Biocall manual

Biocall

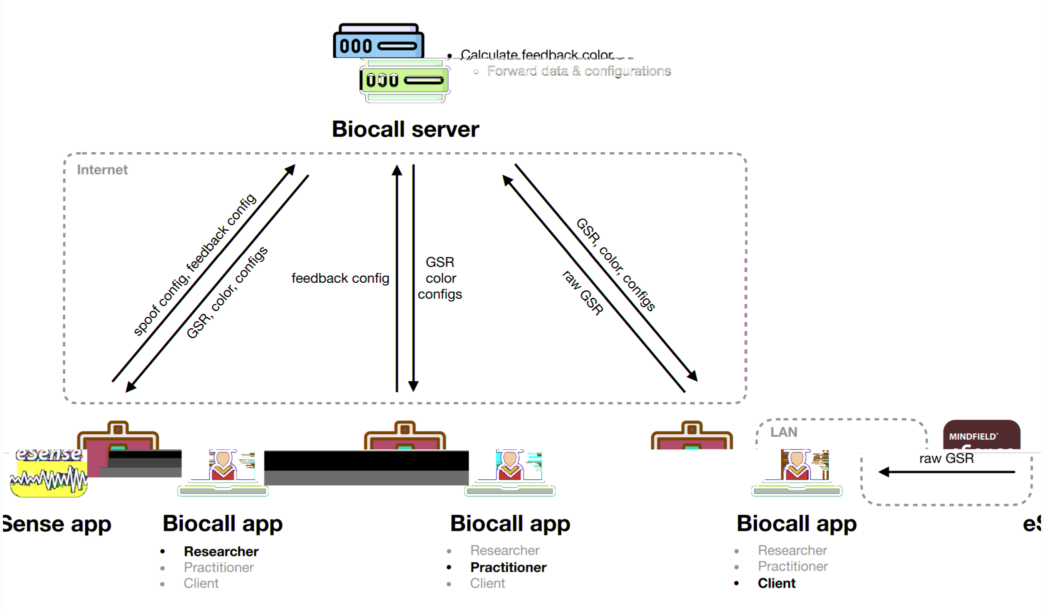
1 Overview

Biocall integrates real-time GSR data into video calls. The application is designed for psychotherapy experiments, in which a typical scenario involves 3 types of user: the researcher, practitioner, and client. In addition to seeing GSR feedback, users depending on their roles are able to decide the type of feedback that is available to the client and to send fake GSR data to spoof users for research use.

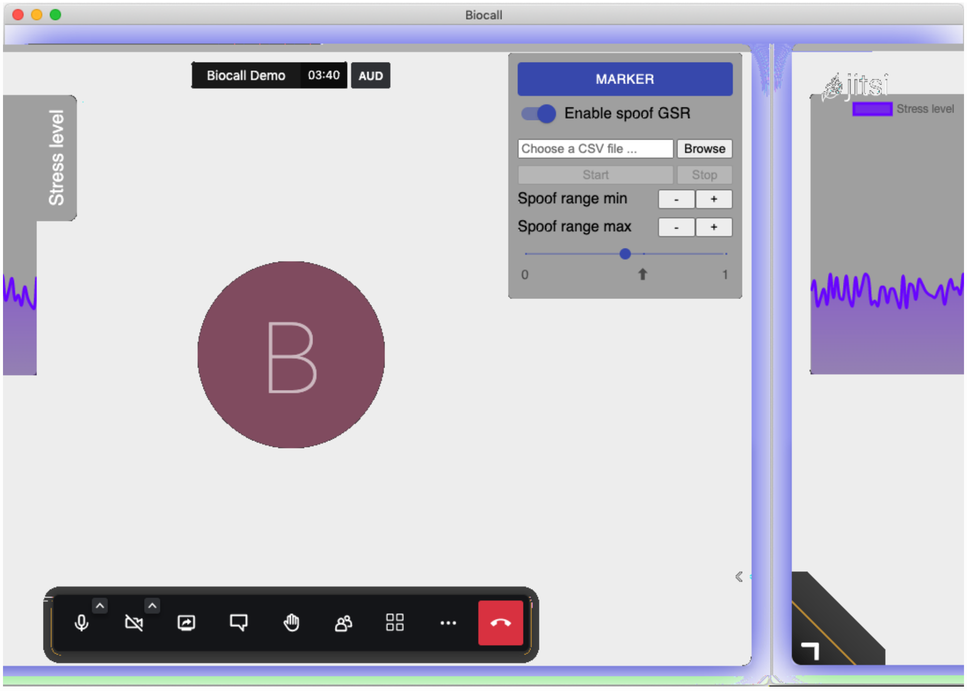
The complete Biocall application consists of 2 components, the Biocall desktop app (`Biocall app`) and the Biocall server (`Biocall server`).

- **Biocall-app** (source code): An all-in-one desktop solution made for users of different types to make video calls, collect GSR data, see feedback and perform various operations.
- **Biocall-server** (source code): A NodeJS server that relays data (GSR and configs) to all users in a video call.

Below shows the structure and data flow of Biocall.



2 Biocall Desktop Application



- **Client-view**
- **Practitioner-view**
- **Researcher-view**

2.1 Features

Role	Feedback visible	Configure feedback	Configure spoof	Export data
`researcher`	✔	✔**	✔	✔
`practitioner`	✔	✔		✔
`client`	✔*			

** Not always. Depends on feedback config.*

*** Not recommended. The feature is designed mainly for practitioner use.*

2.1.1 Physiological feedback

Users are able to see client's GSR data in real time. The GSR feedback is in the form of a line chart and a color border. Both form of feedback is always available to `researcher` and `practitioner`, whereas the feedback availability to `client` is determined by `researcher` and `practitioner`.

- **Line chart**: A line chart showing client's GSR values in the recent 60 seconds.

- **Color border**: A color-changing border around the main video view. The color is calculated based on the ratio of the current GSR to the maximum GSR received since the start of the call. The algorithm is designed to show a bluer color for a smaller GSR value, and a redder color for a larger GSR value. Check Color Calculation for the default algorithm.

2.1.2 Feedback control

`practitioner` and `researcher` can decide which type of feedback is visible to the client. Check 2.2.3 for user instructions.

2.1.3 Spoof

`researcher` is able to spoof other users with fake GSR data. Given a spoof value (`sValue`), the server will randomly pick a number between `sValue+0.1` and `sValue-0.1` to be the spoofed GSR value that will be shown in the feedback. The spoof value can be set from the spoof panel, which supports the following configurations. Check 2.2.4 for user instructions.

- **Marker**: Users are able to set a marker for the current second.
- **Input from file**: Users can get a series of spoof values from a file and send them one after a1er sp

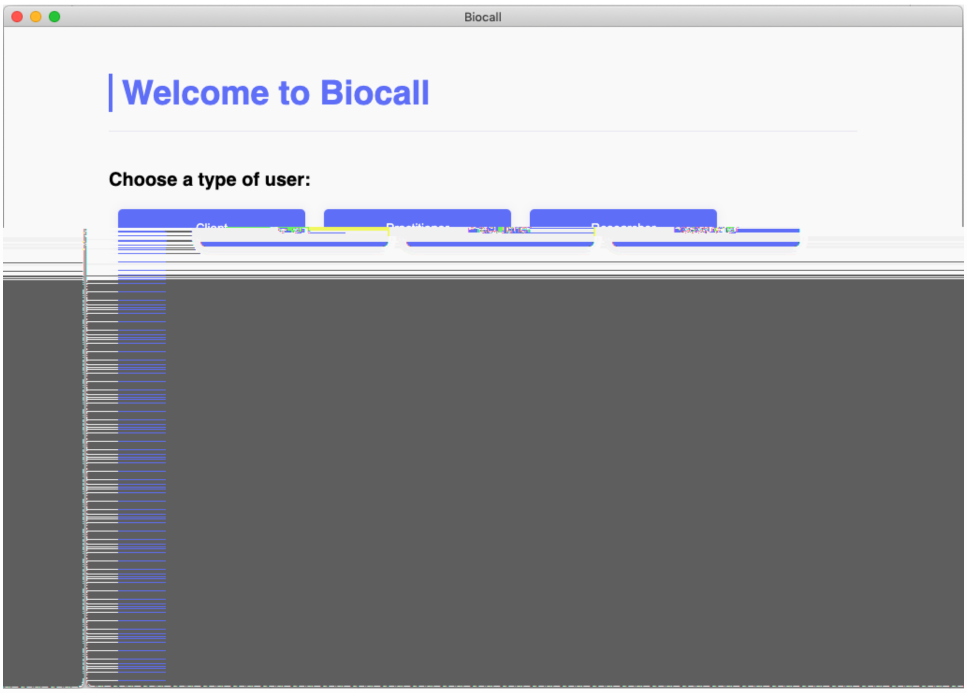
Field name	Value	Content
`time`	numbers	Number of seconds elapsed since user joined the room. Value rounded to the nearest integer.
`people`	numbers	Number of users in the room.
`displayData`	numbers	The data displayed to all users. It is either the GSR data or the spoofed data.
`bioData`	numbers	The GSR data.
`spoofData`	numbers	The spoofed data.
`isSpoofed`	true/false	Whether the spoof function is enabled, i.e., whether the data is spoofed.
`marker`	0/1	Whether the marker is clicked. *
`showBorder`	true/false	Whether the border is shown to the client.
`borderColor`	#rrggbb	The color code of the border color.
`showChart`	true/false	Whether the chart is accessible to the client.
`userOpenControl`	true/false	Whether the control panel is opened on the practitioner's side.
`userOpenChart`	true/false	Whether the chart is opened on the practitioner's side.

** Only recorded locally on the researcher's side.*

The export feature is available to `researcher` and `practitioner`.

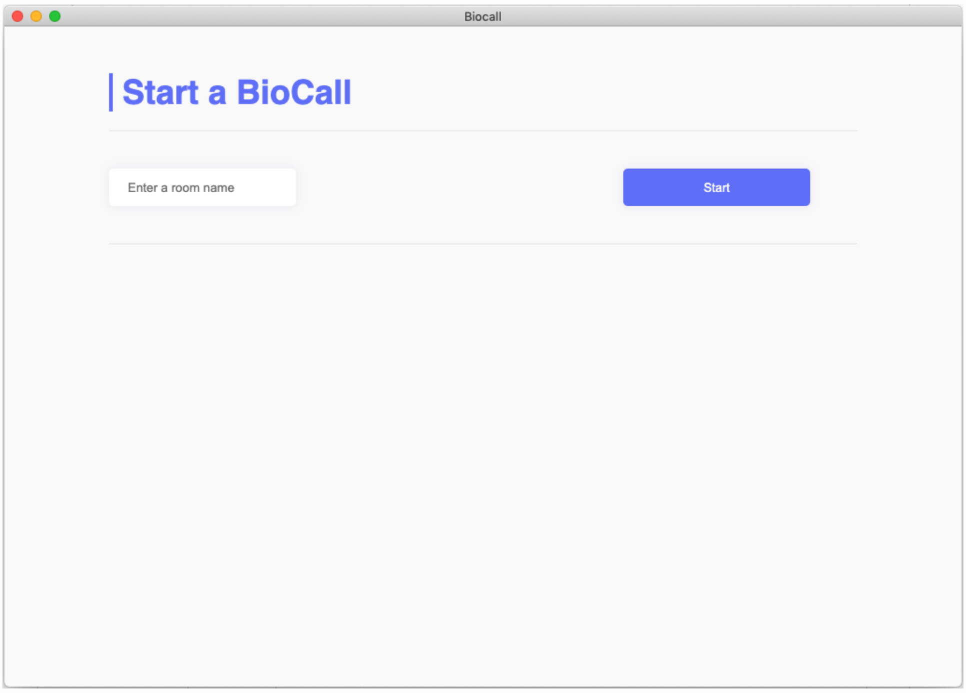
2.2 User guide

2.2.1 To make a call

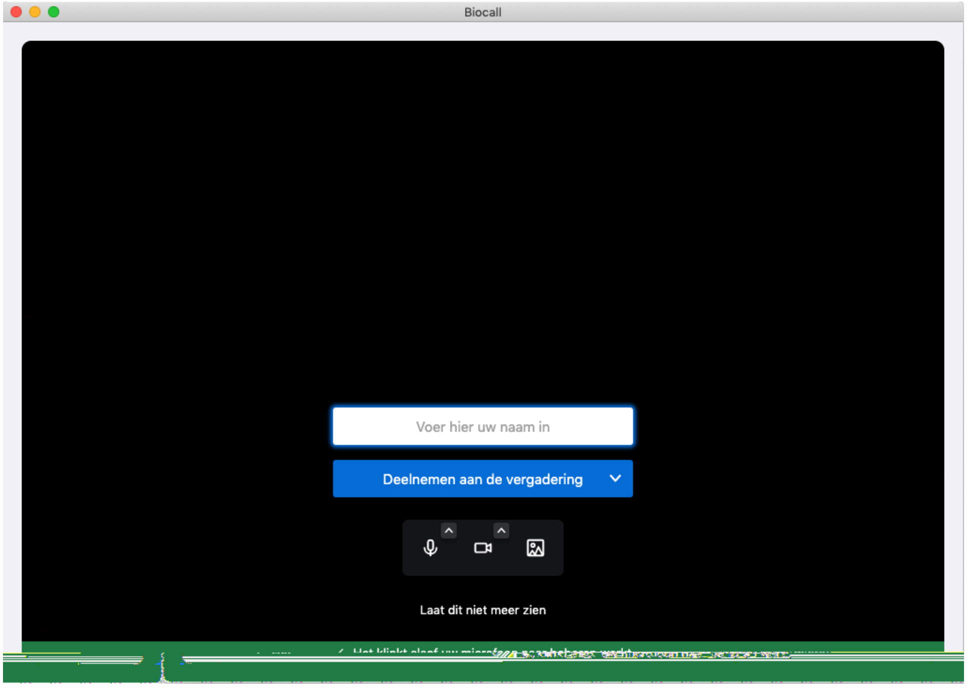


Welcome to Biocall

Choose a type of user:



Start a BioCall

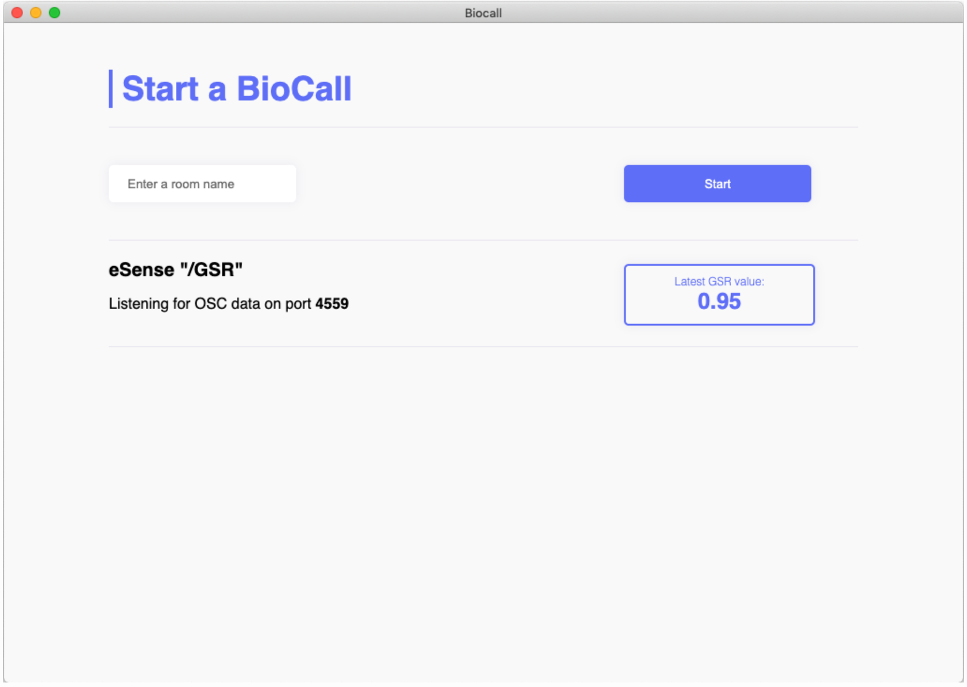


2.2.2 To connect to eSense

The client would have to connect eSense, which measures GSR, to the Biocall app. To do this, open the eSense app and enable Hidden Mode by tapping "Mindfield Biosystems" 5 times at the bottom right. Then set **OSC transfer** in **eSense Skin Response**.

- IP address: Local IP address of the PC on which Biocall app is running.
- Port: `4559`
- Path: `/GSR`

Make sure that the smartphone and the PC are on the same network. The latest GSR value should be updated if connected successfully.



Start a BioCall

Enter a room name

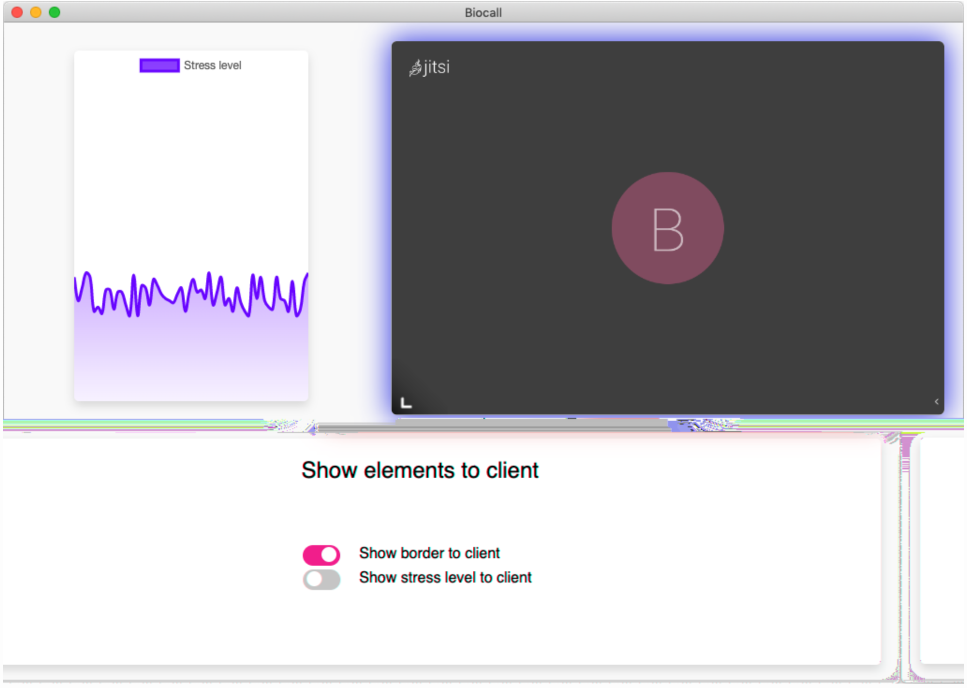
eSense "/GSR"

Listening for OSC data on port **4559**

Latest GSR value:
0.95

2.2.3 To control feedback

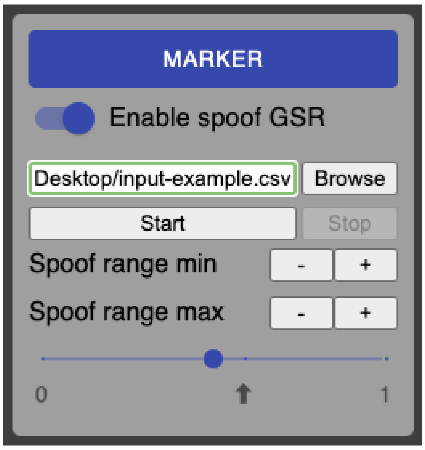
The feedback control panel can be accessed by clicking the arrow icon at the bottom left corner in a call. The screenshot below shows the control panel.



Show elements to client

☑ Show border to client
☐ Show stress level to client

2.2.4 Spoof panel



Desktop/input-example.csv

Press `Ctrl` + `Q` on the keyboard to open the spoof panel. If it doesn't work, move the cursor and click on the color border and then try again.

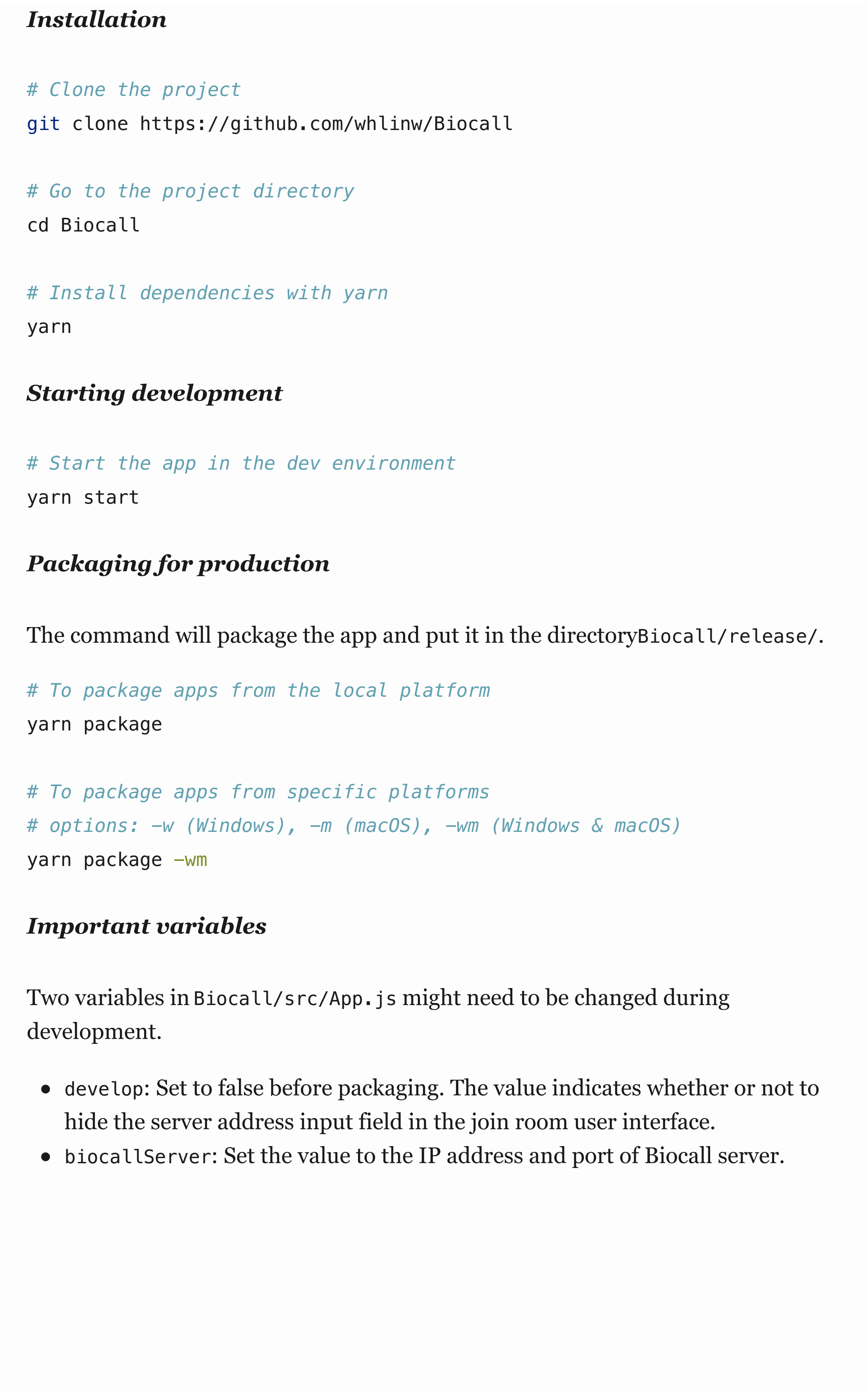
- **Marker**: Press the `MARKER` button to place a marker for the current second.
- **Enable spoof**: Toggle on/off to enable/disable the spoof feature.
- **Get spoof input from file**: Press the `Browse` button to choose a file. Only `.csv` format is supported. Press `Start` to start sending one value per second from the begining of the data sequence. Press `Stop` to halt the sending process.
- **Set spoof range**: Mannually increase/decrease the current min and max GSR value. The value could be replaced if the most recent value displayed is smaller/larger. Changing spoof range will affect the input slider range, color calculation, and the range of y-axis of the line chart.
- **Get spoof input from slider**: Pick from the slider a spoof value to send. The numbers on the two sides endindicate the current min and max GSR values. The upward arrow 🠉 shows the current real GSR value.

2.3 Developer guide

2.3.1 Development

The following software packages should be installed for developing Biocall app:

- Git
- Python
- NodeJS & npm
- Yarn
- node-gyp: Depending on your operating system, you may need to install additional software and configure settings, e.g., Xcode or Visual C++ build tools. Please read through their installation instructions.



Installation

```
# Clone the project
git clone https://github.com/whlinw/Biocall

# Go to the project directory
cd Biocall

# Install dependencies with yarn
yarn
```

Starting development

```
# Start the app in the dev environment
yarn start
```

Packaging for production

The command will package the app and put it in the directory`Biocall/release/`.

```
# To package apps from the local platform
yarn package

# To package apps from specific platforms
# options: -w (Windows), -m (macOS), -wm (Windows & macOS)
yarn package -wm
```

Important variables

Two variables in `Biocall/src/App.js` might need to be changed during development.

- `develop`: Set to false before packaging. The value indicates whether or not to hide the server address input field in the join room user interface.
- `biocallServer`: Set the value to the IP address and port of Biocall server.

```
this.state = {
  develop: true /* ALLOW SERVER INPUT. CHANGE TO FALSE FOR PRODUCTION */
,

  role: 'guest'
  room: 'default',
  sessionName: '',
  sessionTime: 0,
  biocallServer: 'http://127.0.0.1:4001' /* SERVER ADDRESS */,
  serverConnected: false,
  inCall: false,
  //...
}
```

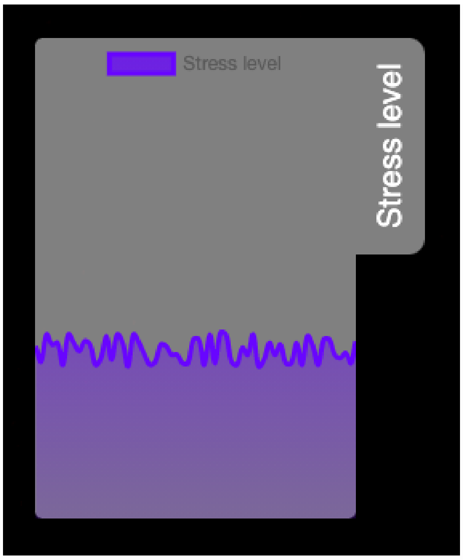
2.3.2 Color calculation

By defualt, the color feedback is calculated based on the ratio (`ratio`) of current GSR to the max GSR. The red value in RGB is set to 255 times `ratio`, and the blue value is set to 255 times `1-ratio`.

To apply your own color calculation algorithm, change the code between `Algorithm start` and `Algorithm end` in the function `setBorderStyle` in `Biocall/src/App.js`. The main task is to decide how `redVal`, `greenVal` and `blueVal` should be calculated.

```
/* [EDIT] Change algorithm & border color in the function setBorderStyle

/* Set border color given the data. */
setBorderStyle(data) {
    /* Default R (redVal), G (greenVal), B (blueVal) values = 0 */
    let redVal = 0;
    let greenVal = 0;
    let blueVal = 0;

    /* Algorithm start. Decide how redVal, greenVal, blueVal are derived

    let percOfMaxVal = data.value / data.max;
    redVal = Math.floor(255 * percOfMaxVal);
    blueVal = 255 - redVal;

    if (data.value <= 1) {
        redVal = 0;
        blueVal = 225;
    } else if (data.value >= data.max) {
        redVal = 225;
        blueVal = 0;
    }
    /* Algorithm end */

    /* Set color based on RGB values. */
    // ...
}
```

2.3.3 Chart name



To change the tab name shown on the right of the chart, replace "Stress level" in `Biocall/src/User.js`.

```
<div
  className="chart-toggle-button"
  onClick={this.props.toggleShovedStressChart}
> {/* [EDIT] Change chart tab name here. */
}
    Stress level
</div>
```

To change the legend name, replace "Stress level" in `Biocall/src/StressLevelChart.js`,

```
this.config = {
  type: 'line',
  data: {
    labels: [ '', '', '', ...],
    datasets: [
      {
        label: 'Stress level', /* [EDIT] Change label name here. */
        // ...
```

3 Biocall Server

A NodeJS server in Biocall responsible for relaying GSR data and feedback configurations to app users. The server listens on port `4001` by default. The value of `state.biocallServer` in the Biocall app should be the IP address and port on which this server runs.

3.1 Installation

```
$ git clone https://github.com/whlinw/Biocall-server.git
$ cd Biocall-server
$ docker-compose up --build -d
```

3.2 Logging

Log files are stored at `/var/log/biocall-server` in the container. The directory is mapped to `biocall_data` on the host machine, so the logs are preserved and accessible. Two files `info-YYYY-MM-DD.log` and `error-YYYY-MM-DD.log` will be generated per day while the server is up. Logs in the recent 14 days are maintained.

4 Note

- This document is written for Biocall v1.0.1 by Wei Lin in May 2021.
- Contact the developer if you have further questions.
- The Biocall application is inspired by and built upon theoriginal prototype developed by Arne Reijntjes.
- Icons in the Biocall structure figure are made bybqlqn and prettycons from Flaticon