Chair of Connected Mobility
TUM School of Computation, Information and Technology
Technical University of Munich

# Cheers

The smart party cup that delivers historic evenings

**Gustav Lund, Aziz Marzouki, and Daniel Mair**

TUM School of Computation, Information and Technology, Technical University of Munich, https://github.com/Team-GAD

August 24, 2022

**Abstract** — The internet connects people all over the world and allows access to networked services pretty much anytime, anywhere. The concept of the Internet of Things (IoT) connects such functionality to any device. In this project, smart cups were developed using IoT as the base premise. They act as actors at a party or event environment and can provide various benefits to the user as well as the organizer. Due to the large amount of data that can be generated by such smart devices, this project relies on the infrastructure paradigm of edge computing to prevent network reliability and reduce network traffic, where the data is processed by an edge gateway and then if necessary aggregated and sent to the cloud.

## 1 Introduction

The urge to interact with other people at social events is a standard, and it has always been an issue of how to act on, and thus execute it, successfully. Therefore, organized activities are very welcomed since they bring interesting and fun ways to enjoy yourself and meet new people.

The Cheers project offers a "smart" way to host social activities at parties and other gatherings. Take an evening at a club for example. Not only are new activities fun and prevent boredom whilst enticing attendees to continue visiting, but such events can also offer an opportunity to interact with other people and make new friends or break the ice towards a new relationship.

### 1.1 Basic structure

Cheers - the smart cup system, offers different games during an evening's stay to compete with other people and earn points, has a quick and cashless payment system, and offers a cloud service to calculate and store personalized data on each club visit.

The games offer points upon completion, which increase someone's leaderboard score and might result in the host gifting the winner a free drink. Games can be individual, team-based, or everyone playing as one team.

The barkeeper is in control of the flow of Cheers and in addition to ordering drinks they can start games, control the individual cups and display the current player leaderboard (e.g. on a large second screen).

While an automatic cashless payment service is future work, ordering drinks with a digital tab and paying at the end of the night is already a functioning feature. Ordering drinks is done by placing the cup on a scanner in order to identify the user. The assignment and return of the total owed money to the establishment by an individual is done at the end of the evening after handing in the cup. Adding the functionality of a PayPal automatic payment is possible.

Every user has the option to either participate as a guest or setup a profile in the cloud. Cloud functionality allows the user to access their profile from their phone at any point in time and track their statistics over many different cheers evenings, such as tracking their blood alcohol concentration by entering their age, sex, height and weight into their profile upon registering. Since the cup is scanned for every ordered drink, Cheers can measure the amount of alcohol consumed and calculate their per mille. Another cloud feature is the sharing of profiles when two people cheer together to keep in contact with each other, if both are interested.

### 1.2 Benefits

Both the participants and the hosts benefit from using Cheers.

The participants get to partake in fun activities, make new friends and even meet potential partners in a fresh and neutral way. Hosts can offer free drinks to winners which raises the stakes and makes the participation lucrative. Users can even track their progress for a specific evening, as well as all evenings over time, including their blood alcohol percentage, the drinks they drank and shared contact information on the users they cheered with. Tracking payment using the user's cup and profile and paying once at the end is another pro for using Cheers.

The host can track beverage quantities, thus purchasing the right amounts of ingredients before they run out. They can even use Cheers to perform analytics, such as figuring out what drinks are the most popular on what days of the week or events, and therefore for example purchasing more ingredients of one drink, changing drink prices or removing a unpopular drink from the menu. The analytics can also be performed on the customer behavior, according to sex, drunkness and age. By starting fun games, players will be forced to purchase new drinks if they want to stay in the games, thus increasing sales. Offering a new and modern vibe at an event can also interest people to visit, thus increasing popularity.

The host could also provide a prevention towards being drugged. Details can be found under Section 4.4.

### 1.3 Games

Cheers has a collection of playable games that can be extended at any time. When games are won or drinks are ordered, the leaderboard increases a player's score according to the game and feat. The following games are included:

**Rainbow dance mode:** The rainbow dance mode game set the led ring of all cups to a animated rainbow colored wheel for a set period of time.

**Team cheer:** When the team cheer mode is started, pairs of cups light up, each with their individual color. The designated partners have to find each other and cheer before the timer runs out.

**GuessWho:** The GuessWho game is the most attentive game and starts off with the barkeeper asking the participants to upload a picture to the platform if a picture is missing in the Cheers cloud. The participant's face must be clearly visible. A target is chosen random, and a monitor displays the target's picture with the face blurred and a countdown below the picture. The first person to find the target in the crowd and cheers with her/him before the time runs out wins. In that case, the target's picture is revealed without the blur, with the winner's picture beside it. The exact logic of the game can be seen in the diagram in the appendix. A. 10

**Drunkness:** The drunkness game mode lights up all cups according to a color that represents a set of possible german traffic violations that could be issued with the current blood alcohol percentage of the individual user.

The color roughly follows the rainbow wheel, with white being sober, more than that violates driving rules for drivers under probation and colors the cup green, as of 0.3 per mille proves liable in case of a car accident and turns the cup turquoise, operating a car above 0.5 is a violation for any driver and sets the color to light blue, 1.1 and above allocates more points and can throw drivers under that amount of influence into jail, thus colors the cup dark blue, riding a bicycle drunk is illegal as of 1.6 and sets the cup to purple. Anything above does not indicate specific additional traffic violations, but modifies the color for the fun of the game. 2 and above sets the cup to red, 2.5 to orange, 3 to an animated rainbow and 4 to grey; you are doing well, you are not doing well, you have an issue with alcohol and you are dead, accordingly.

## 2 Design

The entire system basically consists of 3 different components. The cups, a gateway[1], and the cloud. However, since the cups are wireless and many clubs do not have a network connection, offline functionality is a must. That is why the whole system works also without the cloud component. The general concept describes a design with smart cups that can interact with each other and a hub that sends and collects data from the cups and allows the bartender or an authorized person to register guests, start games, and manage drink orders on the Cheers interface. To reduce network traffic and support full offline functionality, the hub has a local database and runs all relevant components on it´s own instead of routing traffic between the cups and a cloud service. To allow participants to access their statistics on their phones at any time, a cloud service is provided that receives the user's data once they leave the event. The cloud can also be used to communicate with the hub and to support online registration. The general architecture and its communication flow are shown in Figure1.

Basically, an account can be created on the Cloud Platform at any time. This account information can be downloaded from the hub if the corresponding user visits this event. The hub acts as an edge device here. From now on, there is no more communication between the hub and the cloud. There is only local communication either between the individual cups or between a cup and the hub. As soon as the user leaves

[1] also referenced as a hub

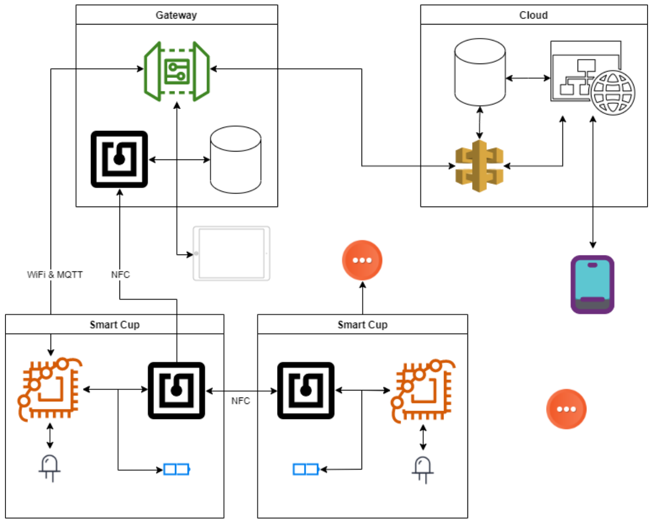


**Figure 1** Cheers' architectural flow structure.

an event, the most important data is summarized and sent to the cloud.

# 3 Implementation

In this section, the implementation is described, whereby a distinction can be made between the software and the hardware implementation.

## 3.1 Software

For the implementation different programming languages and technologies were used, in the following it is described how the individual components were implemented. The Cheers' system architecture is visualized in Figure 2. It shows the individual components and where which technologies are used.

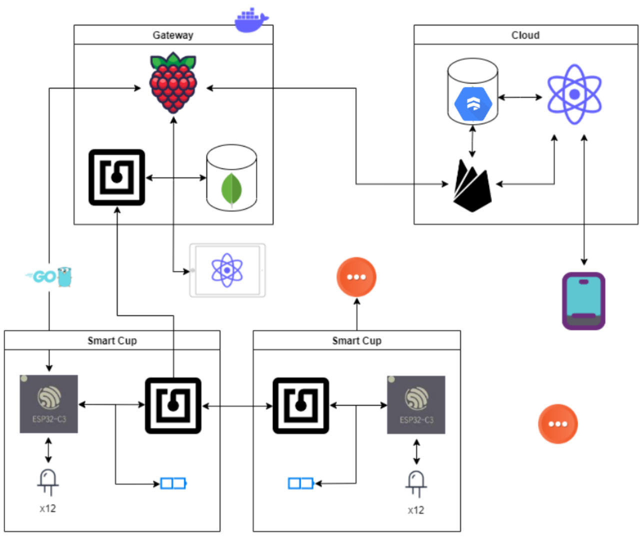


**Figure 2** Cheers' architectural tech-stack structure.

## 3.2 cups

The exact hardware needed for the cups is described in more detail in Section 3.5. The software for the cups was written in Arduino and loaded onto the ESP32. When a cup is initially started, it connects to the gateway's access point via Wi-Fi. If this access point is not found or the connection cannot be established the Wi-Fi Manager switches on. It creates its own access point on the cup's ESP and allows the cup to be configured. Such a configuration page can be viewed in Appendix in Figure 12. Once the connection with the gateway has been successful, a cup communicates with the gateway via MQTT and with the other cups via NFC.

## 3.3 Gateway

The hub can run on any pc in the Club. A Raspberry Pi is particularly well suited for this purpose. The Hub consists of several components, all of which are deployed into individual Docker containers.

One component of this is the MongoDB database, which stores the temporary data of the currently running Cheers instances as well as the permanent bar or event data.

Another component is the MQTT broker. It is used to provide communication between the gateway and the cups. All participants can push data to a specific topic and subscribe to specific topics.

Another component is the react frontend for the event operator. Here only authorized people can access it and it can be used to (de-)register participants, order drinks, manage games, the leaderboard as well as manage Smart cups.

The last and most important component of the hub is the gateway itself. It was written in Golang and manages all communication between the components. The gateway fetches user data from the cloud and sends the processed data back to the cloud after a user logs out. Meanwhile, the gateway writes all data to the database and communicates with the cups via the MQTT broker. It receives either status information such as battery status or scan information from the cups and based on this information sends the cups the appropriate colors and modes in which they should go. In addition, the gateway provides various API endpoints that the frontend can access to get information and can control the system.

### 3.4 Cloud

The cloud is split into frontend and backend. The frontend is developed in ReactJS and is deployed using Firebase. The backend is run, written, and stored using both NodeJS and Firebase and provides REST API endpoints for the gateway and the users. The person using the Gateway can identify, connect, disconnect, and update user profiles in the cloud as well as start a game for users to participate.

The user authentication is handled using Firebase Authentication. The images were stored using Cloud Storage for Firebase. The data is stored using Cloud Firestore, which is a cloud-hosted, NoSQL database built for automatic scaling, high performance, and ease of application development. Each document has a unique ID as a primary key. Users can also be identified with a unique handle (username). The whole database scheme can be seen in the appendix in Figure 11. Cloud Functions for Firebase was used to automatically run backend code in response to events triggered firebase features and HTTPS requests.

In order to take full advantage of the benefits of the cups, regular users and event organizers have access to user-friendly cloud interfaces. After logging in, users can visualize all the information which has been collected during their sessions. They can access a list of people they cheered with and drinks they ordered. They can edit a profile which will be displayed to other people they cheered with. And, they can participate in games started by event organizers.

### 3.5 Hardware

Various hardware elements are required to make a cup smart. The entire hardware diagram can be viewed in more detail in Figure 3. All hardware elements are listed below and their use is briefly described.

**ESP32:** The ESP32 is the central element of a smart cup, it links all the hardware controls the LEDs and communicates with the gateway.

**NFC Module:** The PN532 module was used as an NFC module in this project. It has several modes but, in this system, only the I2C (Inter-Integrated Circuit) mode was used for P2P communication.

**LED:** The LED ring from Adafruit was used as the LED. It has 12 individual LED's which can be used to display various animations.

**Battery:** A LiPo battery with 1100 mA was used, it has an output voltage between 3.3 and 4.2 V depending on how strongly the battery is charged. It supplies the entire hardware with power.

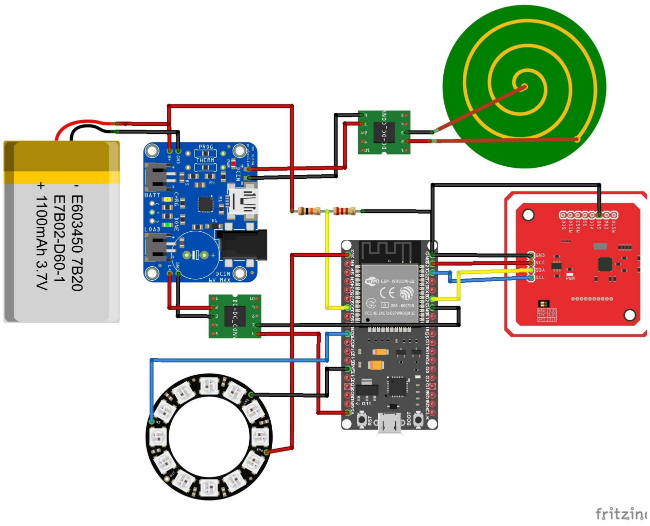



**Figure 3** Wiring plan of a smart cup.

**Battery Board:** For charging the battery you need a charger board. In this project the LiPo charger from Adafruit was used.

**Charging Coil:** The entire case should be closed in the final product, so an induction charger was installed so that the battery can still be charged without having to disassemble the cup again.

**DC to AC Module:** Since the induction charger outputs AC, but a smart cup works with DC, the current must first be converted. For this the AC to DC converter was used.

**Step Up Module:** The used battery works with an output voltage of 3,7V but the ESP32 and the rest of the hardware works with a voltage of 5V therefore the Step-Up module is used to supply the hardware properly.

**Resistors:** To find out the state of charge of the battery, its output current is measured. The 10k ohm resistors are used to measure this current with the ESP32. From this value the charge percentage can be calculated.

All hardware elements are housed in a 3D printed case. Such a housing as shown in Figure 4 is then used to attach the hardware compactly under a glass.

The gateway also needs its own hardware. In principle it doesn't matter which computer is used but in this project a Raspberry Pi 3B was used to host all the components of the gateway. In addition, another NFC module was connected to the Pi so that IDs can be exchanged between the cup and the gateway whenever something is ordered or a cup is assigned to a user.

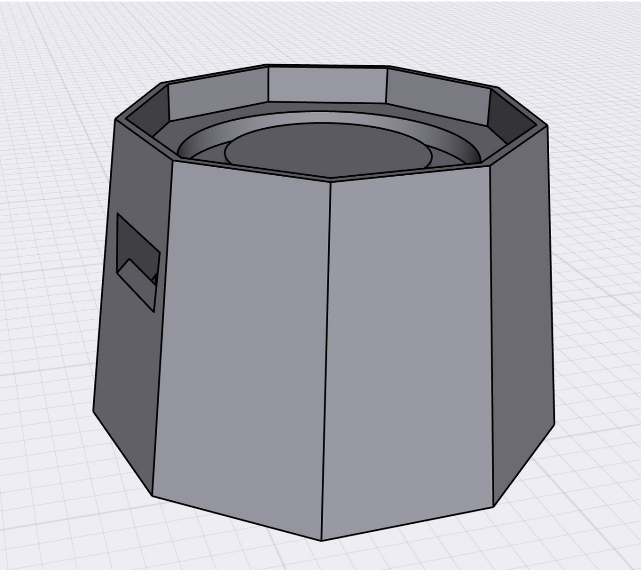


**Figure 4** 3D model of the case.

## 4 Evaluation

The basic goal was as described in Section 1 not to solve a specific problem, but to combine the party experience with technology. The prototype is a proof of concept and offers some features and games described before. However, there is still a lot of potential. A variety of other functions and games could be realized with the smart cups.

A particular advantage of the cups is their expandability. The ability to easily change the modes and LED color of the cups via the gateway, as well as the recognition of other NFC scans, makes it very easy to develop new games. In the future, it may even be possible for operators to create new games through a simple configuration page.

Another advantage is that the individual components of the system - the cloud, the gateway, and the cups - are decoupled from each other. The whole system can be used without the cloud and a user can change the cup as often as he wants during an event. So, if there are problems with a cup or the battery is empty, the cup can be changed, and all the data remains with the user.

Another advantage of the system is that the gateway has its own database and thus only relevant data is sent to the cloud. Thus, when using the cloud, the data transfer is reduced to the minimum.

Since this is a prototype, there is still plenty of room for improvement. First, it would be advantageous to make the entire attachment a bit smaller by building a dedicated PCP board. Furthermore, the current prototype is not yet waterproof, which could be problematic. However, these are all minor problems and could be solved with a little more time. The biggest drawback of smart cups is the reliability of the NFC modules. It is very random how the modules work. In some cases, they work fine, and sometimes they don't work at all or are not recognized by the ESP after a few scans. Initially, the problem was that cables behind the modules were creating interference. When the problem was detected, a new case design was developed, which significantly increased reliability, but it was still not perfect. This is probably ultimately due to the fact that the P2P functionality [Man22] is currently still under development.

Another big disadvantage of the NFC modules is that they must be switched to different modes to communicate with each other via P2P. One cup must be switched to the initiator mode and the other to the target mode. This means that the system loses a lot of flexibility, and it must always be known in advance in which mode the cup must be switched.

### 4.1 ZigBee

When selecting the various technologies, ZigBee came up again and again. ZigBee is a specification for wireless networks with low data volume and low power consumption which is mainly used in smart home devices. In addition to the aforementioned properties, ZigBee creates a hash network for communication, which would be a great advantage in the application area of the cups. In the beginning, WLAN was used for communication between gateway and cup and this should be changed to ZigBee later. As explained earlier in Section 3 the gateway uses an MQTT broker. Here ZigBee2Mqtt could be used to write all data into the broker which would finally make no difference for the cup and the gateway through which physical communication technology is used. Unfortunately, when using the ZigBee modules, there were problems with the connection establishment. The modules could recognize each other, but no simple ZigBee connection could be established by the modules entering the hash network. Due to these problems and the lack of time, ZigBee had to be abandoned.

## 4.2 Data transfer

Since ZigBee allows a significantly lower data transmission rate, it was first necessary to consider which data and how large data packets are sent. For this purpose, the transmitted packets were examined with Wireshark. There were no anomalies. Initially, DHCP and MQTT handshake packets are sent, and as soon as the connection is established, a cup can be configured. Such a configuration packet is 123 bytes in size, which can be sent over ZigBee without any problems. Also, all update packets like scan information or battery status, which are sent from the cup to the gateway, do not exceed the ZigBee limitations, because they are all smaller than a configuration package.

## 4.3 Power consumption

A special important aspect is the power consumption of a cup, the goal was to be able to run the cup for a whole evening. For the power consumption analysis, the Power Monitor from Monsoon [Mon22] was used. Here, all possible functions of the cup were measured over a longer period of time. Figure 5 shows the boot process of a cup. First, the ESP32 is started which is the first big spike, then the ESP connects to the WLAN which is the second spike. After that, the ESP connects to the MQTT broker which is the third block of spikes. Here a cup gets assigned a mode and then stays in that mode. After that, the power consumption stays at a certain base line which depends on the set mode and the set color of the LED. The two arcs at the beginning of the boot are caused by the LED animation. All the smaller spikes that can be seen again and again are caused by the WLAN.

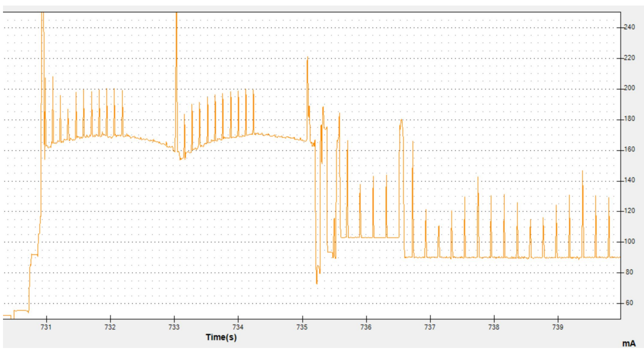



**Figure 5** Avg consumption in boot process of the cup.

During the boot process of the ESP32 there are more spikes, but they only last for a very short time. The measurements have shown spikes of up to 1200 mA which are shown in Figure 6.

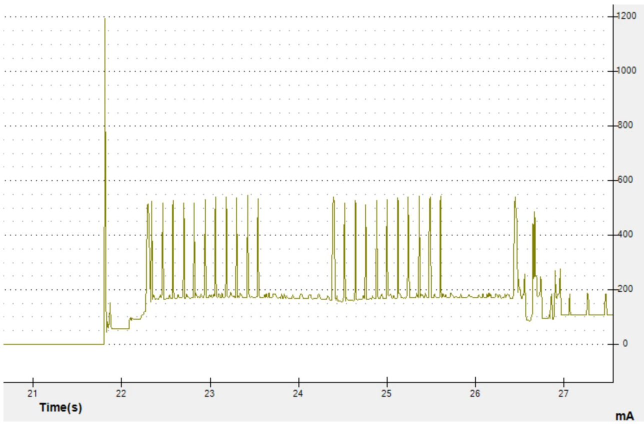



**Figure 6** Spikes in the boot process.

The LED has a decisive influence on the current consumption of the system. In the Figure 7 you can see a jump of about 25mA from light off to light full on.

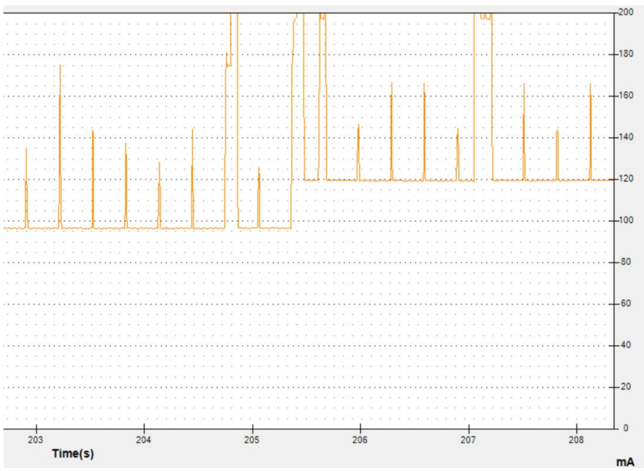



**Figure 7** LED change

Equally important measurements are those taken during an NFC scan. As can be seen in Figure 8, at the beginning the power consumption is at a certain baseline. During the scan, the cup then requires 20 mA more current for about 2 seconds and then the current consumption goes back to the baseline. In this figure, however, the baseline is slightly raised after the NCF scan because the LED color was also changed.

In addition to the measurements just presented, a long continuous measurement was also made to find out the battery run time of the cup. The result can be seen in Figure 9. For this, we switched back and forth between the different modes for over 40 minutes, performed scans and continuously changed the color of the LED. This measurement showed that an average of 535mW was consumed during this period. Since the system uses a 1100 mAh battery, the average battery

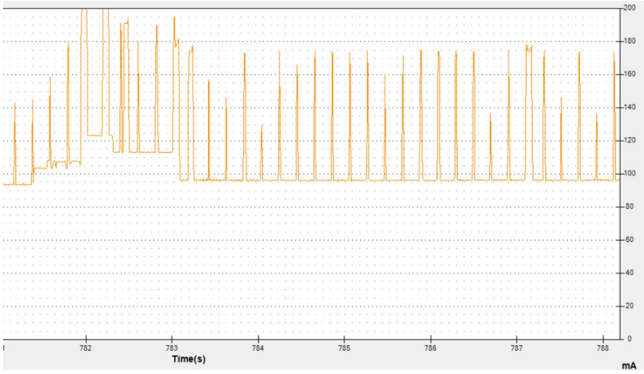
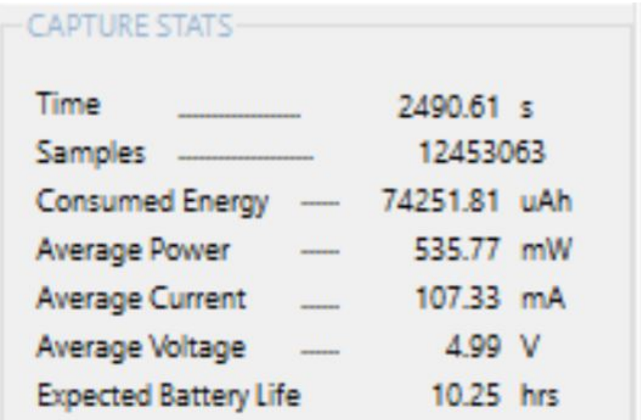



**Figure 8** Consumption during NFC scan.

life is 10.25 hours. However, you have to be careful with this value because it was calculated by the Power Tool and doesn't take into account that we use a 3.7 V battery and first have to convert the voltage to 5 V with a voltage converter. The manufacturer of the converter specifies an efficiency of 80-90 percent, which means that the entire system still has a battery life of over 8 hours, which is completely sufficient for the planned application purposes.



CAPTURE STATS

| | | |
|---|---|---|
| Time | 2490.61 | s |
| Samples | 12453063 | |
| Consumed Energy | 74251.81 | uAh |
| Average Power | 535.77 | mW |
| Average Current | 107.33 | mA |
| Average Voltage | 4.99 | V |
| Expected Battery Life | 10.25 | hrs |

**Figure 9** Capture stats of the power monitor.

However, if the battery life is not enough for an entire event, one could always either install a larger battery, or simply change the cup during an event, which works without any problems since they can be bound to a user and unbound again.

### 4.4 Future work

While visitors can pay at the cashier after returning the cup, PayPal integration could automate this step by automatically billing the visitors after leaving the event if their PayPal is linked to their cloud profile.

Sharing a microphone among several persons during a karaoke party can be very inconvenient, primarily because of the exchange of germs and because organizers have to keep track of their positions. The integration of a wireless microphone which is connected vie Bluetooth would make the product more adapted for its major purposes, which are user entertainment and better organization.

Being drugged at a club or a bar is a big issue and something that both customers and hosts want to prevent. While one-use fluid drug tests exist, they are unpractical and dismissed under influence of alcohol. A crowd funding campaign [Ind22] presents a developed reuseable stick, similar in size to Covid rapid antigen tests, that lights up after being put into a liquid if the liquid has been drugged. A device with similar functionality could be glued to the bottom of each smart cup and connected to the ESP32, thus alerting the cup holder using the led ring, as well as the barkeeper, if the cup has been contaminated.

## References


[Ind22] Indiegogo. *Empower yourself against date rape drugs with pdid*. 2022. URL: https://www.indiegogo.com/projects/empower-yourself-against-date-rape-drugs-with-pdid#/.

[Man22] Manuals+. *ELECHOUSE PN532 NFC RFID Module User Guide - Manuals+*. 2022. URL: https://manuals.plus/de/elechouse/pn532-nfc-rfid-module-manual#p2p_nfc.

[Mon22] Monsoon Solutions. *High Voltage Power Monitor*. 6.08.2022. URL: https://www.msoon.com/online-store/High-Voltage-Power-Monitor-p90002590.

# A Sequence Diagrams

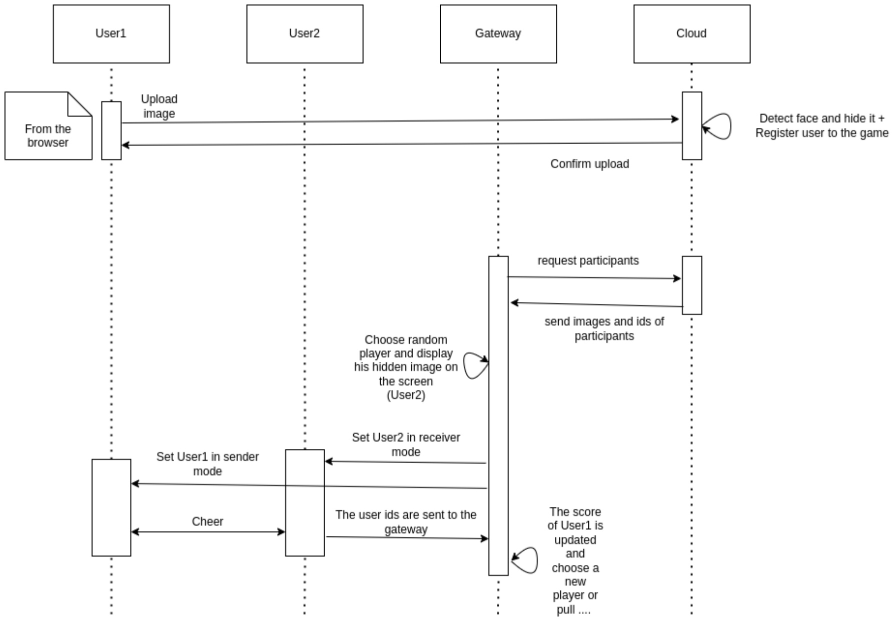



**Figure 10** Sequence diagram describing the logic of the GuessWho game.

# B Database schema

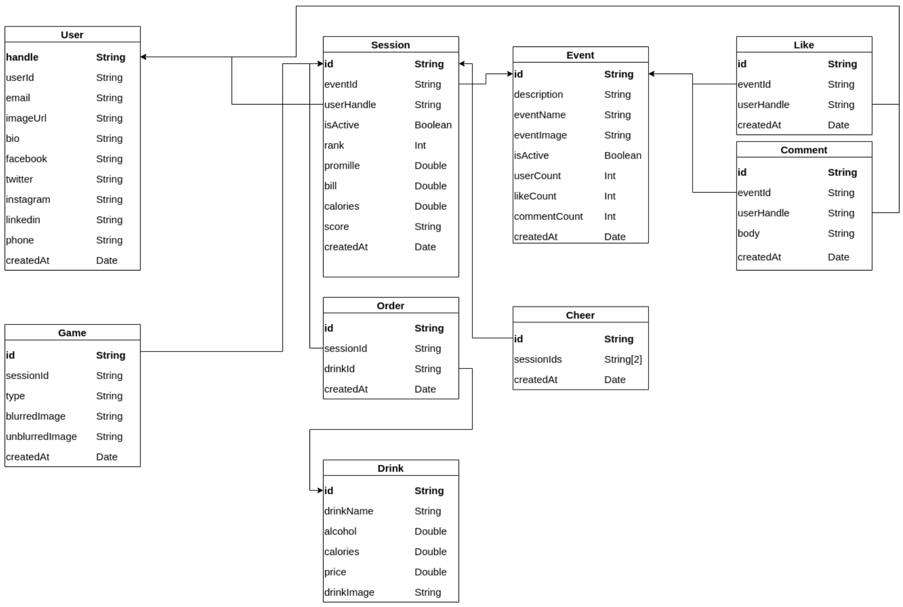



**Figure 11** Cheers' Cloud Firestore database.

## C WiFi Manager

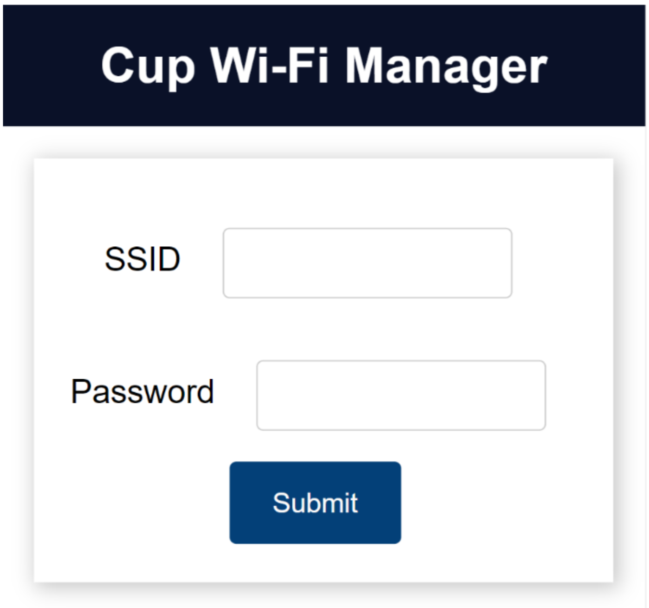



**Figure 12** WiFi Manager of a smart cup.

The Figure 12 shows the Wifi Manager hosted on the cup if it cannot connect to the gateway. Here the SSID and the password of the gateway can be configured.