Smart Shelf: Report

Md. Abdul Kadir
Saarland University
Saarbrücken, Germany
maktareq@gmail.com

Kevin Denk
Saarland University
Saarbrücken, Germany
denk.kevin@web.de

Atika Akmal
Saarland University
Saarbrücken, Germany
atikaakmal19@gmail.com

ABSTRACT
We are on the apex of interaction technology integration. Nowadays, people are integrating digital technology with every rigid device to introduce efficiency and make life easy . Working in big lab with lot of consumable equipments in shelves or finding a product in supermarkets is time consuming and can be exhausting. It's hard for a person to find an object from a gigantic shelf. Also, it's hard to keep supply continuous of consumables by checking every individual slots. To ease the hard work of management and user we introduce *Smart Shelf*, a interactive shelf that supports user in their search for items and operators in supply management. This paper presents the first prototype of the *Smart Shelf*. Advantages and disadvantages of that shelf are shown and further improvements are demonstrated. Additionally to implementation details a evaluation with a short user study is done and described.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation : Human Computer Interaction

Author Keywords
Smart Fabrication, Smart, Shelf, Drawer, HCI, Human Computer Interaction, Physical Computing

INTRODUCTION
Today the word *Smart* is almost everywhere. There are *Smart Homes* and *Smart Fabrication*. But from the last few years, HCI researchers have developed new interactive interfaces, synthesize from different perspective of humans for example Psychology, Economically and Social that can be integrated with modern technologies (Ubiquitous technology). People prefer to use these technologies, but sometimes face problems to obtained desired results. For example, most of the time people don't like to use shelves, because it is hard to find the desired item. By using ubiquities technology, combined with technologies of the *Smart Home* and *Smart Fabrication* domain an interactive shelf, called *Smart Shelf*, can be developed. This shelf can enhance the utility of regular shelves. This paper describes the approach of the implementation of the *Smart Shelf*. Different approaches of the interactive design, how the *Smart Shelf* is implemented and how to interact with it is discussed. One focus is the interaction of users and operators with the shelf. In the end there is an evaluation of the design decisions regarding the project and an outlook for further work. The evaluation contains a short user study, too.

ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

Problem
Most people when they hear about a shelf they think about their bookshelf or some shelves in the kitchen. Almost everyone who have a bookshelf searched at least one time in his/her life for a book in it and wished to have a guideline how to find it the fastest way. Imagine big shelves with a lot of small drawers. Every drawer is only labelled with a small name that describes what in that drawer is. Searching for items in these shelves can be hard and cost a lot of time. An additional scenario is if you apply this concept to big warehouses with hundreds of shelves and more drawers or places where you can place items. Finding an item in such a warehouse can be still harder.

Shelves used in companies or research institutes bring more problems to the surface. Often there are shelves used for storage of electronic components [1] for example. The drawers contained in these shelves are often a lot and small. Searching for a specific item/drawer in such a shelf with for example fifty drawers can be exhausting. But this is not only one problem. Is the searched drawer found, the contained items are maybe out of stock and the user wasted his/her time for search. This is not only exhausting, it is also wast of time if the user could directly see if the searched item is out of stock.

Not only warehouses or storage rooms with shelves have those problems with the inefficiency in finding items or the premise if one item is out of stock. The same problems appear in retail. Customers which can't find their favourite product in a shop are unsatisfied. Maybe they go to another shop and don't come back. This problem can be tackled with *Smart Shelves*, too. The shelf itself could detect if some products in it are only available in a small amount. In this case the shelf could order new products or at least send an information to an operator who can order supplies. With this strategy there will be no more empty shelves in shops and customers can find their favourite product all the time. This strategy is very similar to the current trend of automation called *Industrie 4.0*.

Smart Shelf should be a solution for these problems. It could observe the amount of items in itself, help people to

[1] e.g. Resistors, Capacitor, Micro Controllers or Integrated Circuits

find products and also order supplies if the amount of items is low. The project described by this paper tries to give a solution for the mentioned problems.

RELATED WORK

There are several development work happened last few year in human computer interaction (HCI), home automation and embedded technology. A big set of these work is giving intelligence to rigid objects and allow humans to communicate with them and vice-versa by applying noble HCI techniques. Moreover, post-WIMP devices also offer some features that can be integrate with the modern computer technology development (Ubiquitous computing). However, this post-WIMP GUI concept is only applicable if there is a metaphor available in digital or analogue world. For example, searching the meaning of a word in digital dictionary (e.g. Smart phone dictionary). We want explain decent amount of successful research work that overlap at least in certain area with our Smart Shelf framework; however, there is no implementation or ground work fully overlap with our concept. A technical definition of our project is "Combining different interaction technique to innovate a device that follow the guideline of ubiquitous computing". The most related topic that are already known by design community are: QR-Code for presenting information, automatic amount calculation and controlling device. Finally, the one market leading company Amazon popped out the idea of *Amazon Go Store* that is completely automated without any checkout counter. However, some big stores like this required navigation to finding items a shopper wants, as the the Amazon's slogan of this store is "No Lines, No Checkoutänd their objective is to reduce time consumption in shopping. We really appreciate their idea, however we must say still this smart store have lack in finding items in a minimum amount of time without searching here and there. Our design idea could combine with their innovative design and pull out completely efficient shopping environment in the palm of shoppers.

QR-Code for presenting information

Nowadays application of QR-Code become very popular and common due to the smart phone technology. Now people don't need to type search. Pressing a key is enough to get information based on QR-Code. A very innovative application is using QR-Code in library management. In a case study "Application of QR Code Technology in providing Library and Information Services in Academic Libraries" [4] by Sandeep Kumar Pathak showed that important information can be presented by QR-Code and users can easily get all those information by scanning QR-Code. We are implying this idea in to completely different perspective. In our case every drawer will have individual QR-Code. Each code will represent individual information about items stored in the drawer.

Automatic amount calculation

One major objective our implementation is representing empty or not empty drawers. As it's a very ground level work of many automation project, there are many project information available regarding weight measurement. However, in our project we are counting the objects based on the overall weight. We don't see this sort of work is not very common to automation community. Although, the most related work to that sub task is counting weight based on resistive sensor. An example of this work presented in [5]: Arduino Weight Measurement using Load Cell and HX711 Module [2]. Here they use Load Cell, but we will also use resistive load sensor to calculate the weight signal to test how successful to calculate amount of elements. Also, their project does not include counting.

Figure 1. Example for used Shelf as prototype

Controlling Device

The most important human to machine interaction task in this project is giving command to the system by the user. There are several ways to build up this interaction system. One could be developing from scratch and another is building up over existing individual system. It's very common in *Internet of Things* (IoT) community to use a smart phone for controlling a electronic system. For example there are lot of projects that use Android devices for home automation. However, our approach is similar but objective is different.
We see there are many existing work happened in granular level, but here we are bringing these granular ideas to build a completely new noble system.

CONCEPT

This section describes the basic concept of the *Smart Shelf*. The basic idea of the Smart Shelf is to enhance the interaction with shelves to work with these more effective and to improve the interaction. To satisfy these objectives some core features are proposed in this section. Figure 1 shows a similar shelf to the one used for the project prototype which is small but satisfies the requirements for this prototype. However, one important fact which should be considered is the scalability. The applied techniques should be feasible in a technical and economical way.

The first problem the Smart Shelf wants to address is the fact that it is hard to find a specific item in a high amount of drawers. To do this we propose visual feedbacks combined with an user interface to submit search queries. The user interface can be used to insert a search word, for example the name of the searched item. This search query results in a visual feedback on the Smart Shelf, or in detail, on the specific drawer where the item is contained.

The user interface will be provided as webapp. The platform independence of a webapp leads to this design decision. In a first iteration mobile apps are considered, but were discarded because of the platform dependency (Android, Apple iOS, Microsoft Windows, etc.). Additionally, more features are provided by the webapp. Overall, it provides all functionalities to interact with the Smart Shelf. The search, control the shelf or getting more data about different items. For getting more information about a specific item the user interface provides additionally to a datasheet-like page the capability to scan QR-Codes. These QR-Codes are mounted on each drawer and encode the identifier of the drawer. A central server, which also serves the webapp holds the state of the Smart Shelf and has a mapping of these scanned identifiers to drawers and items. The different items contained in the shelf, the amount of them and current searches are called here as *state*.

As mentioned before an unsatisfying fact is, if the user search for a while for a specific item and finally find it, the drawer is empty. To address this problem the Smart Shelf provides a *predictive management system* (abbr. PMS). This PMS enables the system to send notifications to operators if a drawer is empty or almost empty. Notifications enables the operators to react early to items with low or zero amount. This decreases the chance of out of stock items. To realize such a PMS data are needed for analysis to determine different reactions. Every drawer will get a electronic unit to measure the weights of the items in a drawer. Theses measurements are send to the server, which calculates with the stored weight of the specific item, the amount of items in that drawer. This enables the PMS to have an overview of the state about all items contained in drawers. Additionally to notifications a visual feedback is given at the shelf itself. Empty drawers are marked with a red light to indicate this to users directly.

As mentioned before visual feedback is a central point in the design of the Smart Shelf. Therefore, it is used to indicate the searched drawer to the user and if a drawer is empty. However, the prototype uses visual feedback additionally for a so called *service mode*, which can be switched on and off. During service mode every drawer visualizes its state with different colours. Blue colour, if the drawer contains more items than a specified threshold. Yellow colour as warning, if there are less items than the threshold and a red colour, if there are zero items left. For this service mode the data acquired with the weight measurement are consulted.

Visual Feedback Colours

There are several ideas of giving notification for search item in drawers. For example, sound, fancy display could be used for notifying users. However, we used LED for fulfilling that purpose because LED catches more human attraction towards the point where led glowing compared to sound happening at a point. Moreover a display could do mentioned task to notify a place, but displays are not as bright as LED in lighting condition. Besides, implementing visual feedback system with LED, LEDs are cheaper than the other mentioned options. So, we choose the best option, LED that fulfils our objective. Now, the problem is which colour can provide information meaningfully to user. We found red LED is the best option for giving notification about empty drawer, because culturally humans are familiar with red colour as stopping point. For example traffic light. Moreover, we have one more information to pass to user that is search item available in a specific drawer. We chose blue as a representative for that information. The reason of choosing blue is visually near to the colour of green, but blue LED looks brighter than green LED. So, by breaking the cultural analogy we came up with the idea that a blue light could be good replacement of green light that represent availability or allowed access in real life situation. Furthermore, we allow two users using the system at the same time. We can notify empty drawers to both users by only red LEDs because the users don't have to check the drawers (no collision) if the drawer is already empty. On the other hand, if the tow users' search item is already available we must have to show them which drawer they should check out for their desired item. As it mentioned blue will still used for the for the first user, but for the second user use yellow. The users are also notified by the webapp which coloured LED give them the hint about their desired items. More description of choosing colours can be found in A Review on the Role of Color and Light in Affective Computing [7].

Assumptions

To realize this prototype some preconditions are assumed which are listed in this section.

All items in the shelf are either without packaging or with packaging. There will be no drawer where items are mixed. Furthermore, at any time there is only one sort of items in one drawer. Every drawer contains different items. The weight of a specific item is stored in the server for the amount calculation.

IMPLEMENTATION

This chapter shows the important details of the Smart Shelf implementation. The system of the Smart Shelf itself consists of three main components. The backend server with a MySQL database, a webapp served by the backend and Smart Shelf itself with an Arduino as main microcontroller connected to some electronic components placed in the drawers. Basically, the Smart Shelf is contained of several Smart Drawers. These components and how they communicate with each other are described in the following.

Backend

For the backend implementation the Spring framework is used [6]. This enables to build a lightweight backend server that can serve web pages, a REST API and can connected to a MQTT[2] broker. The communication between backend and Arduino with MQTT is explained in section 4.2 in detail. To serve the webapp the backend implements several REST Services which can be consumed via HTTP. Some of this services produce the

[2] Abbr. Message Queuing Telemetry Transport Protocol

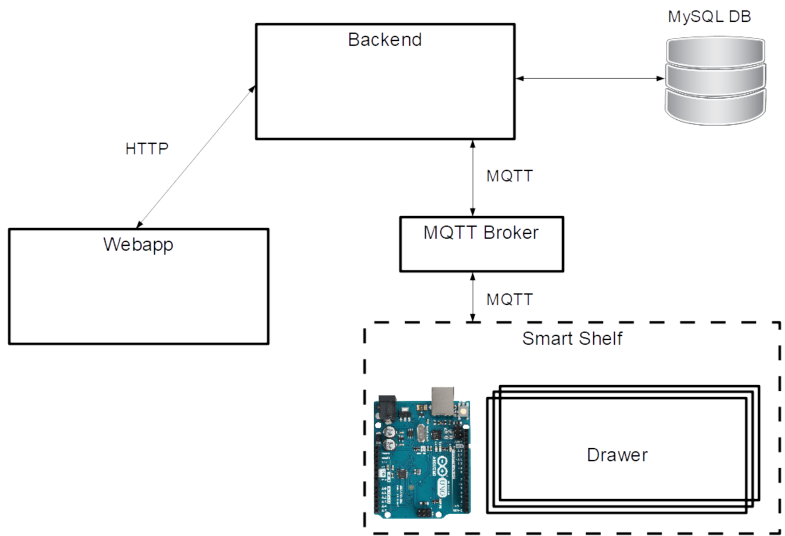

Figure 2. Infrastructure of the Smart Shelf system

different web pages for the webapp. Others produce just data to keep the information in the webapp up to date.

As mentioned before the backend keeps the state of the whole system. For that it implements some queues for all current search queries and has a connection to a MySQL database with all the essential data needed for the Smart Shelf. The database contains for example all items contained in the shelf, in which drawer these items are and what boxes are there. Additional information like the amount of different items are stored, too. To enable the PMS the backend implements also some services to mark an item as out of stock and to send an email to defined operators as notification. These operators can subsequently order supply for the appropriate items. In addition to the sent notification the backend sends a command to the Arduino to switch on the red LED on that drawer. This signals directly to each user that this drawer is empty.

To use more of the capabilities the visual feedback provided via the LEDs on the drawers the Smart Shelf implements a so called service mode. This mode can be started from the webapp. After started the backend checks the state of all drawers and send the appropriated commands to the Arduino to switch on the correct LED on all drawers. As mentioned before there is a mapping between the colours of the LED and the amount of items in a drawer. Blue colour signals that the amount of items in that drawer is above a defined threshold. Yellow colour indicates that the amount is below the threshold and red visualizes that this drawer is empty.

Communication

For the communication between Arduino and backend the lightweight M2M[3] communication protocol MQTT was chosen. MQTT works with the publish and subscribe pattern. There is a central broker where clients can subscribe topics. Additionally, clients can publish messages to different topics. If a client subscribed the topic under which a message is published, the client will receive it [8]. For further information about this protocol check [8].

With MQTT a bidirectional connection can be established between backend and Arduino. Primary, the backend send commands to the Arduino which LEDs the Arduino should switch on or off. But additionally, the Arduino send in a defined interval the measured weights of items contained in different drawers. How the weight measurement is done is explained in following chapters.

Control System of Drawers

The whole concept of smart shelf follows a IoT principle. According to previously mentioned MQTT protocol the Arduino which is the main controller of the smart shelf is subscribed with defined topics in the MQTT broker. The server itself publish to topics based with on the identity of each drawer so that Arduino can receive the user command from the MQTT broker and send corresponding to each of the drawers. After receiving the information Arduino parse the JSON text and invoke command to glow any led based on the server side information. Similarly, the reading from sensor send to the backend in the same way the we receive information from backend, but this time the backend act as a subscriber to topics and Arduino publishes on that topics. The communication tool and hardware used for communicating with server is Ethernet communication which solely operated by the Ethernet card Arduino Ethernet Shied [1]. We used Arduino Client for MQTT [3] library for Arduino to MQTT Broker communication.

[3]Abbr. Machine-to-Machine

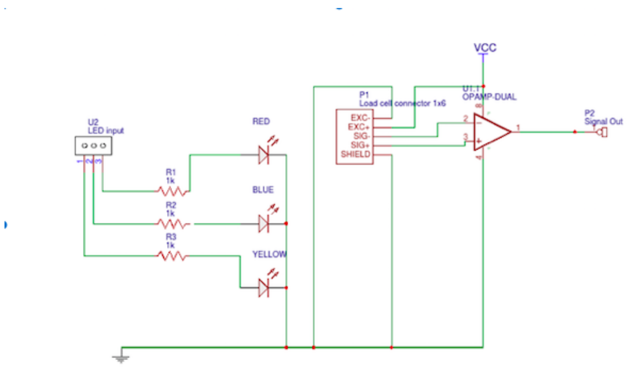

Figure 3. Circuit diagram of smart drawer

Smart Drawer

To fulfil the purpose of prototype and reduce the cost of implementation we implemented two fully featured drawers that includes weight sensing and visual output and four basic drawers that consists of only visual output modules. The fully featured drawer contain one load cell (Model), the load cell is employed to measure the pressure of equipments that is generally stored in the drawers. The signal of the load cell is then amplified via an amplifier to feed into the Arduino. Moreover, The visual feed back module is made of three LEDs that are operated by the Arduino. Similarly, the basic drawers work based on same principle; however the there is no weight sensing technique. The schematic 3 shows the hardware implementation of the full featured drawer. Also the figure 4 shows the visual demonstration of drawer. The transparent plate attached with the load cell is used to place equipment to over it to measure the wight. The load cell below measures the pressure due to the placement of equipment over the visible board.

Webapp

To control the Smart Shelf a webapp is served by the backend. This webapp can be opened via an URL from almost every browser which is connected to the network the backend is in. Implementing the user interface as webapp brings the advantage of platform independence. It can be used form a Android or iOS phone as well as with computer.

Figure 5 shows the main page of the user interface. The main function of the system is to make it easy searching for a item in the shelf. Therefore, a big search field can directly be seen at the main page. Inserting there the name of the searched item results in a list of items matching the inserted name. Additionally, all LEDs of the drawers which contain the displayed items in the webpage will be turned on. By using the SSelect-button for the item to choice all LEDs will be switched off except the selected one. With this functionality the user can easily find the search item without scanning the all drawers for the specific name.

To get more information about an item one can uses the data sheet provided by the webapp. This can directly reached from

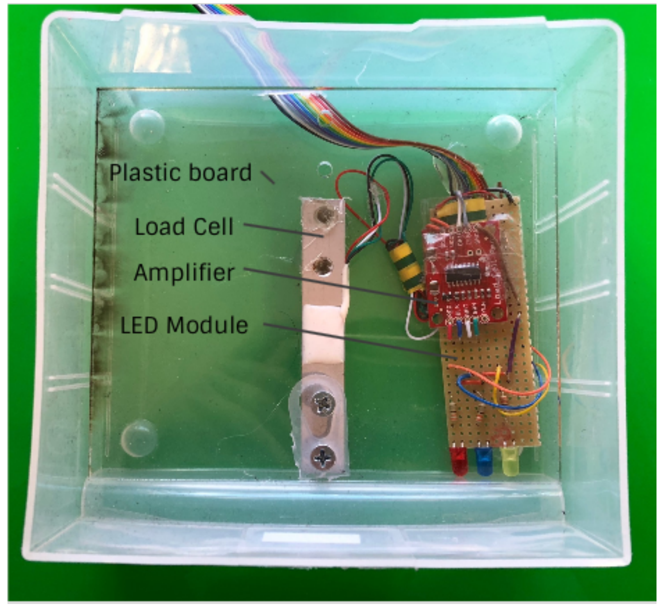

Figure 4. Hardware configuration of a smart drawer

the search result, also seen in figure 5, or by scanning the QR-Code mounted on the appropriate drawer. The QR-Code brings the advantage that if a user need more information about a specific item and is standing right in front of the shelf he/she doesn't have to search again and afterwards select the data sheet. The data sheet can easily get by scanning the QR-Code with the smart phone which used to operate the webapp. As mentioned before the QR-Code encodes the ID of the drawer. By scanning this ID is decoded, send to the backend which responds with the needed information about that drawer to visualize the correct data sheet.

EVALUATION

To evaluate interactive Smart Shelf, we tested it in real world scenarios. We found some strengths and weaknesses during the evaluation of the project. Moreover we did four activities during evaluation that are listed in table 1. In the following some strength and weaknesses are described and if the functionalities described in previous sections are implemented.

Strength

- The drawer design is reliable and long lasting. The system has a user interface for multiple users.
- In service mode, colour of LED's follows cultural analogy, i.e. Red (no item), Blue (amount of items is above a defined threshold) and Yellow (amount of items is under threshold) in each drawer.
- If there isn't any item in a drawer, system send notification email to operator and marks the drawer with a red LED.
- User can start service mode via the webapp that depicts the current state of Smart Shelf.
- More than one user can search items simultaneously.

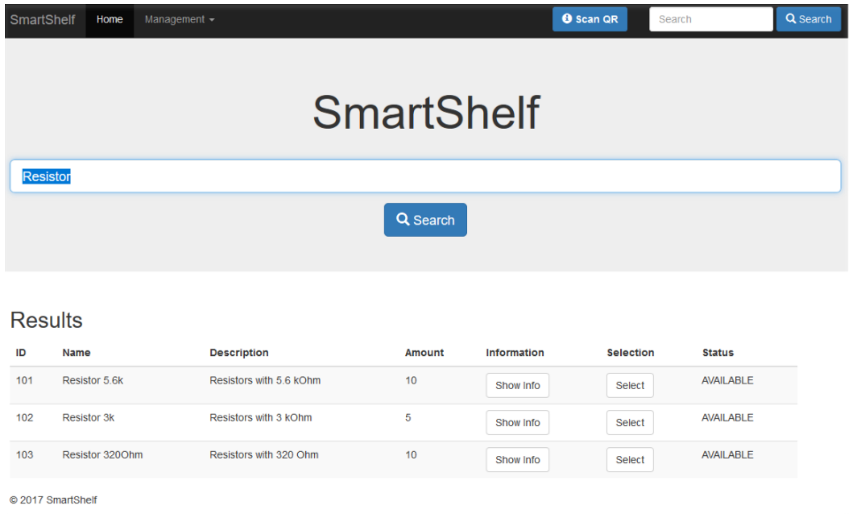

Figure 5. Mainpage of the webapp

- Using QR-Code user can get detailed information of items i.e. capacity of resistors or amount of items in drawer etc.

Weakness

- LED's inside drawer is not clearly visible for visualization.

  Solution
  Use huge diffused LED's (10mm diameter). These are really bright so these can be seen in daytime. These have 465-470nm wavelength, 3.0-3.4V forward voltage and require 20mA current. Typical brightness is 1000mcd instead of using 5mm. Additionally, the LEDs could be placed outside of the drawer.

- Load Cell carries a plastic plate inside the drawer to measure the weight of the items in it. Because of the soft plastic the drawer are made of sometimes the plastic plate sticks or tilt with the drawer walls. This leads to false measurements of the weights.

  Solution
  Drawers made of more solid material should be used.

- Power supply is a issue if higher amount of LEDs are switched on.

  Solution
  The origin of this problem is the limited power supply by the Arduino. To tackle this problem an external power supply can be added to the LED circuits.

Best Scenarios

During the evaluation some scenarios has shown they satisfies the problems explained in previous sections. Searching an item in a shelf works best and was one of the main objectives of this project. Additionally, the service mode showed good results to give a visual feedback to users about the status of the drawers.

Activity No.	Successfully fulfilled task
1	4
2	4
3	3
4	2

Table 1. Overview of the user study results

User Testing

To evaluate how the webapp of the Smart Shelf system performs with real users a small user studies was done. The results of the activities done in that study are listed in table 1. All activities are done two times. The first time the user was able to make himself/herself familiar with the task. In the second round with an slightly other objective it is measured if the task could be fulfilled successfully in a given amount of time.

Activity 1
The use had the task to search for a specific item with the provided user interface.

Activity 2
To get more information about a specific item the user had to scan the QR-Code on a specific drawer to get the datasheet.

Activity 3
Objective of this activity was to obtain the amount of a specific item in the shelf.

Activity 4
With this task the user should enable the service mode and give a feedback about empty drawers, half full and full drawers.

We tested all these activities with four different participants and interviewed these afterwards to collect feedback. The recoded data is shown in this section. The age distribution of the participants were 20 to 29 years, were the people are either employees or students of the University of Saarland.

Overall for the basic functionality of the Smart Shelf we observed almost 100% of successfully fulfilled task for all participants. All participants were able to fulfil the first two tasks within a short amount of time. For Activity 3 only one participant was not able to solve the task within the given time. After interviewing the participant we got the answer that it was not clear to the participant that he/she has first to search for an item to get the amount of items in the shelf. The participant was searching for an overview page of all drawers/items to get the information about the amount of items. To solve Activity 4 only two of four participants were able. Reason for that could be the menu to enable the service mode. To enable the service mode the user has to click the collapsable menu “Management“ and follow the link “Service Mode“. One participant was not able to find this link. For the other participant the information about the meaning of the colours on the drawers was not clear to him/her.

One limitation of the proposed solutions in the implementation section is the size of the drawers. Electronic components for weight measurement consumes a lot of space in the drawer. This results in a very limited space for items itself. Because of that reason we decided to apply the weight measurement in the prototype only to the bigger drawers. The small drawers contains just the board with the LEDs as can be seen in figure 4.

The evaluation of the project reveals some important aspects in which the Smart Shelf can be improved. The results described in this section show that the basic functionality reached good result with users which never got in touch with the new Smart Shelf. But other, more complex functionalities like the service mode need more work to improve the usability. During the interviews with the participants two minor improvements were proposed, too. If a user searches for an item, first all LEDs on drawers which are in the result list switched on their LED. Only after selecting the wanted item in the result list (see figure 5) all LEDs are switched of and only the LED on the correct drawer is switched on. Some users were confused about this and proposed to switch on only one LED after the selection. Furthermore, the buttons in the menu bar on top of the webpage are not that big. These could be bigger that they be more easy to detect and pressed, especially on mobile devices.

A important point for improvement is the usability of the drawer itself. Currently all drawers wired with wires and can not easily taken off the drawer or for example to another place. In a next version of the drawers these should be wireless to handle them more easily and more convenient. This is also explained in the section Future Work.

FUTURE WORK

This chapter describes future work which can be applied to Smart Shelf to add more features and make the user interaction better.

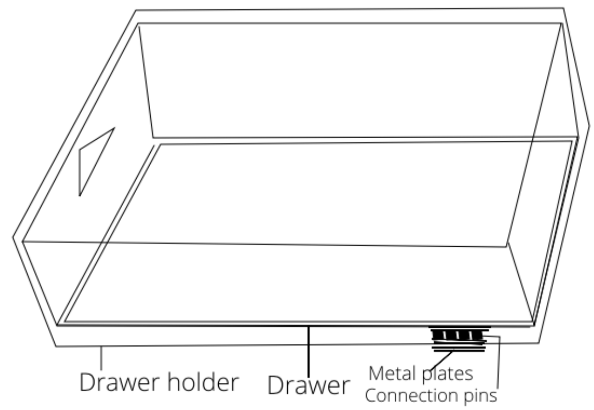

Figure 6. Sketch of connector for removing dependency on connecting wires

Wireless Drawer

Currently, every drawer is connected with the main controller via wires. However, this wires reduce the user friendliness, if an user pull out the drawer completely. For the purpose of fast and cost effective prototype implementation we restrict ourself to implement such a sophistical system. The idea of making each drawer independent of wire is placing connection pins bottom of the drawers and also the out going wire to the controller should be placed with small plate in the main structure of the shelf, so that the when the drawer is closed each of the drawer get a connection between the pins and plate and make bridge between the drawers and controller. The figure 6 shows the idea of implementing such a drawer with a very minimalistic complexity.

Showing direction

The main motivation of this prototype design was to show the visual output only on the drawer where the search item belongs. However, we noticed that we can use other LEDs from other drawers to visualise the direction of the drawer which consists search items. This idea is really helpful when the shelf is very big and we can only see a portion of the shelf if we stand closer to it.

Large Scale Production

We implemented the prototype only for six drawers, but we designed the framework in such a way that it easily expendable to hundred of drawers. Due to the limitation of numbers I/O pins in Arduino we have still limitation to extend the number of drawers. This problem can easily be solved by mapping output into a multiplexed form in the Arduino and from the shelf side by demultiplexing to decode the multiplexed output into the I/O mapping of the drawers.

Increasing Form Factor

Currently the shape weight cell consume a lot of space inside the drawer, for example it reduces the overall from factor of a drawer by thirty percent from the original size of that drawer. Solving this issue fairly a large scale production issue. And this can be solved embedding light weight and thinner shape weight sensor with the drawers.

Scalability

Currently the amount of drawers is limited to the amount of provided pins by the board. In a next version a form of modulation can be applied, so that not every LED needs its own wire/pin at the Arduino board.

CONCLUSION

This paper shows a solution for finding items fast in efficient in a big shelf by combining topics from the domains Internet of Things, Human Computer Interaction, Automation and Ubiquitous computing. The proposed solution *Smart Shelf* offers a user interaction in two ways. First via a webapp where the user can search for items and get information about them. Secondly, the Smart Shelf gives visual feedback using LEDs on the drawers to indicate if a drawer is empty or where the searched item for a user is. These solutions enhance the usability of such shelves. The system propose functionality to search with multiple users for different items. Additionally, a predictive management system is implemented to reduce items out of stock. Furthermore, in a evaluation with a user study some drawbacks and advantages are shown. These led to different aspects of further work.

REFERENCES

1. 2017. Ethernet / Ethernet 2 library. *URL https://www.arduino.cc/en/Reference/Ethernet* (2017).
2. Sarah Al-Mutlaq. 2017. Load Cell Amplifier HX711 Breakout Hookup Guide. (2017).
3. Nick O'Leary. 2017. Arduino Client for MQTT. *URL https://pubsubclient.knolleary.net* (2017).
4. Sndeep Kumar Pathak. 2017. Application of QR Code Technology in providing Library and Information Services in Academic Libraries. (2017).
5. Saddam. 2017. Arduino Weight Measurement using Load Cell and HX711 Module. (2017).
6. Pivotal Software. 2018. Spring Framework. (2018). `URLhttps://spring.io/`
7. Marina Sokolova and Antonio FernÃąndez-Caballero. 2015. A Review on the Role of Color and Light in Affective Computing. 5 (08 2015), 275–293.
8. OASIS Standard. 2014. MQTT version 3.1. 1. *URL http://docs.oasis-open.org/mqtt/mqtt/v3* 1 (2014).