Smart Speaker Integration with LetsHub

By
Michael Diep

Senior Project
California Polytechnic State University
San Luis Obispo
Spring 2018
Table of Contents

Smart Speaker Integration with LetsHub

Table of Contents
Introduction 2
Application 3

Image 1: High level overview of LetsHub’s Property Management workflow 3

Background

Image 2: Screenshot of the Google Home dialogue workflow (what a tenant would hear after mentioning the intended command to their smart speaker) 5

Image 3: Screenshot of the LetsHub Dashboard 6

Image 4: Screenshot of how a contractor would interact with the LetsHub Property Management BOT 7

Design 8

Implementation 9

Dialogflow 9

Image 5: Integration of Google Assistant with the LetsHub platform as done by Dialogflow 10

LetsHub Dashboard 11

Image 6: The LetsHub dashboard containing open incidents for the property manager to manage 11

Twilio 12

Image 7: Twilio Dashboard (Programmable SMS interface) 12

Analysis / Verification 13

Related Work 14

Image 8: A screenshot of the Rentalutions Dashboard 14

Future Work 16

Conclusion 17
Introduction

Imagine you are a property manager, and you are currently renting your property to tenants. These tenants, who pay rent continuously for use of the property often leave the burden of upkeep to the property owner (or property manager). If the tenant were to run into an issue with their rental (such as a broken sink, replacement of a faucet, leaking of the roof, … etc.) they would have to coordinate with the property manager, who would then in turn have to manage the logistics of a service contractor and the tenants availability. Eventually, after logistic juggling from the tenant, property manager, and the contractor, the tenant would have their request resolved. Now, imagine the same issue, but multiplied by 10, 100, or even 10,000 rental properties. Each tenant could potentially have a different issue or request that needs to be serviced. This is where LetsHub’s property management platform comes in.

Currently, in the industry of property management, property managers have to manually manage tenant service requests and coordinate those requests with contractors. This involves complex task management from tenants, property managers, and contractors to service tenant requests. The LetsHub platform serves to use computer technology and artificial intelligence to streamline the management of tenant issues. LetsHub’s platform serves to make it easier for tenants, contractors, and property managers to engage and coordinate with each other. The LetsHub platform is up 24/7 and can use natural language artificial intelligence to service tenant requests, streamlining the work for the property manager. Through the property management BOT, a tenant can be sure that their service request is indexed, updated, and accounted for by the property manager. This is because the property management BOT automates the process of creating service requests and updating tenants.

In this senior project assignment, I implemented a segment of the entire LetsHub property management workflow. The workflow will use smart speakers (Google Home and Amazon Alexa) to directly submit and update tenant requests. With this proposed workflow, tenants can now report rental issues and problems through the use of their smart speaker. With smart speaker integration, we allow the tenant to access a new set of BOT features with their voice.
Application

LetHub is a company that creates software tools to streamline services. LetHub strives to use artificial intelligence to simplify the redundant work that a property manager has to do. LetHub’s Virtual Assistant would automatically manage service requests between the tenant and a contractor. The LetHub product can be evaluated from three different customers and standpoints: the tenant, property manager, and contractor. The tenant is a person who rents a property for either the short-term or long-term. In the specific workflow that I implemented on the LetHub platform, the tenant would only interface through one portion of our application, the smart speaker (Google Home or Amazon Alexa). The property manager is a person (or persons) hired by the owner of the rental company to manage the rental listings. The property managers will interact with the LetHub platform through the dashboard (as shown in image 1). Thirdly, a contractor is referred to as a person who the property manager contracts service requests to. Such requests to a contractor could be a broken sink, leaky roof, utility replacement, … etc. In the diagram shown in image 1 you can see how all the customers interact with the LetHub platform, each at a different level and step of the process.

![Image 1](image.jpg)

**Image 1:** High level overview of LetHub’s Property Management workflow

The goal of the senior project design is to create a fully functioning end to end workflow of automated incident handling. The workflow starts with the conversion of conversational commands to computer instructions. Then, in the second portion of the implementation, we will see the dashboard, where the property manager able to view open incident requests and manage them accordingly. Lastly, we will observe the flow where the property manager (through LetHub) coordinates with a contractor to service request at the tenant’s residence.
Background

The intended use of the product is that a tenant would be able to use their Google Assistant and Amazon Alexa device to report incidents to database. The smart speakers serve to convert human conversational commands to computer interactions. The human conversation commands to computer interactions is done through a human-computer interaction API called Dialogflow. Dialogflow is a tool that has automated machine learning and natural language understanding so we do not have to define the exhaustive list of commands to interact with technologies (such as a smart speaker). This allows LetsHub to utilize the benefits of an AI tool such as Dialogflow to extract computer instructions from human voice. An example that illustrates this interaction is the tenant telling their Google Home device, “Ok Google, Talk to LetsHub Property BOT.” The smart speaker would respond with “Hi I'm Joe, Worldwide Properties' Assistant. How may I help you?” (The entire interaction is shown on Image 2 below)
Afterwards, through the tenant’s smart speaker, they are able to access features such as getting real time status updates on their incident reports, schedule services ahead of time, and check the availability of their contractor.

From the property manager’s standpoint, they are able to monitor and manage all service requests on LetsHub’s dashboard web interface. Incident reports can come from multiple workflows, the first being an already implemented Facebook Messenger BOT. In the senior
project design, the added workflow allows tenants to make incident requests through their smart speaker. All incident requests appear on the property management dashboard, shown on image 3.

Image 3: Screenshot of the LetsHub Dashboard

After the property manager assigns the incident to a contractor, the contractor would get a text from the LetsHub BOT to schedule a time to service the request. The BOT then communicates with the contractor to schedule a time they are available to service the request. As the contractor updates the LetsHub BOT, the status will be shown for the property manager to see, and the tenant to request when they ask for a status update. The BOT will text and be in constant communication with the contractor throughout the entirety of the service request. The screenshot below shows how a contractor would interact with the LetsHub property BOT.
When the property manager assigns a task to the contractor, the contractor immediately receives a text of the open service request. The BOT asks if the contractor is available in three time slots, 2 hours apart from each other and within business hours (Monday to Friday 9am - 5pm). The contractor is able to then select the 2 hour block that they are free to service the request. The BOT then updates the property management dashboard with the appropriate response. As the contractor proceeds to complete the task, the status updates appear on the LetsHub Property Management dashboard. The tenant is also able to get updates on their incident request through an implemented “Get Recent Updates” workflow that we have designed.
Design

The background provided us with a high level overview of the project, multiple user requirements, and the intended functionality of each medium. For the design, we require the integration and inclusion of all of these tools at once. Following the description of tools, we are going to dive into the specific implementation of this workflow.

For the first part of the workflow, the tenant facing API/Tool we use is called Dialogflow. Dialogflow is setup to adequately work down a tree of case statements that we have create. We have to manually create the tenant interaction flow in Dialogflow to gather the information that we need from the tenant. Dialogflow needs to be built and altered to work with the correct user commands and webhooks to the LetsHub backend server. Once the text flow is correctly implemented, we can deploy it to various platforms by integration (Google Assistant, Facebook Messenger, Twilio, Telegram, … etc.). So, through Dialogflow’s integration channel, the smart speakers can be set up to work according to the text workflow. Once the tenant performs the correct command on their smart speaker, the appropriate request would be sent to the LetsHub database in the form of a webhook.

The webhook will then arrive on the LetsHub Property Management Dashboard backend. The backend is an Amazon Web Server (EC2) running MongoDB and Node.js. The frontend property management dashboard interface is made in AngularJS. The backend has a separate workflow implementation to handle requests from both Dialogflow’s (tenant) webhooks and Twilio’s (contractor) API requests. Information is then displayed on the LetsHub dashboard interface along with the appropriate information being parsed. If the request is an emergency, the database will forward the query to the contractor directly.

The LetsHub dashboard also uses Twilio API to service SMS and MMS requests between our platform and the contractor. The information passing between the LetsHub database and the contractor would be managed by Twilio (in the form of SMS and MMS). The implementation of the message registering and passing between the BOT and the contractor is done from the backend and in Node.js. The Property Manager can oversee this entire process through the LetsHub Dashboard and apply administrative support at any time. We will go into more detail on each interface in the implementation portion of the report.
Implementation

As part of the user requirements, we have to service multiple customers: the tenant, property manager, and contractor. Each of these customers interact with our platform on different levels, and the BOT needs to ensure that all three players are property addressed in the property management workflow. LetsHub

LetsHub’s backend runs on an Amazon Web Services EC2 server. The server uses MongoDB to facilitate JSON and RESTful requests. The majority of the backend utilities and scripts are written in Node.js. To get computational information from the tenants, we will be using an artificial intelligence tool called Dialogflow. Dialogflow has integration with many frameworks such as Facebook Messenger, Amazon Alexa and Google Assistant. LetsHub leverages these larger platforms to get user input. Dialogflow will pass requests from the tenant to our backend in JSON format. LetsHub’s backend will process the request accordingly. In some cases, we will have to schedule a contractor to service a request from a tenant. That process would be facilitated by communication between LetsHub’s backend and Twilio’s API. Below, we will go into detail on each subsystem of the workflow and how the interfaces are set up.

Dialogflow

The tenant interfaces directly with their smart speaker. This is done through the tenant’s Google Home and Amazon Alexa. Dialogflow manages the behavior between the smart speaker and a webhook.
Dialogflow manages the human to computer interaction aspects of our workflow. Dialogflow provides a tool to directly integrate commands with Google Assistant (which in turn, interfaces with Google Home). For Amazon Alexa, we require an command file export in the form of Alexa skill files (.zip). Inside the .zip file contains an IntentSchema.json, SampleUtterances.baf, and numerous .txt files containing entities (commands for the smart speaker). These commands would then have to be uploaded to the smart speaker: through a terminal for the Google Home, and through USB for Amazon Alexa. Each workflow command has to be specified directly, as Dialogflow will not know when to move from state to state.
The next interface is the LetsHub Property Management Dashboard. This is where the property manager receives incident requests and organizes them to resolution.

Image 6: The LetsHub dashboard containing open incidents for the property manager to manage

The dashboard is a comprehensive service engine that provides the processed information in the backend to the property manager. The dashboard runs on an Amazon EC2 server using MongoDB and Node.js to perform all necessary backend services. The dashboard is where we catch webhooks from Dialogflow and management workflow processes for the property manager.
Twilio

LetsHub uses Twilio to handle the contractor to dashboard interactions. Below (Image 7) is a screenshot of what the Twilio dashboard looks like. We program the text commands that the dashboard will query when a text is sent to the LetsHub Property Management Twilio number.

![Twilio Dashboard (Programmable SMS interface)](Image 7)

Twilio requests are handled by the LetsHub Dashboard via API calls to between our backend and Twilio’s API. The above image is a screenshot of the Twilio dashboard, where SMS and MMS messages are received and sent. On the Twilio Dashboard, we can track the interactions between our dashboard and the contractor. All of the API requests (send and receive SMS) are written in Node.js on LetsHub’s backend. LetsHub uses Twilio to communicate between the dashboard and contractor.
Analysis / Verification

For testing, we have a person on the LetsHub team, working on creating unit tests for default use cases throughout our platform. The tests are written using the Selenium/Appium framework and are coded using XML, Java, Python, and Microsoft Excel. The tests are performed all together prior to rollout of the product and test the extensive application features such as BOT, dashboard, and contractor interactions. The automated tests run scripts that simulate an external user going through all the features on the platform with end to end (E2E) use cases. The tests are meant to verify the continuity of the workflow from when a tenant talks to our BOT, to the property manager who delegates tasks externally, to a contractor servicing the tenant’s request. From these tests, we can verify if changes that we made affect the the wider platform.

Currently, we do not have a way to test voice inquiries through Google Home and Amazon Alexa. All of the testing on this interface has to be done manually by the team. The intended use of the smart speaker is to submit incidents, and get the status of their service request. If the tenant asks the BOT to get the status of their service request, the smart speaker should locate and respond with the appropriate information.
Related Work

A similar platform called Rentalutions (https://www.rentalutions.com/) has created software that assists property managers with their services. Although this company is servicing a similar industry. LetsHub’s approach is to leverage external APIs such as Dialogflow and Twilio to facilitate communications between the tenant, property manager, and contractor. Additionally, we will be leveraging existing code from our food delivery workflow (see https://www.letshub.com/otherproducts/). This proposed workflow is similar to our existing interface in terms of contractor (delivery) tracking, a web management interface, and automated customer updates. A majority of the features will need to be altered and reimplemented for property management.

There are many property management softwares available that compete directly with our product. Some of these products include Buildium, AppFolio Property Manager, Maintenance Connection, SiteLink, storEDGE, Rentec Direct, Rent Manager, Propertyware, Total Management, SimplifyEm Property Management Software, Escapia, roomMaster, MRI
Residential Management, Property Matrix, Hemlane, Property Vista, ResMan, Axserion, and SKYLINE Software. These software provide different rental solutions for various different industries such as hotel management, rentals, vacancies, and home maintenance.
Future Work

As of right now, the speaker can only handle two workflows. The first workflow being the submission of an incident to the property management dashboard. And the second workflow being requesting the updated status of the report from the speaker. These are the only features requested by LetsHub currently, to establish a proof of concept and bare minimum add on functionality to the company’s workflow.

Some future work that can be done on this product can be enhanced feature development for the speaker functions. This includes the implementation of features that currently exist on the BOT to a Google Voice command. Features such as, submitting specific types of requests, entering phone numbers for verification, updated responses from the BOT and more.

So far, the newly implemented features have been only tested on Google Home devices and not Amazon Alexa devices. The additional speaker functionality would be useful and could potentially reach a wider customer base with the added support. To add additional features for the Amazon Alexa speaker, each speech workflow needs to be coded accordingly.
Conclusion

Overall, the entire project required integration on multiple levels of software and hardware. The hardware level included the programming of the actual commands to the smart speaker. And the software aspect included modifying the backend to receive webhook requests from the smart speaker. All aspects of the workflow, including edge cases, had to be accounted for when integrating the new smart speaker functionality. Each response to and from the tenant and contractor has to be crafted to receive the correct response and handle commands accordingly.

The project serves to create more avenues for tenants to interact with the property dashboard. The idea is if we make it easier for tenants to submit incident requests, they would be more inclined to use the LetsHub system, streamlining the entire property management service. The platform provides a way for property managers to better manage their tasks, contractors, and view analytics on their operations. The benefit of applying technology to this industry is that the computer can be running 24/7 to handle service requests from the tenant and contractor. This allows for improved productivity of the industry in terms of faster scheduling of contractors, increased productivity from the property manager in terms of coordinating repairs, and integration to many APIs through our digital platform.

Implementation of our workflow to a smart speaker creates a new way for tenants to interact with technology. We want to apply the latest software technology available to us and apply it for practical use. From the Google Home, to software API tools such as Dialogflow, the LetsHub Dashboard and Twilio. Each requiring a different level of communication with each other. The more each processes is automated, the more we are able to eliminate the need for human processing in the property management space.