Schrute Farms

Pam - Virtual Secretary

Authors:

Jonathan Segovia
Tristan Starck
Kunal Naik
Michael Caccamo
Yuki Mano

Introduction & Goal

Our goal is to make Pam the default voice assistant secretary for small businesses. Our vision for Pam is to be able to make appointments and answer general questions regarding the business so that real employees can focus on doing what makes their business great. Customers will be able to call in and hold a natural conversation with Pam in order to gain the information they need. Many larger companies have home grown systems for handling phone calls, however smaller businesses may not have the resources to establish their own. We aim to package Pam as an easy to use ecosystem that can be managed with a web dashboard.

Who is it for?

1. **Small Businesses** - those who provide an impactful service to the community, but don't necessarily have the resources to have an employee always man the phone or set up their own automated system.
2. **Customers** - those who want to interact with their favorite local business by phone and have a natural experience with a virtual assistant.

Why Build It?

1. Improving the customer experience standard for voice assistant secretaries
2. Provide way for small businesses to easily handle phone calls
3. Monetization opportunities through the SaaS model

What is it?

Glossary

1. **Client:** Small business contracting Pam to be their secretary
2. **Customer:** people who want to call into business to get information/make appointments
3. **Call:** A call into a business Twilio account to interact with Pam
4. **Appointment:** Appoints made for specific services offered by the business
5. **Confirmation:** Text message sent to the caller after the appointment is confirmed
6. **Business Homepage**: Homepage for business employees
7. **FAQ View**: A page for a business to set answers for FAQs

User Types

1. **Callers**: people calling into a business to set up an appointment or find info on the company (store hours, locations, currently open?)
2. **Businesses**: Registered business' who can receive automated calendar updates with appointments made from incoming callers

Call

- A call can answer simple questions from the caller, or set up an appointment time.
- To set up an appointment time, a time must be provided that fits into the businesses schedule. If the given time isn't available for booking, other times are suggested to the caller.

Appointment

- Appointments made for services at a business.
- To book an appointment, a beginning time must be provided by the caller, and then confirmed by the business.
- The duration of appointments will be set by the business depending on the type of service, and Pam will articulate this duration to the caller at the time of set-up.
- Appointments can be cancelled by both the business and the customer.

Confirmation

- A text message is sent to caller after the appointment is confirmed, and then again a few days or hours before the appointment. The calling phone number is used as the SMS endpoint. A different phone number can be set by the user.

- If an admin cancels an appointment, a text message stating that the appointment is canceled is sent to the client who had that appointment.

Business Homepage

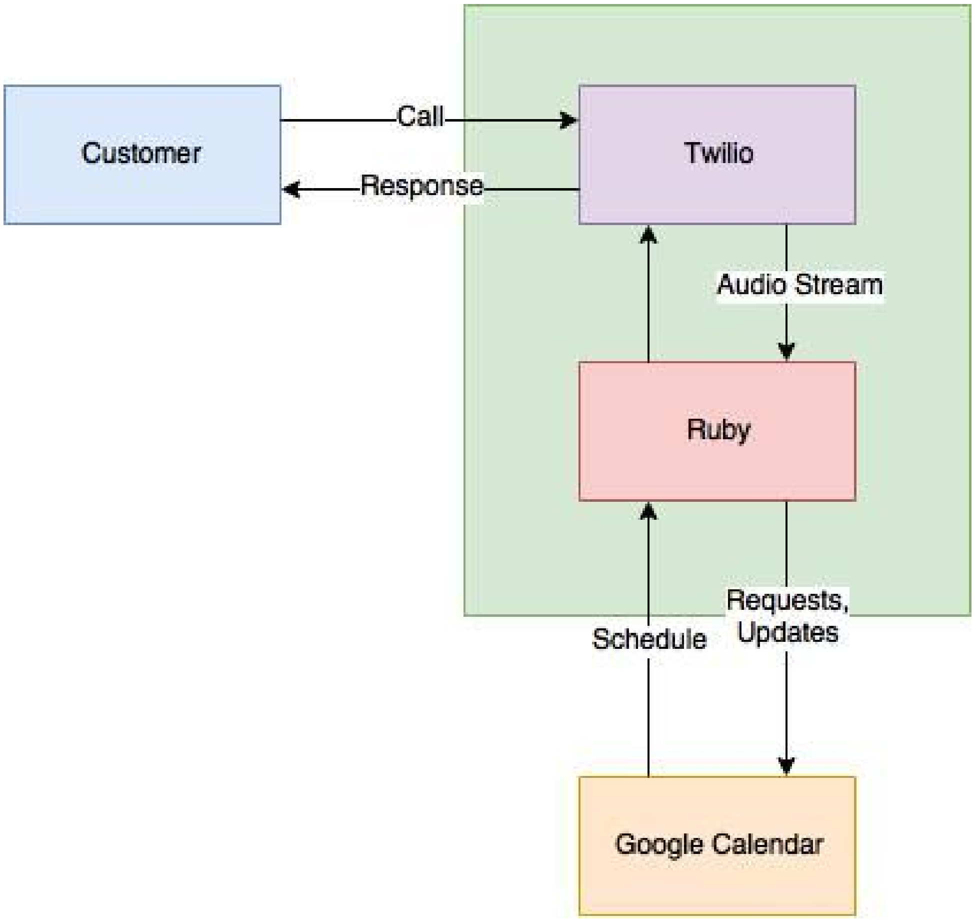
- Admins of the Business Homepage can view all of their customers' appointment times, including customer's' phone number and name.
- Admins can label what days and times their business are open.
- Admins have the ability to cancel a customer's appointment.
- Admins have the ability to change a customer's appointment days and times.

FAQ View

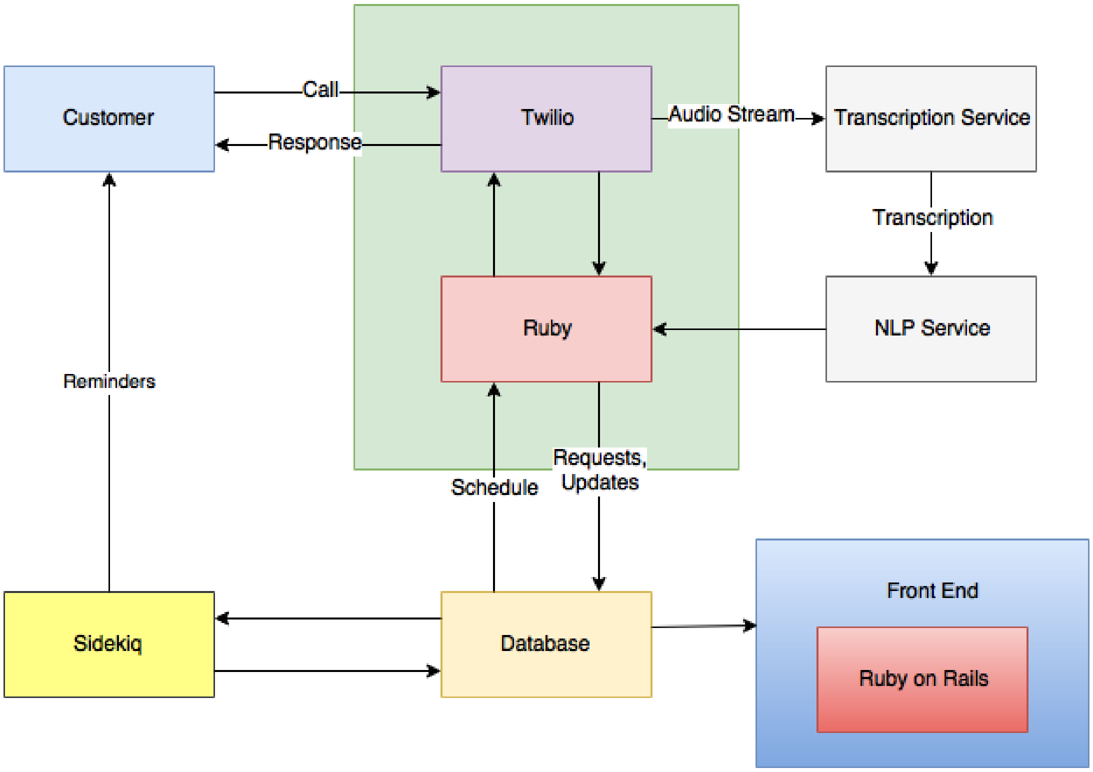
- Businesses can enter answers to commonly asked questions, including store address, hours, closures, nearby parking, directions,

System Architecture:

High Level Diagram Fall Quarter Goal



High Level Diagram Winter Quarter Goal



User Interactions:

Examples of a customer successfully scheduling an appointment:

1. Customer will call the client, and Pam will answer the phone giving an introduction tailored to the client's business.
2. Customer will request to schedule an appointment.
3. Pam will check the client's schedule to see if the time is available. (The time is available)
4. Pam will respond saying the the time is available, and will reconfirm the day and time of the requested appointment.
5. Customer confirms that the appointment is the correct time.
6. Pam will schedule the appointment under the customer's phone number.
7. Pam will ask if the customer has any other questions or concerns.
8. Customer will either hang up or answer "No" and Pam will hang up after thanking the customer for calling.
9. Pam will then send a text to the Customer with the information regarding their appointment.

   Notes:

    1. *If the requested time is not available, Pam will offer another possible time for that day. If this time does not work with the Customer, Pam will ask for another date/time.*

Examples of a customer successfully checking when their appointment is:

1. Customer will call the client, and Pam will answer the phone giving an introduction tailored to the client's business.
2. Customer will request to check when their appointment is.
3. Pam, using the customer's phone number, will check to see when the customer's appointment is.
4. Pam will respond to the customer with their appointment date and time.
5. Pam will ask if the customer has any other questions or concerns.
6. Customer will either hang up or answer "No" and Pam will hang up after thanking the customer for calling.
   Notes:
    1. *If Pam doesn't find a schedule under the Customer's phone number, Pam will ask the Customer for the phone number that they scheduled the appointment with.*

Examples of a Customer cancelling their appointment:

1. Customer will call the client, and Pam will answer the phone giving an introduction tailored to the client's business.
2. Customer will request to cancel their appointment.
3. Pam, using the customer's phone number, will search the client's schedule for the appointment.
4. Pam will confirm with the Customer the date and time of the appointment they originally scheduled.
5. Customer will respond with a "Yes".
6. Pam will cancel the appointment.
7. Pam will ask if the customer has any other questions or concerns.
8. Customer will either hang up or answer "No" and Pam will hang up after thanking the customer for calling.
   Notes:
   1. If Pam doesn't find a schedule under the Customer's phone number, Pam will ask the Customer for the phone number that they scheduled the appointment with.

Examples of the Client logging into their website to manage their appointments:

1. Client goes to their website from their browser on their local PC

2. Client enters login credentials (Username and password) and presses ‘Log In’
3. Client is brought to their Business Homepage.
4. Client clicks on ‘Manage Appointments’ button.
5. Client is shown the next 7 days of appointments on a Calendar view.
6. Client clicks on the appointment they would like to change, and is brought to a new view showing details of the appointment, all editable.
7. Client changes time of appointment and presses ‘Save Changes’
8. Client is brought back to Calendar View.
9. A text message is sent to the Customer whose appointment got changed.

Requirements:

User Stories:

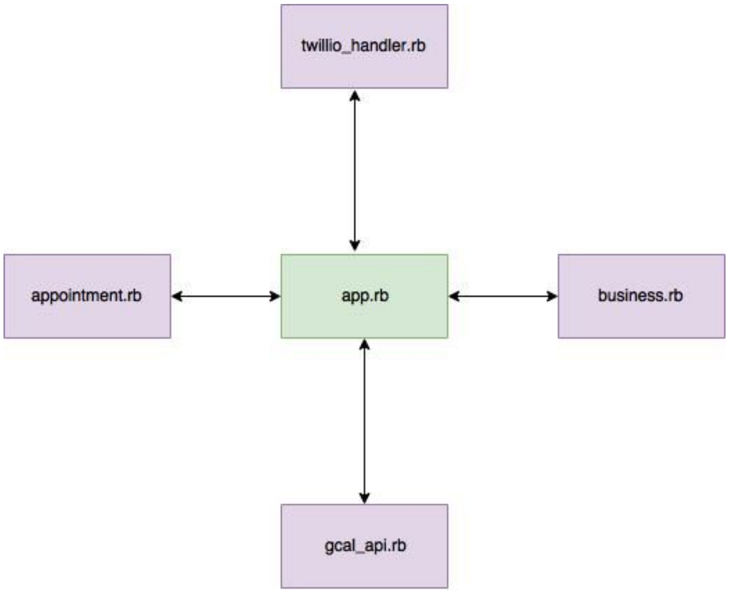
1. As a Pam Developer, I want to have a testing framework in place for unit and functional tests because we need our code in master to always be tested.
2. As a Pam Developer, I want to be able to talk to Pam and have whatever the user says be printed out so it can later be transcribed with a NLP Framework
3. As a Pam Developer, I want to be able to integrate the Google Calendar API so we have a place to store the appointments that are made through Pam
4. As a Pam Developer, I want to hard code certain use cases for appointments so we can test the functionality of the Google Calendar API
5. As a Pam Developer, I want to be able to research different NLP frameworks to see which would be best to use with our transcribed data
6. As a Pam Developer, I want to create a Business class that will model a Business (store hours, address, phone number, etc)
7. As a Pam Developer, I want to create an Operating Hours class that will store the business hours (open time, close time, etc)
8. As a Pam Customer, I want to be able to ask Pam what the business’ current store hours are and have her tell me what they are
9. As a Pam Customer, I want to be able to ask Pam what the business’ current address is and have her tell me what it is
10. As a Pam Customer, I want to be able to tell Pam when I would like an appointment to be made and have that appointment saved to Google Calendar
11. As a Pam Customer, I want to be able to tell Pam that I would like to delete an appointment and have that appointment deleted from Google Calendar
12. As a Pam Customer, I want to be able to tell Pam that I would like to update the time of an appointment and have that appointment time be updated on Google Calendar

13. As a Pam Customer, I want to receive text message confirmation when I create/delete/update an appointment
14. As a Pam Customer, I want to receive text message reminders an hour before my scheduled appointment
15. As a Pam Client, I want to be able to integrate Pam into my current business phone number and have it handle appointments for me
16. As a Pam Client, I want to have a web app that allows me to view the current appointments scheduled and allows me to customize Pam to fit my business needs
17. As a Pam Developer, I want to research different UI frameworks for the web app
18. As a Pam Developer, I want to implement a database to store certain user and customer data so that we can store more data than just appointments (i.e. business specific requirements, phone numbers, etc.)
19. As a Pam Developer, I want to learn to use the Sequel library for use with the database
20. As a Pam Developer, I want to research different database solutions and decide on which one to use

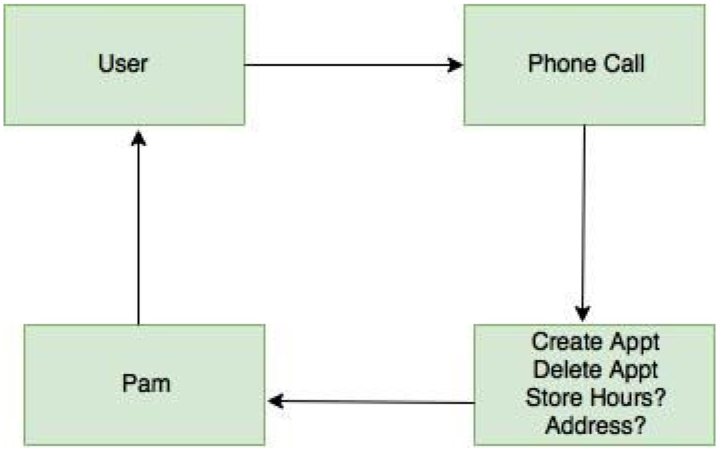
Current Trello Board (Green Boards are Done):
https://trello.com/invoca2018

System Models:

Class Diagram



Interaction Diagram



Appendices:

Technologies Used:

- Docker
- TravisCI
- Ruby
- Rspec
- Sinatra
- Twilio
- Github
- Google Calendar API