I 💚 U
(I See You)

A pre-consultation tool for virtual health care
404: Team Name Not Found

Teladoc Mentors:
- John O'Donovan
- Sushil Bharati
- Matias Massetti
- Juan Jouglard
- Javier Iparraguirre
- Gonzalo Arro
- Gabriel Salazar
- Catherine Ho
- Ole Eichhorn

Ishana Narayanan
Team Lead Chatbot

Sophie Groenwold
Scribe Chatbot

Steven Huynh-Tran
Frontend

Anthony Palomera
Backend

Evan Yip
Frontend

Katelyn Zhang
Backend
What We’ll Cover

01 Problem & Solution
02 Demo
03 Chatbot
04 Backend
05 Challenges
06 Milestones
The Problem & Solution

Transparency and efficiency in telehealth consultations
Background

Telehealth: medical care without in-person visits
Since March 2020, the use of telehealth services has expanded to 38 times its pre-pandemic size.

Teladoc: a major player in the telehealth industry
Teladoc delivered 10.5 million virtual visits in 2020.
The Problem

→ Transparency in patient-physician communication
  - Transparency in medical visits is extremely important to a patient's understanding and sense of autonomy.\(^1\)
  - In a telehealth context, communication issues are heightened by “black box” worries.\(^2\)

→ Efficiency through pre-consult screenings
  - Screenings done during consultations are often repetitive and use up time that could be better spent for both patient and physician.
  - Current examples of pre-consult screening software do not draw conclusions from raw patient data.

\(^1\)Identifying Transparency in Physician Communication, National Library of Medicine (2011)
\(^2\)Privacy and Security Concerns in Telehealth, AMA Journal of Ethics (2014)
Our Solution

01 Pre-consultation
Patient completes symptom screening in portal

02 Analyze Info
Chat logs stored, analyzed, and initial diagnosis made

03 Before Appointment
Physician reviews pre-screen and diagnosis

04 After Appointment
Physician inputs diagnosis and care steps in portal
02
Demo
03

Chatbot

How the patient completes their pre-consult
Chatbot Architecture

- **Amazon Lex**
- **DynamoDB**
- **AWS Lambda**
- **Infermedica API**

**Process Flow**

1. **Amazon Lex**
   - Elicit appointment ID
   - Provide patient symptoms

2. **AWS Lambda**
   - Retrieve medical information
   - Store patient diagnosis
   - Return appointment info

3. **DynamoDB**
   - Retrieve medical information
How our patient-physician information is stored
User logs in

Bot asks for symptoms

Schedule appointment and retrieve information

Gets and stores patient information

Patient Sequence
Challenges

Hurdles we've faced so far

→ Getting started

→ Reconciling aspects of our tech stack

→ Sifting through outdated documentation
Milestones

Our progress so far and plans for Winter 2022
Milestones reached

**End of October**
Clear vision on project functionality

**End of November**
Frontend, Backend, Chatbot integration for basic user experience

**Mid November**
First chatbot flow

**Today**
Clear vision on project functionality

**Start**

---

**End of November**
Frontend, Backend, Chatbot integration for basic user experience
Key Next Steps

**Chatbot**
Make more dynamic, currently deterministic. Include chat log summarization

**Seamless Communication**
Examine better communication architectures

**User Experience**
Refine user experience for a pleasant flow
Thank You

In particular, to:
Our Teladoc mentors
Professor Su & Mason

Any Questions?
Physician logs in

Select an appointment

Retrieve upcoming appointment information

Gets patient information and consult summary
Backend

1. Patient
   - PatientID
   - name
   - age (birthday)
   - gender
   - pre-existing conditions — array of strings
   - active medications

2. Appointment Table
   - appt ID
   - patient ID
   - doctor
   - app type
   - day/time

3. General Consult
   - appointment ID
   - patient ID
   - symptom
   - initial diagnosis (intermediate)
   - doctor diagnosis

- given to relay at when signing up
- for account

JSON object

{id: name, }
Query an Item from a Table

{"appointment_id":{"N":"2222222"},"appointment_type":{"S":"General Consult"},"doctor":{"S":"Carol Jackson"},"appointment_time":{"S":"1838555300"},"patient_id":{"N":"1"}}
Insert Data to a Table

The image shows a website interface for inserting data to a table. The interface includes a form with fields for appointment_id, patient_id, appointment_time, and doctor. It also displays a table with columns for appointment_id, patient_id, appointment_time, and doctor, showing some sample data:

- Appointment ID 1010, Patient ID 1010, Appointment Time 1638558920, Doctor Steve Robins
- Appointment ID 123, Patient ID 456, Appointment Time 1641847500, Doctor Ryland Gerald
- Appointment ID 4, Patient ID 1, Appointment Time 1638558900, Doctor Ruth Robinson
- Appointment ID 1011, Patient ID 1011, Appointment Time 1638558950, Doctor Kelly James
- Appointment ID 1012, Patient ID 1012, Appointment Time 1638558940, Doctor Jerry Main
- Appointment ID 1, Patient ID 2, Appointment Time 1638225000, Doctor Christopher Anderson
- Appointment ID 0, Patient ID 2, Appointment Time 1637872200, Doctor Christopher Anderson