**Compassionate Officer Program (COP)**

**Technical Partnership Document**

Do you get excited about the idea of gamifying good communication skills? Then there is a way for you to contribute.

# Goals of this Document

This document assumes that you have read Compassionate Office Program (COP) Overview document which describes the desired *end state* of this project. This is a living document. As the project progresses, additional information will be added to this document

Table of Contents

1. Describe the current state of development of the software
2. Explain the expected phases of development of the software, as it grows from a simple computer-based application to a VR experience.
3. **Provide the information needed to enable partners participate in the development**

Terms used in this document.

* **NPC** = Non-Player Character. This is the computer-generated personality
* **Trainee** = The person doing the training with the software.
* **Prompt** or **NPC prompt** = The programmed prompt that the trainee should respond to. (e.g. “I feel scared when you talk like that.”)
* **User input** or **Trainee input** = This is the trainee’s response. (e.g. “Do you need safety?)
* **Conversational Unit** = It is the NPC prompt AND the trainee’s response. Several scenarios can be linked together to create a conversation.
* **Contributor** = This is a person (like you!) that wants to contribute to this project
* **Cookie** = It is a small piece of data that is stored while a trainee is practicing. The cookie data keeps track of the current state of the game.
* **NVC** = nonviolent communication

# Prerequisites

Have a basic understanding of nonviolent communication techniques

* Read Thom Bond’s book to get an understanding of Non-Violent Communication. We are happy to provide you the book to read if you want to contribute.

Understand that we will be taking a systematic approach to NVC based on the experience from first responders. Which follows this model:

1.) Be able to objectively understand the environment without judgements or assumptions  
 2.) Correctly identify the feelings involved and validate the feelings with the person  
 3.) Correctly identify the needs involved and validate the needs with the person  
 4.) Negotiate a strategy with other person that meets your needs and their needs

Underlying this approach is the assumption that the first responder has done the 4 steps above on themselves (before doing them with someone else), such that they understand their own feelings and needs, and appropriate strategies in the situation.

# 1.) Current State of Development of the Software

The 1st phase of development is to create The Needs Game. The Needs Game is a simple game which gives a prompt and requires the trainee to guess the correct need. There are currently 7 conversational units that are based on the content provided by Thom Bond, each requiring the trainee to guess a different need.

Software is ready to play with at: *https://the-needs-game.uc.r.appspot.com*

Features:

* Basic game-play format: NPC prompt 🡪 Trainee Response 🡪 *repeat.*  \\
* There are 7 prompts that have a defined order.
* The system simply looks for the keyword in the answer. For example, if the NPC prompt is “I feel scared when you talk like that.” The keyword would be “safety”. So, ANY of the following would the trainee responses would be acceptable:
  + “Do you need safety”
  + “Safety”
  + “I’m guessing you need safety”
  + “Anything safety anything”
  + *Even*, “ You don’t need safety” *Future enhancements can remove this option.*

Limitations:

As of March 1st , the software only can support 1 user at a time. (If multiple people interact with it at the same time, then the software treats all the inputs as coming from the same person, which can be confusing.) Said technically, currently the application can support only one user session at a time.)

Next steps:

By March 20th, the software should be able to support an infinite number of trainees, limited only by the amount of computers we want to rent for the purpose. Mike’s project for March is to enable the software so each trainee has a separate session to interact with the system. (Mike intends to use client-side or server-side cookies to maintain the state for a user.)

Possible Enhancements:

There are a million little enhancements that could be made. Such as

* Add more ‘need’ prompts
* making it look prettier,
* showing questions in a random order
* working more on the word recognition for issues, like “You don’t need safety”
* automatically forward trainee to next prompt if they make too many bad guesses

Cost

* The software is currently hosted on Google Cloud. The costs are *currently* only about $75 dollars a month, but it will grow depending on many people use it. For the next several months, Mike will absorb the costs, and keep a record of them.

# 2.) Phases of Development

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Title | Description | Where | When |
| 1 | Guess the need | A simple computer game where the trainee is given a single prompt (e.g. “I feel scared when you talk like that.”) and the trainee needs to type the single correct needs as a response.(“Safety”)  This is the current stage as of March 10th, 2022 | Normal computer or mobile device | March  2022 |
| 2 | Multi-step conversations | Like above but a more complex version that can allow multiple steps (feelings and needs) in a conversation | Normal computer or mobile device | May |
| 3 | Simple VR version of game | Like above and create a VR environment where an NPC embodies the conversation | Computer and VR | Sept |
| 4 | Complex VR | Create more complex models based on AI or other technology | VR |  |

For each phase assume:

* It will work, but it may not be pretty. It may take a little extra time to beautify the project.

# 3.) Enabling Partners to Participate in Development

**Both technical and non-technical people can contribute to this project.** As Marshall Rosenburg said, “We live in an abundant world.”

Below is a description ***some*** of the technical ways to contribute to the game development. As new team members are added, training sessions will be scheduled to deep dive into these areas where they would like to contribute. **The ways to help fall into 3 categories:**

## **1.) Developing new scenarios for trainees to play with.**

1. **Add additional NPC prompts for trainees to guess the need**. – *Non-Technical*
2. Develop models for NPC responses. This can be based on AI, or other methods. *Technical.*

## **2.) Application Development**

1. Create nicer HTML interface for the Needs Game. *Technical*
2. Enhance model for scoring the trainee’s response. *Technical*
3. Enhance the core software – Add new features *Technical*
4. Clean up the Code to make it cleaner and easier to read and use. *Technical*
5. Optimize infrastructure – Ensure the software is running in the right place and right cost. *Technical*
6. DevOps Design and Management, *Technical*
7. Develop the VR design, artwork, or gameplay. *Technical*

## **3.) Other**

1. Program management - *Non-Technical & Technical*
2. General Software Testing – *Non-Technical & Technical*
3. Technical writer - Create documents for other contributors or trainees. *Non-Technical & Technical*
4. Logos and art development - *Non-Technical & Technical*
5. **Bring your skills!** Maybe you have a skill to contribute? *Non-Technical & Technical*

Each of these are discussed in detail below.

## **The ways to help fall into 3 categories: Detailed Below**

## 1.) Developing new scenarios for trainees to play with.

* **Add additional NPC prompts of needs for trainees to guess**. – *Non-Technical*
* Each one scenario(prompt-response) is controlled by a single Google spreadsheet. The spread sheet allows the author to define the prompt and the possible responses.
* The process for creating a new ‘need’ prompt is:
  + The contributor creates a new scenario based on a google spreadsheet template
    - Example spreadsheet can be found here : *Insert URL*
  + The contributor posts the spreadsheet to a common location
  + Thom reviews and approves the spreadsheet. Thom provides additional information, if necessary, such as the order of the prompts.
  + Mike converts spreadsheet to JSON document and imports into the application. (This process currently takes about 30 minutes to convert the spreadsheet to the required format, import it, then re-deploy the application with the changes. This is something that can be automated later.)
  + The new changes are tested by someone.
* **Develop new models for NPC responses**. This can be based on AI, or other methods that might enable a chatbot. *Technical*
* Assumptions –
  + The front end (user facing) of the Needs Game would remain the same.
  + The application code is stateless. Application state is managed in a cookie.
  + The new model does not need to worry about, starting, stopping, or scoring in the application, unless desired.
* The software will make a call to the new model (This can be an API call to an external location or can be supplied as a custom Python module to be added to the original software.) The call to the new model will be sent with user input and a cookie containing the application state.
* The new model will return the NPC response and a cookie.

## 2.) Application Development

* **Create nicer HTML interface for the Needs Game**. *Technical*
  + Assume a Python/Flask model.
  + The general look and feel are controlled by a HTML page and a CSS style sheet. Some formatting is embedded in the application, which is not ideal.
  + Logos to be developed also.
* **Enhance model for scoring the trainee’s response.** *Technical*
  + Assume:
    - The current scoring is based on TextBlob python module for word and sentiment analysis. There is also a scoring approach based on pattern matching that is embodied in a module, but not completely implemented.
    - The raw scoring is sent to a log, but nothing further, at this time.
    - Scoring is on both the trainee’s original input AND on a cleaned version that removes stop words (based on a partial list of NLTK stop words) and punctuation.
    - The scoring model will get user input and a cookie.
    - The code should be a python, as it will be added to the current code base.
  + Output
    - Scores returned in a JSON format or dictionary value.
* **Enhance the core software – Add features** *Technical*
  + Example types of feature enhancements
    - Showing questions in a random order
    - Making it easier to add additional questions by converting google spreadsheets to JSON documents and adding to the github repository.
    - Automatically forward trainee to next prompt if they make too many bad guesses
  + Mike can have a training session with a contributor that wants to make these types of enhancements.
* **Help Clean up the Code** *Technical*
  + A contributor can help clean up the code to make it cleaner and easier to read and use.
  + This can be a good first step for any technical contributor
* **Optimize infrastructure** – Make sure the software is running in the right place and at the right cost. *Technical*
  + Currently the application runs on Google AppEngine (Flex) with a defined set of resources. The app.yaml file, which controls deployment, can be changed so that the application resources scale up and down gracefully as needed. Scaling up and down as needed will help control costs for this project.
* **DevOps Design and Management** – If you don’t know what DevOps is then you can skip this section.
  + Since we might have several people contributing software code, we want a strategy for pulling together the different software branches, approvals, and deployment. *Technical*
* **Develop the VR** design, artwork, or gameplay. *Technical*
  + This is a little farther out, but if we identified a resource with VR development skills, we could start working on the design and artwork. *Technical*

## 3.) Other

* **Program Management** *Non-Technical*
  + If there are a several contributors, then it may become necessary for someone to fill the role of program manager for tracking changes, keeping schedules, projecting trainee usage, tracking costs, and such.
  + Like any team project, we can create a master list of potential enhancements to the system
    - The potential enhancements can be prioritized
    - The prioritized enhancements can be assigned to contributors to work on.
  + The program manager could meet with core team members to collect and document feature requests and coordinate meetings to discuss.
  + Provide a status update to help inform team members of new developments.
* **General Testing** – *Non-Technical & Technical*
  + We need people to check that the application works in a variety of computers and looking for spelling mistakes and other usability issues. - *Non-Technical*
  + Develop automated system tests so that we can quickly test new changes to the software. - *Technical*
* **Technical writer to create documents** to help people create question prompts or to contribute to the software. *Non-Technical & Technical*
  + Documents, like this one, need to be updated.
* **Logos and art development** - *Non-Technical & Technical*
  + There will be minimal need for artwork for the computer-based training, because I’m assuming that the computer-based training will be mostly text based.
  + As the project evolves into VR, there will be a large need for artwork because each scene will need a different visual environment. For example, for a VR scene where a trainee is called to help a domestic disturbance, we will need the artwork of the domestic scene that the trainee is walking into.
* **Bring your skills!** *Non-Technical & Technical*
  + Maybe you have a skill that you would like to contribute?

# Enabling Contributors

What do the core team members need to do to enable the contributors to work easily?

**Who to contact:**

The technical lead for the project is currently

Michael Roberts

[MikeRobertsIsHappy@gmail.com](mailto:MikeRobertsIsHappy@gmail.com)

Ph. 831-801-0807

**Create 2 versions/environments of the needs game**

Currently deployed to Google AppEngine as a Flex python/Flask app.

1.) Production Version – What trainees will use.

*https://the-needs-game.uc.r.appspot.com - Exists currently*

2.) Development Version – Where we can test new things inside our team.

*https://the-needs-game****-dev****.uc.r.appspot.com -* ***This needs to be created***

This will be our main working area.

**A place to keep track of all the enhancement requests with priority for each.** There are many different ways we can improve this software. We should make a list of all the different enhancement ideas and prioritize them. There are several different options for tracking enhancement requests. The simplest would be a spreadsheet, but we could also use [Github](https://docs.github.com/en/issues/trying-out-the-new-projects-experience/about-projects), JIRA or something else. (As the project evolves our way of tracking work will probably evolve.) The Program Manager will own the priority list.

**Mailing list of contributors.**We can create a way to communicate and keep the team up to date.