SmartBrew: Automated Beer Brewery

Bc. Frniaková Alexandra, Bc. Krchňavý Marek, Bc. Nemcová Michaela,
Bc. Novota Lukáš, Bc. Stríž Peter and Bc. Vajda Marek

Faculty of Informatics and Information Technologies STU in Bratislava
tim6_tp2021@googlegroups.com
http://team06-21.studenti.fiit.stuba.sk/

Abstract. Automating processes in the brewing industry brings many advantages. From saving resources such as water or energy, to potentially making sure that the quality of the resulting product remains constant over many brews. Automation can also allow us to perform the brewing process at a smaller scale and create breweries for personal use at home. In this paper, we propose a software control system for a miny brewery called SmartBrew. Our solution provides many customization options and a semi-automatic brewing process. The final system is deployed to an existing brewery hardware.

Keywords: Brewery · Micro-services · Internet of Things.

1 Introduction

Automating processes in the brewing industry brings many advantages. From saving resources such as water or energy, to potentially making sure that the quality of the resulting product remains constant over many brews. Automation can also allow to perform the brewing process at smaller scale and create breweries for personal use at home.

Mini breweries need to be easily controlled. They may be purchased by beginners, so it is only favorable to automatize the process as much as possible for ease of use. Amateur brewers should be able to choose pre-prepared settings, where the system then takes care of most of the processes, from setting the correct temperature, to adding the ingredients in the correct time intervals.

There are only a number of existing mini breweries, mostly for personal use at home. A big disadvantage of these systems is the lack of customization of brewing recipes, as well as the small capacity per batch, which makes them unsuitable for professional use in bars or restaurants.

In this paper, we propose a software control system for a mini brewery called SmartBrew. Our solution provides more customization options and a semi-automatic brewing process, for when the user needs to add a special step, which is not automatized. The control system is deployed to an existing brewery hardware built by our colleagues at the Faculty of Mechanical Engineering STU in Bratislava.

2 State of the art

There are several automation brewery systems on the market. Most of them are focused for personal use. There are far less options when we talk about restaurant and bar applications.

Fully automated systems use microcontrollers which monitor and adjust the variables during the brewing process of a selected recipe. For homes, bars and restaurants, following options are available [6]:

2.1 PicoBrew

PicoBrew [6] is an all grain brewing machine with a maximum capacity of approximately 9.5l finished brew per one batch. It has an additional 11.36l keg for water supply. This makes it a much more portable due to its smaller size (see Figure 1). Main disadvantage of this machine is that it uses proprietary plastic container for brewing which can break over time. Another important thing to note about PicoBrew is 'PicoPacks' - customized pre-prepared mix of ingredients that need to be ordered and delivered. You can only brew all grains recipes and you can not use your own ingredients for brewing with PicoBrew machine. The price of PicoBrew is approximately 2000 USD.

Fig. 1. PicoBrew device [6]

Fig. 2. MiniBrew device [1]

2.2 MiniBrew

This home brewing system (Figure 2) is semi-automated, because the user needs to add yeast and hops at specified intervals. It has 5l capacity and there is active cooling feature on the keg which helps MiniBrew to maintain fermentation temperatures without an external refrigeration unit. It retails for approx. 850 USD without taxes. Additionally, the machine comes with a mobile app for showing information during the brewing process or other pre-filled recipes. The app is available only for Apple devices and currently not supported on Android [1][6].

2.3 Automatic brewing based on the Arduino platforms

This is a master's degree project [5], which uses five buckets (Figure 3) for brewing the beer. First bucket is for water, second for malt, third is the boiler bucket, forth bucket collects the wort. The last bucket situated on the left side holds cold water, which is being pumped into the heat exchanger. Unfortunately, the presented product is only a prototype for demonstrating purposes only. It is not available for the purchase.

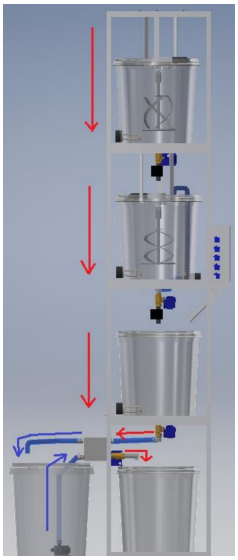

Fig. 3. Rpi brewing machine [5]

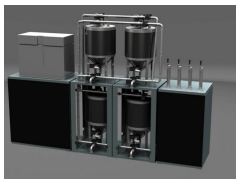

Fig. 4. BrewiePro machine visualization [3]

2.4 BrewiePro

The most comparable project with our solution is BrewiePro [3]. This project from 2018 was sponsored by European union with 50 000 EUR. BrewiePro is a new version of an already existing automated home-brewing system. Their main customer base should be the beer-loving culture. The whole idea is to bring bars and restaurants a way to relatively easily create their own signature beer recipes. All this to impress and satisfy their customers' taste.

This system is promising not only automation of the brewing process, but also the fermentation part. At the time of writing this paper, we cannot say if the project was successfully finished, as the BrewiePro project seems to be not available for sale. This is the most comparable project with our SmartBrew automatic brewery by far. A visualization of the product (Figure 4) also shows beer draught pipes which could be really useful for bar or restaurant applications.

3 Our solution

The main goal of our project was to create a comprehensive solution for process automation in the industry, in our case the brewing industry with focus on market with small breweries. The available steps include choosing a recipe, the brewing process, continuous displaying of the brewing state and creating a brewing report after finalizing the recipe. The principal component of our system is Raspberry Pi, on which the whole system is running. It is also utilizing containerization by running the system in two Docker containers. The system architecture consists of 4 layers:

1. Devices
2. Modules
3. Backend
4. Frontend

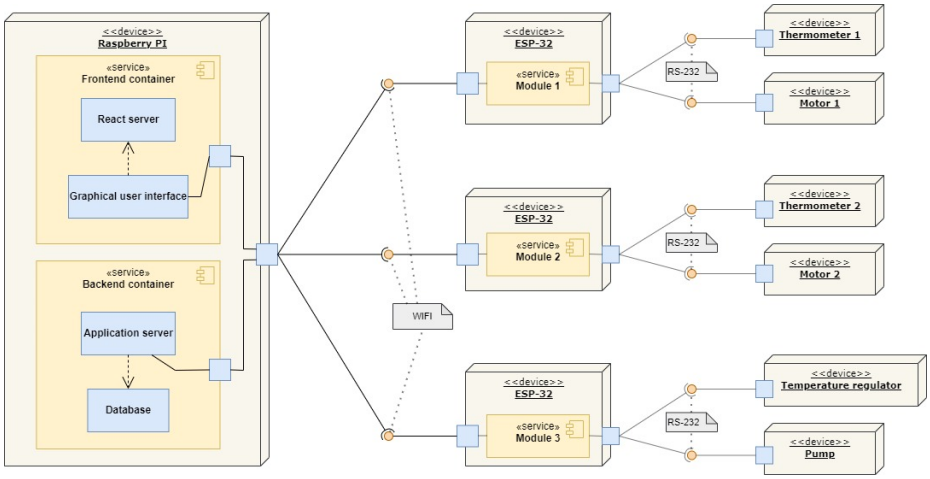

Fig. 5. System diagram

Figure 5 depicts a diagram of our system architecture and its parts. Figure 6 shows our brewing device - the hardware.

3.1 Modules

Modules are used for controlling parts of the brewing device such as motors, temperature regulators or pump, as well as collecting data from sensors inside the device. This data is being periodically sent to backend where it is stored and processed. Modules also accept instructions during brewing to which they reply by taking control of the corresponding part of the device based on the

Fig. 6. Brewing device

instruction and its values. At first, however, modules have to connect to a WIFI according to configuration and then to the backend server by using WebSocket. Modules are running on an ESP32 and the code is written in C++.

3.2 Backend

Backend is integrated together with a database containing data such as lists of ingredients and instructions of the recipes. Communication with modules is done via WebSockets. Backend and user interface are communicating through REST API and structured JSON objects consisting of brewing data. Backend acts as a connector between modules and user interface and allows seamless brewing and user experience. For backend we used NodeJS with Typescript which we mainly used because of its design to build scalable network applications. The database uses PostgreSQL which best suited our needs. For data manipulation we made use of Prisma ORM which helps to build faster apps and make fewer errors.

3.3 Graphical user interface - Frontend

Our brewing device is controlled by a touch-screen monitor so it was important to adjust the frontend accordingly. Our aim was to have a pleasant and easy

to understand GUI so that anyone could use the device. We were following the SCADA control system architecture principles. User can see the state of modules while brewing in real time. The design has been through a number of iterations, to focus on the user need and demands. Our use cases include:

1. Picking a recipe and use it to start brewing
2. Creating a new recipe
3. Editing a recipe
4. Deleting a recipe
5. Choosing ingredients for a recipe
6. Choosing stages that form a recipe
7. Choosing instructions that form stages
8. Showing the statistics of finished brews

For the GUI development, React 17 framework was used. Figure 7 shows the GUI of our system - the overview page. This page may go through more iterations to become even more user friendly and more SCADA-compliant. GUI contains 2 more pages:

– Recipes - list of recipes, option to load, create, edit or delete a recipe
– History - history and statistics of finished brewings

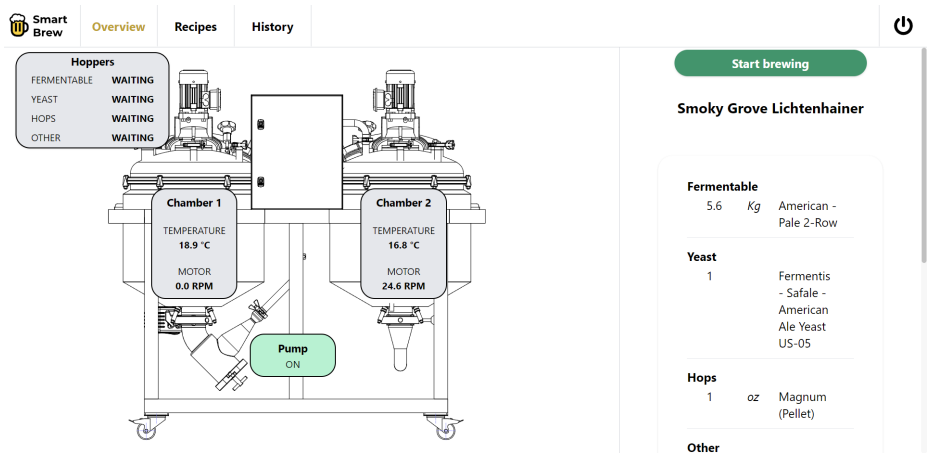

Fig. 7. Graphical user interface of our system

4 Comparison to state of the art

Similar solutions can use different technologies depending on their main purpose. The control and maintenance of automated systems can be also a Java-based

projects. When analyzing and processing inquired data, many approaches are also using Python. The obtained information is often used to make the automated processes even more effective and easier for a brewmaster to use. Our approach used Javacript on both backend and frontend (TypeScript) as it was very easy to work with and we did not need the added performance of lower level languages.

Making the new recipe for the MiniBrew device can be done on their portal. Similarly to our solution, their portal uses React framework while serving different purposes. Those include recipe making, monitoring state and showing statistics. The main difference with our solutions is the ability to customize the beer depending on the style of the beer. Input in the form of knowledge of the brewing professions allowed MiniBrew portal to suggest amount of water, and other ingredients to fit the basic characteristics of the chosen beer style. The steps are divided into brewing stages, adding instructions and ingredient of the same brewing stage at the same time. Our solution separates these steps, allowing more complex combinations of instructions to be created. Our solution provides better insights to the brewing process itself. The MiniBrew app only provides the basic information such as the stage of the brewing, current temperature and required temperature. Our SCADA-based design also provides information on the current state of the modules, which can be very useful, when we want to check the state of the brewery as a whole [2].

Bigger breweries use software such as Zenon[4]. Zenon is used in bigger scale automatization projects, meaning it provides high variety of features and settings. Such complex system may not be needed in the case of smaller breweries. Our solution speeds up the process by not having to log in and manage your profile. Many, often not needed, features can also extend the initial loading time.

5 Conclusion and Future work

In our work we focused on the implementation of a software system needed for automatizing a small brewery, but there are still many possibilities to explore. There is always room for improvement in case of security and performance of our software. We would like to gather feedback from the real use of the brewery and use it to help us improve the experience for users.

There are plenty of features, which would be great candidates for further development. For instance, remote control of the brewery from a smartphone or a different remote device. We could also improve communication with the user and provide him with push notifications on his mobile phone for example. Depending on testing with actual users, there is an option to give more control of individual brewery modules into the hands of the user for finer tuning.

An interesting aspect of our system that we could work on is the analytical processing of each brewing and provide a structured or even interactive report for examination. This would require us to further improve on our data collection or might need additional hardware for the brewery. Currently we only provide a display of individual sensor values and their change in time of brewing.

We developed our software with expandability and portability in mind, therefore we are ready to apply our solution on possible revisions of the current brewery or using our solution on different one, with as little changes as possible, should the need for such transfer arise. This would allow us to connect more breweries to our system as well and manage them together from one place. We can still try to improve on decomposition and disaggregation of modules of our software.

References

1. Craft gen 3 — minibrew. https://www.minibrew.io/product/craft-gen-3, (Accessed on 03/22/2022)
2. Minibrew webpage. https://www.minibrew.io/, (Accessed on 04/01/2022)
3. A new automated system makes home-brew much simpler and quicker — brewiepro project — results in brief — h2020 — cordis — european commission. https://cordis.europa.eu/article/id/250859-a-new-automated-system-makes-homebrew-much-simpler-and-quicker, (Accessed on 03/23/2022)
4. Zenon software. https://www.copadata.com/en/industries/food-beverage/food-beverage-solutions/automation-in-the-brewery-industry/, (Accessed on 04/01/2022)
5. Mårtensson, D.: Principles of an automatic brewing machine: By using codesys with raspberry pi (2017)
6. Parekh, P., Agrawal, A., Bagale, G.: A review of fully automated home brewing systems for craft beer. Journal for Research— Volume **2**(12) (2017)