applications.

[0052] Robots **130** are execution agents that run workflows built in designer **110**. One commercial example of some embodiments of robot(s) **130** is UiPath Robots™ In some embodiments, robots **130** install the Microsoft Windows® Service Control Manager (SCM)-managed service by default. As a result, such robots **130** can open interactive Windows® sessions under the local system account, and have the rights of a Windows® service.

[0053] In some embodiments, robots **130** can be installed in a user mode. For such robots **130**, this means they have the same rights as the user under which a given robot **130** has been installed. This feature may also be available for High Density (HD) robots, which ensure full utilization of each machine at its maximum potential. In some embodiments, any type of robot **130** may be configured in an HD environment.

[0054] Robots **130** in some embodiments are split into several components, each being dedicated to a particular automation task. The robot components in some embodiments include, but are not limited to, SCM-managed robot services, user mode robot services, executors, agents, and command line. SCM-managed robot services manage and monitor Windows® sessions and act as a proxy between conductor **120** and the execution hosts (i.e., the computing systems on which robots **130** are executed). These services

are trusted with and manage the credentials for robots **130**. A console application is launched by the SCM under the local system.

[0055] User mode robot services in some embodiments manage and monitor Windows® sessions and act as a proxy between conductor **120** and the execution hosts. User mode robot services may be trusted with and manage the credentials for robots **130**. A Windows® application may automatically be launched if the SCM-managed robot service is not installed.

[0056] Executors may run given jobs under a Windows® session (i.e., they may execute workflows. Executors may be aware of per-monitor dots per inch (DPI) settings. Agents may be Windows® Presentation Foundation (WPF) applications that display the available jobs in the system tray window. Agents may be a client of the service. Agents may request to start or stop jobs and change settings. The command line is a client of the service. The command line is a console application that can request to start jobs and waits for their output.

[0057] Having components of robots **130** split as explained above helps developers, support users, and computing systems more easily run, identify, and track what each component is executing. Special behaviors may be configured per component this way, such as setting up different firewall rules for the executor and the service. The executor may always be aware of DPI settings per monitor in some embodiments. As a result, workflows may be executed at any DPI, regardless of the configuration of the computing system on which they were created. Projects from designer **110** may also be independent of browser zoom level in some embodiments. For applications that are DPI-unaware or intentionally marked as unaware, DPI may be disabled in some embodiments.

[0058] FIG. **2** is an architectural diagram illustrating a deployed RPA system **200**, according to an embodiment of the present invention. In some embodiments, RPA system **200** may be, or may be a part of, RPA system **100** of FIG. **1**. It should be noted that the client side, the server side, or both, may include any desired number of computing systems without deviating from the scope of the invention. On the client side, a robot application **210** includes executors **212**, an agent **214**, and a designer **216**. However, in some embodiments, designer **216** may not be running on computing system **210**. Executors **212** are running processes. Several business projects may run simultaneously, as shown in FIG. **2**. Agent **214** (e.g., a Windows ° service) is the single point of contact for all executors **212** in this embodiment. All messages in this embodiment are logged into conductor **230**, which processes them further via database server **240**, indexer server **250**, or both. As discussed above with respect to FIG. **1**, executors **212** may be robot components.

[0059] In some embodiments, a robot represents an association between a machine name and a username. The robot may manage multiple executors at the same time. On computing systems that support multiple interactive sessions running simultaneously (e.g., Windows ° Server 2012), multiple robots may be running at the same time, each in a separate Windows ° session using a unique username. This is referred to as HD robots above.

[0060] Agent **214** is also responsible for sending the status of the robot (e.g., periodically sending a "heartbeat" message indicating that the robot is still functioning) and downloading the required version of the package to be executed. The communication between agent **214** and conductor **230** is always initiated by agent **214** in some embodiments. In the notification scenario, agent **214** may open a WebSocket channel that is later used by conductor **230** to send commands to the robot (e.g., start, stop, etc.).

[0061] On the server side, a presentation layer (web application **232**, Open Data Protocol (OData) Representative State Transfer (REST) Application Programming Interface (API) endpoints **234**, and notification and monitoring **236**), a service layer (API implementation/business logic **238**), and a persistence layer (database server **240** and indexer server **250**) are included. Conductor **230** includes web application **232**, OData REST API endpoints **234**, notification and monitoring **236**, and API implementation/business logic **238**. In some embodiments, most actions that a user performs in the interface of conductor **230** (e.g., via browser **220**) are performed by calling various APIs. Such actions may include, but are not limited to, starting jobs on robots, adding/removing data in queues, scheduling jobs to run unattended, etc. without deviating from the scope of the invention. Web application **232** is the visual layer of the server platform. In this embodiment, web application **232** uses Hypertext Markup Language (HTML) and JavaScript (JS). However, any desired markup languages, script languages, or any other formats may be used without deviating from the scope of the invention. The user interacts with web pages from web application **232** via browser **220** in this embodiment in order to perform various actions to control conductor **230**. For instance, the user may create robot groups, assign packages to the robots, analyze logs per robot and/or per process, start and stop robots, etc.

[0062] In addition to web application **232**, conductor **230** also includes service layer that exposes OData REST API endpoints **234**. However, other endpoints may be included without deviating from the scope of the invention. The REST API is consumed by both web application **232** and agent **214**. Agent **214** is the supervisor of one or more robots on the client computer in this embodiment.

[0063] The REST API in this embodiment covers configuration, logging, monitoring, and queueing functionality. The configuration endpoints may be used to define and configure application users, permissions, robots, assets, releases, and environments in some embodiments. Logging REST endpoints may be used to log different information, such as errors, explicit messages sent by the robots, and other environment-specific information, for instance. Deployment REST endpoints may be used by the robots to query the package version that should be executed if the start job command is used in conductor **230**. Queueing REST endpoints may be responsible for queues and queue item management, such as adding data to a queue, obtaining a transaction from the queue, setting the status of a transaction, etc.

[0064] Monitoring REST endpoints may monitor web application **232** and agent **214**. Notification and monitoring API **236** may be REST endpoints that are used for registering agent **214**, delivering configuration settings to agent **214**, and for sending/receiving notifications from the server and agent **214**. Notification and monitoring API **236** may also use WebSocket communication in some embodiments.

[0065] The persistence layer includes a pair of servers in this embodiment—database server **240** (e.g., a SQL server) and indexer server **250**. Database server **240** in this embodiment stores the configurations of the robots, robot groups,

associated processes, users, roles, schedules, etc. This information is managed through web application **232** in some embodiments. Database server **240** may manages queues and queue items. In some embodiments, database server **240** may store messages logged by the robots (in addition to or in lieu of indexer server **250**).

[0066] Indexer server **250**, which is optional in some embodiments, stores and indexes the information logged by the robots. In certain embodiments, indexer server **250** may be disabled through configuration settings. In some embodiments, indexer server **250** uses ElasticSearch®, which is an open source project full-text search engine. Messages logged by robots (e.g., using activities like log message or write line) may be sent through the logging REST endpoint (s) to indexer server **250**, where they are indexed for future utilization.

[0067] FIG. **3** is an architectural diagram illustrating the relationship **300** between a designer **310**, activities **320**, **330**, drivers **340**, and AI/ML models **350**, according to an embodiment of the present invention. Per the above, a developer uses designer **310** to develop workflows that are executed by robots. Workflows may include user-defined activities **320** and UI automation activities **330**. User-defined activities **320** and/or UI automation activities **330** may call one or more AI/ML models **350** in some embodiments, which may be located locally to the computing system on which the robot is operating and/or remotely thereto. Some embodiments are able to identify non-textual visual components in an image, which is called computer vision (CV) herein. Some CV activities pertaining to such components may include, but are not limited to, click, type, get text, hover, element exists, refresh scope, highlight, etc. Click in some embodiments identifies an element using CV, optical character recognition (OCR), fuzzy text matching, and multi-anchor, for example, and clicks it. Type may identify an element using the above and types in the element. Get text may identify the location of specific text and scan it using OCR. Hover may identify an element and hover over it. Element exists may check whether an element exists on the screen using the techniques described above. In some embodiments, there may be hundreds or even thousands of activities that can be implemented in designer **310**. However, any number and/or type of activities may be available without deviating from the scope of the invention.

[0068] UI automation activities **330** are a subset of special, lower level activities that are written in lower level code (e.g., CV activities) and facilitate interactions with the screen. UI automation activities **330** facilitate these interactions via drivers **340** and/or AI/ML models **350** that allow the robot to interact with the desired software. For instance, drivers **340** may include OS drivers **342**, browser drivers **344**, VM drivers **346**, enterprise application drivers **348**, etc. One or more of AI/ML models **350** may be used by UI automation activities **330** in order to determine perform interactions with the computing system. In some embodiments, AI/ML models **350** may augment drivers **340** or replace them completely. Indeed, in certain embodiments, drivers **340** are not included.

[0069] Drivers **340** may interact with the OS at a low level looking for hooks, monitoring for keys, etc. They may facilitate integration with Chrome®, IE®, Citrix®, SAP®, etc. For instance, the "click" activity performs the same role in these different applications via drivers **340**.

[0070] FIG. **4** is an architectural diagram illustrating an RPA system **400**, according to an embodiment of the present invention. In some embodiments, RPA system **400** may be or include RPA systems **100** and/or **200** of FIGS. **1** and/or **2**. RPA system **400** includes multiple client computing systems **410** running robots. Computing systems **410** are able to communicate with a conductor computing system **420** via a web application running thereon. Conductor computing system **420**, in turn, is able to communicate with a database server **430** and an optional indexer server **440**.

[0071] With respect to FIGS. **1** and **3**, it should be noted that while a web application is used in these embodiments, any suitable client/server software may be used without deviating from the scope of the invention. For instance, the conductor may run a server-side application that communicates with non-web-based client software applications on the client computing systems.

[0072] FIG. **5** is an architectural diagram illustrating a computing system **500** configured to perform digital assistant using RPA, according to an embodiment of the present invention. In some embodiments, computing system **500** may be one or more of the computing systems depicted and/or described herein. Computing system **500** includes a bus **505** or other communication mechanism for communicating information, and processor(s) **510** coupled to bus **505** for processing information. Processor(s) **510** may be any type of general or specific purpose processor, including a Central Processing Unit (CPU), an Application Specific Integrated Circuit (ASIC), a Field Programmable Gate Array (FPGA), a Graphics Processing Unit (GPU), multiple instances thereof, and/or any combination thereof. Processor (s) **510** may also have multiple processing cores, and at least some of the cores may be configured to perform specific functions. Multi-parallel processing may be used in some embodiments. In certain embodiments, at least one of processor(s) **510** may be a neuromorphic circuit that includes processing elements that mimic biological neurons. In some embodiments, neuromorphic circuits may not require the typical components of a Von Neumann computing architecture.

[0073] Computing system **500** further includes a memory **515** for storing information and instructions to be executed by processor(s) **510**. Memory **515** can be comprised of any combination of Random Access Memory (RAM), Read Only Memory (ROM), flash memory, cache, static storage such as a magnetic or optical disk, or any other types of non-transitory computer-readable media or combinations thereof. Non-transitory computer-readable media may be any available media that can be accessed by processor(s) **510** and may include volatile media, non-volatile media, or both. The media may also be removable, non-removable, or both.

[0074] Additionally, computing system **500** includes a communication device **520**, such as a transceiver, to provide access to a communications network via a wireless and/or wired connection. In some embodiments, communication device **520** may be configured to use Frequency Division Multiple Access (FDMA), Single Carrier FDMA (SC-FDMA), Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), Orthogonal Frequency Division Multiplexing (OFDM), Orthogonal Frequency Division Multiple Access (OFDMA), Global System for Mobile (GSM) communications, General Packet Radio Service (GPRS), Universal Mobile Telecommunications Sys-

tem (UMTS), cdma2000, Wideband CDMA (W-CDMA), High-Speed Downlink Packet Access (HSDPA), High-Speed Uplink Packet Access (HSUPA), High-Speed Packet Access (HSPA), Long Term Evolution (LTE), LTE Advanced (LTE-A), 802.11x, Wi-Fi, Zigbee, Ultra-Wide-Band (UWB), 802.16x, 802.15, Home Node-B (HnB), Bluetooth, Radio Frequency Identification (RFID), Infrared Data Association (IrDA), Near-Field Communications (NFC), fifth generation (5G), New Radio (NR), any combination thereof, and/or any other currently existing or future-implemented communications standard and/or protocol without deviating from the scope of the invention. In some embodiments, communication device **520** may include one or more antennas that are singular, arrayed, phased, switched, beamforming, beamsteering, a combination thereof, and or any other antenna configuration without deviating from the scope of the invention.

[0075] Processor(s) **510** are further coupled via bus **505** to a display **525**, such as a plasma display, a Liquid Crystal Display (LCD), a Light Emitting Diode (LED) display, a Field Emission Display (FED), an Organic Light Emitting Diode (OLED) display, a flexible OLED display, a flexible substrate display, a projection display, a **4**K display, a high definition display, a Retina ° display, an In-Plane Switching (IPS) display, or any other suitable display for displaying information to a user. Display **525** may be configured as a touch (haptic) display, a three dimensional (3D) touch display, a multi-input touch display, a multi-touch display, etc. using resistive, capacitive, surface-acoustic wave (SAW) capacitive, infrared, optical imaging, dispersive signal technology, acoustic pulse recognition, frustrated total internal reflection, etc. Any suitable display device and haptic I/O may be used without deviating from the scope of the invention.

[0076] A keyboard **530** and a cursor control device **535**, such as a computer mouse, a touchpad, etc., are further coupled to bus **505** to enable a user to interface with computing system **500**. However, in certain embodiments, a physical keyboard and mouse may not be present, and the user may interact with the device solely through display **525** and/or a touchpad (not shown). Any type and combination of input devices may be used as a matter of design choice. In certain embodiments, no physical input device and/or display is present. For instance, the user may interact with computing system **500** remotely via another computing system in communication therewith, or computing system **500** may operate autonomously.

[0077] Memory **515** stores software modules that provide functionality when executed by processor(s) **510**. The modules include an operating system **540** for computing system **500**. The modules further include a RPA digital assistant module **545** that is configured to perform all or part of the processes described herein or derivatives thereof. Computing system **500** may include one or more additional functional modules **550** that include additional functionality.

[0078] One skilled in the art will appreciate that a "system" could be embodied as a server, an embedded computing system, a personal computer, a console, a personal digital assistant (PDA), a cell phone, a tablet computing device, a quantum computing system, or any other suitable computing device, or combination of devices without deviating from the scope of the invention. Presenting the above-described functions as being performed by a "system" is not intended to limit the scope of the present invention in any way, but is intended to provide one example of the many embodiments of the present invention. Indeed, methods, systems, and apparatuses disclosed herein may be implemented in localized and distributed forms consistent with computing technology, including cloud computing systems. The computing system could be part of or otherwise accessible by a local area network (LAN), a mobile communications network, a satellite communications network, the Internet, a public or private cloud, a hybrid cloud, a server farm, any combination thereof, etc. Any localized or distributed architecture may be used without deviating from the scope of the invention.

[0079] It should be noted that some of the system features described in this specification have been presented as modules, in order to emphasize their implementation independence more particularly. For example, a module may be implemented as a hardware circuit comprising custom very large scale integration (VLSI) circuits or gate arrays, off-the-shelf semiconductors such as logic chips, transistors, or other discrete components. A module may also be implemented in programmable hardware devices such as field programmable gate arrays, programmable array logic, programmable logic devices, graphics processing units, or the like.

[0080] A module may also be at least partially implemented in software for execution by various types of processors. An identified unit of executable code may, for instance, include one or more physical or logical blocks of computer instructions that may, for instance, be organized as an object, procedure, or function. Nevertheless, the executables of an identified module need not be physically located together, but may include disparate instructions stored in different locations that, when joined logically together, comprise the module and achieve the stated purpose for the module. Further, modules may be stored on a computer-readable medium, which may be, for instance, a hard disk drive, flash device, RAM, tape, and/or any other such non-transitory computer-readable medium used to store data without deviating from the scope of the invention.

[0081] Indeed, a module of executable code could be a single instruction, or many instructions, and may even be distributed over several different code segments, among different programs, and across several memory devices. Similarly, operational data may be identified and illustrated herein within modules, and may be embodied in any suitable form and organized within any suitable type of data structure. The operational data may be collected as a single data set, or may be distributed over different locations including over different storage devices, and may exist, at least partially, merely as electronic signals on a system or network.

[0082] FIG. **6** is a flow diagram illustrating a method **600** for executing one or more tasks using robotic processing automation (RPA), according to an embodiment of the present invention. The one or more tasks include calendaring, travel scheduling, expense report generation, and message (e.g., electronic mail) monitoring, to name a few. In some embodiment, method **600** begins at **605** with the user assigning a workflow, such as a personal digital assistant workflow, to a robot. The assignment of the workflow allows for the monitoring of one or more events. In some embodiments, the one or more triggers may be defined as, or may include, scheduling a meeting, scheduling a flight, scheduling a hotel, generating an expense report, re-organizing communications, or any event that would cause a robot to

perform an automated task on behalf of the user. At **610**, the robot identifies the one or more triggers during the monitoring of the one or more events. In this embodiment, the robot continuously monitors for one or more triggers to identify the one or more triggers. At **615**, the robot loads a workflow associated with the one or more identified triggers, and at **620**, executes the loaded workflow to perform one or more tasks associated with the one or more triggers. The loaded workflow may include a calendaring workflow, a travel scheduling workflow, an expense report generation workflow, and a message (e.g., electronic mail) monitoring workflow, to name a few.

[0083] The process steps performed in FIG. **6** may be performed by a computer program, encoding instructions for the processor(s) to perform at least part of the process(es) described in FIG. **6**, in accordance with embodiments of the present invention. The computer program may be embodied on a non-transitory computer-readable medium. The computer-readable medium may be, but is not limited to, a hard disk drive, a flash device, RAM, a tape, and/or any other such medium or combination of media used to store data. The computer program may include encoded instructions for controlling processor(s) of a computing system (e.g., processor(s) **510** of computing system **500** of FIG. **5**) to implement all or part of the process steps described in FIG. **6**, which may also be stored on the computer-readable medium.

[0084] The computer program can be implemented in hardware, software, or a hybrid implementation. The computer program can be composed of modules that are in operative communication with one another, and which are designed to pass information or instructions to display. The computer program can be configured to operate on a general purpose computer, an ASIC, or any other suitable device.

[0085] Calendaring Workflow

[0086] In some embodiments, when the calendaring workflow is executed, the robot may populate a graphical user interface (GUI) for the user to enter calendaring information. This information may include a meeting subject line, a proposed date, and a proposed time. The robot uses the information inputted by the user and transmits the request to another user's email account. For example, the robot sends a calendar request in an email format, allowing the other user to accept, deny, or modify the proposed meeting. In some embodiments, however, prior to submitting the calendar request, the robot generate a GUI with the proposed calendar request for the user to accept, deny or modify. In another embodiment, the robot proceeds with transmitting the request with the other user or the other user's robot without the user's involvement.

[0087] In embodiments where the user did not input a proposed date and time for the proposed meeting, the robot may review user's electronic calendar to determine user availability. In some further embodiments, the robot may access a database, which includes previous meetings with the other user, other meetings with different users, and so forth. By accessing the database, the robot may use a ML algorithm to determine the users preferred dates/times for conducting the meeting. For example, the robot determines that the user prefers to schedule meetings in the mornings or late afternoons. Using this data, the robot automatically determines a proposed date and time for the meeting, and then sends the request to the other user's email, allowing the other user to accept, deny, or modify.

[0088] In yet another embodiment, the robot may communicate with another robot associated with the other user. In this embodiment, the robot sends the calendar request directly to the robot via a separate communications channel or via email. In embodiments that use emails, the other user's robot may scan the inbox for calendar requests. Regardless of the communication method, the other user's robot may automatically accept, deny, or modify the invite based on the other's user's schedule.

[0089] In yet an alternative embodiment, the robot may support automatically coordinating a meeting between the user and one or more external users. Such an embodiment would replace functionalities in software applications such as Calendy™ In this embodiment, the user finds it difficult to coordinate or schedule a meeting with people outside of (or external to) his or her organization. In this case, the robot retrieves meeting times that are available for each external user. Upon retrieving the meeting times, the robot constructs an email containing dates and times for the meeting, and sends the email to the user, and in some embodiments, also to the external users. See FIG. **7**, which is a GUI **700** illustrating an email containing possible dates and times for the user to select, according to an embodiment of the present invention. In GUI **700**, the email instructs the user, and in some embodiments the external users, to respond to the email with an order of preference for dates and times.

[0090] The robot, using a listener module, may listen for a reply to this email and may coordinate the meeting in the background. After a predefined period of time has elapsed, or all responses have been handled or retrieved, the robot prompts the user with a calendar invite that is the best for everyone, leaving the user with the option to hit send.

[0091] To handle the response, the robot may, in order to intelligently monitor who has responded, require some orchestration to keep track of who has been sent what, and who has responded. This might require orchestrator queues. For example, the robot may keep a log of who has responded and the dates/times that are available of the user and the external user. This may, however, be handled locally if needed. For example, a temporary Excel™ file may be used to handle the state of each reply or non-reply.

[0092] In some embodiments, the listener module detects when an email has arrived in the user's inbox. In such an embodiment, the robot parses the response (e.g., email), and compares the response to the originally suggested dates/times. The robot then prompts the user with a created calendar invite for the user to send. See FIG. **8**, which is a GUI **800** illustrating an email reply from an external user, according to an embodiment of the present invention.

TABLE 1

System Requirements for Calendaring

	Requirement
1	Determine available meeting times for all internal users.
2	Contact external users with determined internal availability by sending a polished email
3	Track who has responded to the email sent, and begin scheduling based on either: Everyone has responded Some threshold time has been met (e.g., 12 hours)
4	Create a draft calendar invite, based on all responses, ready to hit send (e.g., for the user's final sign off)

[0093] Travel Scheduling Workflow

[0094] In some embodiments, the robot upon finalizing the schedule on the user's calendar scans the calendared schedule for travel related information. This may include the location of the meeting. For example, when the robot identifies a location outside of the user's home or office location, the robot may load the travel scheduling workflow. Under this workflow, the robot may search for transportation by way of air, rail, or vehicle, and provide the user with options. In another embodiment, the robot may use the user's travel preferences to reserve a flight. For example, the robot identifies, from the user's travel preferences, the preferred airline and hotel. Using this information, the robot access one or more travel webpages and populates the appropriate fields on the one or more travel webpages. The robot may then launch a GUI showing the populated information, allowing the user to secure the reservation. In another embodiment, the robot may secure the reservation without user involvement or input.

[0095] Expense Report Generation Workflow

[0096] In some embodiments, the user may upload one or more receipts to a receipt folder. The robot, which runs in the background, may detect the uploaded one or more receipts in the receipt folder. The robot scans through each receipt and parses through information contained therein. For example, the parsed information includes vendor information, description of goods or services rendered, and amounts for each item in the receipt, to name a few. Using this parsed information, the robot identifies categories to which the receipts apply to. For example, using the parsed information, the robot identifies whether the receipts relate to a business travel trip. In this example, all receipts related to the business travel trip are sorted by the robot. The robot may then, using optical character recognition, create an expense report and insert the parsed information into the expense report.

[0097] In another embodiments, when a user selects the expense report workflow, the robot retrieves the expense report workflow. The robot populates a GUI, allowing the user to upload receipts that have been previously scanned. In another embodiment, the GUI may allow the user to scan and upload the receipts for the robot to analyze. The receipts may be saved in a database. Similar to the above embodiment, once the receipts are saved in the database, the robot access the receipts and parses the data from the receipt. Data includes vendor information, description of goods or services rendered, and amounts for each item in the receipt, to name a few. The robot retrieves an expense report, and populates the fields in the expense report with the data that was parsed. Lastly, the expense report is saved for the user's review and final approval.

[0098] Regardless of the embodiment, the robot may send a message to the user for approval. In another embodiment, the robot may generate the expense report without the user's involvement.

[0099] Communication Monitoring Workflow

[0100] In some embodiments, the robot continuously monitors communications that are sent from, and received by, the user. The robot, for example, may monitor communications on any medium. This may include email communication, Slack™ communication, iMessaging™ and so forth.

[0101] As messages are received by the computing device (or mobile device), the robot, which runs in the background, scans through the contents, and in some embodiments, the subject line of the message. Using this scanned information, the robot may identify communications that should have a higher priority. For example, the robot scans the body of the message and identifies any response deadlines. After compiling a list of response deadlines, the robot sorts the communications in order, and assigns a priority number. The robot may then generate a GUI that includes the communications and ranks them in order of importance for the user to select from.

[0102] Depending on the embodiment, prior to re-organizing the communications, the robot may send a request to the user for approval or denial of the proposed reorganized communications. In other embodiments, the robot may re-organize the communications without user involvement.

[0103] FIG. **9** is a GUI **900** illustrating a toolbar **905** embedded with a JavaScript™ for monitoring one or more triggers, according to an embodiment of the present invention. In this embodiment, the JavaScript™ causes the robot to monitor for one or more triggers when a user is accessing one or more applications. The JavaScript™ may be enabled at startup of the computing device, or in some embodiments, may be activated by the user. In those embodiments, the user may select pop-up menu **910** from toolbar **905**, which includes an activation button **915** and a deactivation button **920**. When the user presses on activation button **915**, the robot loads a workflow that will be monitoring for one or more triggers (or events). When the user presses on deactivation button, the robot ceases all monitoring of the one or more triggers.

[0104] It should be appreciated that this embodiment is not limited to JavaScript™ but other technologies such as .Net may be used, or may be used in conjunction with or more other technologies.

[0105] In another embodiment, and although not illustrated, toolbar **905** may receive an input from the user instructing the robot to execute a process. The input may be received via text input or voice input. In an alternative embodiment, other media (e.g., video, documents, images, etc.) may be used to instruct the robot. In embodiments with voice input, the robot uses voice to text recognition in order to process the voice input. The voice input may be for a specific instruction, i.e., "please schedule a meeting for July 8 at 10 am with John Doe at john.doe@domain.com". Regardless of whether the input received is a voice input or a text input, the robot loads the workflow to begin monitoring for one or more triggers based on the received input. In some embodiments, there may be a database that includes a plurality of stored inputs that are associated with a corresponding workflow. This way, when an input is received, the input is compared against the stored inputs allowing the robot to identify the appropriate workflow.

[0106] FIG. **10** is a GUI **1000** illustrating a popup window (or chat box) **1005**, according to an embodiment of the present invention. In some embodiments, the robot listens to or monitors for one or more triggers. When a trigger is identified, the robot causes popup window **1005** to launch within an application or anywhere on the operating system. Popup window **1005** may include an input section **1010**, a submit section **1015**, and a dialogue box **1020**. In some embodiments, input section **1010** receives voice input or text input. Submit section **1015** may submit the input from input

section **1010** for robot to process. Dialogue box **1020** may display the submitted input from the user and the response from the robot.

[0107] Popup window **1005** is configured to communicate between the user and the robot a plurality of instructions, each of which may be associated with different triggers. In one example, when robot causes popup window **1005** to launch, robot may communicate a request to the user. The request may be shown through dialogue box **1020**. Further, the request may be for an action associated with an identified trigger from the one or more triggers. The user may respond via input **1010** using text or voice instructing the robot to proceed. In an alternative embodiment, other media (e.g., video, documents, images, etc.) may be used to instruct the robot. The robot, upon receipt of the instructions, loads and executes the appropriate workflow.

[0108] In some embodiments, the user may instruct via input section **1010** the robot to monitor for a predefined set of triggers. In this embodiment, the robot loads the appropriate workflows for monitoring purposes. This way, the user is not constantly inundated with notification messages from the robot and only receives notification messages of interest to the user.

[0109] Popup window **1000** may include additional features such as a button to silence notifications from the robot, a button for the user to select only those notifications that the user deems important, or an upload button that allows the user to upload a media file instructing the robot to execute a workflow or a set of instructions.

[0110] FIGS. **11**A-D is a GUI **1100**A-D illustrating a next generation digital assistant, according to an embodiment of the present invention. In GUI **1100**, the next generation digital assistant helps the user walk through a process by way of recommendations, consuming context on a webpage, and also allows the user to use functionality such as smart skills. Smart skills may be defined as generative AI skills that have been predefined (e.g., summarized, explain, Q&A, etc.). A smart skill may include the user uploading a document and the user asking the next generation digital assistant a question. On the back end, the request is transposed so next generation administrative assistant interprets the question as planning the next steps for the user. Next generation administrative assistant is configured to determine if the request falls into a predefined bucket, and if so, next generation administrative assistant utilizes one of the buckets to prepare a response. In short, the user may by voice or data input instruct the next generation digital assistant to perform a task, causing the next generation digital assistant to perform the task.

[0111] In another embodiment, next generation digital assistant may provide additional recommendations (or the next best step) upon completion of the task. For example, if the next generation digital assistant was instructed to book travel, then the next generation digital assistant would recommend the user with the next best step (i.e., book a car rental and/or accommodations).

[0112] Invoking and Execution of Next Generation Digital Assistant

[0113] In some embodiments, next generation digital assistant may be invoked through context within a webpage. For example, when the user highlights text on the webpage, an option drop down is provided to the user. See, for example, item numeral (drop down) **1105**, which shows next generation digital assistant providing a series of actions such as reply, remind, summarize and paste, etc. In another embodiment, a 'read page' button is available as an option to the user, and upon clicking on the 'read page' button, next generation digital assistant may assist the user with the recommendations.

[0114] Upon selection on the reply (e.g., action available in the list), the context is sent to next generation digital assistant (i.e., a backend prompt). Next generation digital assistant then pops up a side panel **1110**, and performs the selected action.

[0115] Under side panel **1110**, a reply to be generated in response to the selected text is populated. This allows the user to modify the generated reply using options such as casual, neural, formal, brief, etc. See, for example, FIG. **11**B. The reply is generated using smart skills such as a generative pretrained transformer (GPT). In this embodiment, a user is able to modify an output generated by the GPT or Smart skills.

[0116] In some embodiments, there are two methods for performing the selection action. For example, the first action is triggered by the user. This may include highlighting text or selecting a screen and sending this information to the next generation administrative assistant. Another embodiment includes automatic, i.e., capturing the active screen automatically. Let's say for example the user typed in, 'can you help me plan for this trip', the captured text is sent to next generation administrative assistant for analysis.

[0117] Once the user selects the option, the next generation digital assistant recommends a next action under side panel **1110** (e.g., 'Find a similar Candidate' and 'Open Workday and Create candidate'). See, for example, FIG. **11**C. The recommended actions are available as automated response, and based on the context of the process being performed by the user, next generation digital assistant recommends the action as next steps to the user. See also FIG. **11**C. In some embodiments, next generation digital assistant considers multiple factors—personal context (e.g., user's role), external data providers (e.g., travel policy or knowledge base), etc. For purposes of explanation, next generation digital assistant may look at two types of recommendations. First type being a direct recommendation, i.e., where the user submits a request for what he or she can do, and next generation digital assistant providing a recommendation for said request. The second type being next best step, i.e., where the user requests for a travel check list, and upon next generation digital assistant completing the travel check list, next generation digital assistant also providing or asking the user if they would like flight availability for a particular destination based on the travel checklist.

[0118] In some embodiments, the user select any of the recommended actions, next generation digital assistant requests the user to input required arguments to execute the process before proceeding with 'Run'. In an embodiment, next generation digital assistant infers the input values to the arguments with the context it has related to the process and/or page. After executing the recommended action by the available automation, next generation digital assistant outputs additional next steps. In some additional embodiments, next generation digital assistant updates the user on successful execution of the task and allows the users to view the updates. See FIG. **11**D.

[0119] Intent Processor

[0120] When a user sends a request to next generation digital assistant (see FIG. **11**A-D, side panel **1110**), intent

processor determines how next generation digital assistant should respond to the request. For example, intent processor may determine whether to just respond to the user using generative AI or other smart skills. In another example, intent processor may determine whether to run an automation or provide a series of recommendations (according to relevancy) to the users if the intent processor is unsure about the request. In this example, if the user requests 'Run the open workday—Pick a candidate automation for me', intent processor may determine that the user is requesting for an automation to be executed, and next generation digital assistant executes the automation. Intent processor may use different large language model (LLM) models to complete different actions recommended by next generation digital assistant in backend.

[0121] In some embodiments, intent processor is under the intent level, i.e., the user submits a request, and next generation digital assistant determines what is the intent of the request. In this embodiment, next generation digital assistant determines if a smart skill should be used, what type of automation should be used, etc. These are executed when next generation digital assistant determines that the request submitted passes a predefined threshold. The predefined threshold provides a guarantee that the user expects a certain result back from next generation digital assistant. If next generation digital assistant determination is below the predefined threshold, next generation digital assistant may provide the user with some options to select. This may be displayed in side panel **1110**.

[0122] In certain embodiments, relevant automation available in the next generation digital assistant is identified by selector mapping between the page and the automation, or else intent processor can also refer to the page's object repository to determine the relevant automations.

[0123] AI UI Automation

[0124] With the given NLP statement (a high level command), next generation digital assistant starts performing the automation and re-assesses what it should do next. The user can record the steps and save the recorded steps as a workflow. In certain embodiments, the workflow can be modified by the user.

[0125] In certain embodiments, a user may upload a file to a webpage or an application, and next generation administrative assistant may generate a question and answer (Q&A) about the uploaded file. For example, a user may upload (or drag and drop) a copy of the user's passport in side panel **1110**. The user may further type in side panel **1110***a* request for further information, (in this case, ask if a visa is required). Next generation administrative assistant may analyze the content of the file (e.g., passport) to answer the question. In another embodiment, multiple files may be uploaded and next generation digital assistant may determine the difference in the file and generate a Q&A for the user.

[0126] FIG. **12** is a flow diagram illustrating method **1200** for invoking a next generation administrative assistant and executing a series of actions, according to an embodiment of the present invention. In some embodiments, method **1200** begins at **1205** with invoking next generation administrative assistant. Next generation digital assistant may be invoked by a user action. The user action may include highlighting of text in an application such as a web browser, email application, messaging application, etc.

[0127] At **1210**, method **1200** includes generating a drop down menu, which may be part of the application or separate from the application. This drop down may include a summary of the user action and one or more options for a user to select from in response to the user action.

[0128] At **1215**, method **1200** may include, upon receipt of the user selection, launching a side panel that populates a draft response. This draft response is generated by the next generation digital assistant. The side panel may also include options for the user to select from. These options may include how the reply should be drafted (e.g., casual reply, neutral reply or formal reply, and/or brief reply, average reply or long reply).

[0129] At **1220**, method **1200** may include executing (including sending) the reply by next generation digital assistant after the user selects the option(s). At **1225**, method includes generating, by next generation digital assistant, additional recommended steps for the user to consider.

[0130] Below is an embodiment illustrating bi-directional communication between the user and next generation digital assistant. FIG. **13** is a GUI **1300** illustrating setup and deployment of next generation digital assistant, according to an embodiment of the present invention. In this embodiment, next generation digital assistant may be deployed and updated to all users with an assistant. Next generation digital assistant also has all the capabilities of a running process such as Orchestrator™ and Integration Service access. UiPath® Assistant may be set up to boot up with the next generation digital assistant process, so next generation digital assistant is running when the user logs in.

[0131] FIG. **14** is a GUI **1400** illustrating launching next generation digital assistant when minimized, according to an embodiment of the present invention. In this embodiment, when next generation digital assistant is minimized, users can access the side panel by clicking or selecting on a notch on the right side of the screen or through an in-context interaction (see below).

[0132] FIG. **15** is a GUI **1500** illustrating a first run start up experience, according to an embodiment of the present invention. In some embodiments, when a user opens next generation digital assistant from the notch, the user is greeted with a sample prompts and actions that next generation digital assistant can help the user with. These prompts are curated by the team to be useful for all users.

[0133] FIG. **16** is a GUI **1600** illustrating in-context interactions, according to an embodiment of the present invention. In this embodiment, when an email arrives with details about the meeting, a user may highlight the relevant text and send it to the next generation digital assistant. In another embodiment, the user can use next generation digital assistant to take a screen grab and transfer context into the chat.

[0134] FIG. **17** is a GUI **1700** illustrating a smart skill being executed by next generation digital assistant, according to an embodiment of the present invention. In some embodiments, using context about the meeting and the enterprise travel policy, a travel checklist is created using a smart skill (e.g., pre-canned prompts engineered by the generative AI model). Based on the checklist, next generation digital assistant may recommend follow-up actions to the user.

[0135] FIG. **18** is a GUI **1800** illustrating a user uploading a context (e.g., a data file) using the side panel of FIGS. **11**A-D, according to an embodiment of the present invention. In some embodiments, the user may upload a document

(e.g., photo of his or her passport) and next generation digital assistant may use the uploaded document to extract the information. This information may then be added as context for next generation digital assistant and may also be used for input parameters in subsequent automations.

[0136] FIG. **19** is a GUI **1900** illustrating an application programming interface (API) automation, according to an embodiment of the present invention. In some embodiments, a user may email his or her document (e.g., passport information) to the travel team using an integration service connector for Outlook™. In this embodiment, next generation digital assistant may automatically infer argument values from the conversation.

[0137] FIG. **20** is a GUI **2000** illustrating prebuilt automation, according to an embodiment of the present invention. In some embodiments, the user may continue down the list by selecting a recommendation to run a UiPath™ automation that sets his or her out-of-office message. In this embodiment, next generation digital assistant may infer argument values for this automation similar to the argument values shown in FIG. **19**.

[0138] FIG. **21** is a GUI **2100** illustrating AI automation, according to an embodiment of the present invention. In some embodiments, the user may view information about the customer on LinkedIn™ by asking next generation digital automation to use AI automation to navigate to the customer's profile on LinkedIn™.

[0139] It will be readily understood that the components of various embodiments of the present invention, as generally described and illustrated in the figures herein, may be arranged and designed in a wide variety of different configurations. Thus, the detailed description of the embodiments of the present invention, as represented in the attached figures, is not intended to limit the scope of the invention as claimed, but is merely representative of selected embodiments of the invention.

[0140] The features, structures, or characteristics of the invention described throughout this specification may be combined in any suitable manner in one or more embodiments. For example, reference throughout this specification to "certain embodiments," "some embodiments," or similar language means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, appearances of the phrases "in certain embodiments," "in some embodiment," "in other embodiments," or similar language throughout this specification do not necessarily all refer to the same group of embodiments and the described features, structures, or characteristics may be combined in any suitable manner in one or more embodiments.

[0141] It should be noted that reference throughout this specification to features, advantages, or similar language does not imply that all of the features and advantages that may be realized with the present invention should be or are in any single embodiment of the invention. Rather, language referring to the features and advantages is understood to mean that a specific feature, advantage, or characteristic described in connection with an embodiment is included in at least one embodiment of the present invention. Thus, discussion of the features and advantages, and similar language, throughout this specification may, but do not necessarily, refer to the same embodiment.

[0142] Furthermore, the described features, advantages, and characteristics of the invention may be combined in any suitable manner in one or more embodiments. One skilled in the relevant art will recognize that the invention can be practiced without one or more of the specific features or advantages of a particular embodiment. In other instances, additional features and advantages may be recognized in certain embodiments that may not be present in all embodiments of the invention.

[0143] One having ordinary skill in the art will readily understand that the invention as discussed above may be practiced with steps in a different order, and/or with hardware elements in configurations which are different than those which are disclosed. Therefore, although the invention has been described based upon these preferred embodiments, it would be apparent to those of skill in the art that certain modifications, variations, and alternative constructions would be apparent, while remaining within the spirit and scope of the invention. In order to determine the metes and bounds of the invention, therefore, reference should be made to the appended claims.

1. A computer-implemented method for executing one or more digital assistant tasks using robotic processing automation (RPA), comprising:

activating a robot to monitor for one or more triggers while a user of a computing device is performing one or more tasks on the computing device;

identifying, by the robot, one of the one or more triggers for execution of the one or more digital assistant tasks; and

loading, by the robot, a workflow associated with the identified one of the one or more triggers to complete one of the one or more digital assistant tasks, or

creating, by the robot, a new workflow associated with the identified one of the one or more triggers based on bidirectional communication between the robot and a user of a computing device.

2. The computer-implemented method of claim **1**, wherein the activating the robot comprises

activating, by the user of the computing device, the robot to perform monitoring of the one or more triggers for executing the one or more digital assistant tasks.

3. The computer-implemented method of claim **1**, wherein the activating the robot comprises

activating the robot to perform monitoring of one or more triggers automatically at startup of the computing device for executing the one or more digital assistant tasks.

4. The computer-implemented method of claim **1**, further comprising:

upon activating of the robot, a toolbar is launched facilitating communication between the robot and the user, wherein

the toolbar is configured to facilitate two-way communication between the robot and the user.

5. The computer-implemented method of claim **1**, further comprising:

upon activating of the robot, a graphical user interface is launched facilitating communication between the robot and the user, wherein

the graphical user interface is configured to facilitate two-way communication between the robot and the user.

6. The computer-implemented method of claim **1**, further comprising:

communicating, by the robot, with the user via a toolbar or a graphical user interface for execution of one of the one or more tasks when the robot identifies the one of the one or more triggers.

7. The computer-implemented method of claim **6**, wherein the communicating with the robot comprises

sending, by the robot, a message suggesting a digital assistant task associated with the identified one of the one or more triggers.

8. The computer-implemented method of claim **1**, further comprising:

receiving, by the robot, a message comprising digital assistant task to be performed from the user via a toolbar or graphical user interface.

9. The computer-implement method of claim **8**, wherein the message comprises a media file.

10. The computer-implemented method of claim **9**, further comprising:

upon receipt of the message, comparing, by the robot, the media file with other media files stored in a machine learning (ML) database;

identifying, by the robot, a workflow associated with the media file;

loading, by the robot, the workflow associated with the media file; and

executing, by the robot, the workflow associated with the media file.

11. The computer-implemented method of claim **10**, further comprising:

when the robot is unable to load a workflow, creating a new workflow for the media file and storing the new workflow in the ML database.

12. The computer-implemented method of claim **11**, wherein the creating of the new workflow comprises

analyzing, by the robot, at least two workflows in the ML database;

extracting, by the robot, at least one or more sections from each of the at least two workflows to create the new workflow; and

sending, by the robot, the new workflow to the user via a graphical user interface or a toolbar for the user to confirm and prior to execution of the new workflow.

13. A computer-implemented method for executing one or more digital assistant tasks using robotic processing automation (RPA), comprising:

populating, by a next generation digital assistant application, a drop down menu comprising a series of options when a user performs an action in a software application;

populating, by the next generation digital assistant application, a side panel comprising a task to be performed by the next generation digital assistant application and a series of options for the user to select from; and

updating, by the next generation digital assistant application, the side panel with additional options in response to a selected option from the series of options, wherein the additional options are recommended actions for the user to select from in addition to the task performed by the next generation digital assistant application.

14. The computer-implemented method of claim **13**, further comprising:

triggering, by the next generation digital assistant application, one or more workflows through bi-directional communication with the user by way of a chat interface within the side panel.

15. The computer-implemented method of claim **14**, wherein the triggering of the one or more workflows comprises

creating a new workflow based on a prompt or text submitted by the user.

16. The computer-implemented method of claim **15**, wherein creating of the new workflow comprises

identifying a workflow from a catalog of pre-existing workflows from the prompt or text.

17. The computer-implemented method of claim **16**, wherein the creating the new workflow comprises

creating the new workflow from information provided by the user through prompt, context, and actions.

18. The computer-implemented method of claim **13**, further comprising:

capturing or storing, by the next generation digital assistant application, text communication submitted by the user via a chat interface within the side panel for developing contextual history.

19. The computer-implemented method of claim **18**, further comprising:

inputting, by the next generation digital assistant application, the developed contextual history into a workflow loaded in response to a user input.

20. The computer-implemented method of claim **13**, further comprising:

performing, by the next generation digital assistant application, a task without a pre-existing workflow on an application based on actions, prompts or requests submitted by the user.

* * * * *