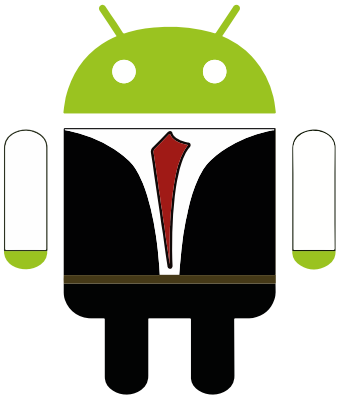
PROJECT DOCUMENTATION

FREE ORDERMAN SYSTEM

THE APPLICATION



a free solution for order management in bars, pubs, restaurants & clubs

Author: Alex Mazzon

Contents

1 Introduction

Welcome to FOS! This is the project documentation of the *Free Orderman System* (FOS), a sophisticated software package that wants to be a free solution for order management in bars, pubs, restaurant or clubs. As an open source project it is a fast, secure and simple software accessible to everyone.

1.1 Overview

This project documentation wants to explain what an order management system (OMS) is and present our solution - the Free Orderman System. This introduction gives you an idea about the project and the motivation why we created FOS.

Chapter *2 Order Management System* should be read by everyone that has no idea what an OMS is or what it is used for. The chapter explains every step of the catering process while using an OMS and illustrates a study with all success-related factors that are important to a catering company that wants to introduce and use an OMS.

Chapter *3 The Free Orderman System* reveals all secrets of our solution for a complete new software package that intends to be a real and professional solution for every catering company. It illustrates the goals of the project, the FOS application itself plus the functional interaction between every device and the database management. Technical details are also included.

Chapter *4 Development* tells something about the making of FOS. From the idea to the design and implementation plus the tools, resources and utilities that were used to develop FOS. Here you can also find links to the online documentation of the project and even to the source code.

To show that the project was feasible we decided to test it in a catering company. Chapter *5 FOS as Business Solution* includes the test report.

Finally I tried to comment where our project is going from here (chapter *6 Start Up*). As we do not want to quit at this stage. We want to present our new and highly sophisticated product to the world. It may seem exaggerated, but it is always worth trying.

1.2 The Idea

First I would like to introduce everyone to my project by telling where the idea comes from and what's its essence. I am proud to present an idea that nobody has ever thought about before. It is a completely new concept that is able to support not only big catering companies, but also smaller ones since it has no cost. Let me surprise you.

1.2.1 How and where was the Free Orderman System born?

Illustration 1: "Die Pizzerei" Innsbruck - The birthplace of the Free Orderman System

Everything started in Innsbruck when I worked in an Italian restaurant last summer: *Die Pizzerei*. There I discovered that a professional order management system is very useful but also really expensive. I never thought about it, but when I realized the real cost I had a vision: Why not use smartphones or tablets as ordermans[1] and connect them to a common computer? I presented the idea to my classmate Florian and we decided to realize the project together.

1.2.2 The Essence

Nowadays lots of companies reject taking an order from their customers with paper and pen, replacing it with electronic ordering to accelerate the catering process and be simply more efficient. If we consider how expensive a professional order management system is, there is no doubt that it is not profitable for small companies. They are excluded from using this technical device system.

The essence of the idea is to create a software that connects already available devices and infrastructures, creating no further costs. Basically we need a hand-held electronic device (e.g. smartphones, tablet...) to take the orders from the customers, a computer to evaluate

1 Electronic device for order taking in bars, pubs or restaurants

and process the order and finally a printer to print the order. Actually these three things are already available in nearly every bar or restaurant. From a small village pub to the most remote alpine pasture counter in the mountains.

It doesn't matter what brand, how small or weak your devices are, FOS wants to support every computing platform and every operating system without limiting the use of it only to big companies. The acronym FOS stands also for Free. A software for everyone.

Illustration 2: FOS uses common smartphones or tablets, computers and printers to create a free OMS

1.3 The Motivation

In a world where nearly everyone owns a smartphone, a tablet and/or a computer, in most cases only the software is missing to create something that supports some kind of process. For instance there are applications that have access to the train stations and train departures in the whole of Europe and provide this information through a simple application that people use everyday. There are personal wireless networks at home, in restaurants, bars, pubs and even networks that cover a whole city.

We see that the infrastructure is already available to create an order management system that everyone can introduce without further devices. But nobody has ever done it yet.

This is our motivation.

1.4 The Goal

The goal is to implement an open source order management software which is free but still professional. It should definitely not limit anything related to other OMS and even improve functionalities. FOS should be a complete solution for order entry and processing. It also includes a payment system.

Since FOS uses already available infrastructures it should be installable on multiple platforms. The aim is to make it available to every common operating system.

2 Order Management System

This chapter explains what an *Order Management System* (OMS) is and how it works. It shows up the limitation of an OMS and talks about expandable sectors. Finally the Hospitality Study illustrates which factors a catering company needs to consider in order to increase profits.

2.1 What is an Order Management System?

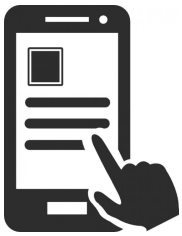
An *order management system* (OMS[2]) is a computer software system used in a number of catering companies for order entry and processing.

The catering process can be divided in three important steps: Take an order from the client. Prepare the order. Bring it to the table. It may seem very simple and primitive, but there are many different ways to administrate this process.

Paper and Pen

The most simple way of taking an order is the classic way with paper and pen. The waiter writes all products that the client wishes on a piece of paper and brings it to the bar, the kitchen or other places where the order will be prepared. When the order is ready to be consumed the waiter brings it to the table and hopefully the customer will be satisfied and leave a big tip.

Paper and Pen versus OMS



Just this short description of the process with *paper and pen* is enough to understand that there could be lots of problems during this process. The waiter could misunderstand his client, the barkeeper or the cook could misunderstand the waiter and finally the waiter could forget the table on which he has to bring the order. An order management system wants to make sure that this common problems never happen. It is an electronic system that never forgets the table number and can support the waiter in many different ways. The system provides an intuitive way of entering an order and an automatic way of processing it.

The OMS tries to minimize the mistakes or misunderstandings between waiter, barkeeper, cook and client. But it also makes the catering business faster and simply more efficient.

2 cf. [OMS]

The catering process	using an OMS
taking an order from the client	with an electronic device
prepare the order	by reading the order on the printed receipt
bring it to the table	by reading the number on the receipt

Table 1: The catering process using an OMS

The following chapters tell everything about these three important steps of the catering process using an OMS.

2.1.1 Taking an Order with an Electronic Device

The ordering device allows the waiter to review the menu, select food and beverages from the menu and electronically place the orders. Devices are either hand-held or fixed. They can provide touch or keypad. The choice of the type depends on the needs of the establishment. Generally the waiter takes the order from the customers on the table with a hand-held device and the barkeeper takes the orders behind the bar through a fixed device.

All devices are part of the same system, so that every waiter can see the orders made by all the other waiters. It could be that one waiter is allowed to view only a certain number of tables, because he doesn't need to see the other tables. For instance if one waiter works in the bar and the other waiter on the terrace, the waiter on the terrace doesn't need to see the tables of the bar and vice versa. This way the waiter maintains a good overview of his tables and cannot be disturbed by other waiters.

The Interface

On the electronic device a software is installed that provides the user interface through which the waiter enters the orders he receives from the client. This user interface must be as simple as possible and provide a good overview of the locations, tables and products of the business.

Locations, Tables, Categories and Products

The locations are all separate places of the catering company where clients are served. For example the bar, the hall and the terrace are three separate locations of a restaurant. Each location has its own tables. For instance the bar could have ten tables and the terrace twenty tables. Every table is identified through its own unique number.

A catering company has a menu with its products. The OMS holds a list of these products and every product belongs to a category. For example the product *ice tea lemon* could be part of the category *cold drinks* and the product *espresso* of the category *coffee*.

Note that every business has its own and different locations, tables, categories and products. That means that the OMS has to be very flexible and easy to personalize, for instance not only contain some standard products.

2.1.2 Evaluate and Print the Order

When the order has been taken the OMS will print it. Normally the OMS prints the order with a receipt printer[3] on paper. On this receipt you can generally find various information. The most important information on the receipt are:

(1) the location name, where the table is located

(2) the table number

(3) the waiter's name or waiter-number who has taken the order

(4) the order itself with all its products and quantities

(5) the date and time of the order

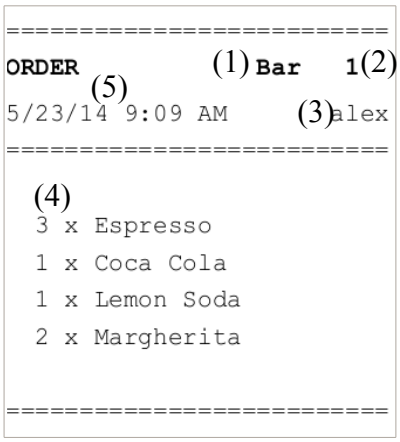


Illustration 3: Example for an order receipt

Before the OMS prints the receipt, the order must be evaluated. The evaluation process consists in finding out how to split the order and where it has to be printed. The catering company could have more places where the food or beverage will be prepared, for instance the kitchen and the bar. Normally every different activity area has its own receipt printer to accelerate the process. That is why the OMS must find out how to split the order, evaluate its products and decide on which printer to print each part of the order.

2.1.3 Prepare the Order and Bring it to the Table

After the order has been taken and printed the OMS has finished its work. The barkeeper or the cook reads the printed receipt and prepares all products on the order. When it is ready the waiter gets the food or beverage and serves it. The whole process of one order is now completed.

2.1.4 Counterorder

Waiters are human beings and humans make mistakes. For instance they could enter a wrong product into an order and print it. If a waiter notices his mistake he can do a counterorder that cancels the order. If he is fast enough he can prevent the barkeeper or cook from preparing the wrong thing.

The counterorder can be printed in the right activity area and looks similar to an order receipt. Generally it contains the title *counter*, written bold on the top of the list with all products to counter from an order on a certain table.

3 A small printer for receipts

2.2 Limitation and Expandable Sectors

The limit of an order management system is clearly recognizable: It cannot know if something has been served. This would be an important information, especially during a stressful situation. This could be realized by installing sensors on the table which tell whether the order is served or not.

Reminders

Another possible improvement would be the introduction of a reminder, which is an additional option that the waiter has to remind the barkeeper or cook to prepare an order of a certain table. If the sensors would be installed on a table and working it would be possible to send these reminders after a certain time if nothing has been served.

QR codes

The introduction of QR codes[4] painted on the table could be another solution for recognizing the table on which the waiter wants to take orders. This would be a much faster and less impersonal solution to find a table, because the waiter just needs to hold the camera on this QR-code and spend less time watching on the screen searching the table.

These functionalities would bring an order management system to the next level.

Illustration 4: QR code for the URL of the English Wikipedia Mobile main page

2.3 The Hospitality Study: Trends, Success Factors, Technologies

The Hospitality study[5] explains which factors a catering company needs to consider in order to increase profits, what type of assistive technology could help and how the owner of a catering company should deal with the shortage of qualified personnel.

More than 350 hospitality industries professionals have been asked success-related questions. A main finding of the study is as follows: Service quality is, according to 94% of those surveyed, the most important factor for success.

2.3.1 Target Group for Study

352 restaurant owners and operators were asked, 71% of those surveyed are in a management position. The type of hospitality business surveyed ranges from those with

4 QR or Quick Response Codes are a type of two-dimensional bar-code that can be read using smartphones and dedicated QR reading devices, that link directly to text, emails, websites, phone numbers and more. cf. [QRCode]

5 cf. [The Hospitality Study]

less than 50 seats to those with indoor and outdoor seats for more than 400 guests. It was an online survey with direct mailing to target group. The study has been made in 2012.

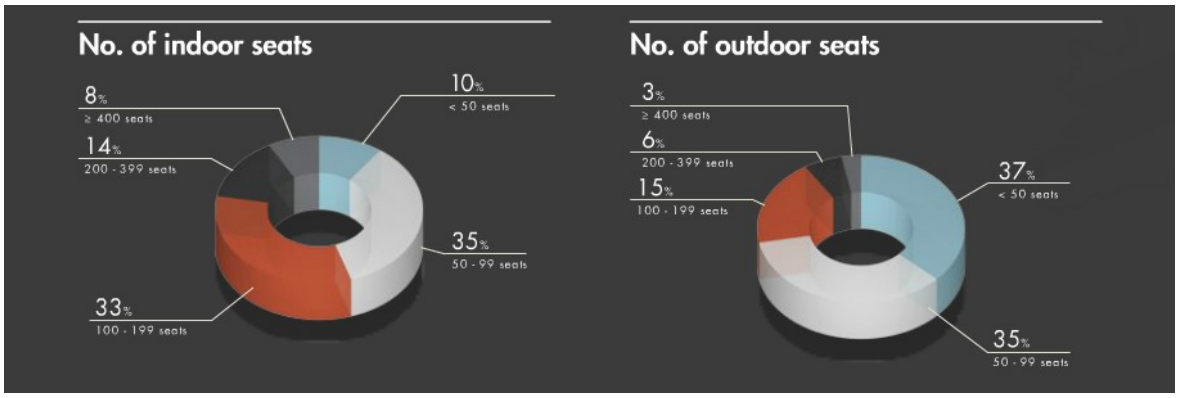


Illustration 5: Hospitality Study – Target group for the study

2.3.2 The Opinion of the Guests

The first thing to consider is the opinion of the guests. What do the guests think about the use of an OMS? The study showed that only 12% of the guests think that using an OMS when taking an order is impersonal. The majority (55%) think that it is accelerating the catering process and every third thinks that it is modern.

Conclusion: If we consider these facts there is no more doubt that an OMS has a positive impact on the vast majority of the guests.

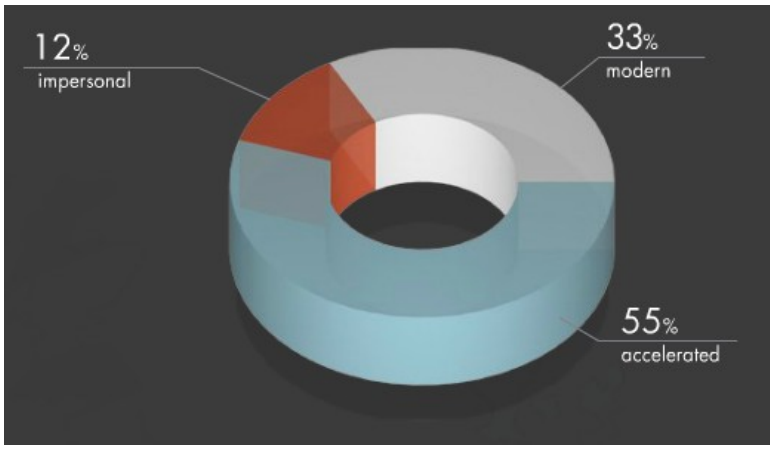


Illustration 6: Hospitality Study – The opinion of the guests

2.3.3 Success Factors in the Hospitality Industry

What does the future hold? What really matters and how can all this knowledge lead to more success?

Success is defined primarily through service: 94% of the survey participants named service quality as the number one criterion for success, particularly the food and beverage quality. The third most important factor is the efficiency of service process, which can be increased by using an OMS. Price is not the main issue.

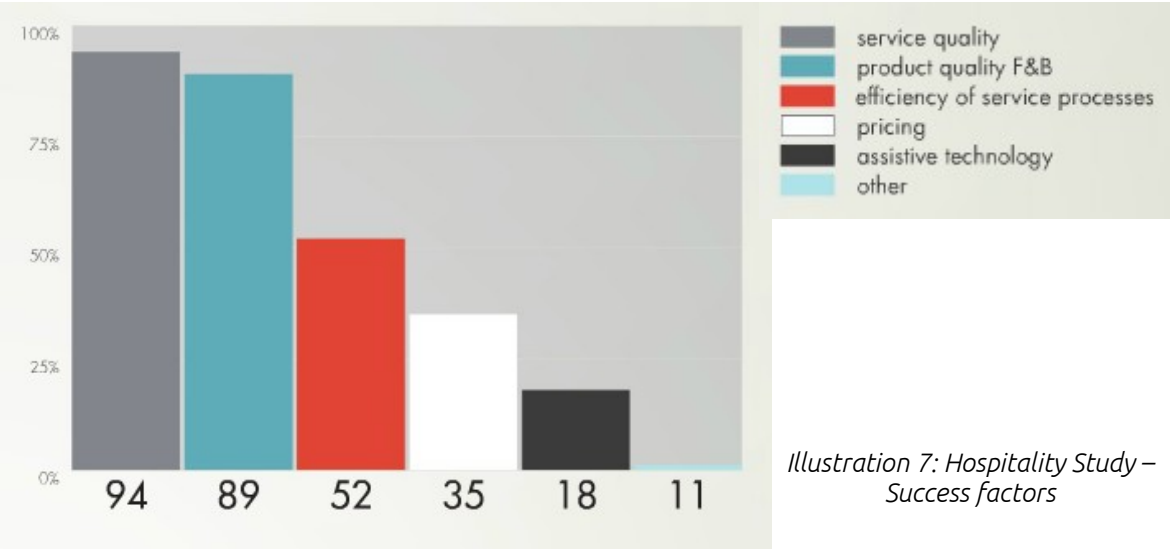


Illustration 7: Hospitality Study – Success factors

2.3.4 Tools that Support First Class Service

An important element for first class service is the interaction of kitchen and service (or bar and service): Incoming orders have to be forwarded immediately to the kitchen and bar. It is best if this information is communicated via OMS.

2.3.5 Range of Benefits from using an Order Management System

More than three-quarters of restaurant owners could identify a significant increase in speed. More than one-third perceive positive effects on the employee satisfaction.

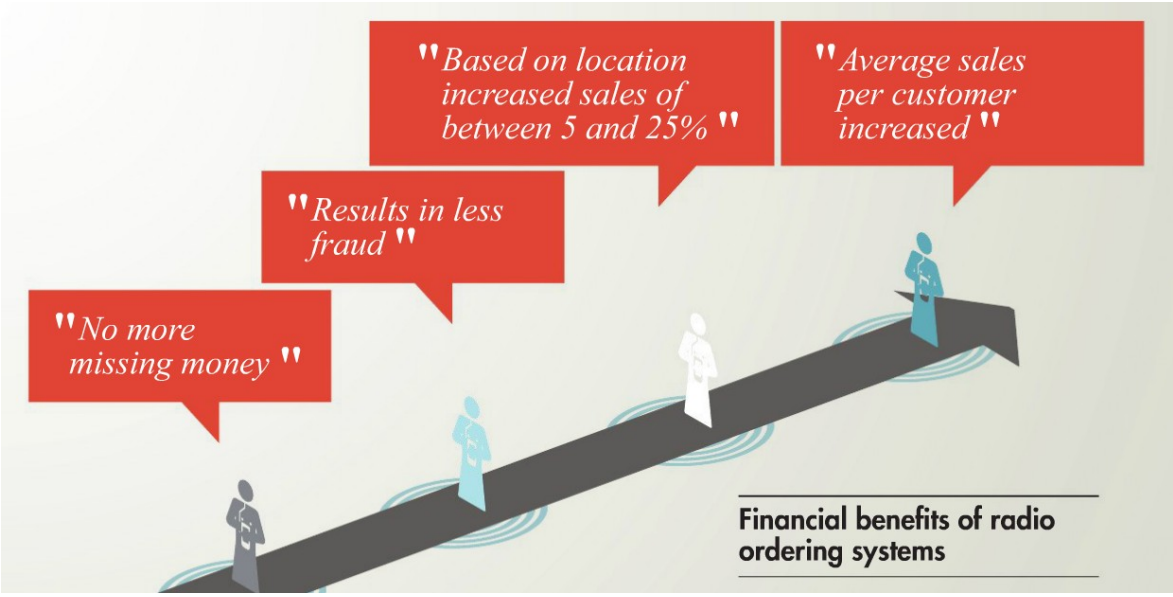
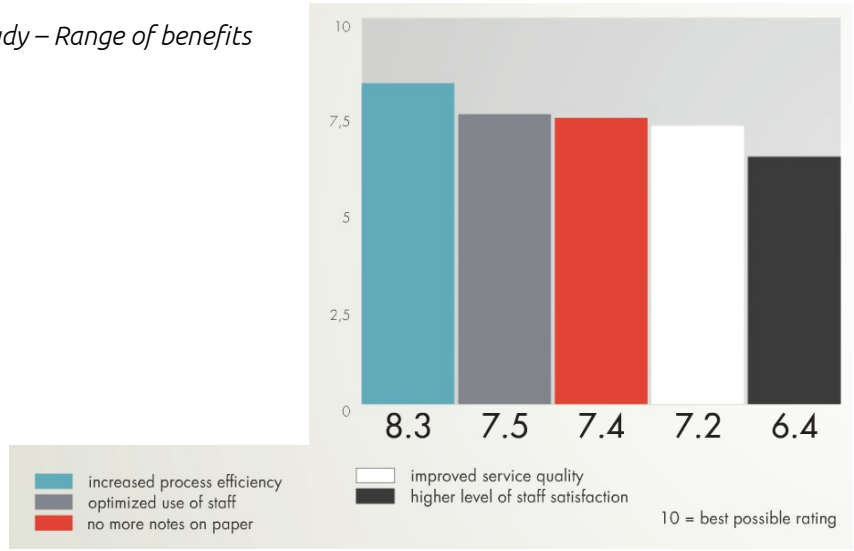


Illustration 8: Hospitality Study – Range of benefits from using mobile systems



3 The Free Orderman System

The Free Orderman System (FOS) wants to be a free solution for ordering management in bars, pubs, restaurants, clubs etc. As an open source project it will be a fast, secure and simple software accessible to everyone.

FOS is a software package that consists of the *FOS Application* and an administrative program (the *FOS Server*).

3.1 The FOS Application

The Application is that part of FOS that makes the waiter able to take the orders from the customers through an electronic device. Generally the waiter takes the order from the customers on the table with a hand-held device and the barkeeper takes the orders behind the bar through a fixed device (see chapter *2.1.1.Taking an Order with an Electronic Device*).

3.1.1 The Goal

Fast, secure, user-friendly & good-looking

The primary aim of the application is to provide a fast, secure, user friendly and good-looking application. It is important that the waiter doesn't loose time during his work, because an order management system must accelerate and not decelerate the catering process (see *Table 1: The catering process using an OMS*). It should also support the waiter as much as possible. He must have a good overview of all his tables and do fast ordering.

Every click is an expenditure of time

The waiter is obviously much faster if he gets used to FOS and knows all its functionality, but FOS should not limit, constrain or even stop anything during the work of the waiter. In less clicks as possible the waiter must be able to administrate his work. For instance the application must only show those animations that are really necessary to understand what happens.

3.1.2 Login – The FOS Authentication System

The Free Orderman System is able to manage various waiter accounts. With these accounts every waiter gets access to the system through the application.

Every waiter can work on some tables and these tables are linked to the account of the waiter. He can either work on all tables of the restaurant or work just on those tables that were assigned to him. Generally the boss decides which tables belong to which waiter and he can regulate this limitation in the administration tool of the server (see *3.2.1 The Administration of FOS*).

FOS needs to manage various waiter accounts

It often happens that the boss decides that some waiters can work on the same tables. Therefore it must be possible for two or more waiters to work on the same table at the same time. For instance this happens when two waiters want to get the orders together from a table with many people.

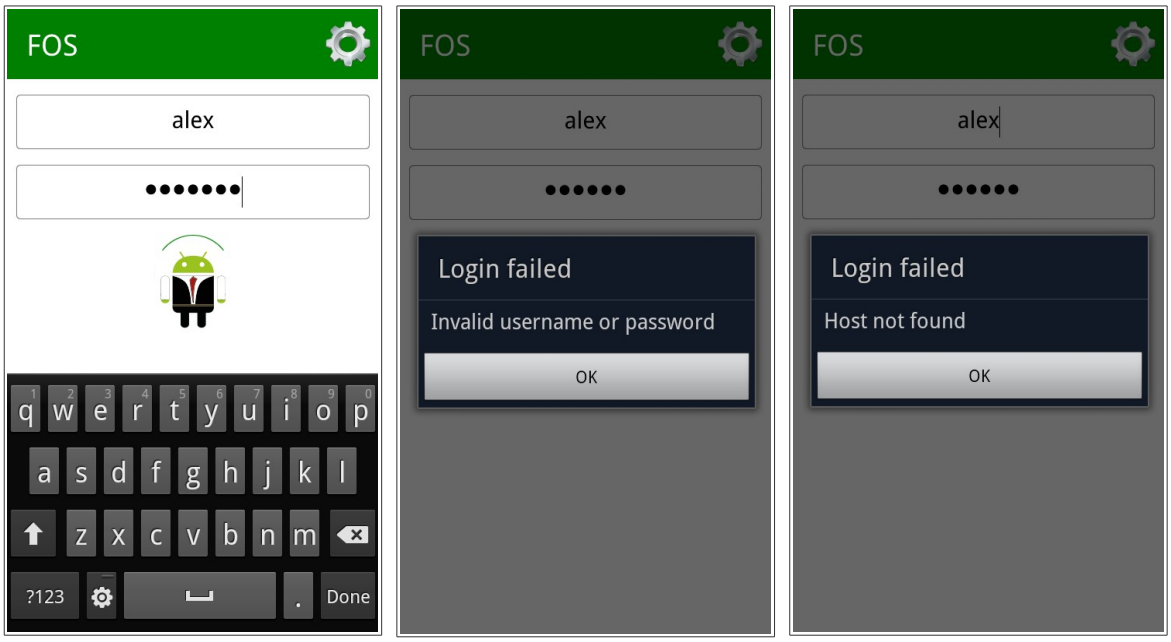


Illustration 9: The FOS authentication system

Another reason why FOS needs to manage various waiter accounts is because we have to prevent the system from being hacked or abused. Someone could sit down in the restaurant and use his smartphone, where the FOS application is installed, to make orders and even virtually pay his orders without really paying them. That would definitely not be a positive increase of the earnings. That is why every catering company has its own users that access the system with user-name and password.

Let's say that waiter A can work on the tables 1, 3, 5 and waiter B can work on the tables 2, 4, 6. In this case the waiter A doesn't even see the tables 2, 4 and 6 on his application. He only sees his three tables, the same for waiter B.

What happens after the login and possible problems during the authentication

During the authentication the application could encounter some problems:

- The device is not connected to the same network as the FOS server

- The waiter has inserted the wrong user-name or password

In these cases the application shows up an error dialogue which indicates the problem.

If the login was successful the application shows all the locations with all the tables assigned to the waiter allowing him to start his work.

3.1.3 The Locations Interface

The waiter is now logged into FOS and connected to the server. The application receives all the initial data from the server, which also contains the locations of the catering company (see *2.1.1 Locations, Tables, Categories and Products*).

Illustration 10: Choosing locations and tables

Choosing a location

At this point the application visualizes all locations through a simple user interface. A location is clearly represented by a button that contains the abbreviation of the location name.

The waiter can choose a location by clicking on it. The application will change the interface and visualize all tables of the chosen location.

Logout

From the locations interface it is also possible to logout from the system. When a waiter logs out he will be asked if he wants to print his clearance. The clearance contains the daily total earning of the waiter. Normally the waiter brings his clearance with all the money, which must correspond to the total earning on the clearance, to the boss. This hand over generally happens when the waiter finishes his working day. This is a good check, if all the

money is at hand. The waiter can check if he has collected all the money during his work and also the boss can review if something is missing.

3.1.4 The Tables' Interface

After a location has been chosen by the waiter the application visualizes all the tables of this location (see *2.1.1 Locations, Tables, Categories and Products*).

Choosing a table

A table is clearly represented by a button that contains the table number. The tables can have two different colours that indicate whether there are orders on the table or not. For instance white tables are tables where no orders have been made yet and black tables contain minimum one order. The waiter can choose a table by clicking on it. The application will change to the order interface.

The Menu

By press and hold on a table a menu will appear on the top of the screen. This menu allows to split a table or move all orders of one table to another table.

3.1.5 The Order Interface

After a table has been chosen the application visualizes all products that have already been ordered on the table through a clearly arranged list view. If no order has been made the list is empty and no products are visualized. On the top of the screen a green bar shows the chosen table number and the current total price of all orders on the table.

Make orders

On the bottom of the screen the waiter is able to make new orders by choosing a category and finally the product to order. If the product was previously ordered the new count will be added on the left side of the product in the list. Every click increases this number by one. If the clicked product was not previously ordered it will be added to the list with a new count set +1.

By press and hold on a product in the list a dialogue appears. It asks the waiter how many items he wants to order. This allows him to be very fast in ordering a certain quantity of a product which has already been ordered. For instance if the customers on a table say that they want two more *Tiramisù* (they had already ordered one *Tiramisù*) the waiter only needs to press and hold on *Tiramisù* and click on two.

Decrease the quantity of new orders

If the waiter wants to decrease the quantity of a product on the next order, he just needs to click on that product in the list and the new count decreases by one.

This can only be done if the order was not sent and printed out yet, if it was already the waiter has to do a counterorder.

Illustration 11: The order interface allows to order new products and visualize already ordered products through a clearly arranged list view

Orders visualization with a clearly arranged list view

Every product that has been ordered is represented through a row of the orders list view. This row visualizes the following information:

(1) the new count, which visualizes the quantity of the product to order with the next order

(2) the count, which visualizes the amount of a certain product that has already been ordered

(3) the name of the product, or the abbreviation if the name is too long

(4) the price of the product

(5) the order price (the product price multiplied by the count)

The menu

The menu of the table appears by clicking on the green top bar of the screen. It overlays this bar and shows the menu items: counterorder, pay and pay separately. See the following chapters *Counterorder, Pay and Pay Separately*

3.1.6 Counterorder

The waiter can prevent the barkeeper or cook from preparing the wrong thing by making a counterorder (see *2.1.4 Counterorder*). He enters the counterorder mode and chooses the products to counterorder. The waiter enters and leaves the counterorder mode through the menu.

Illustration 12: The countermode allows to counter already ordered products and prevent the barkeeper or cook from preparing the wrong thing. The separate payment allows the customers to pay separately.

3.1.7 Pay and Pay Separately

When a table was served and a customer wants to pay, the waiter has to print the bill and cash the money. If the customer wants to pay a whole table it is simple: He just needs to click on the menu item *pay* and the bill will be printed with all the products and the total table price.

Pay separately

If the customers want to pay separately the process is more complicated, but FOS tries to keep it as simple as possible. The waiter enters the *pay separately* mode through the table menu and sees a split screen. On the bottom half of the screen the list of all products and their quantity is visualized. This is the same list from the order interface.

By clicking on a product in the list the count of that product will decrease by one and the product will be added to the list visualized in the top half of the screen. This list contains all products that the customer wants to pay from his table. For instance if customer A wants to pay a *Pizza Margherita*, a *beer* and an *espresso* the waiter chooses the table of customer A, enters the *pay separately* mode and selects these products. They will jump from the table list with all products to the top list, where now only the pizza, the beer and the espresso are

```
==========================
BILL              Bar   1
5/23/14 8:51 AM        alex
==========================
*******NON FISCALE********
==========================

  1 x Bolognese         6.00
  2 x Coca Cola         4.00
  3 x Espresso          3.30
  1 x Lemon Soda        2.00
  2 x Margherita       12.00
  3 x Tiramisù         12.00
        ------------------
        TOTALE:        39.30
==========================
     HAVE A NICE DAY!
  FREE ORDERMAN SYSTEM
==========================
```

Illustration 13: Example for a bill

visualized. Now the waiter clicks on the pay button and the bill will be printed. It contains only the products that customer A wants to pay.

Every time that a customer pays the products will be deleted from the table. If everything on the table has been paid the table will appear as free.

3.2 The FOS Server

The server has all the data stored in a database and shares it with the clients. This database contains all users, locations, tables, categories and products of the catering company. In addition the FOS server allows to manage all the data through a clearly arranged and intuitive user interface.

3.2.1 Database

The FOS database[6] stores the locations, tables, categories and products of the catering company. It has to be very flexible and not just contain some standard products, because every business has its own and different locations, tables, categories and products. The database management interface allows to insert, update or delete data from the database (see *3.2.2 Database Management*).

Database Design

The database design is an important step. The relation between locations and tables or between categories and products must be structured and has to provide all the data that is needed without limiting anything. The database contains seven tables.

Every data record of the database is uniquely identifiable with its primary key[7].

Waiters

The table *waiters* contains all the waiters that work for the catering company. This table allows to create the login from the FOS application (see *3.1.3 The FOS Authentication System*). In addition the name is used to indicate the waiter on the printed receipts or on the bill. That is why the table *orders* contains the number of the waiter that has taken the order.

Orders

The table *orders* contains all orders that have been made as well as their properties. One order contains the product that the client ordered and the table of the client. The data fields *extra* and *extraprice* contain the information of the extras that the client wishes with this order, for instance *espresso plus water* or *Coca Cola plus lemon*. In this case *water* and *lemon* are the extras and the price of these extras is saved in the data field *extraprice*. All extras are products and normally characterized by the plus sign, added on the beginning of the product name, for instance *+lemon*.

6 Databases are designed to offer an organized mechanism for storing, managing and retrieving information. They do so through the use of tables. cf. [DB]

7 When you create a new database table, you have to select one primary key that will uniquely identify records stored in that table. cf. [PrimKey]

Products and Categories

Every product of the catering business is stored in the table *products*. Two important properties of a product are the product name and an abbreviation of this name. In addition to that every product is part of a category and has its own price.

The table *categories* contains all categories, which are simply represented by the category name and an abbreviation of this name. A category can contain lots of products.

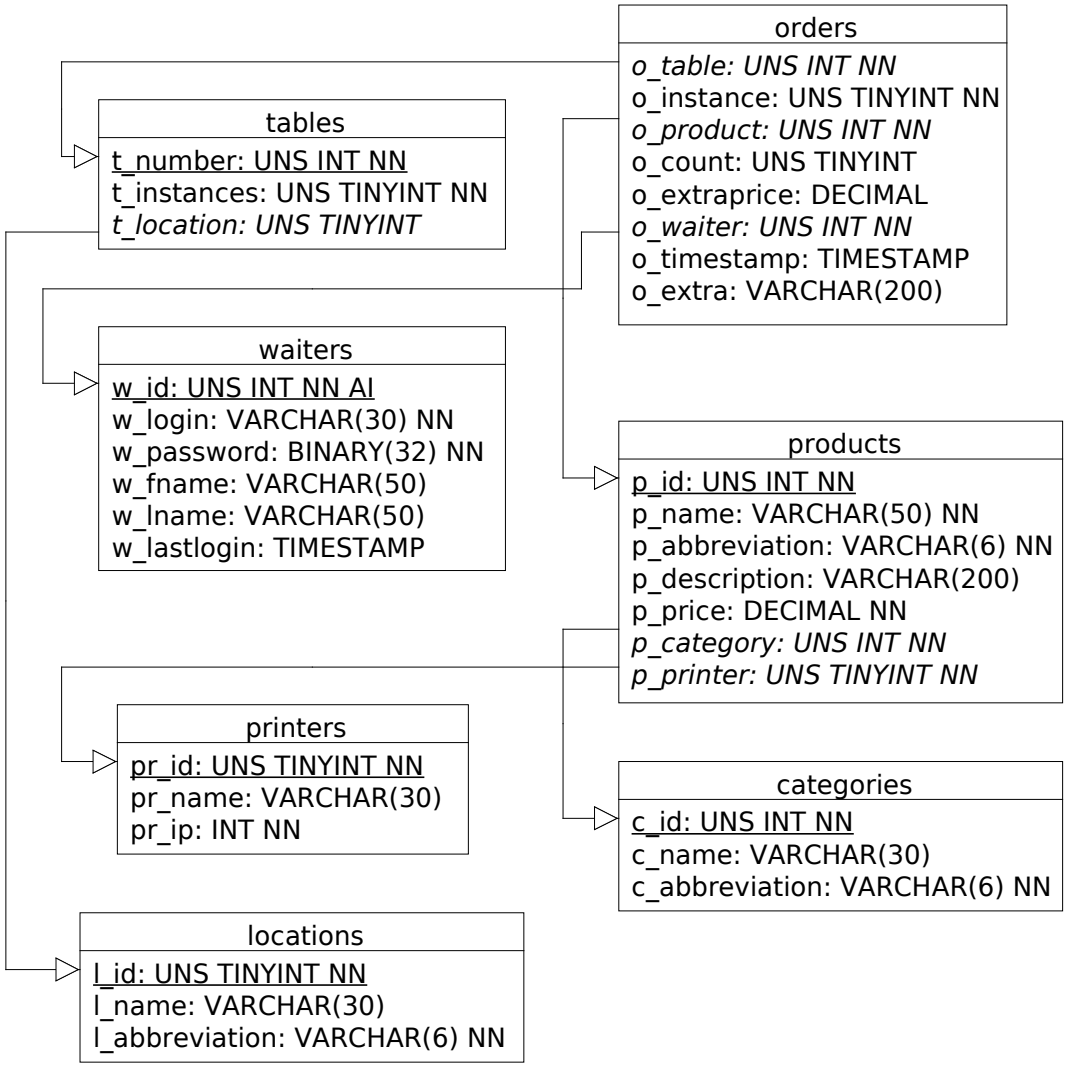


Illustration 14: The FOS database design

Printers

All printers that the catering company owns are saved in the table *printers*. FOS has to know what printers are available to decide where to print the receipts (see *2.1.2 Evaluate and Print the Order*). A printer has an IP address and a name that indicates where it is located. For instance the name *kitchen* which means that the printer is in the kitchen and every product that has to be prepared in the kitchen will be printed with this printer.

Tables and Locations

All tables of the catering company are stored in the database and contain a unique table number. Every table belongs to a location which is stored in the table *locations*. One location can contain lots of tables.

3.2.2 Database Management

The FOS server allows to manage all locations, tables, categories and products through a clearly arranged and intuitive user interface. The catering company is also able to create their own users of the system.

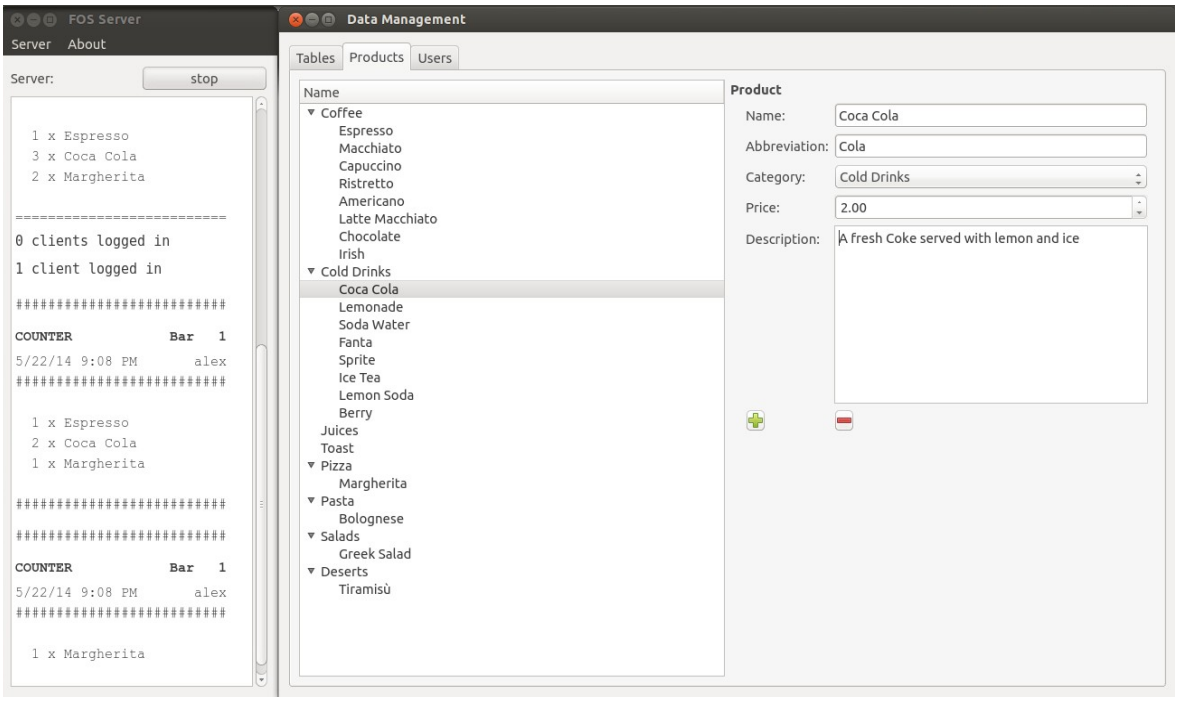


Illustration 15: The FOS server with the data management interface running on Ubuntu

Users generally are: The owner of the restaurant or bar itself and all the waiters that work for him. The boss has more privileges to administrate and manage FOS. Normally he is the only one that has access to the statistics and the database management. But other members of the staff could also be enabled to have access, for instance chef de rang or the accountant.

FOS updates the data and shares it with the clients instantly

The FOS server allows to change the data while the waiters are working and logged into the system. It is not necessary to stop the business when data gets updated. For instance when the boss decides to sell a new product he can enter the data management window and insert this new product. Instantly after this insertion the waiters receive this new product and are able to order it from now on.

The same applies to the categories, locations and tables.

3.2.3 Statistics

The FOS server creates various statistics. These statistics should give the possibility to the owner of the hospitality business to maintain the overview of the earnings and spendings. For instance if a company sells *espresso* it is interesting and success-related for the boss or the accountant to know how many *espressos* on the whole the company sells.

In addition to the spending review statistics, FOS could create other interesting statistics like the medium stay of customers.

3.3 Functional Interaction

The communication between all devices is based on the server-client model. The FOS server stores all the data and shares it with the clients. The application just visualizes the data and does not calculate or create something new. The client gives orders to the server, exactly like the customer gives orders to the waiter. The server stores the data and when the client asks something to the server, the server reads the data from the database and provides it to the clients.

The following activity diagram represents the functional interaction between the FOS server and the FOS client. This is the original protocol of FOS, which explains and defines the application life-cycle:

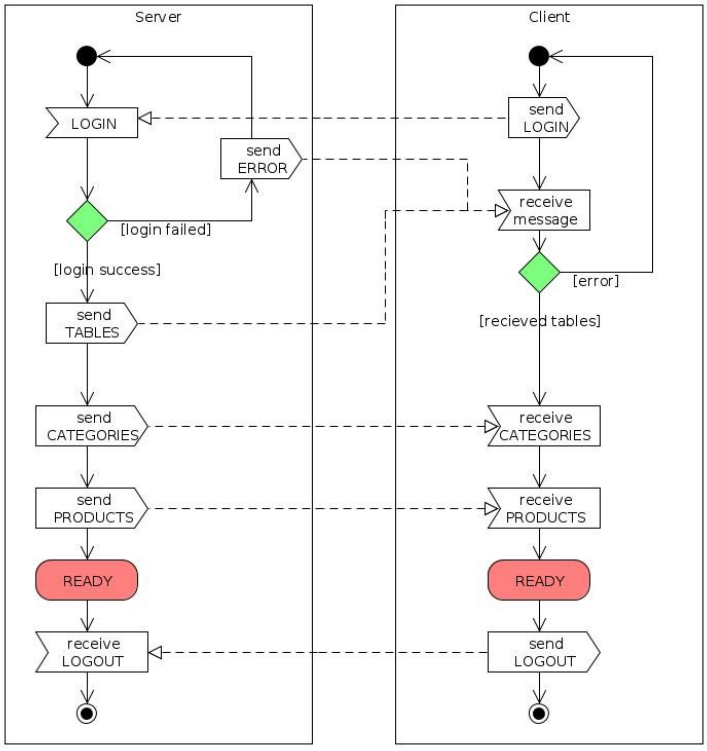


Illustration 16: The original protocol of FOS, which explains and defines the application life-cycle

When the *ready* state is reached, everything has been done and from now on the Free Orderman System is ready to use. The following protocol describes what happens when the waiter orders something:

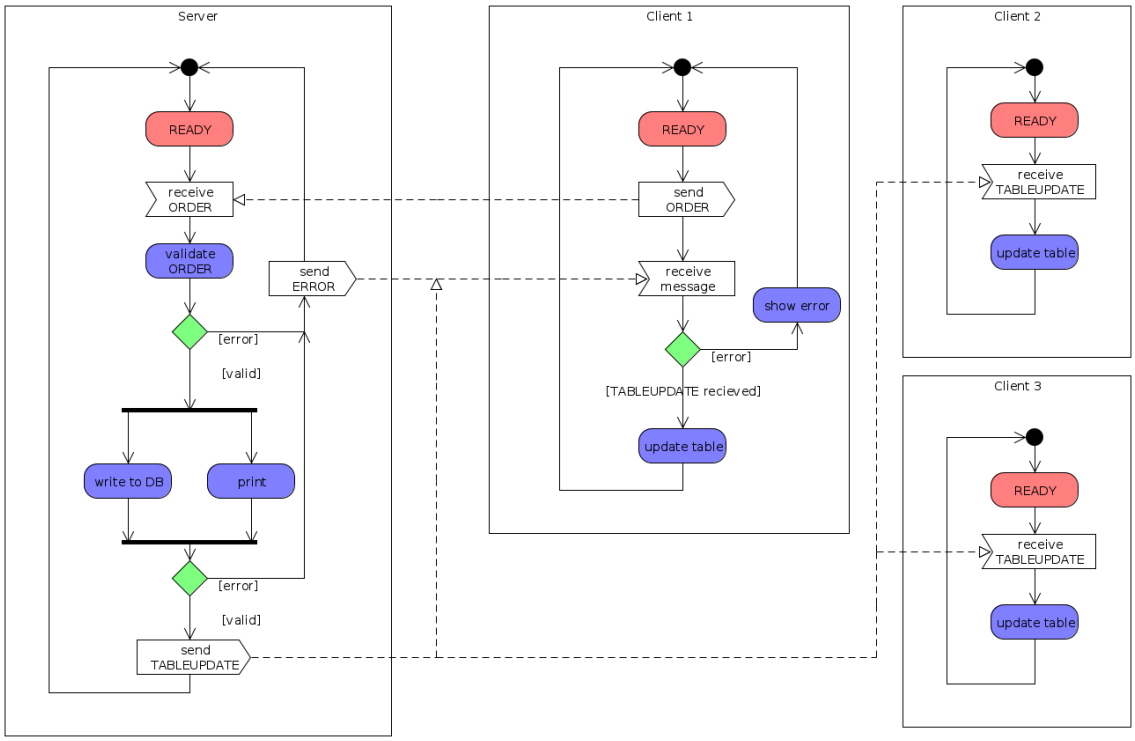


Illustration 17: The original protocol of FOS, which explains and defines the application life-cycle

The waiter is also able to counterorder, split, move or checkout a table (pay or pay separately) and finally logout. The protocol of these activities are stored on the FOS mediawiki web page, at *wiki.fos.fmdevelop.com/FOS*.

3.4 Technical Details and Information

FOS is an electronic order management system based on *Qt/C++* and *QML* functionality. It is available as Desktop version and for every Android device but it is looking forward to support also iOS (Apple) and many other platforms.

Development of FOS is led by the FOS developers team and owned by Florian Mahlknecht and Alex Mazzon. It is licensed under the GNU GPL licence.

3.4.1 Installation and Available Platforms

The application can be installed on a smartphone or any other device that is able to connect to the FOS server (see *3.2.The FOS Server*).

For now the application is available as desktop version and for every android device. Since FOS is developed in *Qt/C++* and *QML* (see chapter *4 Development*) it won't be a problem to make FOS available also for iOS (Apple) and many other platforms[8]. This innovative programming technologies allow to develop cross-platform[9] applications and user interfaces, which basically mean that you write the code only once and then you can export the application on many different platforms.

The Free Orderman System  *official FOS logo*	
Programming Languages	Qt/C++ and QML
Source Model	Free and open source software
Initial Release	v0.1 (3. November 2013)
Latest Release	v. 1 (26. May 2014)
Marketing Targets	Personal computers, Tablets, Smartphones
Available Languages	English (more coming soon)
Licence	GNU GPL
Official Website	Coming soon

Table 2: Technical Details

3.4.2 Development Concepts

To develop FOS we used various development concepts. Here you can find three of them[10].

KISS principle

KISS is an acronym for *"Keep it simple, stupid"* as a design principle noted by the U.S. Navy in 1960. The KISS principle states that most systems work best if they are kept simple rather than made complicated; therefore simplicity should be a key goal in design and unnecessary complexity should be avoided.

RAII

Resource Acquisition Is Initialization (RAII) is a programming idiom used in several object-oriented languages, most prominently C++, where it originated, but also D, Ada, and Vala. The technique was developed for exception-safe resource management in C++ during 1984–89, primarily by Bjarne Stroustrup and Andrew Koenig, the term itself was coined by Stroustrup.

Model-view

Model-View-Controller (MVC) is a design pattern originated from Smalltalk that is often used when building user interfaces.

MVC consists of three kinds of objects. The Model is the application object, the View is its screen presentation, and the Controller defines the way the user interface reacts to user

8 A platform is an underlying computer system on which applications can run.
9 In computing, cross-platform is an attribute conferred to computer software or computing methods and concepts that are implemented for multiple computer platforms (operating systems).
10 cf. [KISS], cf. [RAII] and cf. [MVC]

input. Before MVC, user interface designs tended to lump these objects together. MVC decouples them to increase flexibility and reuse.

If the view and the controller objects are combined, the result is the model/view architecture. This still separates the way that data is stored from the way that it is presented to the user, but provides a simpler framework based on the same principles. This separation makes it possible to display the same data in several different views, and to implement new types of views, without changing the underlying data structures.

3.4.3 Communication

All devices are connected to the same local area network[11], intranet[12] or the public internet. Any device that is able to connect to this network can be part of the system.

The FOS server generally runs on a computer and knows all the IP addresses of the clients that are logged in. The application provides the login and can run on any supported device (see *3.4 Technical Details and Informations, supported platforms*).

The delivery of streams happens over a TCP/IP-socket. TCP provides reliable, ordered and error-checked communication. FOS transmits data between server and client with JSON objects. This human-readable text consisting of attribute-value pairs is easy to use, because it provides libraries for *C++*. Objects are created from the client or the server, converted to a data stream through JSON libraries functions and sent over the TCP/IP socket.

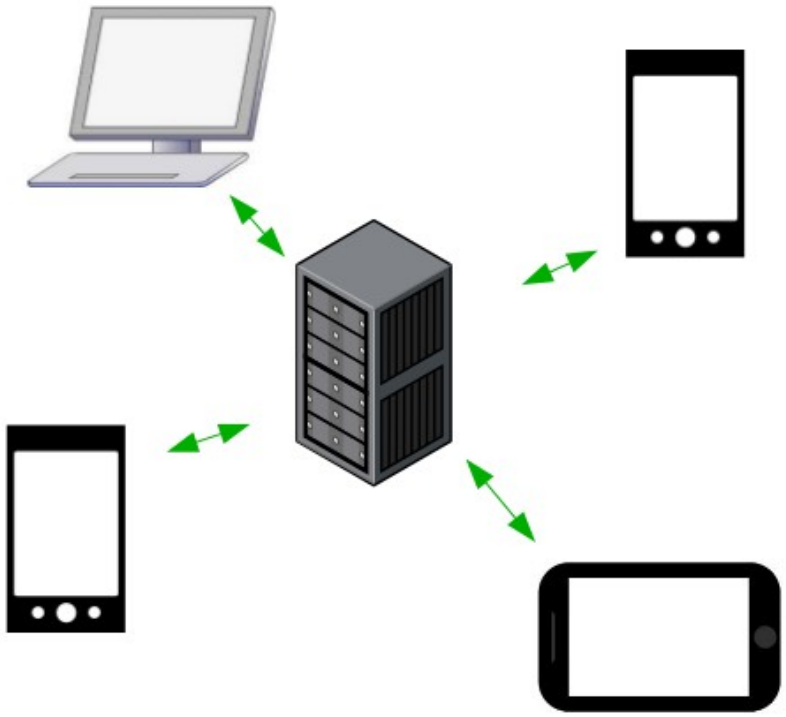


Illustration 18: The communication between the FOS server and the FOS Application (clients)

11 A local area network (LAN) is a computer network that interconnects computers within a limited area such as a home, school, computer laboratory, or office building using network media.

12 Intranet is composed of multiple local area networks

4 Development

Before we began to implement FOS there was a lot of design to do. Everything had to be over-thought a lot of times and there was always something that didn't work. But we consciously spent a lot of time designing FOS. The idea is great and it seemed simple but in the end it was a big challenge to create every single part of FOS.

4.1 Design – From the Idea to the Implementation

There were and still are a lot of problems to solve and questions to think about before coding. In particular we realize the project as a team where everyone has different ideas, wishes and views of how FOS should work.

We brought all ideas together and really had to rack our brains. The following chapters explain which tools, resources and utilities we used to facilitate the development of FOS.

4.1.1 The FOS Development Team

The FOS development team actually consists of the following members:

- Alex Mazzon – 19 years old, Salorno, Italy
- Florian Mahlknecht - 18 years old, Gummer, Italy

Since FOS is an open source project everyone can collaborate with us or develop their own software.

For further reading about the team have a look at the project in Launchpad at *www.launchpad.net/FOS*.

4.1.2 Ascertainment of Goods

First we tried to understand all features of an order management system and how to write down all requirements on FOS. This ideas were concretized by creating a requirements specification sheet, the database design, the protocol design and lots of diagrams (such as use case diagrams).

Every version of FOS defines its own use case diagrams and milestones. The FOS mediawiki page contains all these specifications, diagrams and protocols.

4.1.3 Tools to Share the Code and Develop Together – Launchpad & Bazaar

FOS is developed by a team and not by just one person. The code needs to be shared to all team members and that is exactly what the tools Launchpad and Bazaar allow to do.

Launchpad is a unique collaboration and hosting platform for software projects. It brings communities together — regardless of their choice of tools — by making it easy to share code, bug reports, translations and ideas across projects[13].

We choose Launchpad because it allows us to develop, promote and publish our software. Furthermore it encourages contributions to our project, improves collaboration with other developers and allows to build communities using teams and mailing lists.

Illustration 19: The logo of Launchpad

Bazaar is a version control system that helps tracking project history over time and to collaborate easily with others[14].

4.1.4 Implementation – Qt/C++ and QML

Illustration 20: The logos of Qt and QML: New technologies that allow to create cross-platform applications and aim to: Code less. Create more. Deploy everywhere

Qt allows to develop cross-platform applications and user interfaces. The whole functionality and the user interface of the server is implemented with this innovative technologies using the *Qt creator*. This allows to compile the program for many platforms.

The FOS application is developed with the *Qt* creator using *Qt/C++* and *QML* functionality. *QML* allows to create the user interface for all different platforms. The interface uses the *C++* models for data visualization. If the data is modified in the model, the view is automatically updated (see *3.4.1 Development Concepts, model view*).

4.1.5 Versioning

FOS uses versioning. Versioning is needed to assign the different releases of the software to different versions. Every new version implements new requirements and generally the software is advanced and improved with every new version. A new version includes new

13 cf. [Launchpad]
14 cf. [BZR]

functionalities, old functionalities that were improved or optimized. Possible bugs could be fixed.

Why Versioning?

Since we want to continue developing FOS and present it to the world (see chapter *6 Start Up!*), it is important that FOS be constantly improved and optimized. Versioning is strongly recommended for this.

When you make changes to the code, you could make mistakes and you versioning allows to revert back the code to a certain version. In addition to that by versioning you develop the code step by step (version by version) and in this way you never have too old backups or loose code. Also when you have to maintain multiple versions of a products versioning is useful. Furthermore you can:

- See differences between two (or more) versions of the code
- Prove that a particular change broke or fixed a piece of code
- Review the history of some code
- Submit a change to someone else's code
- Share your code and let other people work on your code
- See how much work has been done, where, when and by whom
- Experiment with a new feature without interfering with working code

4.2 Documentation

The Free Orderman System is well-documented as a project through the FOS wiki page. In addition to the wiki page the code is documented as well as possible to make every FOS development team member able to collaborate.

4.2.1 The Free Orderman System Mediawiki Page

The mediawiki page contains the whole documentation with all use cases and milestones of every version. It explains also how to collaborate with us and create new versions of FOS.

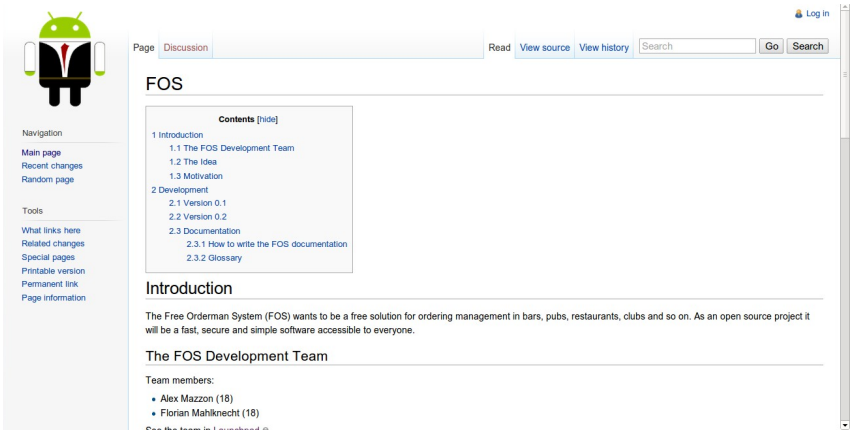


Illustration 21: The FOS mediawiki

5 It Works! - The Free Orderman System as Business Solution

After the release of FOS version 1.0 we tested our order management system in the bar and ice parlour *Salurn*. The test showed that FOS is potentially a good alternative to a professional order management system.

The Free Orderman System was tested over a working week of the catering company. We arranged the following survey:

- What was the first impression?
- How did the customers react to the new order management system?
- Do you think that FOS accelerated the catering process?
- Is FOS a success-related solution for your business?

The following test report reveals the results.

5.1 The Opinion of the Bar Owner

> *"Using FOS I was able to accelerate the catering process, especially on Sunday mornings or during parties, when it gets really stressful."* Enrico

During the whole development time the owner of the bar and ice cream parlour *Salorno*, Enrico Franceschini, was impatient to introduce the Free Orderman System. He never used an order management system in his business, because it was not profitable for his small company.

The first impression was great: Enrico is really enthusiastic to be the first one that tries out our new order system. He is happy to introduce something new and helpful without any cost. The most customers doesn't even noticed that the waiters were using a new order management system, those who did were interested in the project and wanted to know more about it.

5.2 The Opinion of the Waiter

> *"In the beginning I thought: The Free Orderman System? What is that? I often worked with the Orderman system, but I really never thought that it could run so well on my smartphone. I didn't needed to do nothing special, just install it*

and I liked it as soon as I ordered the first beer...”

The waiters of Enrico where also happy. Everything went well and the highly personalization of FOS allowed them to decide their own abbreviations of the products and manage the whole data alone. This made them really fast on recognizing their products and they got surprisingly fast on using FOS.

In the opinion of the waiters our project is absolutely marketable. They found it great to install the application on their own phone and it is the best free solution that they ever worked with. They found it cool because everyone has the possibility to enjoy it.

5.3 Our Personal Opinion

In our personal opinion the test went really well and we were extremely happy that it worked. We waited behind the bar, near the printer and smiled after every order that was printed successfully. It was much work to create FOS, but in the end it is great to see that the whole thing is working.

Anyway there is still tons of work to do before FOS is really ready for other companies. During the test we tried to analyse all problems and to consider the next steps. Read about these thoughts in the next chapter.

6 Start Up! - Where is The Free Orderman System Going from Here?

We want to publish and distribute the Free Orderman System. To realize this we need to make FOS available as a software package. The most important next step is to create an official FOS website where the whole package should be downloadable and simply installable. The application will be submitted to the Android Market so that everyone can download it directly to the Android device, without further effort. As soon as the application will be available for iOS also, it will be downloadable from the App Store.

Furthermore we want to support FOS users as well as possible. Those who are not able to install the system must get help and everyone should receive a reliable service. We want to buy appropriate and convenient printers so that we can sell them to the catering companies and provide a complete service.

We aim for a breakthrough and want to revolutionize the order management system world, especially for small companies.

What does the future hold? Only time will tell.

7 Glossary

8 Bibliography

OMS: Wikipedia Contributors, Order management system,
http://en.wikipedia.org/wiki/Order_management_system, April 17, 2014

QRCode: Piranha, What is a QR Code?, http://www.whatisaqrcode.co.uk, June 2, 2014

The Hospitality Study: Andreas Neuhofer, Orderman, Trends, Success Factors,
Technologies, http://www.orderman.com/en/hospitalitystudy.html, April 20, 2014

DB: Mike Chapple, What is a Database?,
http://databases.about.com/od/specificproducts/a/whatisadatabase.htm, June 2, 2014

PrimKey: Mike Chapple, Choosing a Primary Key,
http://databases.about.com/od/specificproducts/a/primarykey.htm, June 2, 2014

KISS: Wikipedia Contributors, KISS principle,
http://en.wikipedia.org/wiki/KISS_principle, May 22, 2014

RAII: Wikipedia Contributors, Resource Acquisition Is Initialization,
http://en.wikipedia.org/wiki/Resource_Acquisition_Is_Initialization, May 22, 2014

MVC: Trolltech, Model View Controller,
http://doc.qt.digia.com/4.2/model-view-programming.html, May 22, 2014

LAN: Wikipedia Contributors, Local Area Network,
http://en.wikipedia.org/wiki/Local_area_network, May 22, 2014

Launchpad: Canonical Ltd., Publish your work. Collaborate with free software
communities., https://launchpad.net/+tour/index, June 2, 2014

BZR: Canonical Ltd., Bazaar, http://bazaar.canonical.com/en/, June 2, 2014

9 List of Illustrations

10 List of Tables