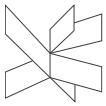

Project description

Arturs Silins - 315226
Bhupas Gautam - 306411
Maximillian Marius Wallin - 315268
Ondrej Klimek - 315255
Siddhartha Jonathan Grasse - 315278

Supervisors:
Ib Havn
Jørn Martin Hajek

Software Technology Engineering
2nd Semester

Horsens, 2022

Table of Contents

List of figures and tables

Background description

Our client, known as Bob, is the head of a Danish restaurant chain. The client currently uses FlexyBox[1]. (FlexyBox, 2022) This software targets businesses in general. This can sometimes be an issue as restaurants have different needs than, for example, grocery stores or gyms. This system is tough to navigate through - it is too complex and not very user-friendly. User-friendliness can be improved, among others, by consistency. There also have been system instabilities. The only thing that the client likes about the current system is that the employees can log in on both laptop and phone.

The hostesses must book tables manually which can be time consuming, and the customers don't get to see where they will sit unless they book their tables in person. Additionally, waiters use it to order food. Some of the problems reported by waiters are chaotic menu item list for taking orders, or troubles with sending requests for the next dish. This often heavily impacts the kitchen staff and may cause chaos.

But this System is inefficient and time-consuming. A wide variety of ordering websites allow users to log in, book food, and submit special requests while collecting points that already exist but without the possibility to book specific tables. Also, a wide variety of restaurant-specific websites allow users to book tables at times already exist, but without the option of ordering food.

Bob is looking for a program that combines the booking and the ordering and digitalizes them to require less manual work. He wants a product that enables guests to book tables and pre-order food themselves while also having the option of receiving reward points, which they can use to get discounts on the following order. Guests should be able to submit special requests, for example, specific diets or allergies.

Additionally, Bob wants to have a comment section where guests can express their wishes, for instance, if the occasion is a birthday or an anniversary. Bob also wants to have the option of granting special discounts to guests in certain conditions. Guests should be able to decide if they're going to dine in or order their food for takeaway. If they choose to dine in, they can select a preferred seating position, for example, one by a window, in a quiet area, or outside. Also, they can choose a specific number of chairs and have baby chairs added if needed.

In the end, guests should be provided with information on payment methods. Once their visit has finished or food has been ordered, guests should have the chance to leave feedback on their dining experience. Bob would like the system language to be in English by default and have the option of Danish as the second language to accommodate a wide variety of guests. If multiple languages are not possible in the System, the client is satisfied with the System being in English.

Definition of Purpose

The purpose is to provide Bob with more flexibility to create an optimal dining experience for his guests by implementing a more efficient booking/ordering system that allows him to be less dependent on employees.

Problem Statement

Currently, various employees require a lot of manual work to book a table and order food for the guests, especially when special requests and other wishes have to be considered.

1. Which features will our program have, that FlexyBox does not?
2. Who will be able to use the System?
3. How will our System differ from previous solutions?
4. How many user roles will our System need?
5. What kind of input from the guests will need to be considered?
6. What

Delimitation

1. We are not able to implement a secure log-in.
2. The System will not be available as a mobile app or a website.
3. The System will only be available in English.

Choice of models and methods

To make our project development as efficient as possible we will be using Scrum with regards to Unified Process.

Scrum is a framework that helps teams and organizations address complex problems step by step, while delivering products. It is used by teams, which consist of a Product Owner, Scrum Master and Developers[2]. (Scrum.org, What is Scrum?, 2022) The Product Owner is accountable for effective Product Backlog management and working together with the customer and development team to deliver the product. His main responsibility is the Product backlog which must be clear, understandable and explain the Products Goal. The Product Owner is the only person that can represent stakeholders need in the Product Backlog.[3] (Scrum.org, What is a Product Owner?, 2022) The Scrum Master is accountable for keeping track of sprints making sure the team works on the goals which are stated in the sprint backlog. The Scrum Master has to help the team with understanding Scrum and with keeping things on track so that the product is finished quickly and efficiently. He also has to help the Product Owner with the Product backlog and define the Product Goals.[4] (Scrum.org, What is a Scrum Master?, 2022)

Unified Process is based on enlargement and refinement of a system having multiple iterations, with cyclic feedback. There are four phases in the iterations and consists of one or more iterations.

Our Scrum Master will be Arturs Silins because he is the most structured and organized person of our group. Ondrej Klimek came up with the project idea, so he is going to be the project owner. Our Sprint length will be 4 days. Definition of Done: All the requirements are met; we have tested the product and it works as intended and when the Documentation is finished.

For keeping track of documentation, design and sprints, we will be using Word, Excel and Astah. For implementation we will be using DataGrip and IntelliJ. The code will be stored at a remote repository on GitHub so that the Developer team can get access to it.

Time schedule

We have an expected workload of 28 hours per ECTS point per student, which adds up to 280 hours per student, 1400 hours in total. To keep an efficient work environment, we will be adding intermittent breaks using an app called “Pomodoro” to keep efficient work and break times. The completed project will be handed in at the latest on 31/05/22 at 13:00.

February 2022			March 2022			April 2022			May 2022		
T 1	Winter break		T 1			F 1			S 1	Første maj	
W 2			W 2	Project Description work		S 2			M 2		18
T 3			T 3			S 3			T 3		
F 4			F 4			M 4		14	W 4	Project Elaboration	
S 5			S 5			T 5			T 5		
S 6			S 6			W 6	Product Backlog feedback		F 6		
M 7	Tuition start	6	M 7		10	T 7	Demonstrate working architecture		S 7		
T 8			T 8			F 8			S 8		
W 9	Group getting to know eachother		W 9	Project Description work		S 9			M 9	Last day of tuition	19
T 10			T 10			S 10	Palm Sunday		T 10	Project Construction start	
F 11			F 11			M 11	No lessons scheduled	15	W 11		
S 12			S 12			T 12			T 12		
S 13			S 13			W 13			F 13	Common Prayer Day	
M 14		7	M 14		11	T 14	Maundy Thursday		S 14		
T 15			T 15			F 15	Good Friday		S 15		
W 16	Groups, Project Inception start		W 16	Project Description feedback		S 16			M 16	Project gives meaning to custc	20
T 17			T 17			S 17	Easter Sunday		T 17		
F 18			F 18			M 18	Easter Monday	16	W 18		
S 19			S 19			T 19			T 19		
S 20			S 20			W 20	Project Elaboration start		F 20		
M 21		8	M 21		12	T 21			S 21		
T 22			T 22			F 22			S 22		
W 23	Project proposal Feedback		W 23	learning Scrum, work on design		S 23			M 23	minimal viable product finishec	21
T 24			T 24			S 24			T 24	Project Transition start	
F 25			F 25			M 25		17	W 25		
S 26			S 26			T 26			T 26	Ascension Day	
S 27			S 27			W 27	Project Elaboration		F 27	Finished construction phase	
M 28		9	M 28		13	T 28			S 28		
			T 29			F 29			S 29		
			W 30	Product Backlog, Scrum roles, Sprint		S 30			M 30		22
			T 31						T 31	Hand in Project	

Risk assessment

Risks	Likelihood Scale: 1-5 5 = high risk	Severity Scale: 1-5 5 = high risk	Product of likelihood and severity	Risk mitigation e.g. Preventive-& Responsive actions	Identifiers	Responsible
New/updated technologies that the client wants us to use, might get released	3	4	12	Make sure that the team is up to date with the newest technologies	Client will contact us	All members
The restaurant shuts down	2	5	10	Look for a new client that needs the same services	Client will contact us	Ondrej Klimek
The client wants more features or changes	4	1	4	The price will increase, and the deadline will be extended	The client talks about changes	Arturs Silins
Member drops out	1	5	5	Good Communications between members removal of unprioritized goals	The person might seem unresponsive	Maximillian Wallin

Table 1 - Risk assessment

Resources

[1]https://flexybox.com/
[2]https://www.scrum.org/resources/what-is-scrum
[3] https://www.scrum.org/resources/what-is-a-product-owner
[4] https://www.scrum.org/resources/what-is-a-scrum-master