HOW-R-U?

Suite of e-coaches aimed to analyse human behaviour

Carlos Sánchez Páez
Oresti Baños Legrán

17th July 2020

Índice

Index

Index

Proposal

- *E-coaches* suite as chatbots.

Proposal

- *E-coaches* suite as chatbots.
- Doctors can assign questions to patients.

Proposal

- *E-coaches* suite as chatbots.
- Doctors can assign questions to patients.
- Data analysis.

Proposal

- *E-coaches* suite as chatbots.
- Doctors can assign questions to patients.
- Data analysis.
- Psychologist bot.

Index

Context

- Mental disorders are very common in our society.

Context

- Mental disorders are very common in our society.
- Doctors have high workloads.

Context

- Mental disorders are very common in our society.
- Doctors have high workloads.
- Mental diseases are taboo.

Context

- Mental disorders are very common in our society.
- Doctors have high workloads.
- Mental diseases are taboo.
- No continuous traceability of patient's health status.

Context

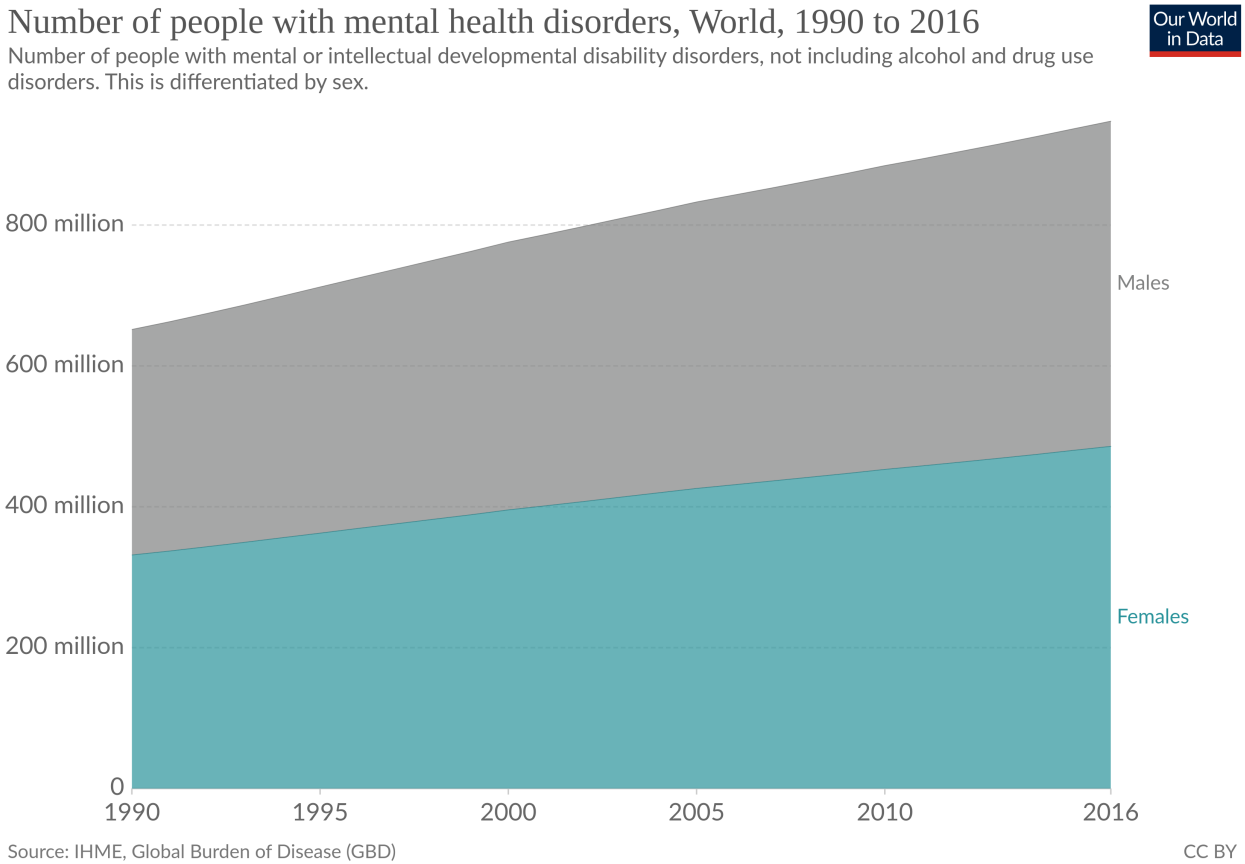

Figure: Number of people with mental health disorders

Reprinted from (Ritchie & Roser, 2018).

Motivation

- Technology is becoming increasingly integrated into our lives.

Motivation

- Technology is becoming increasingly integrated into our lives.
- Smartphones are used in a daily basis.

Motivation

- Technology is becoming increasingly integrated into our lives.
- Smartphones are used in a daily basis.
- Chatbots are becoming growingly becoming popular.

Motivation

- Technology is becoming increasingly integrated into our lives.
- Smartphones are used in a daily basis.
- Chatbots are becoming growingly becoming popular.
- In Spain there are 2 psychologist per 100.000 citizens (Vicente, 2019).

Index

Objectives

- **Main goal**:

Objectives

- **Main goal**:
  - *Conversational-agent-as-a-sensor* that asks questions to patients.

Objectives

- **Main goal**:
  - *Conversational-agent-as-a-sensor* that asks questions to patients.
  - Questions will be defined by specialists.

Objectives

- **Main goal**:
  - *Conversational-agent-as-a-sensor* that asks questions to patients.
  - Questions will be defined by specialists.
- **Secondary goals**.

Objectives

- **Main goal**:
  - *Conversational-agent-as-a-sensor* that asks questions to patients.
  - Questions will be defined by specialists.
- **Secondary goals**.
  - Graphical web interface for doctors.

Objectives

- **Main goal**:
  - *Conversational-agent-as-a-sensor* that asks questions to patients.
  - Questions will be defined by specialists.
- **Secondary goals**.
  - Graphical web interface for doctors.
  - Flexible and scalable architecture to add functionality to the system.

Objectives

- **Main goal**:
  - *Conversational-agent-as-a-sensor* that asks questions to patients.
  - Questions will be defined by specialists.
- **Secondary goals**.
  - Graphical web interface for doctors.
  - Flexible and scalable architecture to add functionality to the system.
  - Architecture based on containers to host the different system modules.

Objectives

- **Main goal**:
  - *Conversational-agent-as-a-sensor* that asks questions to patients.
  - Questions will be defined by specialists.
- **Secondary goals**.
  - Graphical web interface for doctors.
  - Flexible and scalable architecture to add functionality to the system.
  - Architecture based on containers to host the different system modules.
  - Implement a system that covers the previous goals.

Objectives

- **Main goal**:
  - *Conversational-agent-as-a-sensor* that asks questions to patients.
  - Questions will be defined by specialists.
- **Secondary goals**.
  - Graphical web interface for doctors.
  - Flexible and scalable architecture to add functionality to the system.
  - Architecture based on containers to host the different system modules.
  - Implement a system that covers the previous goals.
  - Test a beta version of the system to retrieve target audience's feelings about it.

Index

Index

Number of publications related to chatbots

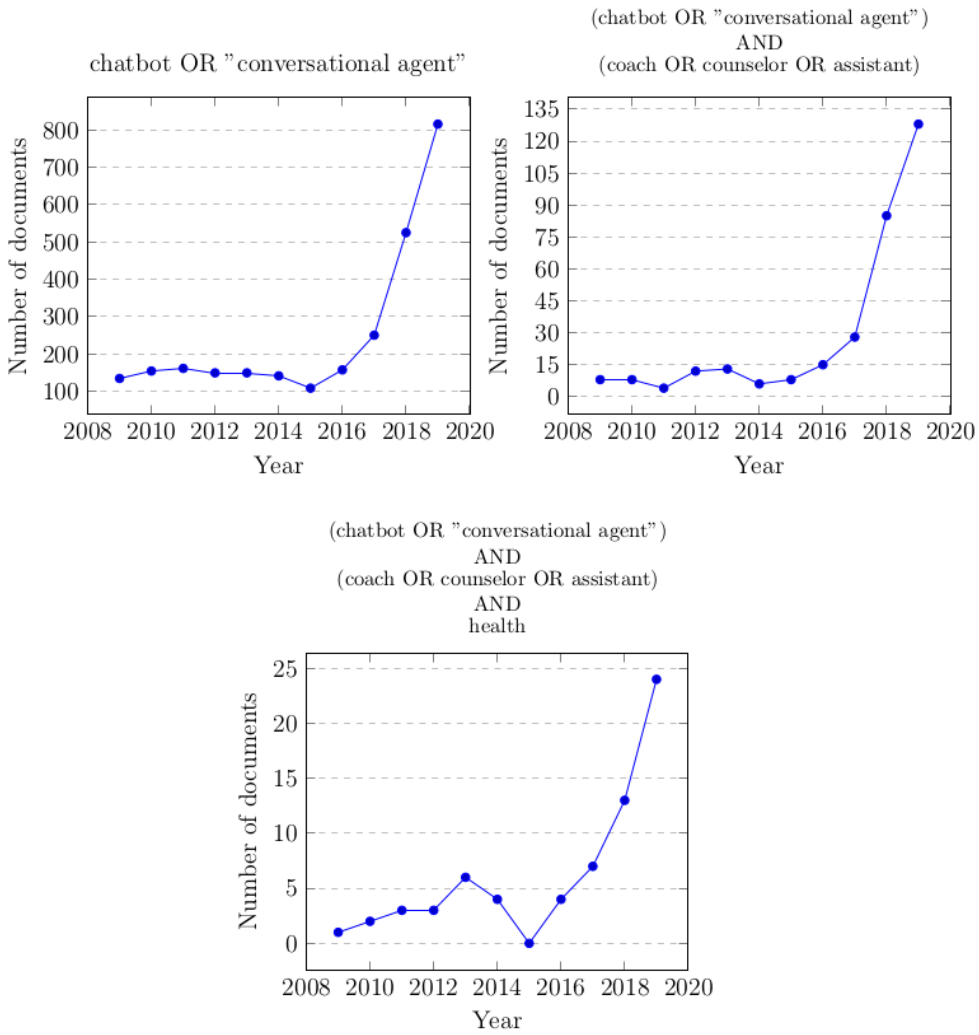

Figure: Search results of different queries performed in *scopus.com*.

Index

Health application domains

- Areas of application.

Health application domains

- Areas of application.
  - Dermathology

Health application domains

- Areas of application.
  - Dermathology
  - Nutrition

Health application domains

- Areas of application.
  - Dermathology
  - Nutrition
  - Psychology

Health application domains

- Areas of application.
  - Dermathology
  - Nutrition
  - Psychology
  - etc.

Health application domains

- Areas of application.
  - Dermathology
  - Nutrition
  - Psychology
  - etc.
- Target group.

Health application domains

- Areas of application.
  - Dermathology
  - Nutrition
  - Psychology
  - etc.
- Target group.
  - Students

Health application domains

- Areas of application.
  - Dermathology
  - Nutrition
  - Psychology
  - etc.
- Target group.
  - Students
  - Doctors

Health application domains

- Areas of application.
  - Dermathology
  - Nutrition
  - Psychology
  - etc.
- Target group.
  - Students
  - Doctors
  - Patients

Index

Conversational agent types

- Coaches: help users to get what they want.

Conversational agent types

- Coaches: help users to get what they want.
- Counselors: help users to identify and solve problems.

Communication formats

- Text

Communication formats

- Text
- Voice

Communication formats

- Text
- Voice
- Multimodal

Index

Technology

Platform	Daily active users (billions)	Free API for chatbots
(*Facebook Messenger*, 2008)	1.66	✔
(*Whatsapp*, 2009)	1.5	✗
(*WeChat*, 2011)	1.083	✔
(*Telegram*, 2013)	0.2	✔
(*Kik*, 2010)	0.015	✔
(*Discord*, 2015)	0.014	✔
(*Slack*, 2013)	0.012	✔
(*Viber*, 2010)	0.008	✔
(*Line*, 2012)	0.00723	✔

Table: Comparison between different chat applications (2019).

Technology

- Facebook likes can be helpful to predict people's sensitive properties (Kosinski, Stillwell, & Graepel, 2013).

Technology

- Facebook likes can be helpful to predict people's sensitive properties (Kosinski et al., 2013).
- Whatsapp's end-to-end encryption methods are not secure enough (Rastogi & Hendler, 2017).

Technology

- Facebook likes can be helpful to predict people's sensitive properties (Kosinski et al., 2013).
- Whatsapp's end-to-end encryption methods are not secure enough (Rastogi & Hendler, 2017).
- Telegram provides more privacy protection (Sutikno et al., 2016).

Index

Index

Requirements

- *Must have* requirements.

Requirements

- *Must have* requirements.
  - Ask questions to the patients.

Requirements

- *Must have* requirements.
  - Ask questions to the patients.
  - Custom keyboard.

Requirements

- *Must have* requirements.
  - Ask questions to the patients.
  - Custom keyboard.
  - User's enrollment and deletion.

Requirements

- *Must have* requirements.
  - Ask questions to the patients.
  - Custom keyboard.
  - User's enrollment and deletion.
  - File with patients' answers.

Requirements

- *Must have* requirements.
  - Ask questions to the patients.
  - Custom keyboard.
  - User's enrollment and deletion.
  - File with patients' answers.
  - Modularity.

Requirements

- *Must have* requirements.
  - Ask questions to the patients.
  - Custom keyboard.
  - User's enrollment and deletion.
  - File with patients' answers.
  - Modularity.
- *Should have* requirements.

Requirements

- *Must have* requirements.
  - Ask questions to the patients.
  - Custom keyboard.
  - User's enrollment and deletion.
  - File with patients' answers.
  - Modularity.
- *Should have* requirements.
  - CC BY-NC-SA 4.0 (*Creative Commons*, 2001) license.

Requirements

- *Must have* requirements.
  - Ask questions to the patients.
  - Custom keyboard.
  - User's enrollment and deletion.
  - File with patients' answers.
  - Modularity.
- *Should have* requirements.
  - CC BY-NC-SA 4.0 (*Creative Commons*, 2001) license.
  - Questions and answers should be modifiable.

Requirements

- *Must have* requirements.
  - Ask questions to the patients.
  - Custom keyboard.
  - User's enrollment and deletion.
  - File with patients' answers.
  - Modularity.
- *Should have* requirements.
  - CC BY-NC-SA 4.0 (*Creative Commons*, 2001) license.
  - Questions and answers should be modifiable.
  - Public questions.

Requirements

- *Must have* requirements.
  - Ask questions to the patients.
  - Custom keyboard.
  - User's enrollment and deletion.
  - File with patients' answers.
  - Modularity.
- *Should have* requirements.
  - CC BY-NC-SA 4.0 (*Creative Commons*, 2001) license.
  - Questions and answers should be modifiable.
  - Public questions.
  - Configurable schedule.

Requirements

- *Must have* requirements.
  - Ask questions to the patients.
  - Custom keyboard.
  - User's enrollment and deletion.
  - File with patients' answers.
  - Modularity.
- *Should have* requirements.
  - CC BY-NC-SA 4.0 (*Creative Commons*, 2001) license.
  - Questions and answers should be modifiable.
  - Public questions.
  - Configurable schedule.
  - Interactive charts.

Requirements

- *Could have* requirements.

Requirements

- *Could have* requirements.
  - Numeric order of questions.

Requirements

- *Could have* requirements.
  - Numeric order of questions.
  - Custom questions frequency.

Requirements

- *Could have* requirements.
  - Numeric order of questions.
  - Custom questions frequency.
  - Other languages.

Requirements

- *Could have* requirements.
  - Numeric order of questions.
  - Custom questions frequency.
  - Other languages.
  - Password change.

Requirements

- *Could have* requirements.
  - Numeric order of questions.
  - Custom questions frequency.
  - Other languages.
  - Password change.
  - Two factor authentication.

Requirements

- *Could have* requirements.
  - Numeric order of questions.
  - Custom questions frequency.
  - Other languages.
  - Password change.
  - Two factor authentication.
  - Groups.

Requirements

- *Could have* requirements.
  - Numeric order of questions.
  - Custom questions frequency.
  - Other languages.
  - Password change.
  - Two factor authentication.
  - Groups.
  - Delete data from users.

Requirements

- *Could have* requirements.
  - Numeric order of questions.
  - Custom questions frequency.
  - Other languages.
  - Password change.
  - Two factor authentication.
  - Groups.
  - Delete data from users.
  - View and modify profile data.

Requirements

- *Could have* requirements.
  - Numeric order of questions.
  - Custom questions frequency.
  - Other languages.
  - Password change.
  - Two factor authentication.
  - Groups.
  - Delete data from users.
  - View and modify profile data.
  - Assign questions to all patients.

Requirements

- *Could have* requirements.
  - Numeric order of questions.
  - Custom questions frequency.
  - Other languages.
  - Password change.
  - Two factor authentication.
  - Groups.
  - Delete data from users.
  - View and modify profile data.
  - Assign questions to all patients.
  - Timezones support.

Requirements

- *Could have* requirements.
  - Numeric order of questions.
  - Custom questions frequency.
  - Other languages.
  - Password change.
  - Two factor authentication.
  - Groups.
  - Delete data from users.
  - View and modify profile data.
  - Assign questions to all patients.
  - Timezones support.
- *Won't have* requirements.

Requirements

- *Could have* requirements.
  - Numeric order of questions.
  - Custom questions frequency.
  - Other languages.
  - Password change.
  - Two factor authentication.
  - Groups.
  - Delete data from users.
  - View and modify profile data.
  - Assign questions to all patients.
  - Timezones support.
- *Won't have* requirements.
  - Cross-platform.

Requirements

- *Could have* requirements.
  - Numeric order of questions.
  - Custom questions frequency.
  - Other languages.
  - Password change.
  - Two factor authentication.
  - Groups.
  - Delete data from users.
  - View and modify profile data.
  - Assign questions to all patients.
  - Timezones support.
- *Won't have* requirements.
  - Cross-platform.
  - Share the retrieved data with third parties.

Index

Architecture

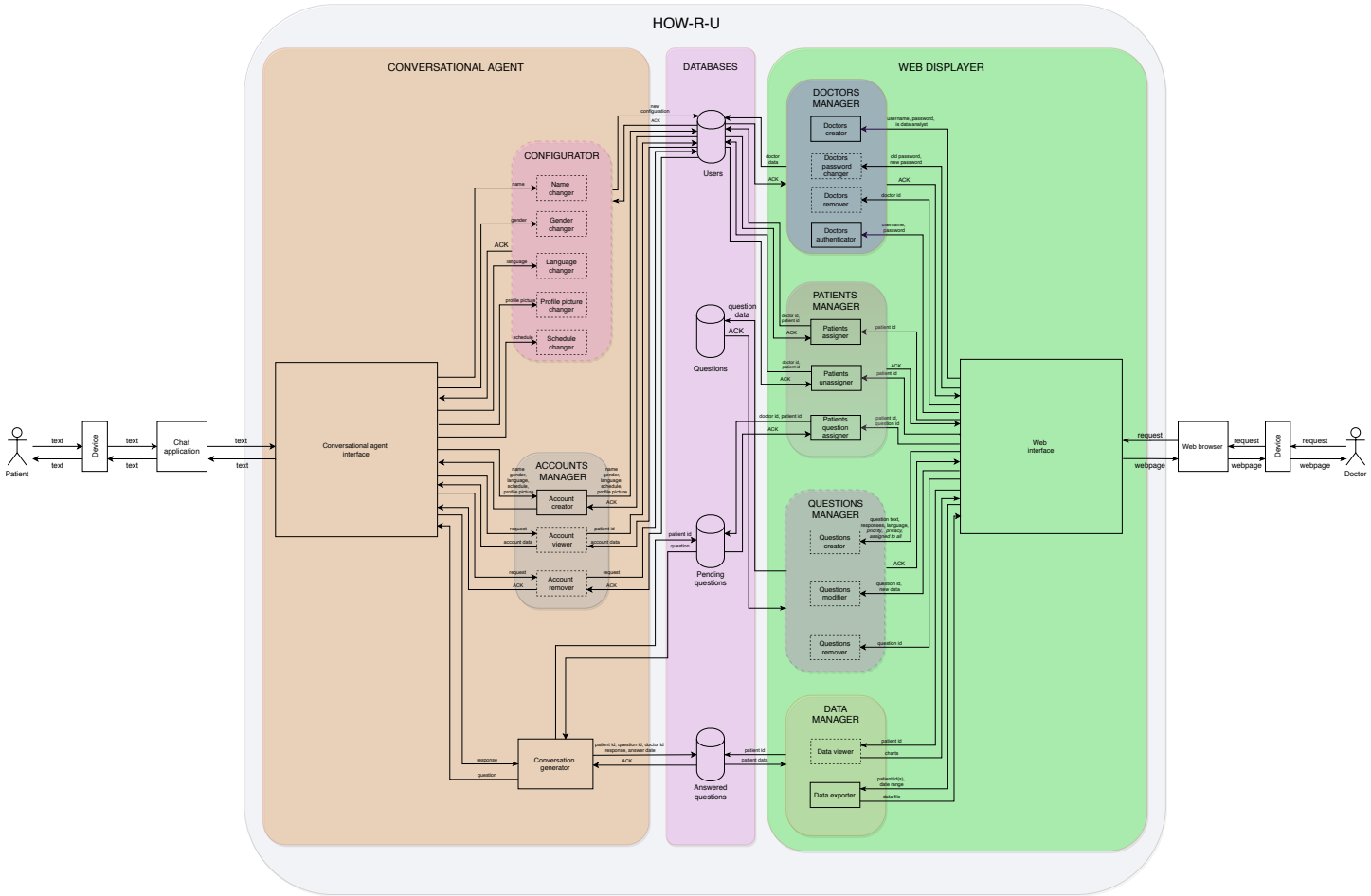

Figure: System architecture. Created using *diagrams.net* (*diagrams.net*, 2020).

Architecture

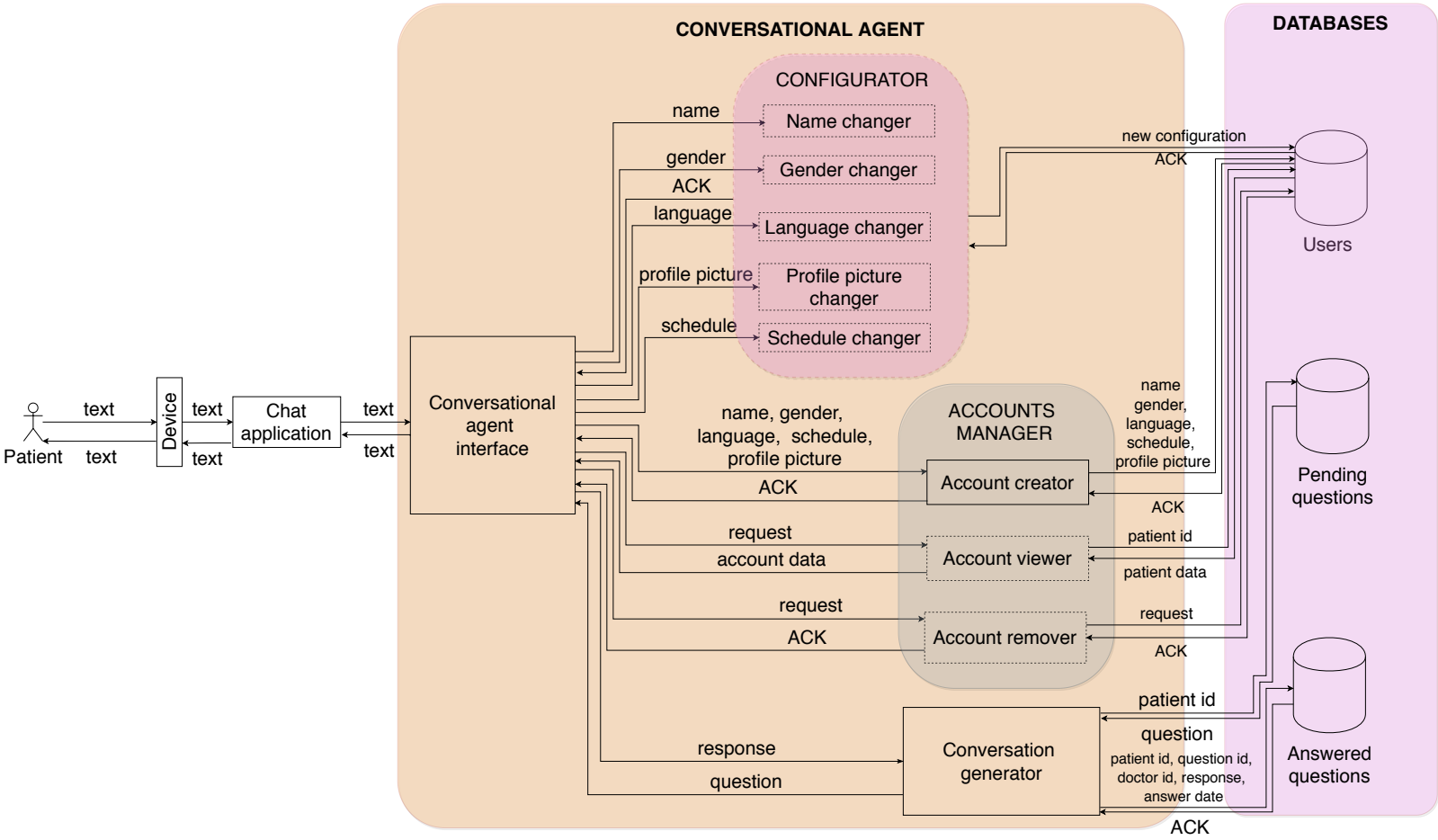

Figure: System architecture (conversational agent and databases).

Architecture

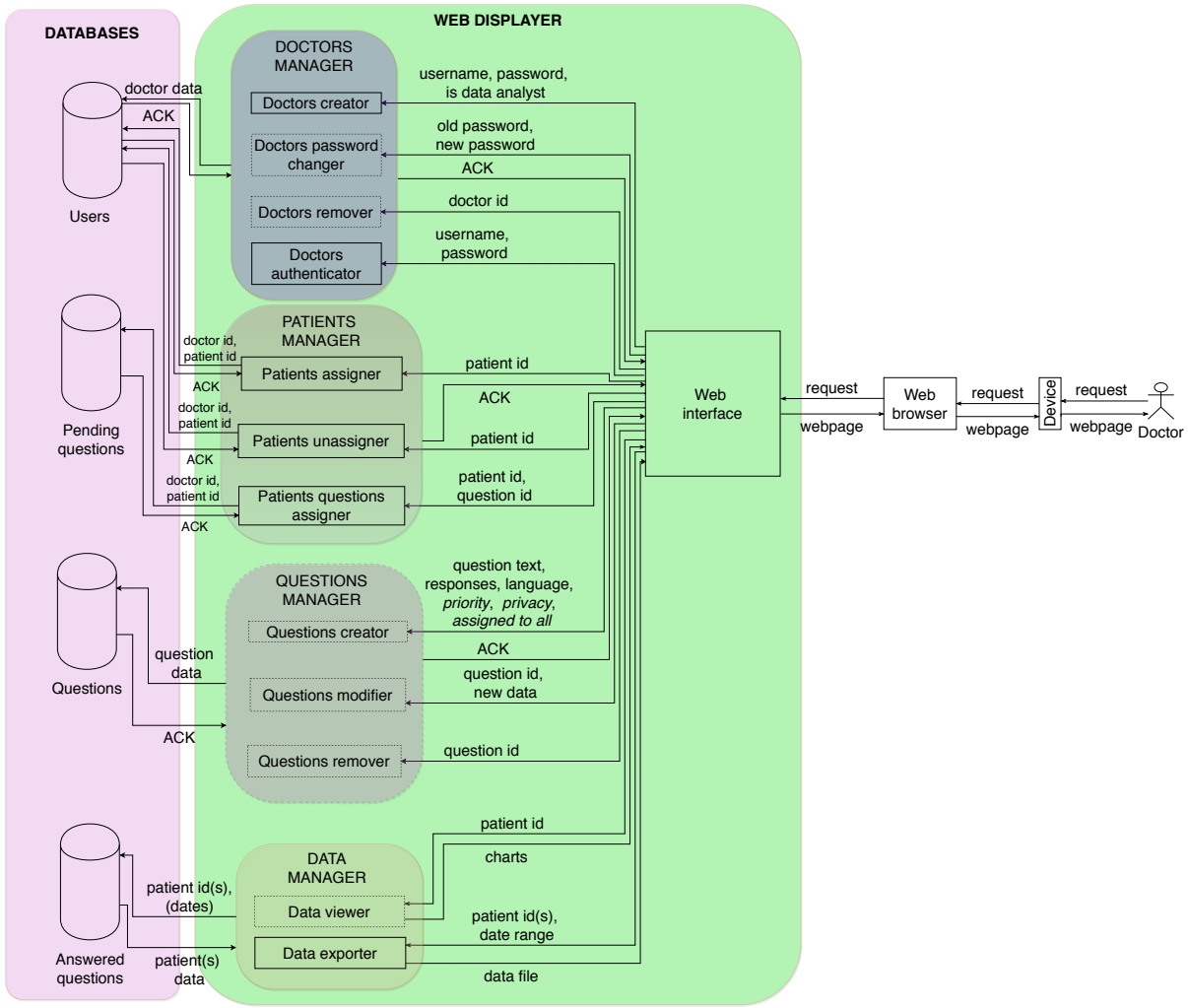

Figure: System architecture (web interface and databases).

Index

Implementation (databases)

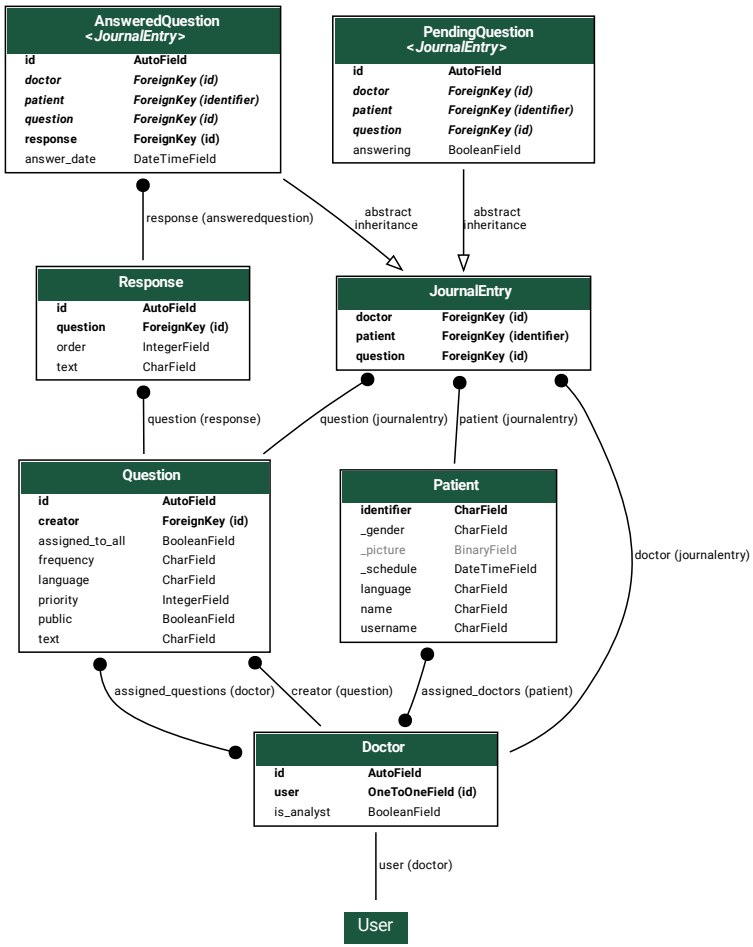

Figure: System class diagram (HOW-R-U classes)

Implementation (conversational agent)

- Handlers.

Implementation (conversational agent)

- Handlers.
  - Start handler

Implementation (conversational agent)

- Handlers.
  - Start handler
  - Config handler.

Implementation (conversational agent)

- Handlers.
  - Start handler
  - Config handler.
  - Question handler.

Implementation (conversational agent)

- Handlers.
  - Start handler
  - Config handler.
  - Question handler.
- Jobs.

Implementation (conversational agent)

- Handlers.
  - Start handler
  - Config handler.
  - Question handler.
- Jobs.
  - PendingQuestionJob

Implementation (conversational agent start handler)


```
GENDER, PICTURE, LANGUAGE, SCHEDULE = range(4)
@send_typing_action
def start(update, context):
    """
    Shows welcome message and asks for language
    """
    # Check that user is not registered
    try:
        patient = Patient.objects.get(identifier=update.message.from_user.id)
        logger.info(  f'User {update.message.from_user.username} tried to register
        ↪  again.')
        update.message.reply_text(text=messages[patient.language]['already_exists'])
        return ConversationHandler.END
    except Patient.DoesNotExist:
        # The user should not exist in DB
        context.user_data['patient'] = Patient(name=update.message.from_user.first_name,
        ↪  identifier=str(update.message.from_user.id),
        ↪  username=update.message.from_user.username)
        logger.info(f'User {update.message.from_user.username} started a new
        ↪  conversation')
        send_welcome_message(patient)
        send_language_selection(patient)
    return LANGUAGE
```

Implementation (conversational agent start handler)


```
start_handler = ConversationHandler(
    entry_points=[CommandHandler('start', start)],
    states={
        LANGUAGE:
        ↪  [MessageHandler(Filters.regex(f'^({Flag.flag("es")}|{Flag.flag("gb")})$'),
        ↪  language)],
        GENDER:
        ↪  [MessageHandler(Filters.regex('^(Male|Female|Other|Masculino|Femenino|Otro)$'),
        ↪  gender)],
        PICTURE: [MessageHandler(Filters.photo, picture), CommandHandler('skip',
        ↪  skip_picture)],
        SCHEDULE: [MessageHandler(Filters.regex('^([0-1]?[0-9]|2[0-3]):[0-5][0-9]$'),
        ↪  schedule)]
    },
    fallbacks=[]
)
```

Implementation (conversational agent PendingQuestionJob)


```python
class PendingQuestionJob(object):

  def __init__(self, context, patient):
      self.patient = patient
      self._create_job(context)

  def _create_job(self, context):
      context.job_queue.run_daily(callback=self.job_callback, time=self.patient.schedule,
      ↪  name=f'{self.patient.identifier}_pending_questions_job')

  def job_callback(self, context):
      pending_questions = self._get_pending_questions()
      for task in pending_questions:
          if not self.is_question_answered(task):
            task.answering = True
            ask_question(task, self.patient)
      send_message(self.patient, "All questions have been answered")
      if was_configurator_running(self.patient.identifier, context):
        reopen_configurator(self.patient)
```

Implementation (conversational agent)

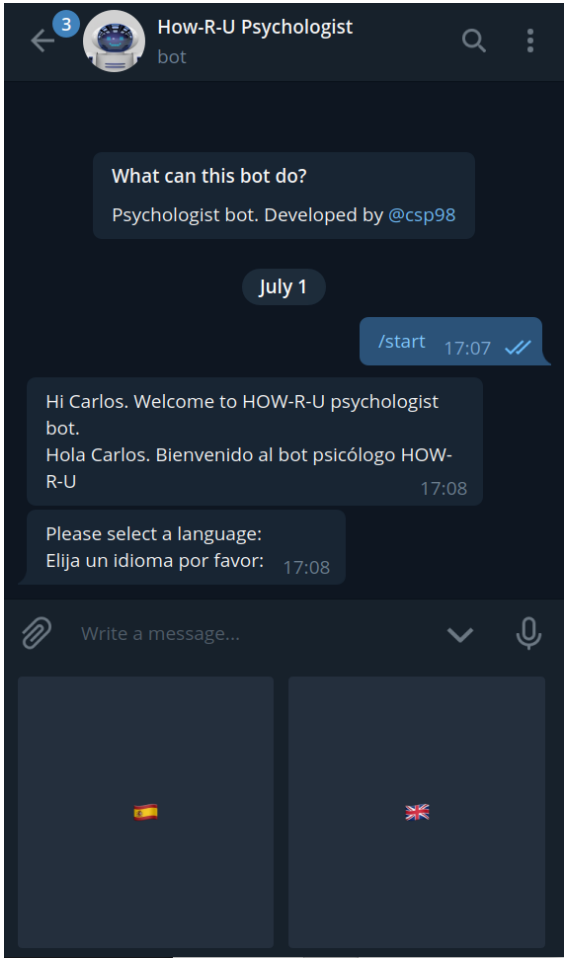

Figure: Agent showing the welcome message and asking for language selection.

Implementation (conversational agent)

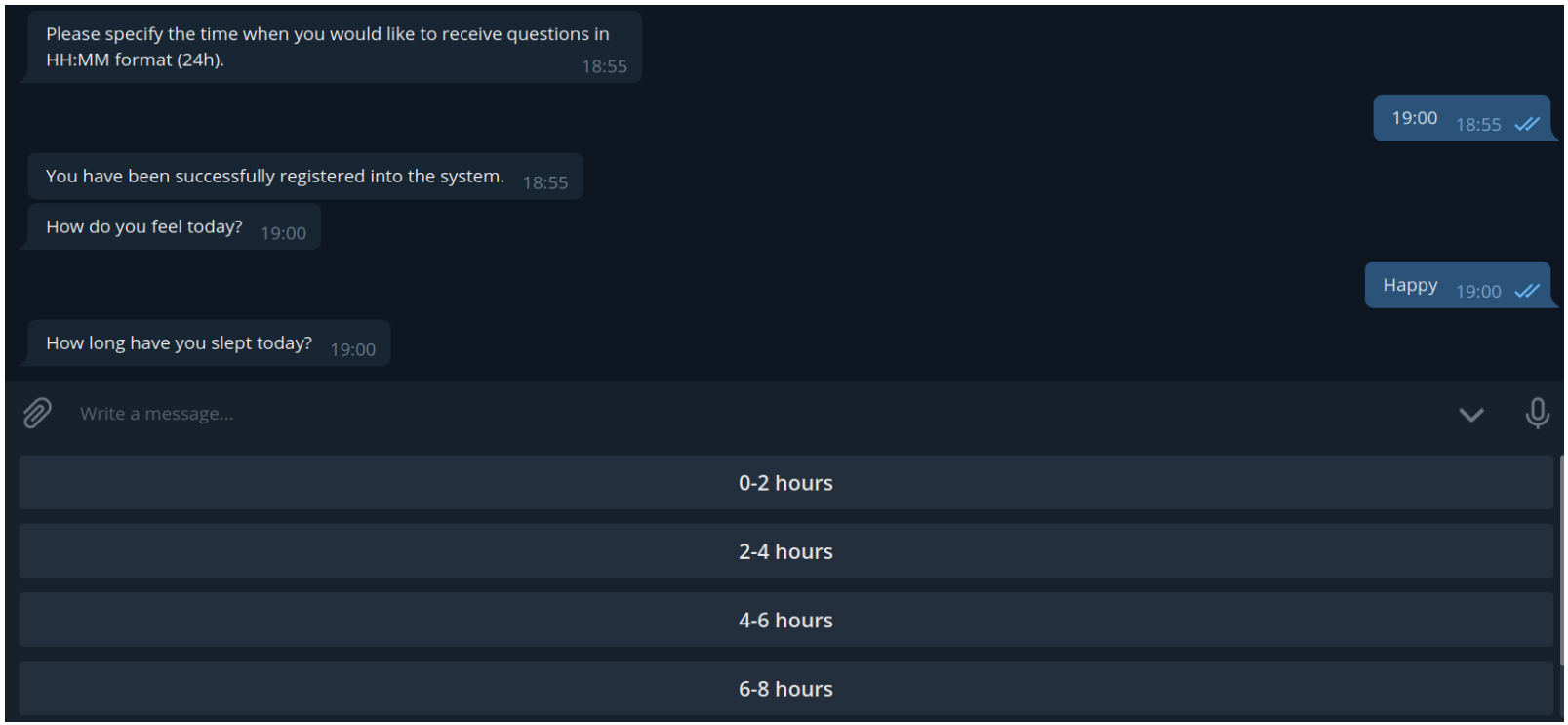

Figure: HOW-R-U converstional agent asking a question to a patient.

Implementation (web interface)

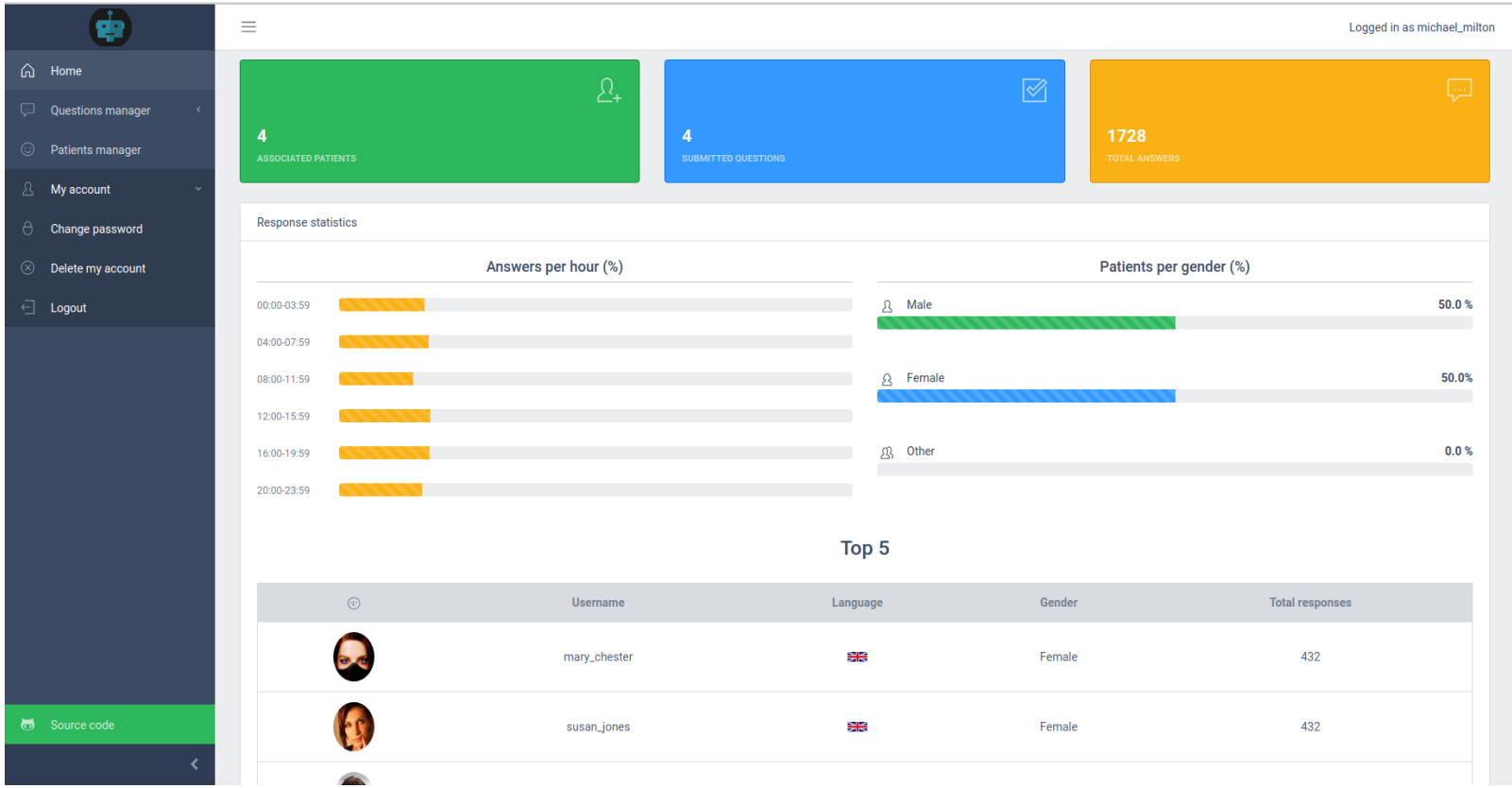

Figure: HOW-R-U homepage.

Implementation (web interface)


```python
@login_required(login_url="/login/")
def index(request):
    """
    Shows the index page, including global parameters (top patients, number of
↪   associated patients, answers, gender and time percentages, etc.,)
    """
    doctor = request.user.doctor
    top_patients = get_top_patients(doctor)
    doctor_patients = doctor.patient_set
    number_associated_patients = doctor_patients.count()
    submitted_questions = Question.objects.filter(creator=doctor).count()
    total_answers = get_total_answers(doctor)
    male_percentage, female_percentage, other_percentage = get_gender_stats(doctor,
    ↪  number_associated_patients)
    answers_per_hour = get_answers_per_hour(doctor)
    context = {
        "top_patients": top_patients,
        "number_associated_patients": number_associated_patients,
        "submitted_questions": submitted_questions,
        "total_answers": total_answers,
        "male_percentage": male_percentage,
        "female_percentage": female_percentage,
        "other_percentage": other_percentage,
        "answers_per_hour": answers_per_hour
    }
    return render(request, "index.html", context)
```

Implementation (web interface)

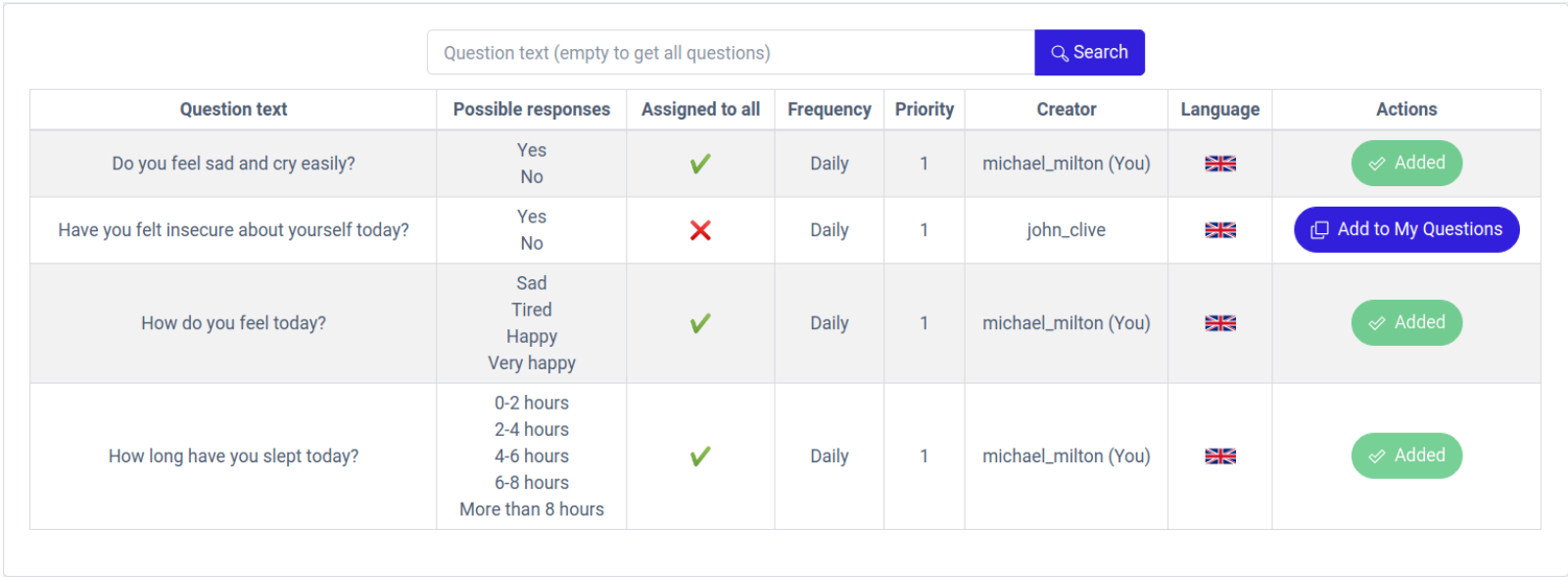

Figure: Public questions page.

Implementation (web interface)

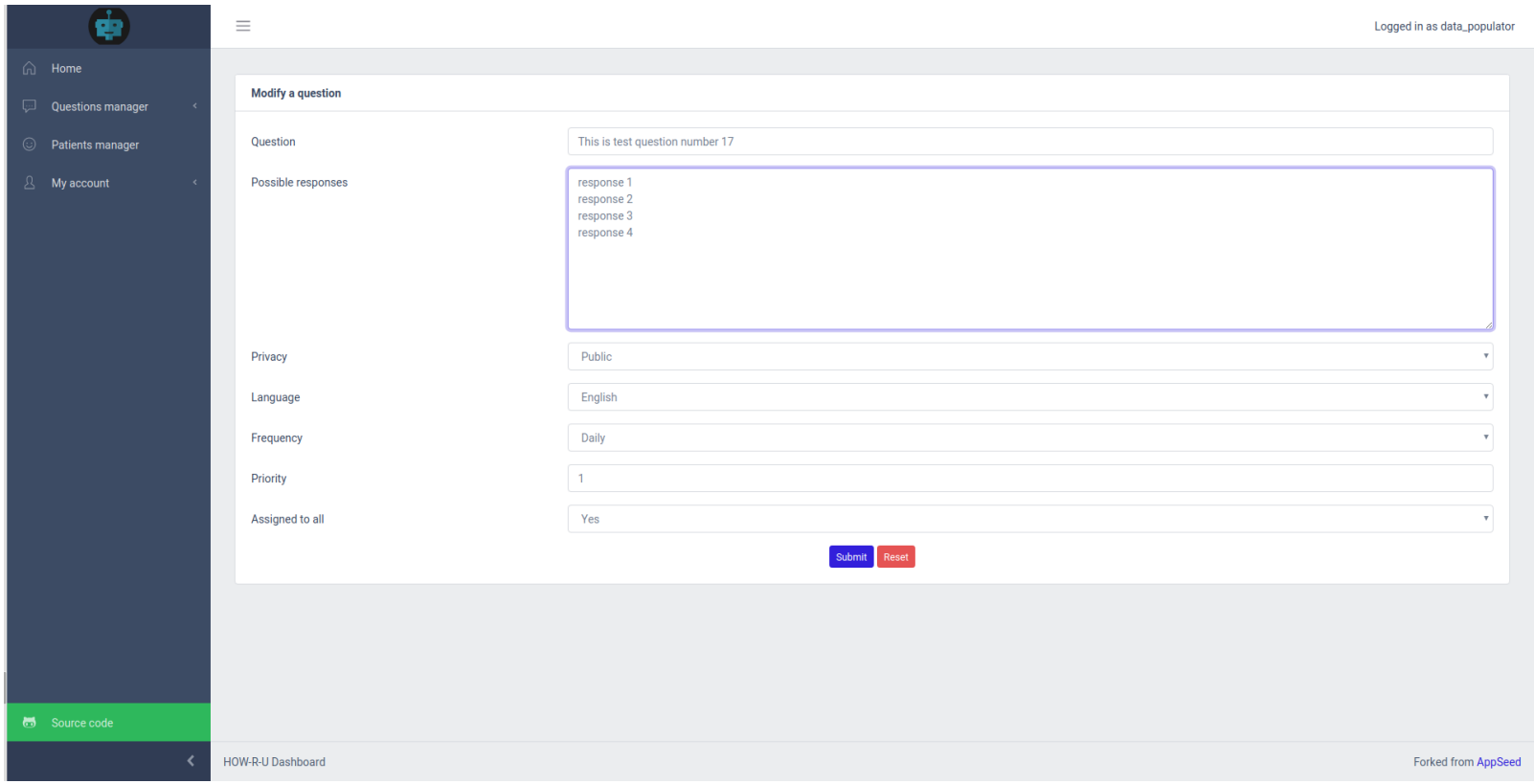

Figure: Questions creator and modifier.

Implementation (web interface)

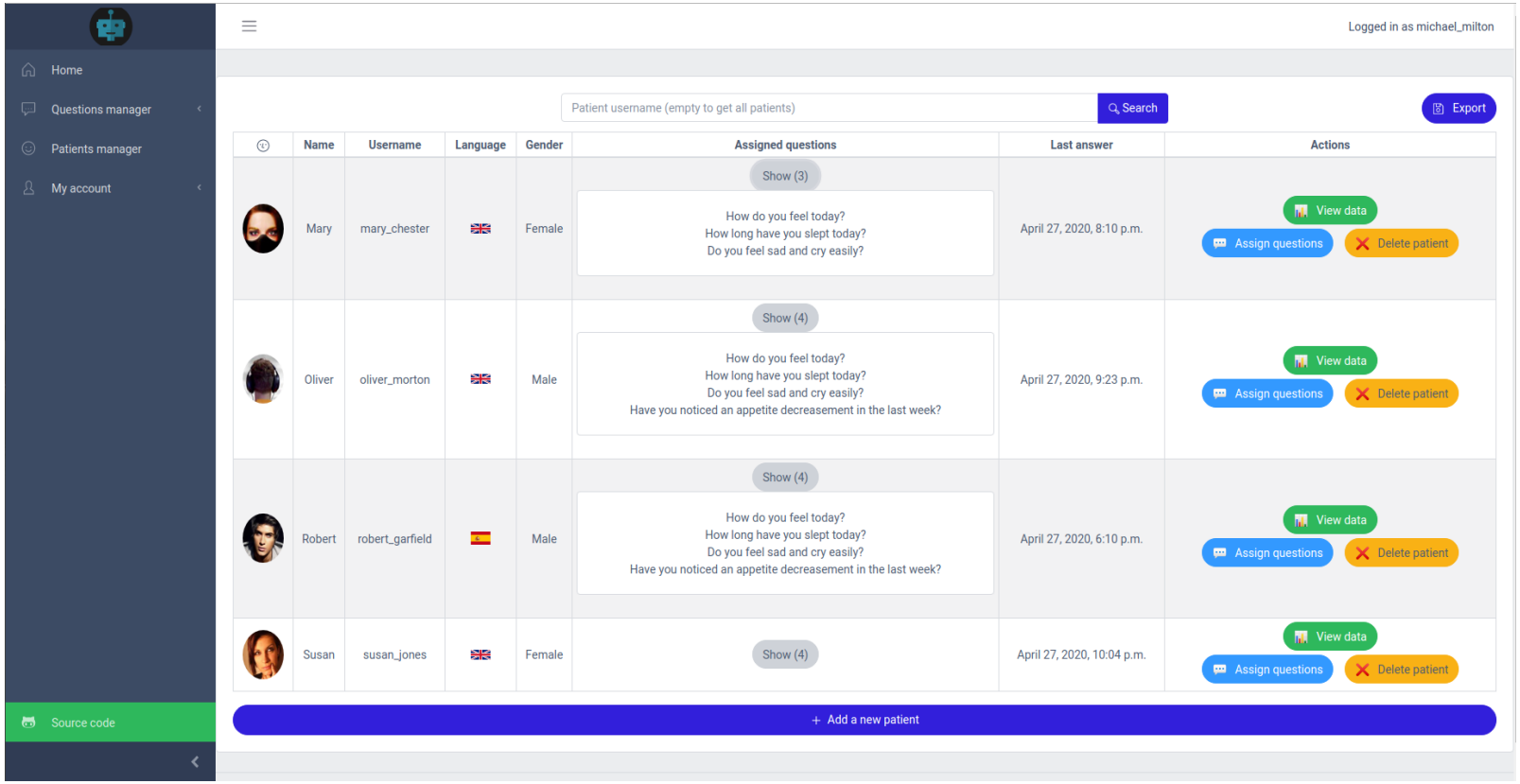

Figure: Patients manager.

Implementation (web interface)

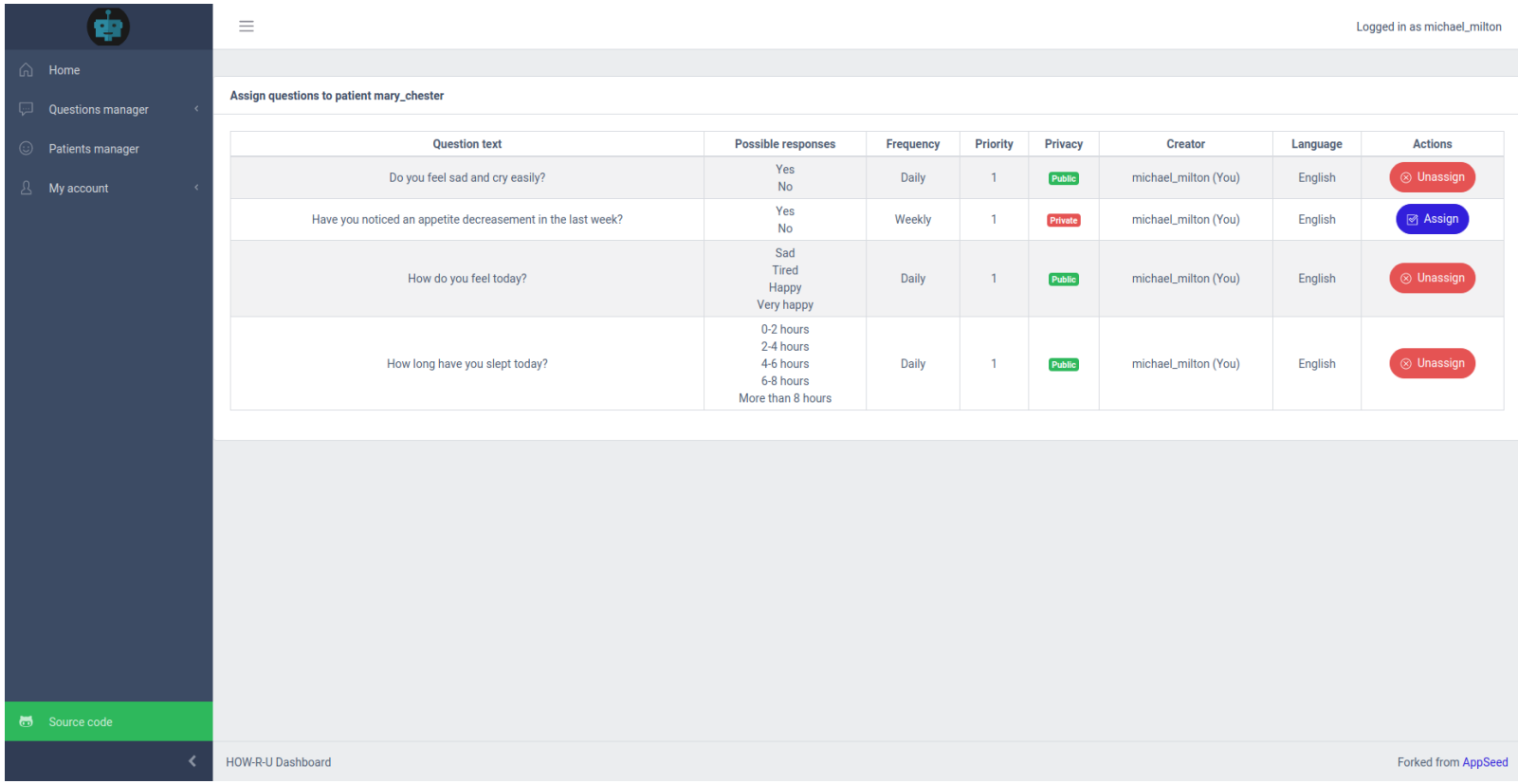

Figure: Patients manager assign questions page.

Implementation (web interface)

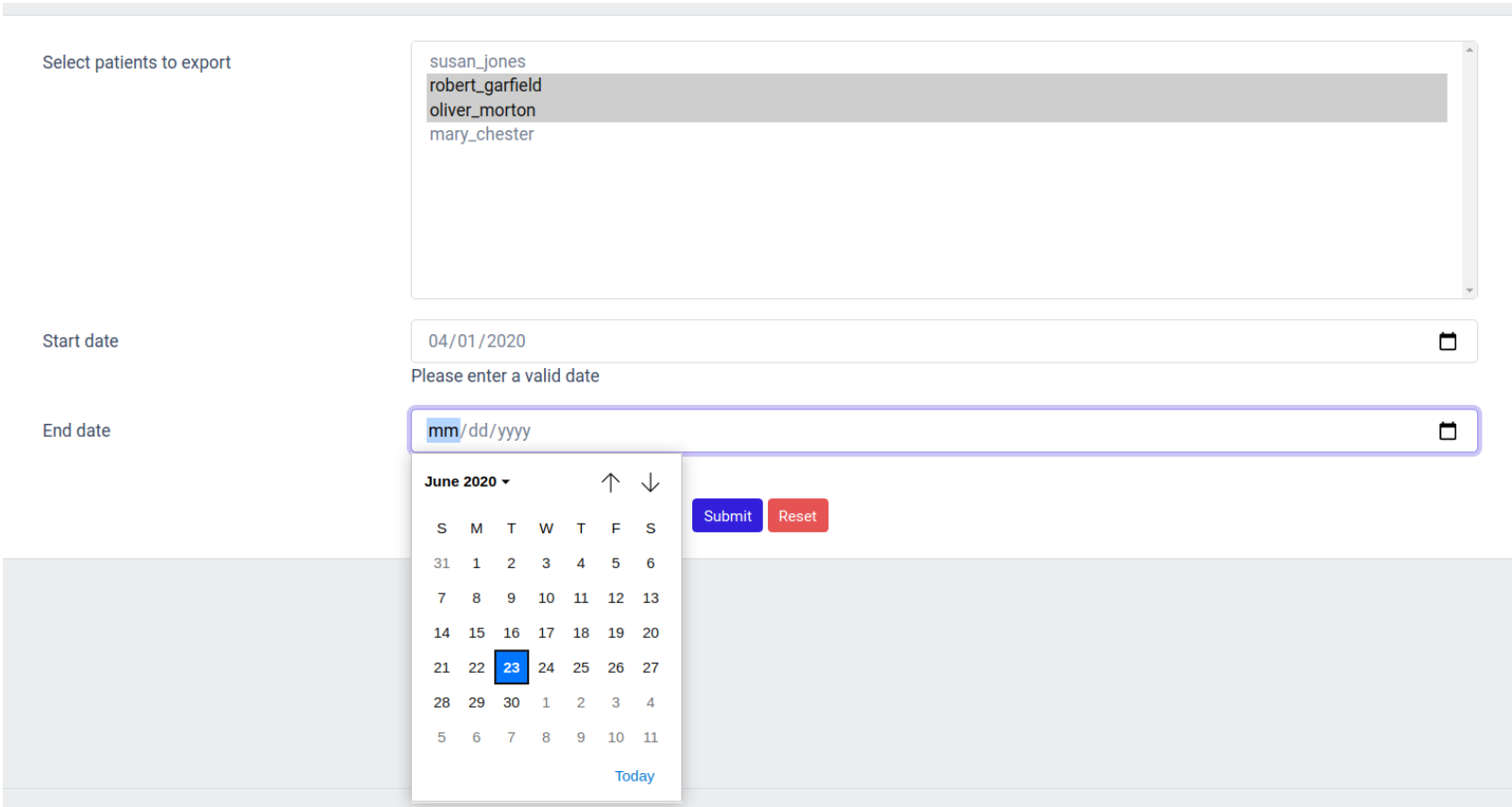

Figure: Export page

Implementation (web interface)

Patient username	Question	Answer	Date
robert_garfield	How do you feel today?	Tired	2020-04-01 04:12:20
robert_garfield	How do you feel today?	Sad	2020-04-02 05:24:20
robert_garfield	How long have you slept today?	0-2 hours	2020-04-01 12:56:20
robert_garfield	How long have you slept today?	4-6 hours	2020-04-02 11:28:20
robert_garfield	Do you feel sad and cry easily?	Yes	2020-04-01 14:22:21
robert_garfield	Do you feel sad and cry easily?	No	2020-04-02 06:31:21
oliver_morton	How do you feel today?	Very happy	2020-04-01 01:45:19
oliver_morton	How do you feel today?	Happy	2020-04-02 02:59:19
oliver_morton	How long have you slept today?	More than 8 hours	2020-04-01 20:51:20
oliver_morton	How long have you slept today?	2-4 hours	2020-04-01 22:24:20
oliver_morton	Do you feel sad and cry easily?	No	2020-04-01 07:47:20
oliver_morton	Do you feel sad and cry easily?	Yes	2020-04-02 17:26:20
oliver_morton	Have you noticed an appetite decreasement in the last week?	Yes	2020-04-01 21:41:20
oliver_morton	Have you noticed an appetite decreasement in the last week?	Yes	2020-04-08 21:09:20

Table: Example data generated with the *Export* feature.

Implementation (web interface)

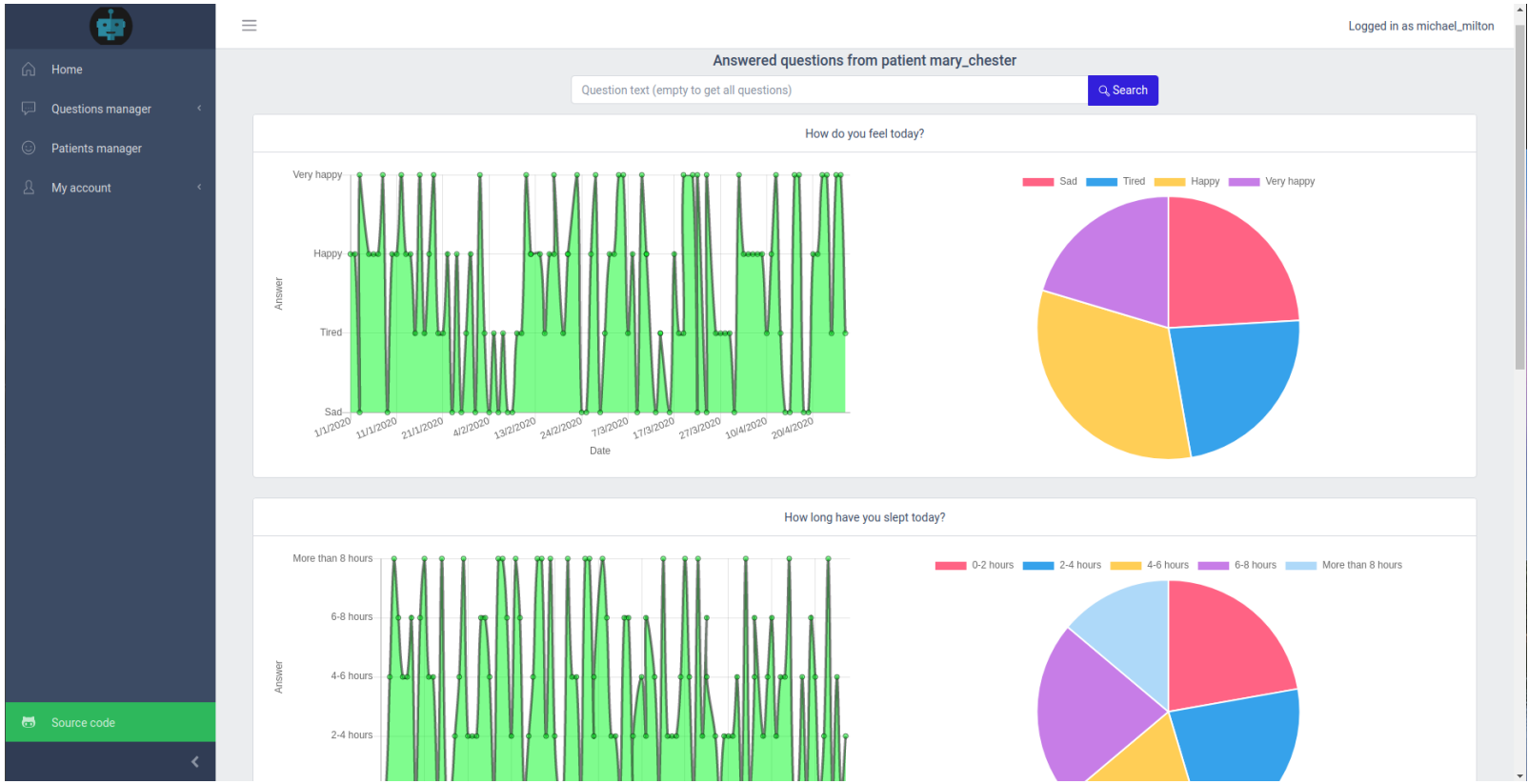

Figure: View data page.

Index

Index

Development environment

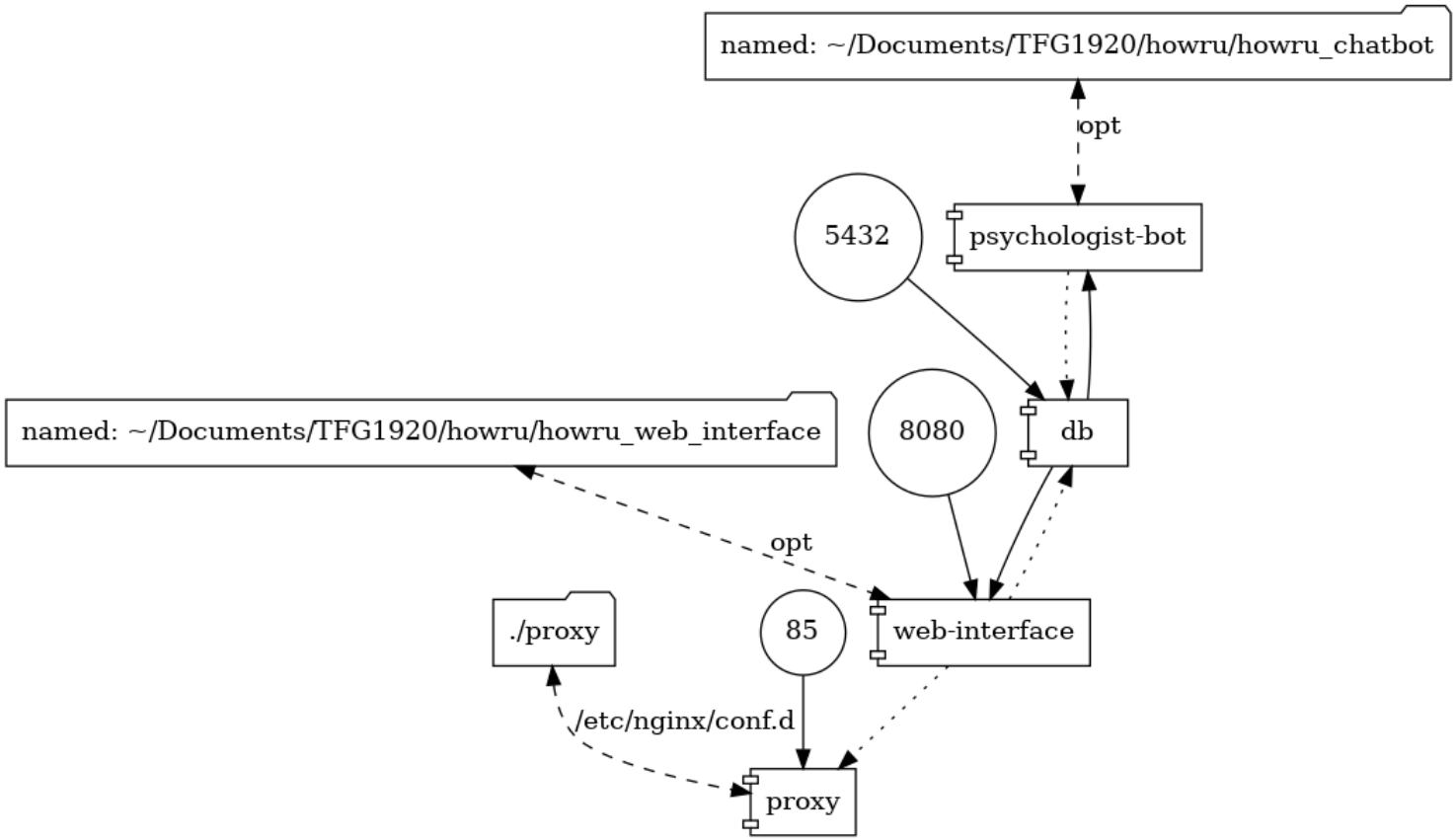

Figure: Docker-compose file schema. Generated by (PSIH, 2016).

Index

Production scalable environment

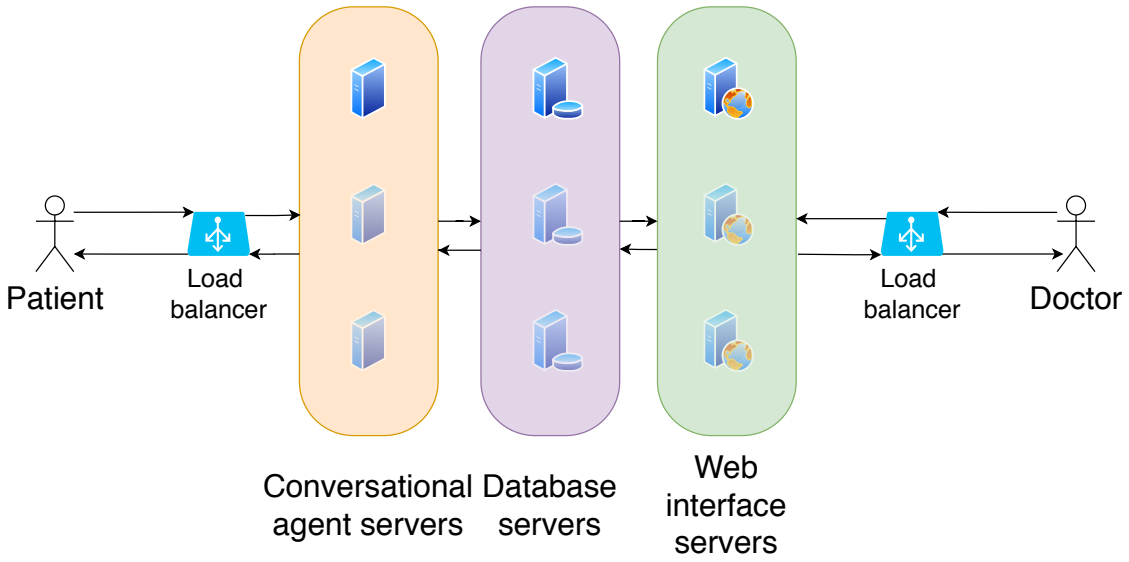

Figure: Scalable environment architecture diagram. Created using *diagrams.net* (*diagrams.net*, 2020).

Production non-scalable environment

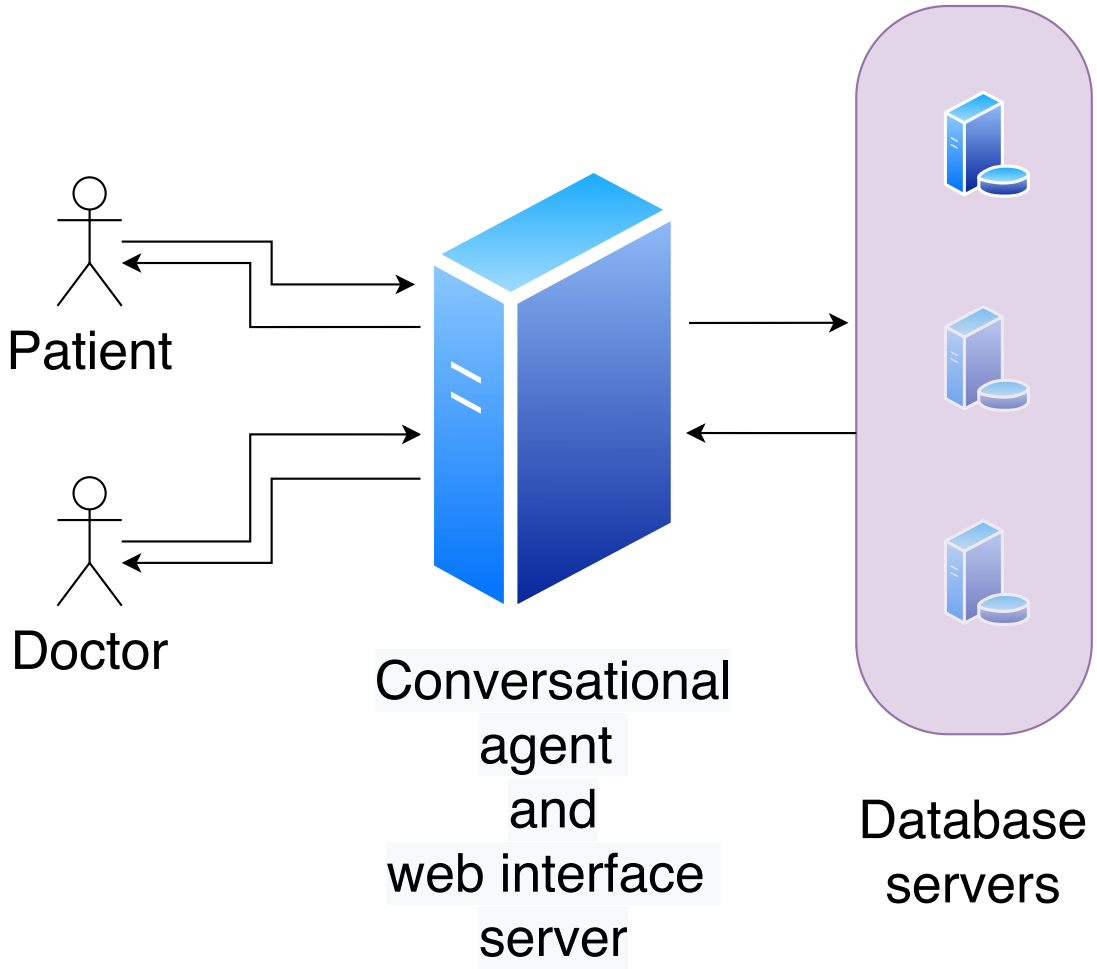

Figure: Non-scalable environment architecture diagram. Created using *diagrams.net* (*diagrams.net*, 2020).

Production one-instance environment

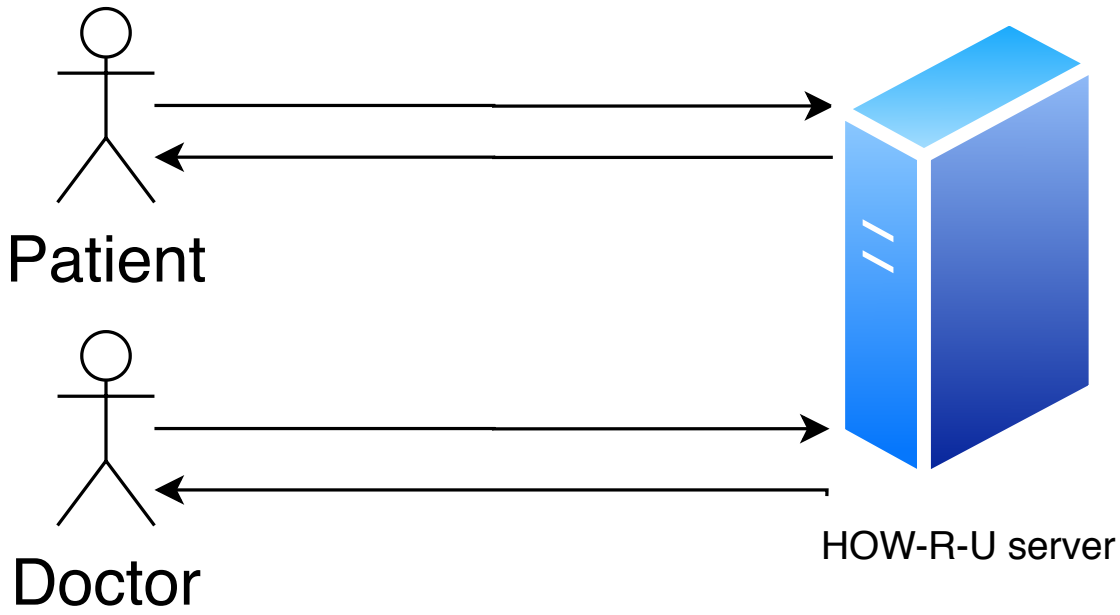

Figure: One-instance environment architecture diagram. Created using *diagrams.net* (*diagrams.net*, 2020).

Index

Discussion

STRENGTHS

1. Different scenarios.
2. Multiple e-coaches.
3. Intuitive interface.
4. No extra software for patients.

WEAKNESSES

1. Telegram is not used by the majority of people.
2. System deployment requires knowledge on computer science.

OPPORTUNITIES

1. COVID-19 data analysis.
2. Telegram ensures privacy.

THREATS

1. Elder people may not know how to use Telegram.
2. If patients do not have Telegram installed, they could install a specific app instead of Telegram.
3. Not all patients have smartphones.

Figure: HOW-R-U SWOT analysis. Based on http://www.mostlycolor.ch/2015/07/swot-matrices-in-latex.html.

Conclussions

- **Main goal**: *Conversational-agent-as-a-sensor* that asks questions defined by specialists to patients.

Conclussions

- **Main goal**: *Conversational-agent-as-a-sensor* that asks questions defined by specialists to patients. ✔
- **Secondary goals**.

Conclussions

- **Main goal**: *Conversational-agent-as-a-sensor* that asks questions defined by specialists to patients. ✔
- **Secondary goals**.
  - Graphical web interface for doctors.

Conclussions

- **Main goal**: *Conversational-agent-as-a-sensor* that asks questions defined by specialists to patients. ✔
- **Secondary goals**.
  - Graphical web interface for doctors. ✔
  - Flexible and scalable architecture to add functionality to the system.

Conclussions

- **Main goal**: *Conversational-agent-as-a-sensor* that asks questions defined by specialists to patients. ✔
- **Secondary goals**.
  - Graphical web interface for doctors. ✔
  - Flexible and scalable architecture to add functionality to the system. ✔
  - Architecture based on containers to host the different system modules.

Conclussions

- **Main goal**: *Conversational-agent-as-a-sensor* that asks questions defined by specialists to patients. ✔
- **Secondary goals**.
  - Graphical web interface for doctors. ✔
  - Flexible and scalable architecture to add functionality to the system. ✔
  - Architecture based on containers to host the different system modules. ✔
  - Implement a system that covers the previous goals.

Conclussions

- **Main goal**: *Conversational-agent-as-a-sensor* that asks questions defined by specialists to patients. ✔
- **Secondary goals**.
  - Graphical web interface for doctors. ✔
  - Flexible and scalable architecture to add functionality to the system. ✔
  - Architecture based on containers to host the different system modules. ✔
  - Implement a system that covers the previous goals. ✔
  - Test a beta version of the system to retrieve target audience's feelings about it.

Conclussions

- **Main goal**: *Conversational-agent-as-a-sensor* that asks questions defined by specialists to patients. ✔
- **Secondary goals**.
  - Graphical web interface for doctors. ✔
  - Flexible and scalable architecture to add functionality to the system. ✔
  - Architecture based on containers to host the different system modules. ✔
  - Implement a system that covers the previous goals. ✔
  - Test a beta version of the system to retrieve target audience's feelings about it. ✗

Bibliography I

Creative commons. (2001). https://creativecommons.org/.

diagrams.net. (2020). Retrieved from https://www.diagrams.net/

Discord. (2015). https://discordapp.com/.

Facebook messenger. (2008). https://www.facebook.com/messenger/.

Kik. (2010). https://www.kik.com/.

Kosinski, M., Stillwell, D., & Graepel, T. (2013). Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, *110*(15), 5802–5805.

Line. (2012). https://line.me/.

PSIH, G. (2016). *docker-compose-viz.* https://github.com/pmsipilot/docker-compose-viz.

Bibliography II

Rastogi, N., & Hendler, J. (2017, 01). Whatsapp security and role of metadata in preserving privacy.

Ritchie, H., & Roser, M. (2018). Mental health. *Our World in Data*.

Slack. (2013). `https://slack.com/`.

Sutikno, T., Handayani, L., Stiawan, D., Riyadi, M., & Subroto, I. (2016, 06). Whatsapp, viber and telegram which is best for instant messaging? *International Journal of Electrical and Computer Engineering (IJECE)*, *6*, 909.

Telegram. (2013). `https://telegram.org/`.

Viber. (2010). `https://www.viber.com/`.

Vicente, D. (2019). La ratio de psicólogos, a 16 puntos de europa. *El Mundo*.

Wechat. (2011). `https://www.wechat.com/`.

Whatsapp. (2009). `https://www.whatsapp.com/`.

The end

Time for questions