

Computer Science Program

2th Semester 2015

Semester Project

Implementation of IT system to “Kad Zebras Deg” Bar

The report has been prepared by :

Ionut Danci-Bumbea

Janis Indrans

Leonard Hirja

Nikita Cimaskevic

Richard Retezi

Supervisor:

Henrik Munk Hvarregaard

Date of submission:

02-06-2015



Real Life Education

Table of Content

Computer Science Program	1
Table of Content	2
1. Introduction	5
Preliminary investigation	5
Inception phase	5
Elaboration phase	6
Construction phase	6
Project planning	6
2. Preliminary investigation	6
2.1 Company Description	7
3 Inception phase	7
3.1 Business analysis and feasibility study	7
Style of management	7
Motivation	9
Culture	10
Problem Statement	11
Problem description	11
SWOT analysis	12
Porters Five Forces	13
Stakeholder Analysis	15
Mission / vision	16
Business case for the system	16
Payback projection	17
Business case conclusion	18
Activity diagram on business level	19
Event tables	19
3.2 Functional Requirements	20
Use case diagram	20
Brief use cases	22
Mock-ups	22
4 Elaboration phase	23

4.1 SCRUM	23
4.2 Use case prioritization	24
4.3 First sprint	25
Fully dressed use case	25
Candidate class	27
Domain model	27
System sequence diagram	31
Operation contracts	32
Interaction diagram	33
Design class diagram	34
Code	35
Test	38
5 Construction phase	40
5.1 Second sprint	40
Data base architecture	40
Relational model	40
Grasp and GoF patterns	41
6 Group Evaluation	42
7 Conclusion	42
8. References	43
9. Appendices	43
9.1 Event Tables	43
As is Table	43
To be table	44
9.2 Fully Dressed use Cases	45
9.3 Interaction diagrams	48
9.4 Operation Contracts	48
9.5 Product backlog	50
9.6 System sequence diagram	51
9.7 Use case diagram	51

9.8 Brief Use Case	52
9.9 Design class diagram	57
9.10 Relational model	59

1. Introduction

“Most good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program.” -Linus Torvalds

Same was in our situation for the second semester's final project. Our task was to develop software for a company in real life situation. As it is not always easy to make an agreement with companies to allow students to use their information for study purposes, because they don't want to expose some important information, we tried different approach by searching for a company within circle of people that our group members is friends with or people we know. This approach was successful and we found friend of Janis that founded his own bar / restaurant “Kad Zebras Deg” (English - When zebras burn) in 2014 and he agreed to provide us with necessary information. After arranging meeting with owner in Skype, the plan for the work process where made. In order to do a good and organized job on project, the plan was divided in 4 phases – preliminary investigation, inception, elaboration and construction. For the work process we chose to follow iterative UP (Unified Process) combined with practice of Scrum.

The Unified process divides the project into 4 parts:

- Preliminary investigation
- Inception phase
- Elaboration phase
- Construction phase

Preliminary investigation

Preliminary investigation contains interview with owner [Appendix 1 - interview], company description.

Inception phase

Inception phase is the shortest one part where it should be set out problem description, business analysis, use cases and domain model.

Elaboration phase

Elaboration contains use case diagram, mock-up, brief use cases, fully dressed use cases and domain model.

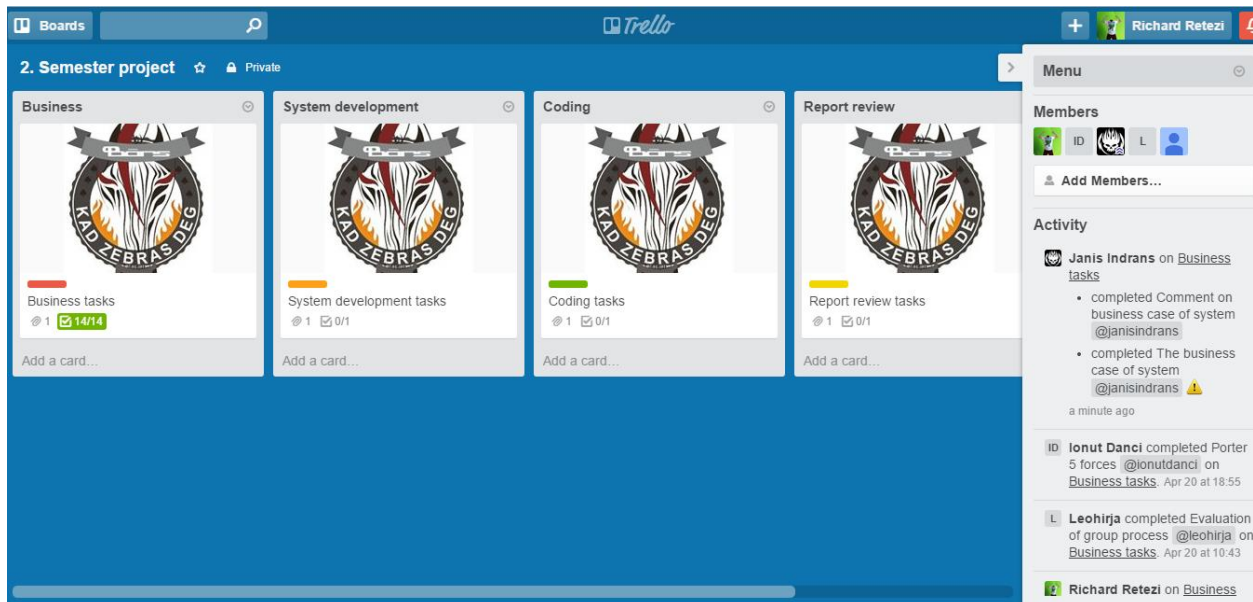
Construction phase

Construction is the largest phase in the project which contains a series of iteration. The result should be an executable code and new features are added during each iteration.

Project planning

To plan our project we have been using Trello as well as assign task to each of our group member. If there were some unassigned tasks the members could choose to take them. If some could finish his work earlier another group member was able to take his job or help him in case of lack of knowledge.

Figure 1: “Trello screen screenShort”



2. Preliminary investigation

The Preliminary investigation phase starts with company description and company owner interview. That information will contribute thorough analysis of the company and based on the analysis, appropriate system requirements will be made.

2.1 Company Description

“Kad Zebras Deg” is a small bar/restaurant which is situated in Riga, the capital of Latvia. It’s been founded in 2014 by Renars Bents. He is the only owner of the company and he has 6 employees: One cashier, two waiters / bartenders, one chef and two kitchen assistants. Renars is strongly involved in his bar/restaurant and is helping as much as he can to his employees. He is very friendly and fair to his employees as he doesn’t fire anyone without a serious reason. Bar / restaurant is quite new so it doesn’t have any IT system. He can’t afford to have a lot of employees so he is in need of IT system to make his bar/restaurant more organized and efficient. At the moment everything is done by writing down on paper and using the computer to record the income and outcome with the help of MS Word or MS Excel. Every weekend Renars is organizing all kinds of events like music bands, stand-up comedians, game nights on Xbox which has prizes at the end, in order to attract more customers. The usual customers are mostly his friends and young people from 16 – 30 years old as it was designed to follow up with current trends.

3 Inception phase

Inception phase is the initial step to establish a common vision and basic scope for the project. Inceptions phase consists business analysis and feasibility study which facilitates setting out functional requirements. That will promote use case prioritization creation.

3.1 Business analysis and feasibility study

Style of management

Renars Bents looks like a fair and open manager / owner according to how he was describing his bar / restaurant and the way he manages his staff. After exploring the information that was given on the interview and experience of Janis who is friend with Renars, conclusions were made. Renars follows “consults” leadership style which means that manager both identifies the problem and makes the decision, but only after listening and possibly adopting solutions that have been suggested [Tannenbaum and Schmidt]. Leadership style was identified based on the interview and the facts that Janis provided. Janis has been friends with Renars for a long time and was on the holiday in Latvia when Renars was just planning to open his own bar / restaurant. Owner was

asking everyone for ideas that he could adopt when opening his bar / restaurant and slowly adopting them. It shows that he first listened to suggestions and possibly adopted them. When there is a problem in his business he tries to ask all employees how they should deal with it and tries not to make blind decisions that could harm his business. As it is a new business everyone is involved in its growth. This means that the staff members are willing to help and try to improve the efficiency. His style of characteristics is “Achiever” [D.Fisher and W.R.Torbert]. This conclusion was made based on that he has long-term goals and he feels like initiator. He was planning to open his bar / restaurant for a long time. Only when he was completely confident that he will be able to found his business, he did it. When he was describing his business he mentioned that he opened his bar / restaurant not because to try to earn a lot of money from the very beginning but because he really loves what he is doing and that he will try to provide customers the best services as the business grows. That shows that he has long term goals. As it was described before he listens to his staff and people around him which indicate that he welcomes feedback which also means that he seeks for mutuality. Sometimes he listens too much to other people and facts that he forgets to think over everything himself and make his own decision which indicates that he appreciates complexity but is blind to subjectivity behind objectivity. He usually feels guilty if something didn't go the way it was planned like it happened when someone suggested to make his bar / restaurants theme as first “velo bar / restaurant” in Latvia where you can come to fix your bike using the tools in his premises outside the bar / restaurant. . It failed because it cost more than he could afford and for a while he could not come up with any new ideas and let the employees make decisions in everyday tasks. This shows that he feels guilt if falls short of his own standards.

From this analysis we can assume that the most valuable information is that his style of management includes long term goals, which means that when developed system will be implemented he will be patient until employees fully adopt to it. That's when efficiency of his business will rapidly increase. As he takes employees feedback in consideration we can assume that they will respect his decision about implementing system as he knows that employees are getting more tired as the bar / restaurant is getting more popular and the amount of customers grows, which means that the employees work increases and they need to find a way to be more efficient and find a way to make the work easier.

Motivation

To motivate staff in fairly small companies like bars and restaurants usually it is not that easy as the growth in that kind of sphere of business is quite small. At this point where the company is still young the promotions are almost impossible. This is the reason why owner / manager are trying different approaches to motivate his staff. Based on Herzberg's two-factor theory Renars provides some of the extrinsic and intrinsic rewards [Herzberg 1968]. By exploring the information provided, following rewards were identified:

Extrinsic

- **Supervision/relationship with supervisor**
- **Working conditions**
- **Remuneration: pay, salary**
- **Relationship with peers and with subordinates**
- **Job security**

Intrinsic

- **Recognition**
- **The work itself**
- **Responsibility**
- **Personal growth**

Supervision/relationship with supervisor was identified by the fact that Renars has a friendly relationship with every staff member and he treats everyone equally. He doesn't have any favorites and he respects his staff. That's also mutual.

Considering that it is newly established business, the owner is trying to provide the staff good working conditions. He provides staff with all the tools they require and tries to make their jobs easier. That helps to keep up a good relationship with staff and good service for customers. This is also a reason for why he would like to have a computer system in his bar / restaurant. Development of this system will clearly benefit of this factor and most likely succeed.

The salaries are quite good as the company tries to pay fair amount and it is the average salary in Latvia. Most people work for the minimum wage in the same sphere. Staff also knows that they will receive tips from customers and it motivates them to provide good service.

Relationship between staff members is really good, not only because the environment is satisfying, but also because the owner organizes parties for company, time to time. This allows staff members to be closer and also to get know each other better. In the result it builds a good teamwork and raises the team spirit.

Staff members feel secure in terms of work. Renars is always around at bar / restaurant and will help if needed. He is a reasonable man and will understand if there were any unexpected factors that led to underperformance. If there appear dissatisfied customer he will listen to both sides of the parties and come up with fair resolution.

Employees know that if they will perform good, owner / manager will notice that and will always give a good feedback. If they have a good ideas he will also consider them and might even take in consideration. This covers recognition award.

The work itself is being made as much satisfying as Renars can make, that includes also plans for the future computer system. Employees know what they have signed up for and keeping in mind that there is big unemployment in Latvia, they are quite happy to work there.

Responsibility reward is seen in fact that Renars trusts his employees considering that they work in food business where customers must get the best service and avoid any kind of food poisonings.

In bar / restaurant works mostly young people which means that owner / manager gives unexperienced staff members a chance for personal growth and might increase efficiency in company as the staff will gain experience. Young staff also might show better performance in busy days by having more energy.

Culture

Company's culture plays a big role in its business. Establishing what kind of culture company is practicing, it gives more knowledge of the possible reactions according to changes. For this

project it was decided to use Kanter's two types of culture to analyze behavior that will affect the business when implementing the new system [Kanter 1983]. He argued that there are, broadly, two types of culture: segmentalist and integrative. While analyzing information provided conclusion where made, that the culture of this project's company is integrative.

Integrative culture

- Sees problems as related
- Views problems and responsibilities as shared and connected
- Has matrix or team/project-based structure
- Innovates and test assumptions: invites experimentation
- Invites confrontation and eventually transcends differences
- Creates mechanisms of coordination for sharing information and ideas
- Looks for novel solutions
- Is outward-looking

This project's company is quite small and new so it has more like a team-based structure. All the problems as described earlier are solved together and they are seen as related. Company is still experimenting with approaches in services provided, time to time there might appear some minor conflicts which are solved with compromising approaches. Information and ideas are sheared and discussed in meetings, which are noted and considered for implementation. Also as described earlier ideas might come from the outside of the company. With all this analysis it was established that there shouldn't be any big resistance against the new system and the staff will share with ideas in its development.

Problem Statement

How should we design a software for a newly established bar without having previous experiences from the customer's side?

Problem description

The problem description gives us a better insight, why the project needs to be made, what is it for, what problems are needed to be solved and can be solved.

The main problem with the company “Kad Zebras Deg” Bar that it is newly founded and does not have an IT system yet. The processing of orders, bookings, the event handling and all service related processes are manually done (per say, written on a piece of paper).

A bar in 2015 has to have an IT system to show quality and prestige in front of the customers. With an IT system it is easier for the customers to reserve a table, it is easier for the employees to register and process orders and to keep track of them. Because there was no previous system existing, there is no previous experience, no software to update, therefore a brand new system needs to be implemented.

If the bar wants be competitive with the rest of the bars in the city it needs to develop and improve continuously in order to meet with modern day standards.

By completing the project and creating the IT system, the bar should have an easy system to use, facilitating the everyday work of employees and let the customers have better accessibility and engagement for the bar (easy and fast way of reserving a table or buying tickets for concert events for instance).

SWOT analysis

SWOT analysis is used to analyze external and internal factors which affects achieving the goal that in turn should be specified for the business.

“Kad zebras deg” bar / restaurant

Table 1: “SWOT Analysis of “Kad zebras deg” bar/restaurant”

Strengths	Weaknesses
<ul style="list-style-type: none">✓ Popular among young adults✓ New and fresh✓ Unique design	<ul style="list-style-type: none">✓ Unexperienced management✓ The lack of advertising✓ The lack of “brand loyalty”✓ Limited budget✓ The lack of an IT system

Opportunities	Threats
<ul style="list-style-type: none"> ✓ Expansion ✓ Hiring more experienced staff ✓ Demand for organizing bigger and better events 	<ul style="list-style-type: none"> ✓ Competition in the city ✓ Economic situation (Crisis) ✓ The threat losing in a price competition

Business goal: Provide high quality bar service for the middle class society.

“Kad zebras deg” bar / restaurant has been a very popular pub among young people ever since its opening. This opinion was formed mainly because of the new design and the fresh look. A new and improved selection of drinks could also be considered an advantage.

Even though the pub is a popular place the lack of experience for the manager is a great disadvantage. Because of a small budget advertising is not an option. What makes it even hard to manage is the lack of an IT system which is needed in order to ensure a nice flow for transactions and very easy management.

Depending on the success and the income of the company the owner could decide to expend the scale of business and to enter new markets. This is a very important step in this case too as the pub needs more clients and more employees. In most of the cases the more clients we have the more profit we can have. If the clients ask for it then the pub could organize bigger and better events.

The biggest threats are losing in a price competition against another pub which can afford to offer better offers or which has a bigger capital, an economic crisis and opening of a huge number of pub in the same area.

SWOT analysis was made based on studies of [Bloisi].

Porters Five Forces

This section is used to analyze all five Porter’s forces to explain the connection between the new entrants and how their represent a threat for the client’s business. This section also explains customer’s purchase power and what a substitute product is.

Threat of new entrants

- Easy to purchase a propriety and get started with a similar pub in a very short time.
- New clubs, bars or restaurants can also be open nearby and it can affect the way the business is evolving.
- Very difficult to predict or anticipate new entrants on the market.
- It is important to create a brand and to maintain.
- It is easy to gain permission from the authorities.

Determinants of supplier power

- Multiple suppliers can be used to provide the same articles. Low influence
- High number of alternatives
- The liberty of ordering online or directly
- Working with global known brands

Rivalry among existing firms

- Medium rivals in the nearby area
- Bigger and better pubs or highly known competitors can open anytime
- Marketing partners and using customer to promote
- Innovation is very important because of better and newer products can attract more clients

Determinants of buyer power

- Vast selection of drinks and foods available
- Buyers don't have bargaining power since prices are set
- Buyers determine the theme for parties/events

Threat of substitute products

- New and better events can be regarded as substitutes
- Difficult to stay in the top of all similar/substitute products

The threat of new entrants is a very serious one. At any given moment a new pub or bar could open in the neighborhood since it's very easy to purchase a property and transform it into a bar. Most of the client could take advantage of this. Since it's very hard to predict new entrants it is better to focus on creating brand and a very high status since the very beginning.

The supplier does not have that much power since the products they deliver can be offered from more than one supplier. This way the manager can choose the supplier according to delivery and product quality, price and so on. A good advantage is having a brand and a very good reputation. The order can be done online or directly.

Innovation is very important regardless of area. The business needs to keep improving in order to keep the customers and acquiring new ones. Every small new entrant can be a threat. Existing rivals could also improve and the competition can become bigger if no measures are taken. The use of marketing partners and promoting using t-shirt or cups can help a lot.

The customers have no influence regarding the price of the place. They are very influential when it comes to parties and events themes or singers. They can also influence the products in the menu.

Some substitute products can be specific food restaurants. They can appear any time and there is nothing to do to prevent their launch. Also, it's very difficult to predict their appearance. The only solution is to race with them and stay in top when talking about customer satisfaction and number of services offered.

Porter five forces analysis was made based on studies of [Lauritzen and Krogager].

Stakeholder Analysis

A stakeholder analysis gives an overview about the interest groups in connection with the project. The two main groups are those who are interested in the success of the project and those who are interested in the failure of the project.

Those who are interested in the success of the project are:

- The owner of the bar: The owner wants to have a successful business.
- The employees of the bar: They want to have a solid workplace.

- The potential customers: They want to have a successful bar with good quality service.
- The suppliers: The more bar they supply to, the more order and income they can receive.

Those who have disinterest in the success of the project are:

- The other bars in the city (Competitors): There is a big competition in the city and they don't want new bars carrying away the customers.
- Possible old people living in the area (negative external effect): They don't want to have a noisy place at night and drunken people disturbing the area.

Mission / vision

The mission vision of the company defines its primary sets of goals, their main objective for the long-term. The vision of the company is how the company wants to see themselves, how they want the customers to think and feel about the company Mission

Mission

The mission of “Kad zebras dag” is to ensure the high quality, delicious, remarkable food as well as professional and friendly service. Assure that clients are provided the fair price. Always work according to customer's expectations and believes.

Vision

We believe that friendly and wonderful service affects person's emotional and physical state. Our vision is to make people happy and deeply satisfied by giving them unforgettable taste and excellent service.

Mission / vision were made based on studies of [Bloisi].

Business case for the system

Almost every company and organization in 21st century has its own system no matter if it's open source or custom made to meet all the necessary requirements in order to profit financially and to gain efficiency on daily operations. System that is proposed to “Kad zebras deg” will succeed in

many ways. As it doesn't have any system at this particular time it will be custom made and will cover all of the required operations. This company is not rich and can't afford a lot of stuff so this proposal is vital to increase its efficiency by letting students to build it.

The system will be built as part of student project and will cost nothing except that the company will have to buy its own computers and acquire licenses for the necessary software. In this case it will require as a minimum of four computers in order to meet the minimum requirements for the system. One computer in the kitchen, one for the waiters, one for the cashier and one in the office. It might be expensive at first but in the long run it will profit much more than now. Considering that the company is still young and there is a lot of competition going on in this type of business, this system will help the company to be more organized and efficient as most of its competitors don't have any system. Resistance from staff is not expected as they are tired of doing everything in the old way of registering orders and booking the tables. This will save them a lot of time, they will avoid misunderstandings and dissatisfied customers because of the long waiting time before receiving their orders.

The kitchen and the waiters will be up to date with courses and their serving order. Table bookings will be organized manually by the staff or the system if it's a standard request, plus customers will be able to preorder courses that the chef will begin to cook as soon as they arrive to the restaurant and check in. In summary this system will gain satisfied customers and staff members to the company and will benefit in many ways. In the long run it might lead the company to grow and expand as the efficiency will rise.

Payback projection

Table below describes payback projection [Cadle and Yeates]. Research shows that the company will profit after the first year. Calculation of prices showed that company does not require high end PCs to operate with software proposed and as a minimum it requires 4 PCs and 1 external hard drive which will allow to back up data in case of accidents. Total amount for hardware is 2540 Euros. Hardware maintenance will cost 1000 Euros a year as there will be only 4 PCs. Only software they need to acquire is antivirus (we chose Avast) to protect PCs from viruses and malwares that could harm data on PCs. Avast license cost 150 Euros a year for all computers and requires renewal every year. Another software that will be required is MS SQL Express for database which is free. Staff savings per year will be at least 12000 Euros which means two extra

employees. As described before Bar / Restaurant can't afford more employees so our software can save some money and increase efficiency. By paying extra 4690 Euros first year the efficiency will increase as there would be 2 more employees and save on those salaries 7310 Euros. Next year and in the future the profit will increase even more as the Bar / Restaurant won't need to buy hardware anymore. Only costs that will be required are hardware maintenance, software support and antivirus license renewal which sums up to 2150 Euros a year. The prices and salaries were examined based on the average wages and prices in Latvia [Trading economics], [X Net].

Table 2: "Payback Projection"

Item	Year 1	Year 2	Year 3	Year 4	Year 5
Hardware purchase	2540				
Hardware maintenance	1000	1000	1000	1000	1000
Software purchase	150				
Software support	1000	1150	1150	1150	1150
Cumulative total costs	4690	6840	8990	11140	13290
Staff savings per year	12000	12000	12000	12000	12000
Cumulative savings	12000	24000	36000	48000	60000
Cumulative savings less	<u>7310</u>	<u>17160</u>	<u>27010</u>	<u>36860</u>	<u>46710</u>

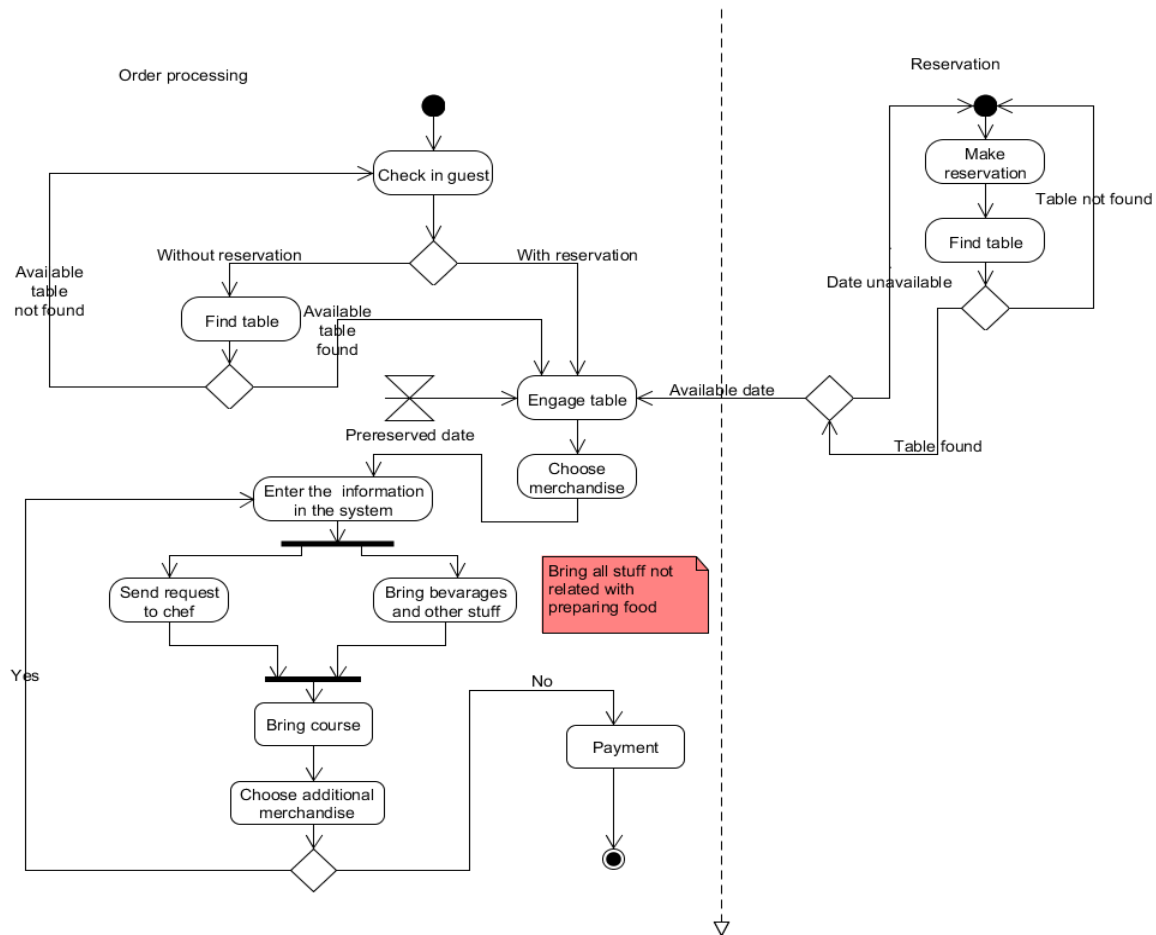
Business case conclusion

According to our cost-benefit analysis and company analysis, the owner will already have profit and savings in the first year even after investing in the software. The following years the bar / restaurant can increase its profit and have more savings. This suggests that the benefit of the software will overcome the costs and the project is worth doing.

Activity diagram on business level

Activity diagram is visual representation of processes occurring in the company.

Figure 2: “Activity Diagram”



Based on this workflow the use case diagram was generated as well as gained an understanding of business processes of the company.

Event tables

The goal of the event tables is to define current business processes in the organization and clarify how exactly it works in order to determine possible changes after IT system would be implemented.

Table 3: “Event (AS IS)”

Event (AS IS)	Activity	Step in activity	Actor
Order is being made	Make order	*Customer sits at the table and chooses courses *Employee / Manager writes down order in the notepad *Employee / Manager goes to the kitchen and gives the order sheet to chef	Employee / Manager
Table is being reserved	Make reservation	*Customer calls Bar / Restaurant and asks for the table *Employee manager checks the reservation list and decides which table can he assign *Employee / Manager writes down the new reservation in	Employee / Manager

Table 4:”Event (TO BE)”

Event (TO BE)	Use Case	Step in use case	Actor
Customer wants to make an order	Make order	*Customer sits at the table and chooses courses *Employee/Manager writes down information in the notepad *Employee/Manager inputs customers desired course or product in the system *Employee/Manager repeats the action until all desired courses	Employee / Manager
Customer wants to reserve the	Make reservation	*Customer calls and specifies date and time for reservation *Employee / manager enters information in the system *Employees / manager save the information on the system	Employee / Manager

Full event table can be found in the appendix.

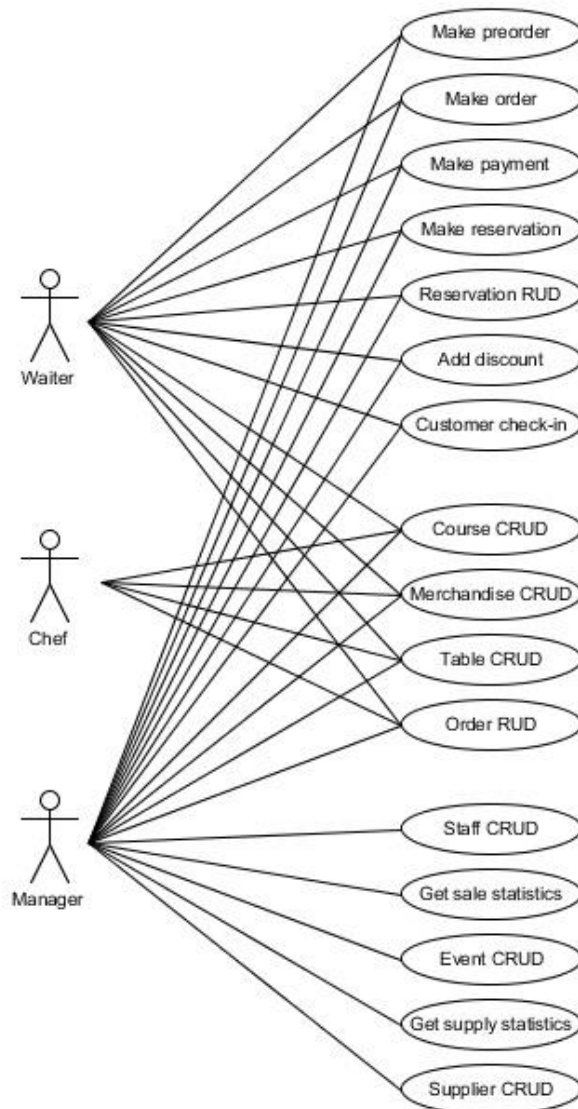
3.2 Functional Requirements

Use case diagram

Use case diagram is the most important component in order to design a good program. There were no time spared in making the best out of the information provided. Based on the event tables and workflow

of the company the use cases were established. There are three actors for this system: manager, waiter and chef. The visibility of the use cases where established as the manager will have access to every use case, waiter will not be able to access, Staff CRUD, Get sales statistics, Event CRUD, Get supply statistics, Supplier CRUD and chef will have access only to few use cases – Course CRUD, Merchandise CRUD, Table CRUD, Order RUD. Corse CRUD, Merchandise CRUD and Table CRUD use cases are accessible to everyone to let company’s staff work together when new merchandise, tables or courses are brought in the bar / restaurant and needs to be registered. This way it will save a lot of time.

Figure 3: “Use Case Diagram”



As long as the use cases were completed there were no doubts about the requirements for the new system and the designing of use cases could begin. To describe shortly every use case in diagram we have used brief use cases.

Brief use cases

The purpose of brief use case description is shortly outline and define what system does in response of actors actions.

Table 5: “Brief Use Case”

Use Case :	Reservation CRUD
	<p>Employee and manager(actors) uses the system and able to do :</p> <ul style="list-style-type: none">• Create reservation• Find reservation• Update reservation• Delete reservation <p>To create the reservation actors register it in the system and save it. System creates and saves the course.</p> <p>To find reservation actors put information in the system. The system finds reservation.</p> <p>To update reservation actors put information. System finds course and actors update it.</p> <p>To delete reservation system finds it and actors delete it.</p>

It helps to prepare use case prioritization in order to start elaboration phase. The rest part of the brief use cases description can be found in appendix.

Mock-ups

Mock-ups is model that shows how program would look like as well as tool which is used to retrieve user feedback during the test. Mock-ups help user understand thoroughly what kind of functions program will provide as well as figure out if we understood everything right, but as long as program will be developed by defining domain model, SDD and etc. functionality might be changed.

Figure 4: “Mock-up”

Waiters menu

Choose action

No of seats: 7

System suggestion

Year: 2015

Month: May

Day: 14

Name: John Smith

Make reservation

Search

Back

Tables

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36
37	38	39	40	41	42

In the mock-up above is shown waiters menu where the waiter can make reservation and assign guests to the appropriate table and search for reservation by clicking on a search button. There is button for system suggestion which helps waiters by using corresponding algorithm choose table. If needed to change order just one of the table should be chosen.

4 Elaboration phase

During the elaboration phase our team had to cover majority of the system requirements. The main purpose in this phase is to establish a preliminary architecture of the program and validate it for the sake of switching to the next phase. In order of being flexible, organized and making decision through collaboration our team has been using SCRUM methodology.

4.1 SCRUM

SCRUM is project development methodology. To apply it as planning tool in our Unified Process we made some variations. We consider each iteration as sprint, product backlog as use cases and etc.

Figure 5: “SCRUM Figure”



Picture above shows our flow of work, after defining use case diagram. In order to track and manage developing process one person was assigned like a SCRUM manager for splitting the work among group members.

4.2 Use case prioritization

To define most complex use case which will be fully implemented and described in further iterations we had to make use case prioritization table and convert it to the product backlog in order to use SCRUM in our development.

Table 6: “Prioritization”

Action	Complexity	Business Value	total value	Priority
Make order	5	4	20	2
Make reservation	5	5	25	1
Create merchandise	1	1	1	27
Find reservation	1	1	1	28
Delete reservation	2	2	4	19
Update reservation	1	1	1	29

As depicted in the table above we have two main use case which either have high complexity and business value assessment, therefore those two most important use cases will be implemented first for generating preliminary architecture of the software.

Table 7: “Product backlog for Reservation System”

Product backlog for Reservation system			
Item	Priority	Estimate	Planned
<i>Make order</i>	1	5 team days	Sprint 1,Elaboration
<i>Make reservation</i>	2	4 team days	Sprint 1, Elaboration
<i>Create merchandise</i>	3	3 team days	Sprint 2, Elaboration
<i>Find reservation</i>	4	1 team days	Spring 2, Elaboration
.....			

According to product backlog we have assigned first two use cases for sprint 1, another use cases to another sprints. Full product backlog can be found in the appendix.

4.3 First sprint

After pointing out most valuable and difficult use cases it is time to start first sprint. The main purpose of sprint was to design and implement first two use cases “Make Order” and “Make Reservation”. To make first sprint as fast as possible the work was divided among our team and we have been having meetings almost every day and discussing what each of the team member has done, clarifying and appointing new goals.

Fully dressed use case

In order to make further design like domain model, System Sequence Diagram, our team needed to totally describe use case step by step, it was supposed to be coherent as well as specific.

Table 8: “Fully Dressed Use Case-Make Order”

Use case name	Make order	
Actors	Employee/Manager	
Pre-condition	Table is created, merchandise is created, reservation is created	
Post-condition	Order was made and added to the system	
Frequency	70 times a day	
Main success scenario (flow of events)	Actor(action)	System(response)
	1.Guests decided on their order and waiter records it	

	4. Waiter clicks on the table and creates order	5. System creates order
	6.Waiter chooses appropriate merchandise and add to the order	7.System add merchandise to the order
	8.Waiter repeat step 6 until all merchandises are added	
	9.Waiter completes the order	10.System saves the order and sends notification to cook
Alternative flows	4a. Guest had the pre-order 1.Waiter asks if guest would like to change pre-order 1a.Guest would like to change pre-order 1.Waiter updates pre-order 2.Waiter completes order 3.System makes order active and send notification to the cook 1b.Gues would not like change anything 1.Waiter confirms pre-order 2.System makes order active with pre-order merchandises already added to the order	
	6a. Merchandise is out of stock (Drinks and Miscellaneous) 1.System informs waiter that product is out of stock 6b. Merchandise is running out of stock (Drinks and Miscellaneous) 1.System informs waiter that product is running out of stock	

Table 9: “Fully Dressed Use Case-Make Reservation”

Use case name	Make reservation	
Actors	Employee/Manager	
Pre-condition	Table is created	
Post-condition	The reservation is created and assigned to appropriate table.	
Frequency	50 times a day	
Main success scenario (flow of events)	Actor(action)	System(response)
	1. Customer calls and provides all information for reservation (name, phone number, amount of people, number).	
	2.Waiter(employee) enters information to the system	3.System checks if there is available table at the desired date and adds the table to reservation
	4. Waiter completes reservation	4.System creates reservation and adds information to the databases

Alternative flows	2a. There are no tables available
--------------------------	-----------------------------------

The tables above describe how business processes interacts with the system for achieving certain goal.

By using fully dressed use case shown above our team was able to design domain model.

Candidate class

The candidate class list represents a list with all the possible class names that can be used in the system. This list is created writing down all the nouns related with the system and only using the most relevant ones. This list represents only a part of the possible classes that can be used in the actual system. Some other names can be added later if needed. Also, a class with the same name is not required for every noun in the table.

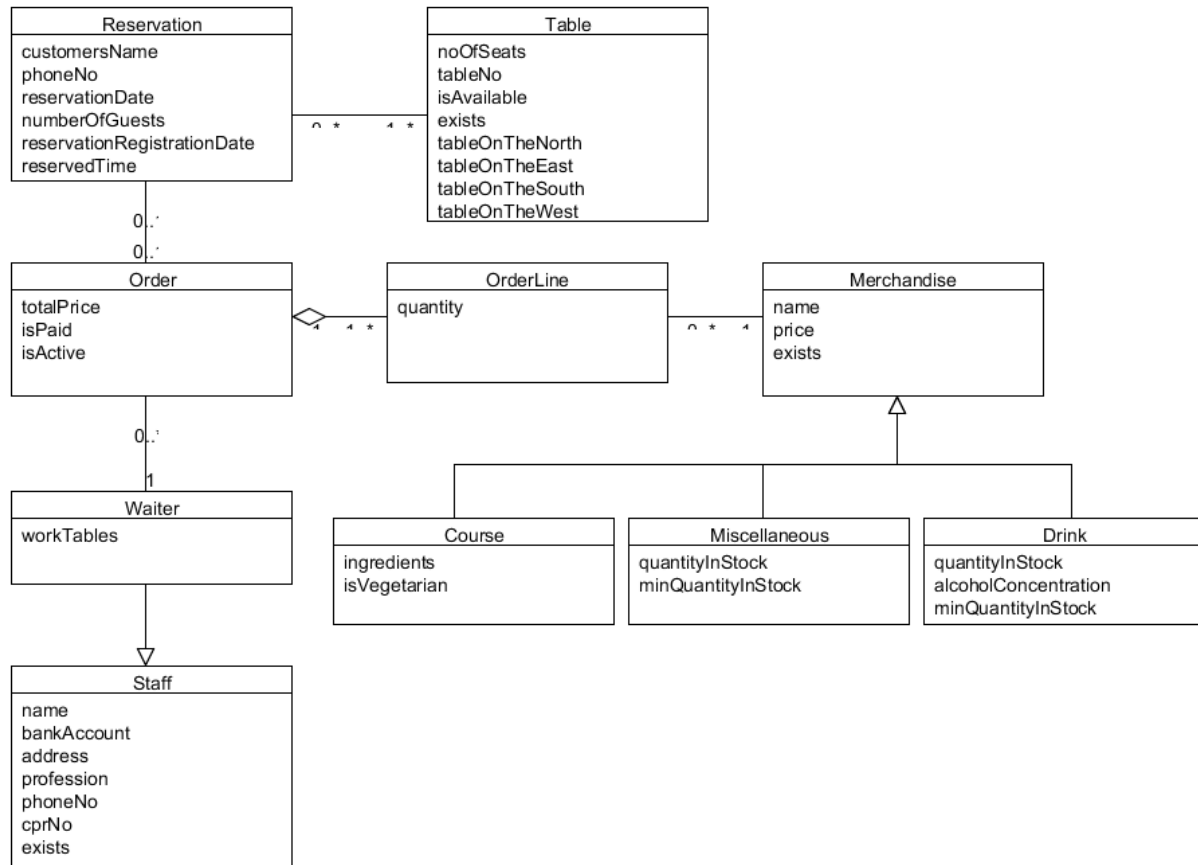
Table 10: “Candidate Class”

Merchandise	Table	Drink	Order
Chef	Miscellaneous	Course	Reservation
Guest	Waiter		

Domain model

Domain model is visual representation of real world. To define domain model we have been using candidate list.

Figure 6: "Domain Model"



Domain model of current project might not look too complex, but in the same time it was challenging. As it is shown in the diagram, there are 10 classes: Reservation, Table, Order, OrderLine, Merchandise, Course, Snack, Drink, Staff and Waiter. To understand what each class is for, description and explanation of the most interesting fields are provided below:

Reservation: this class was made to allow the system to create and manage reservations for customers if they want to book a table in restaurant. Reservation can be created, found, updated and cancelled. It contains following fields – customersName, phoneNo, reservationDate, numberOfGuests, reservationRegistrationDate, reservedTime. When reservation is made depending on time and date table(-s) will change its (their) state in field isAvailable to false (indicating that it can't be reserved for that specific time, latest time it can be booked will be one hour before reservation takes place and only if there are no more free tables available).

Table: this class was made to allow the system to create and manage the tables in restaurant. Table can be created, updated, found, and deleted. It contains the following fields – noOfSeats, tableNo, isAvailable, exists, tableOnTheNorth, tableOnTheEast, tableOnTheSouth, tableOnTheWest.

- isAvailable – provides the field to store the state of the table, either it is available or taken.
- exists – provides the field to store the state of the table, either it exists or not, if it's no more available in the restaurant (removed), so it would not mess up the system in the history of orders and reservations. It will stay in the system until company decides to clean it up.
- tableOnTheNorth – provides the field to store information of the table to the north of current table so the system would know how to combine table reservations if there are more customers than the seats at the table and could find if there is available table next to it which could be also reserved
- tableOnTheEast – same as tableOnTheNorth only checks for available table to the east
- tableOnTheSouth - same as tableOnTheNorth only checks for available table to the south
- tableOnTheWest - same as tableOnTheNorth only checks for available table to the west

Order: this class was made to allow the system to create orders when customers come to the restaurant. Order can be created, found, updated and cancelled. It contains the following fields – totalPrice, isPaid, isActive.

- isPaid – provides the field to store the state of the payment, either customer has paid or not.
- isActive – provides the field to allow the system to store the state of the order. When reservation is made, customer is able to make a preorder so the chef can start cook when customer arrives. isActive field will indicate the system about if the order should be visible to chef.

OrderLine: this class was made to allow the system to add the same item more than one time to the Order. Object of the OrderLine cannot exist without the Order. It contains the following field – quantity.

Merchandise: this class was made to allow the system to create, update, find and remove objects of the merchandise. It is an abstract class which means that whenever you create any of its subclasses it will provide them with the same fields that are implemented in the class Merchandise. It contains the following fields – name, price, exists.

- exists – provides the field to store the state of the merchandise, either it exists or not, if it's no more available in the restaurant (removed), so it would not mess up the system in the history of orders. It will stay in the system until company decides to clean it up.

Course: this class was made to allow the system to create, update, find and remove objects of the Course. Course is one of the main product of the restaurant which allows to create and record dishes that restaurant provides to its customers. It is a subclass of the abstract class Merchandise. It contains the following fields – ingredients, isVegetarian.

- Ingredients – provides the field to store the String of ingredients that describes what the dish is made of. It was a good idea for this field in order if customer has any allergy and he would like to find out what is the dish made of. Waiter will be able to find it out with ease.

Miscellaneous: this class was made to allow the system to create, update, find and remove objects of the Miscellaneous. This object is created for any other products that bar / restaurant is selling besides the courses and drinks. It contains the following fields – quantityInStock, minQuantityInStock.

- minQuantityInStock – provides the field to store the number of minimum amount of products in stock before the system notifies staff that it should be reordered.

Drink: this class was made to allow the system to create, update, find and remove objects of the Drink. With this class system can store all the drinks that bar / restaurant is selling. It contains the following fields – quantityInStock, alcoholConcentration, minQuantityInStock.

- alcoholConcentration – provides the field to store the number of the alcohol concentration in the drink. It waiters with information in case the customer will ask. It also allows the system to filter nonalcoholic drinks or alcoholic drinks.
- minQuantityInStock – works the same way as in the class Miscellaneous.

Staff: this class was made to allow the system to create, update, find and remove objects of the Staff. It is a superclass for class Waiter. It contains the following fields – name, bankAccount, address, profession, phoneNo, cprNo, exists.

- profession – provides the field to store the taken position of staff member in the system
- exists – same as all the other classes with this field it indicates if the staff member still works in the bar / restaurant and is stored in history.

Waiter: this class was made to allow the system to create, update, find and remove objects of the Waiter. This class allows to add waiter to Order to provide it with information of who served it. It contains the following field – workTables.

- workTables – provides the field to store a list of tables that waiter is working on at his working hours.

Associations

Reservation class is associated with classes Table and Order to store the core information about every detail for that specific reservation. Reservation can be made by phone and also we use the same system if the customers comes to the restaurant and wants to receive services. In any way it will be a reservation of a table. Reservation can have 1 to many tables if there is more persons than seats at the table and they are next to each other multiple tables can be reserved for the same reservation. Table can have 0 to many reservations in the same day if no other choices are left.

Order class is associated with classes Waiter and OrderLine to store the information about the order and waiter who served it, if something goes wrong. Order and OrderLine is an aggregate, where Order is the root. The orderLine cannot exist without the order. Order can have 1 to many orderLines. OrderLine can have one and only one order. Order must contain one Waiter. Waiter can have 0 to many orders.

OrderLine class is associated with Merchandise class. OrderLine can have only one kind of merchandise with quantity one or more. Merchandise can have 0 to many orderLines.

Merchandise is generalized in three subclasses – Course, Drink and Miscellaneous.

Class Staff is associated with the class Waiter. Class Waiter is a subclass of the Staff class which makes the Staff a superclass.

Conclusion

Analysis of the business workflow and its requirements for the system gave this project a good overview of the structure and functionality. This led to domain model as it is shown and described in this section. This is the core plan for the classes and their fields for the code and gave all the necessary information for system to develop further. This shows the improved business workflow, which includes the system.

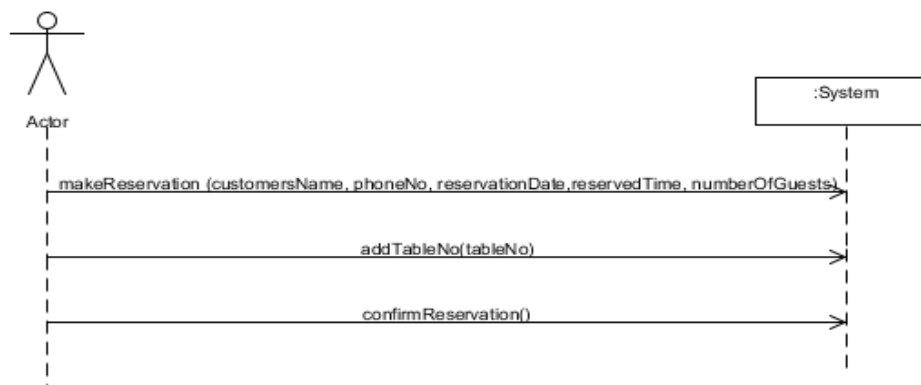
Description also helped to keep all members informed about the design of our system and to understand what the fields are for. This lets the members to be up to date if something changes or if anyone has some doubts about the domain model and how everything works. Also it lets to avoid discussions about the same things between members if they missed anything.

System sequence diagram

The system sequence diagram (SSD) is the artefact which shows inputs and outputs of the system as a black box which means it only describes what system does , no how it does it.

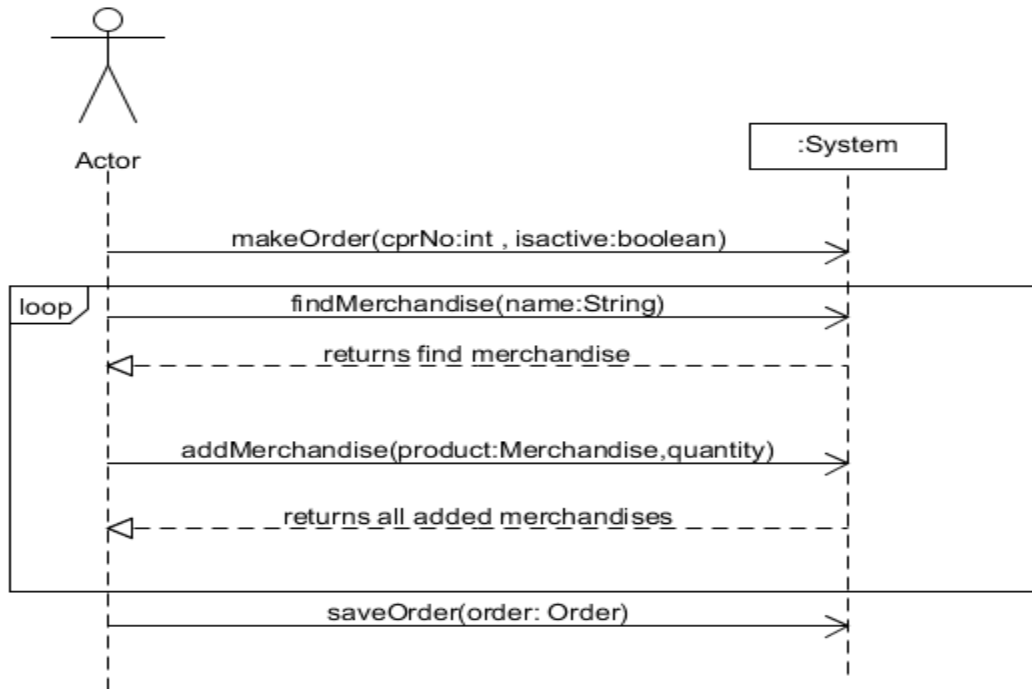
The SSD below was made by following main use case Make Reservation and Make Order.

Figure 7: “SSD-Make Reservation”



In Make Reservation customer provides all required information and actor makes reservation with adding the table afterwards. By confirming reservation actor sends message to the system to save it and insert into database.

Figure 8: “SSD-Make order”



In Make Order actor makes order in which he specifies tables and who is serving those tables. Afterwards actor inputs merchandises until all of them will be added and system shows all products added already. Afterwards actor completes order and system save it and inserts to the database afterwards.

Operation contracts

Operation contracts describe detailed changes to objects in a domain model, as the result of a system operation. [Craig Larman].Pre-conditions and post-conditions should be described as well. The methods for operation contract should be retrieved from System Sequence Diagram.

Table 11: “Operation Contract- Make Reservation”

Operation:	makeReservation(customersName,phoneNo,reservationDate,reservedTime, numberOfguests)
Use cases:	Make reservation
Precondition:	a Table instance is created
Post conditions:	The reservation procedure has started
	New reservation instance is created
	New reservation is saved in the database

Table 12: “Operation Contract- Make Order”

Operation:	makeOrder(cprNo:int, isactive: boolean)
------------	---

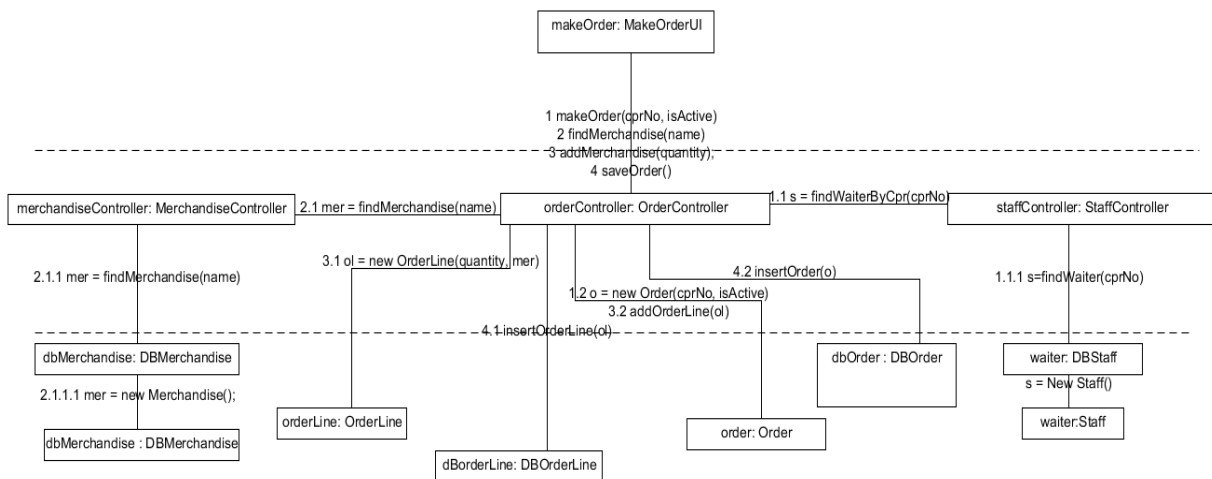
Use cases:	Make order
Precondition:	Stuff is created
Post conditions:	Order object is created and associated with reservation and waiter

We described only most important methods in our use case. The tables above shows how does system state changes depending on pre and post conditions. Additional operation contracts can be found in the Appendix.

Interaction diagram

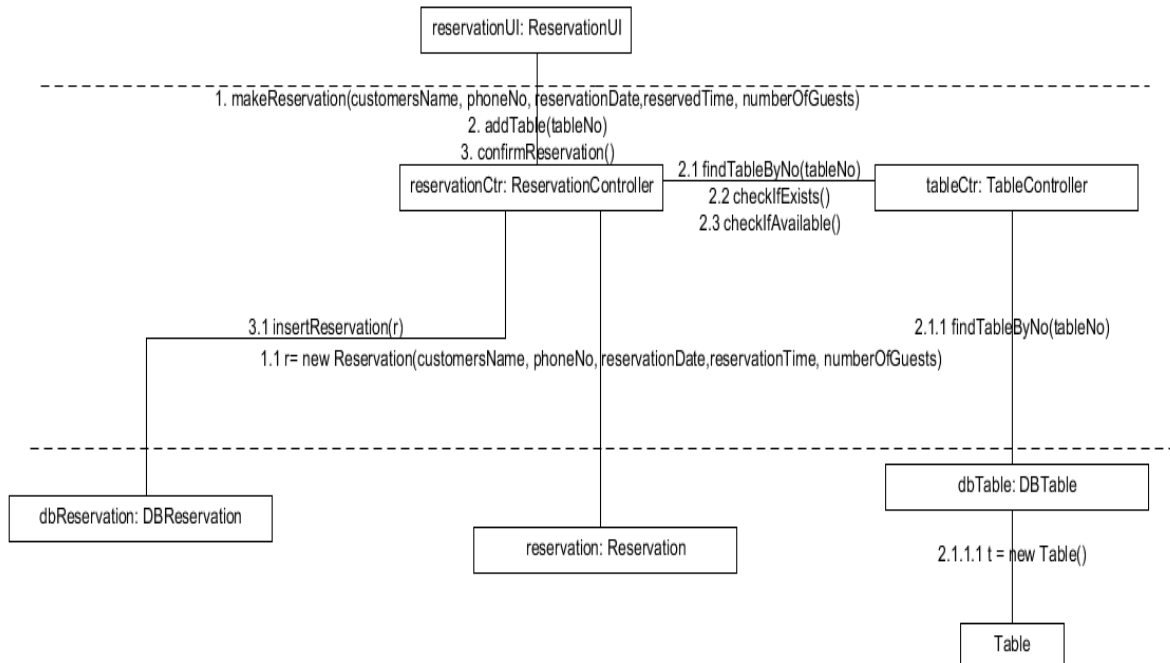
The interaction diagram shows the three-layered architecture (model/database layer, control layer and interface layer), the classes and methods that are in connection with the given use case. The diagram also shows the order of procedure and the interaction between the objects and the layers.

Figure 9: “Interaction Diagram”



In all the cases the order of procedure, the different tasks start from the user interface (generally by an input) and the subtasks are in the control and model layer. In the make order use case the waiter starts new order, system finds waiter by CPR number in DBStaff (Database) via staff controller. Next step is find merchandise by sending message to the merchandise controller which in turn sends message to DBMerchandise and makes merchandise object. To specify quantity we add merchandise to orderLine object which was created in third step by method addMerchandise. The last step is to insert new order object to the database.

Figure 10: “Interaction Diagram”



To make reservation, actor through GUI sends message makeReservation to the reservation controller which creates new reservation object. Next step is to add table to the reservation by sending message to the table controller. Controller finds table in the database and makes table object. Check if exists and checks availability. After confirming reservation, all information is inserted to the database.

Design class diagram

Design class diagram it is almost implemented program, it shows the structure of the system with classes, attributes, methods and relationships among different objects.

Design class diagram shown below describes one use case make order. We have three layers GUI, Controller, Model. Current part of the diagram shows only 3 controllers which are connected from main Order controller. At the same time according to the interaction diagram order controller has connection to model layer classes for retrieving information from Database and create instances for further usage. All parts of Design class diagram can be found in the appendix where we divided all design class diagram in different parts(each controller) in order to fit A4 format.

Figure 11: “Design class diagram with main controllers”



Code

Figure 12: “Code Example of OrderController Class”

```

public OrderController()

public void makeOrder(String cprNo, boolean isActive) throws DatabaseException {
    Staff waiter = staffController.findWaiterByCpr(cprNo);
    order = new Order();
    order.setActive(isActive);
}

public void findMerchandise(String name) {
    mer = merchandiseController.findMerchandise(name);
}

}
  
```

In the figure above it can be seen that all implemented methods are from our Design Class Diagram. According to the diagram table, staff, merchandise and reservation controllers are created to have connection between objects. There are five methods as well according to system design.

1. makeOrder the method with parameters cprNo and isActive. Creates waiter object by finding it by CPR number through staff controller and Staff controller creates DBStaff and retrieves information from Database. Afterwards it creates new Staff object and send it back.

Figure 13: "The piece of code of Staff Controller"

```
public class StaffController {
    DBStaff dbStaff = new DBStaff();

    public Staff findWaiterByCpr(String cprNo) throws DatabaseException{
        return dbStaff.findStaff(cprNo);
    }
}
```

2. FindMerchandise method with parameters name of merchandise, creates merchandise controller object, which searches for merchandise by creating DbStaff object in the Database. In the picture below it can be seen FindMerchandise method in the Merchandise controller searching in three different tables (Course, Miscellaneous, Drink) and gives back Merchandise object which one we are casting afterwards to clarify what type of object is retrieved.

Figure 14: "The piece of code of Merchandise Controller"

```
public Merchandise findMerchandise(String name){
    for(int x = 0; x < dBCourse.getAllCourses().size(); x++){
        if(name.equals(dBCourse.getAllCourses().get(x).getName())){
            return dBCourse.getAllCourses().get(x);
        }
    }
    for(int x = 0; x < dBMiscellaneous.getAllMiscellaneous().size(); x++){
        if(name.equals(dBMiscellaneous.getAllMiscellaneous().get(x).getName())){
            return dBMiscellaneous.getAllMiscellaneous().get(x);
        }
    }
    for(int x = 0; x < dBDrink.getAllDrinks().size(); x++){
        if(name.equals(dBDrink.getAllDrinks().get(x).getName())){
            return dBDrink.getAllDrinks().get(x);
        }
    }

    return null;
}
```

Thread

Threads allow execute multiple tasks at the same time. We created one additional thread for retrieving information from Database and make ArrayList of tables for further usage.

Figure 15: "Example of Table Threads class"

```
public class TableThreads extends Thread {
    private Object obj=new Object();
    private Table table;
    private ArrayList<Table> tables;
    private DBTable dbTable;
    private int numberOfTables;
    public TableThreads(int numberOfTables) {
        public void run() {
            synchronized(obj){
                for (int i=1;i<=numberOfTables;i++){
                    try {
                        table = dbTable.findTableByTable
                        tables.add(table);
                    } catch (DatabaseException e) {
                        // TODO Auto-generated catch bloc
                        e.printStackTrace();
                    }
                }
            }
        }
        public ArrayList<Table> returnTables() {
    }
}
```

Exception class

To make our own logic in handling exceptions in the code we have decided to make our own Exception class which will handle database exceptions. The picture below shows example of Exception class.

Figure 16: "Example of Database exception class"

```
public class DatabaseException extends Exception
{
    String errorString;
    public DatabaseException(String error)
    {
        errorString=error;
    }
    String printError() {
        return "Exception was caught because something when wrong in the"
            + "database ::" +errorString;
    }
}
```

Algorithm

The main algorithm implemented it searches through all the available tables that are currently in the database and, depending on the number of seats needed it occupies a certain number of tables. Every tables has a number and a maximum seats available. Also, for every table the neighbors are known. The search starts by checking if the first table that's available is enough to create the order. If it's now then it saves all the neighbors of that table into an Linked List. The tables are occupied one by one and every time one table is occupied the neighbors are added to the same Linked List. The process continues until there are no more seats needed or until there are no more tables available for booking. The same order is attributed to all the tables that are booked so all of them will show the same interface with the same merchandise.

```
public void createOrderAndAddTables() {
    int numberOfGuests = Integer.parseInt(noOfSeatsTextField.getText());
    int numberOfChairsNeeded = numberOfGuests;
    int i = 0;
    while (!tableController.getAllTables().get(i).isAvailable()) {i++;}
    if (i < tableController.getAllTables().size()) {
        Table table = tableController.getAllTables().get(i);
        table.setTableNo(i + 1);
        table.setExists(true);
        table.setAvailable(false);
        tables.add(table);
        rc.addTableToReservation(table);
        numberOfChairsNeeded -= tableController.getAllTables().get(i).getNoOfSeats();
        if (numberOfChairsNeeded > 0) {
            checkTables(i, numberOfChairsNeeded);
        }
    }
    Order order = new Order();
    order.setOrderId(orderController.getAllOrders().size());
    order.setActive(true);
    order.setTotalPrice(0);
    try {
        orderController.insertOrder(order);
        rc.makeReservation(nameTextField.getText(), phoneNoTextField.getText(), days.getSelectedItem().toString() + "-"
            + monthsYear.getSelectedItem().toString(), time.getSelectedItem().toString(), Integer
                .parseInt(noOfSeatsTextField.getText()), order);
    } catch (NumberFormatException e) {
        e.printStackTrace();
    } catch (DatabaseException e) {
        e.printStackTrace();
    }
}
```

Test

To make sure that system works according to the user requirements and have no bugs or mistakes test are performing. There are different artefacts for testing software such as unit test, integration test, system test, user acceptance test.

User acceptance test

To make proper test for the system, we asked couple of people who is not involved in our project to help us. After preparing few scenarios and giving them short user manual people were asked to accomplish ones. For purpose of tracking comments were written during performing of test.

The following scenarios were given to the testers:

- Create merchandise
- Create reservation
- Create order
- Change order
- Add merchandise to the order
- Delete merchandise from the order

Table 13: “User Acceptance Test”

Problem	Solution
Program crashed when user input wrong data typed	Problem fixed by adding additional Exceptions handling
When user inputs twice the same thing the system crashes	Problem fixed
Not enough prompts for the user in UI	Added more prompts in order to ease user surfing

Unit and Integration test(JUnit)

Using Junit framework in order to identify system bugs JUnits tests were created. Unit test is white box testing which shows internal bugs in the program. JUnits were created for all CRUD operations.

System test

System test is black box testing which purpose is to figure out if made software corresponds its specification.

Table 14: “System Test”

Test case nr	Pre-conditions	Test Steps	Expected results	Verified
1. Make order	merchandise was created, reservation was created	Choose reservation Select make order Add merchandise Click complete order	Order should be created and merchandises added, order added to database.	Yes
2. Make reservation	Tables are created	Waiter clicks make reservation and inputs all information. If there is free table waiter inputs additional information. Waiter asks for pre-order and select make order if answer is true.	Reservation should be created with guests information inserted and with pre-order or without	Yes

5 Construction phase

After implementing main use case through one sprint in the Elaboration phase we had a preliminary implemented program and basic architecture. To make other use cases and features of the system we had to start another sprint according to construction phase.

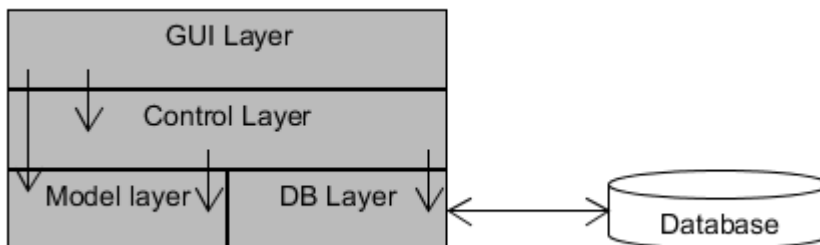
5.1 Second sprint

In the second sprint we implemented the rest of the use cases, describe the relational model for database and add additional features.

Data base architecture

To store all data in the database in our software we have applied Data access layer besides GUI layer and Business Logic Layer (Control Layer).

Figure 17: “Layer Architecture”

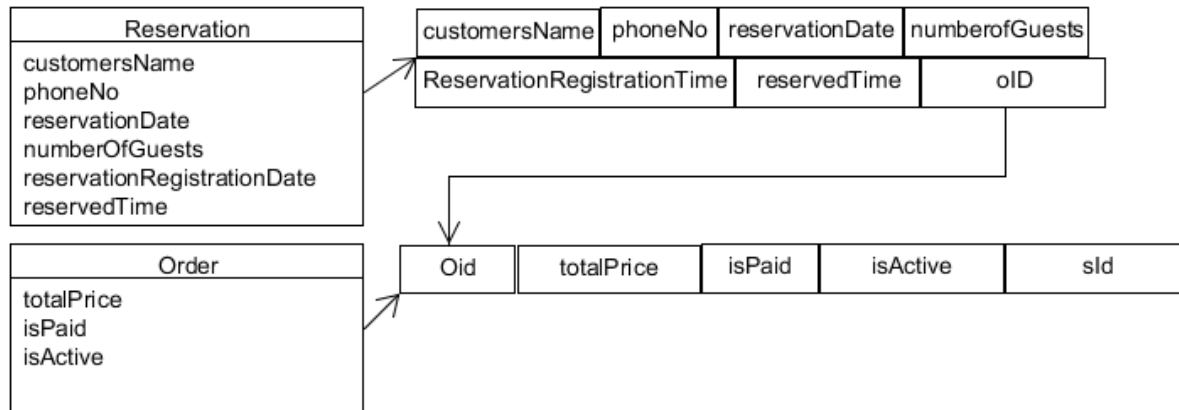


For the sake of making database connection we have used Data Access Object which holds information relating business logic and SQL statements for connection. For each Data access object we have one interface with

Relational model

Relational model it is formal model of a relational database. By using it we convert domain model into relational tables and make relationship between those tables. To show our team consideration and made solutions some converting are shown below.

Figure 18: “Relational Model”



The picture above depicts transformation reservation to the reservation table and order to the order table in the relational model. The last column in the reservation table is foreign key for Order table which represents us connection between those two classes according to the domain model.

Grasp and GoF patterns

To make our code reusable, flexible and maintainable we were designing our software according to the GRASP principles (patterns) which is considering as assign responsibilities patterns. We have been using mostly few of them – Low coupling, High cohesion, Controller and Creator.

Low coupling

To improve reusing potential of our code low coupling principle was applying all the time. We tried to reduce impaction in one class on another.

High cohesion

To make code more comprehensible, manageable and understandable we assigned to each class one or few responsibilities in order to avoid unrelated ones. Therefore our code was strongly related and highly focused.

Controller

In order to avoid appointing responsibilities to UI classes we have been employing controller principle which means that we had additional object for receiving messages from UI and delegate activity along other objects

Creator

We have been following few rules according to this principle whereby we were able to define which class in responsible for creation another class

Gof

Only one pattern Singleton has been employed from GoF collection.

6 Group Evaluation

During the entire project, our approach in working on it was to divide the tasks to each team member, in order to be efficient regarding the time consume. We have been trying to understand each other and make our communication efficient and polite despite of misunderstandings and different colors in our inside profile. To work efficiently each our team member have been using different approach and planning. To make our work more effective we have been planning our meetings every day and trying to as often as we could.

Along the bench of good things we have faced some problems as well. There was difficult to divide all the work among our members in the group because of different experience. During the project some of us had problems and therefore it affected attendance and flow of our project.

Finally we overcome all obstacles and put all efforts for our project. It was excellent experience for further projects and group work.

7 Conclusion

The purpose of this project was to develop appropriate IT system for the “Kad Zebras Deg” bar/restaurant without previous experience from customer side. We have implemented software which partially fulfills the requirements of customer needs. We could not execute full IT program because of lack of time and experience. We defined majority of business requirements and shortly described them in order to derive most important ones for further implementation. By using SCRUM and Unified Process we have done flexible agile software development and preliminary executable program architecture. The final step was to implement the rest use cases. The main part of the program was developed according to the requirements with possibility for further extension.

Each member of our team has obtained a better understanding throughout working on this project. Relying on our previous knowledge gained during the lessons as well as following our supervisor advices we have been trying to overcome the challenges and obstacles which came out within our work. This project was a big impulse for us to future learning.

8. References

[Bloisi] Management and Organisational Behaviour Chp 3, Wendy Bloisi

[Cadle and Yeates] Project Management for information systems Chp 3, Cadle and Yeates (2008)

[Krogager] Management accounting, financial measurement and planning, Volume 1, 2nd edition, Academica, Chp 1.3.6, 2006.

[Trading economics] <http://www.tradingeconomics.com/latvia/wages> (22.04.2015)

[X Net] <http://www.xnet.lv/index.php?menuid=11&mini=7&kas=Galda+datori> (22.04.2015)

“Linus Torvalds” Linus Benedict Torvalds is a Finnish American software engineer, who was the principal force behind the development of the Linux kernel that became the most popular kernel for operating systems.

[Tannenbaum and Schmidt] Organisational behavior, individuals, groups and organization, 4th edition, Ian Brooks. Page 169.

[D.Fisher and W.R.Torbert] Organisational behavior, individuals, groups and organization, 4th edition, Ian Brooks. Page 176.

[Herzberg 1968] Organisational behavior, individuals, groups and organization, 4th edition, Ian Brooks. Page 93.

[Kanter 1983] Organisational behavior, individuals, groups and organization, 4th edition, Ian Brooks. Page 270.

9. Appendices

9.1 Event Tables

As is Table

Event (AS IS)	Activity	Step in activity	Actor
---------------	----------	------------------	-------

Merchandise is delivered to company and needs to be registered	Merchandise registration	<ul style="list-style-type: none"> *Employee receives merchandise *Employee writes everything down on paper *Employee records data of merchandise in MS Excel *Selling price and name of new merchandise is recorded in the book of merchandise 	Employee / Manager
Merchandise needs to be checked	Check merchandise	<ul style="list-style-type: none"> *Employee goes to cellar or looks in the fridge for merchandise *Employee counts how many merchandise have left or checks its expiry date 	Employee / Manager
Employee needs to find price of merchandise	Get price	*Employee opens book of merchandise and finds the price or checks the price label	Employee / Manager
Employee needs to change information about merchandise	Update merchandise	<ul style="list-style-type: none"> *Employee finds the merchandise in the MS Excel and in the book of merchandise *Employee changes data of merchandise in the MS Excel and rewrites information in the book of merchandise. 	Employee / Manager
Merchandise is no more available	Remove merchandise	*Employee crosses out merchandise from the book of merchandise	Employee / Manager
Order is being made	Make order	<ul style="list-style-type: none"> *Customer sits at the table and chooses courses *Employee / Manager writes down order in the notepad *Employee / Manager goes to the kitchen and gives the order sheet to chef 	Employee / Manager
Table is being reserved	Make reservation	<ul style="list-style-type: none"> *Customer calls Bar / Restaurant and asks for the table *Employee manager checks the reservation list and decides which table can be assigned *Employee / Manager writes down the new reservation in the reservation list. 	Employee / Manager

To be table

Event (TO BE)	Use Case	Step in use case	Actor
Create merchandise on the system	Merchandise CRUD (Create)	<ul style="list-style-type: none"> *Employee / manager inputs data of merchandise in the system *Employee saves data on the system 	Employee / Manager

Find merchandise on the system	Merchandise CRUD (Find)	*Employee / manager inputs the name of merchandise in the system *System returns information about the merchandise	Employee / Manager
Update merchandise on the system	Merchandise CRUD (Update)	*Employee inputs the name of the merchandise in the system *System returns information about the merchandise *Employee enters new data in the system *Employee saves new data on the system	Employee / Manager
Merchandise is no more available and is being removed from the available items, but not from the system	Merchandise CRUD (Delete)	*Employee inputs name of merchandise in the system *System returns information about the merchandise *Employee removes merchandise from the available merchandise	Employee / Manager
Customer wants to make an order	Make order	*Customer sits at the table and chooses courses *Employee/Manager writes down information in the notepad *Employee/Manager inputs customers desired course or product in the system *Employee/Manager repeats the action until all desired courses or products are in the order *Employee/Manager complete the order	
Customer wants to reserve the table	Make reservation	*Customer calls and specifies date and time for reservation *Employee / manager enters information in the system *Employees / manager save the information on the system	

9.2 Fully Dressed use Cases

Use case name	Create merchandise
Actors	Employee/Manager
Pre-condition	
Post-condition	Merchandise created and saved in the system

Frequency	1-5 times a day	
Main success scenario (flow of events)	Actor(action)	System(response)
	1.Manager/employee begins to create a merchandise and inputs the name of it	2.System checks if desired merchandise exists
	3. Manager/employee inputs information(name, price) about merchandise.	4. System adds information to merchandise
	5.Manger/empoyee complete action	5.System creates merchandise and saves information to data base.
Alternative flows	2a. Merchandise is exists in the system 1.System asks to add another merchandise or cancel .	

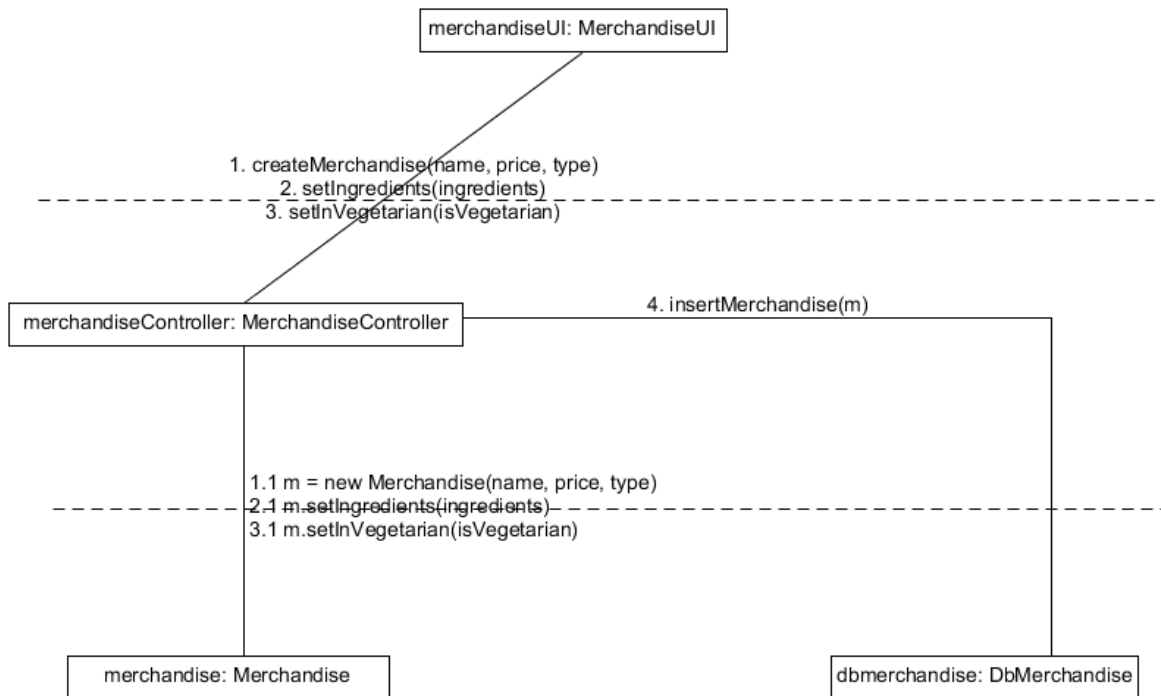
Use case name	Read(Find) merchandise	
Actors	Employee/Manager	
Pre-condition	Merchandise created and saved in the system	
Post-condition	Merchandise is found and system revealed it	
Frequency	100 times a day	
Main success scenario (flow of events)	Actor(action)	System(response)
	1.Manager/employee send request to the system in order to find merchandise	2.System requies to input information about merchandise
	3. Manager/employee inputs information about merchandise.	4. System finds merchandise according to given information
	5.Manger/empoyee complete action	5.System shows merchandise.
Alternative flows	4a. Merchandise was not found 1.System asks to input another information or finish the action	

Pre-condition	Merchandise created and saved in the system	
Post-condition	Merchandise is found and information was changed about it	
Frequency	10 times a day	

Main success scenario (flow of events)	Actor(action)	System(response)
	1.Manager/employee send request to the system in order to update merchandise	2.System requies to input information about merchandise
	3. Manager/employee inputs information about merchandise.	4. System finds merchandise according to given information
	5.Manager/empoyee inputs new information in order to update it	6.System updates merchandise
	7.Manage completes action	8.System saves information
Alternative flows	4a. Merchandise was not found 1.System asks to input another information or finish the action	

Use case name	Delete merchandise	
Actors	Employee/Manager	
Pre-condition	Merchandise created and saved in the system	
Post-condition	Merchandise is deleted from the system	
Frequency	5 times a day	
Main success scenario (flow of events)	Actor(action)	System(response)
	1.Manager/employee send request to the system in order to delete merchandise	2.System requies to input information about merchandise
	3. Manager/employee inputs information about merchandise.	4. System finds merchandise according to given information
	5.Manager/empoyee deletes it	6.System saves changes
Alternative flows	4a. Merchandise was not found 1.System asks to input another information or finish the action	

9.3 Interaction diagrams



9.4 Operation Contracts

Operation:	makeReservation(customersName,phoneNo,reservationDate,reservedTime, numberOfguests)
Use cases:	Make reservation
Precondition:	a Table instance is created
Post conditions:	The reservation procedure ha started
	New reservation instance is created
	New reservation is saved in the database

Operation:	addTableNo(tableNo)
Use cases:	Make reservation
Precondition:	Reservation has started
Post conditions:	table number was entered
	table is reserved and not available for others
	In case of preorder an inactive order has been created

Operation:	confirmReservation()
Use cases:	Make reservation

Precondition:	Table is available
Post conditions:	Reservation confirmed and saved in the database

Operation:	makeOrder(cprNo:int, isactive: boolean)
Use cases:	Make order
Precondition:	Stuff is created
Post conditions:	Order object is created and accosiated with reservation and waiter

Operation:	findMerchandise(name:String)
Use cases:	Make order
Precondition:	Order object is created
Post conditions:	Obejct returns found merchandise

Operation:	insertMerchandise(product:Merchandise,quantity)
Use cases:	Make order
Precondition:	Merchandise created, order object is created
Post conditions:	Obejct Merchandise accosiated with obejct order

Operation:	createMerchandise(name, price, type)
Use cases:	Create Merchandise
Precondition:	Item not in the database
Post conditions:	Item is created and saved The item is defined as one of the three types available

Operation:	setIngredients(ingredients)
Use cases:	Create Merchandise
Precondition:	Merchandise was created
Post conditions:	Item is updated and the ingredients are set.

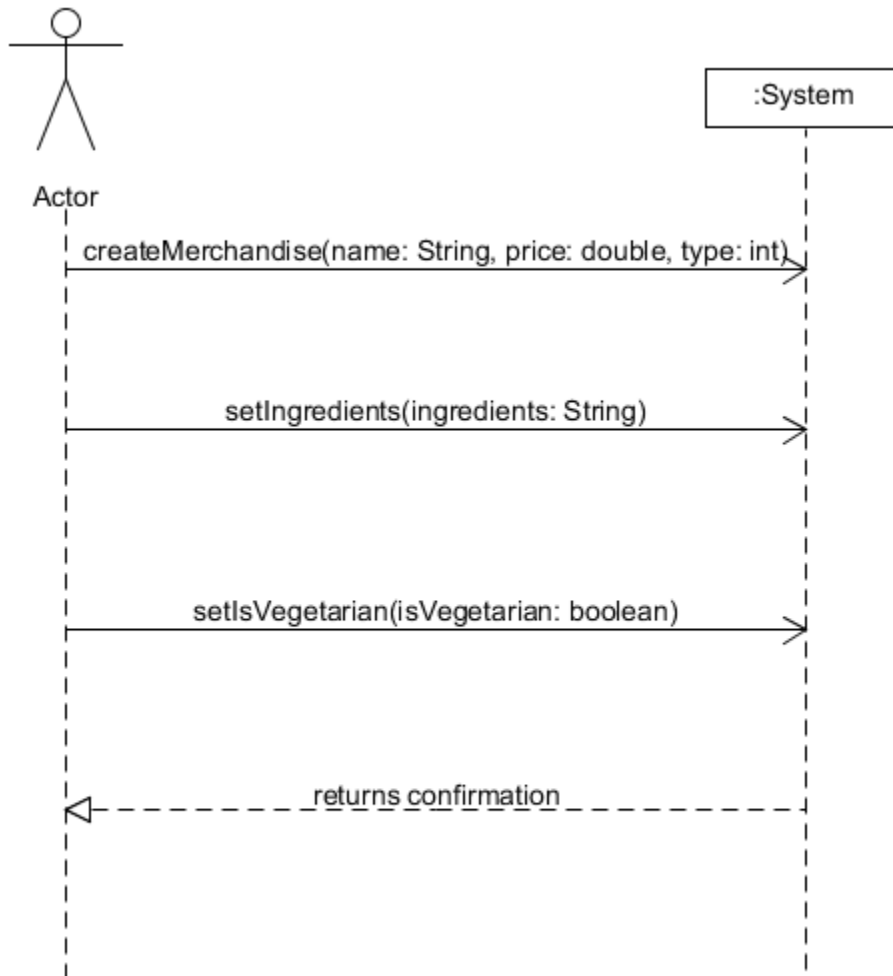
Operation:	setIsVegetarian(isVegetarian)
Use cases:	Create Merchandise
Precondition:	Merchandise was created
Post conditions:	Item is updated.

9.5 Product backlog

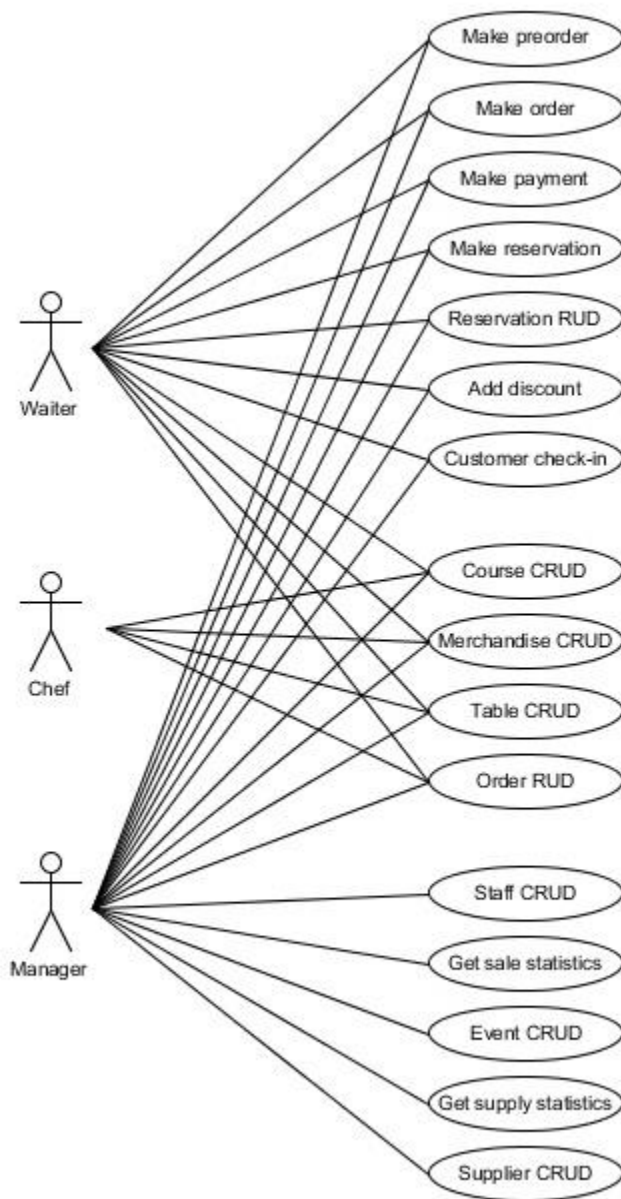
		Sprint			Time	Time left
ID	Description	#1	#2	#3		
	Effort needed for release 1					
Inception phase						
1	Workflow	2	0	0	40	39
2	As is/To be tables	2	0	0		
3	Use case diagram	2	0	0		
4	Brief Use Cases	1				
Elaboration phase						
4	Use case prioritization	3	0	0	39	39
6	Fully dresses use cases	2	0	0		
7	Candidate class list	2	0	0		
8	Mock-ups	4	0	0		
9	Domain model	10	0	0	35	30
10	System sequence diagram Make Order	5	0	0	30	28
11	System sequence diagram Create Merchandise	5	0	0		
12	System sequence diagram Make Reservation	5	0	0		
13	Interaction diagram Make Order	6	0	0		
14	Interaction diagram Create Merchandise	6	0	0	28	27
15	Interaction diagram Make Reservation	6	0	0		
16	Design class diagram	20	5	0		
17	Junit tests and Use case tests	0	5	0		
Construction phase						
18	Relational Model	20	5	0	17	15
19	Database architecture	25	5	0	14	14
20	Model layer	0	10	0	13	15
21	Database layer	0	20	0		
22	Exception layer	0	3	0		
23	Control layer	0	25	0		
24	GUI layer	0	30	0	10	9
Release 1 Testing version						
25	Implementing check	5	0	0	9	8
26	Fixing mistakes	5	0	0		
27	Implementing new small features	5	0	0		

Release2 Final version release						
28	Adding discount	1	0	0	7	5
29	Adding notifications	2	0	0	0	0
Effort in the whole backlog		144	108			

9.6 System sequence diagram



9.7 Use case diagram



9.8 Brief Use Case

Use Case :	Course CRUD
	<p>Employee and manager(actors) uses the system and able to do :</p> <ul style="list-style-type: none"> • Create course • Find course • Update course • Delete course <p>To create the course actors register it in the system and save it. System creates and saves</p>

	<p>the course.</p> <p>To find course actors put information in the system. The system finds course.</p> <p>To update course actors put information. System finds course and actors update it.</p> <p>To delete course system finds it and actors delete it.</p>
--	---

Use Case :	Suplier CRUD
	<p>Employee and manager(actors) uses the system and able to do :</p> <ul style="list-style-type: none"> • Create suplier • Find suplier • Update suplier • Delete suplier <p>To create the suplier actors register it in the system and save it. System creates and saves the course.</p> <p>To find suplier actors put information in the system. The system finds course.</p> <p>To update suplier actors put information. System finds course and actors update it.</p> <p>To delete suplier system finds it and actors delete it.</p>

Use Case :	Merchandise CRUD
	<p>Employee and manager(actors) uses the system and able to do :</p> <ul style="list-style-type: none"> • Create merchandise • Find merchandise • Update merchandise • Delete merchandise <p>To create the merchandise actors register it in the system and save it. System creates and saves the course.</p> <p>To find merchandise actors put information in the system. The system finds course.</p> <p>To update merchandise actors put information. System finds course and actors update it.</p> <p>To delete merchandise system finds it and actors delete it.</p>

Use Case :	Table CRUD
	<p>Employee and manager(actors) uses the system and able to do :</p> <ul style="list-style-type: none"> • Create table • Find table • Update table • Delete table

	<p>To create the table actors register it in the system and save it. System creates and saves the course.</p> <p>To find table actors put information in the system. The system finds course.</p> <p>To update table actors put information. System finds course and actors update it.</p> <p>To delete table system finds it and actors delete it.</p>
--	---

Use Case :	Order CRUD
	<p>Employee and manager(actors) uses the system and able to do :</p> <ul style="list-style-type: none"> • Create order • Find order • Update order • Delete order <p>To create the order actors register it in the system and save it. System creates and saves the course.</p> <p>To find order actors put information in the system. The system finds course.</p> <p>To update order actors put information. System finds course and actors update it.</p> <p>To delete order system finds it and actors delete it.</p>

Use Case :	Make payment
	<p>Employee and manager uses the system to calculate all price and print the receipt. System calculates price prints receipt and saves information to the statistics.</p>

Use Case :	Reservation CRUD
	<p>Employee and manager(actors) uses the system and able to do :</p> <ul style="list-style-type: none"> • Create reservation • Find reservation • Update reservation • Delete reservation <p>To create the reservation actors register it in the system and save it. System creates and saves the course.</p> <p>To find reservation actors put information in the system. The system finds reservation .</p> <p>To update reservation actors put information. System finds course and actors update it.</p> <p>To delete reservation system finds it and actors delete it.</p>

Use Case :	Add discount
	Employee and manager uses the system to add discount to the order if costumer dissatisfied. System saves discount and re-counts a price for costumer.

Use Case :	Customer check in
	Employee and manager uses the system to confirm reservation when customer came to the restaurant

Use Case :	Send notification to waiters
	Chef(employer) after finishing course confirm that by putting information and system send notification to waiters that course is ready.

Use Case :	Stuff CRUD
	<p>Manager uses the system and able to do :</p> <ul style="list-style-type: none"> • Create stuff • Find stuff • Update stuff • Delete stuff <p>To create the stuff , manager registers it in the system and save it. System creates and saves the course.</p> <p>To find stuff, manager puts information in the system. The system finds course.</p> <p>To update stuff, manger puts information. System finds course and actors update it.</p> <p>To delete stuff system finds it and manager deletes it.</p>

Use Case :	Get statistics
	Manager retrieves information from the system in order to trace order statistics. Manager puts period information and system shows it.

Use Case :	Event CRUD
	Manager uses the system and able to do :

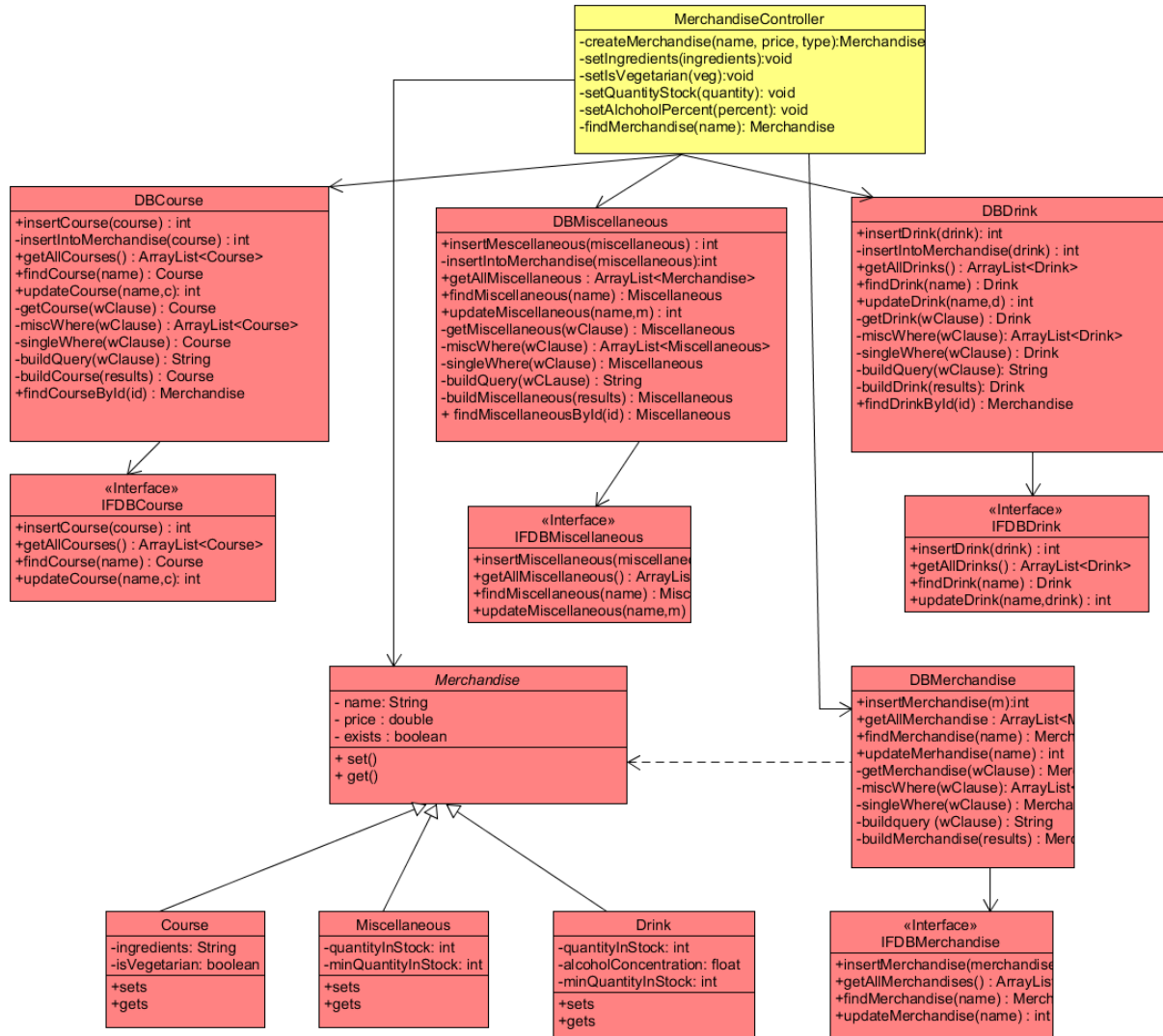
	<ul style="list-style-type: none"> • Create event • Find event • Update event • Delete event <p>To create the event, manager registers it in the system and save it. System creates and saves the course.</p> <p>To find event, manager puts information in the system. The system finds course.</p> <p>To update event, manger puts information. System finds course and actors update it.</p> <p>To delete event system finds it and manager deletes it.</p>
--	--

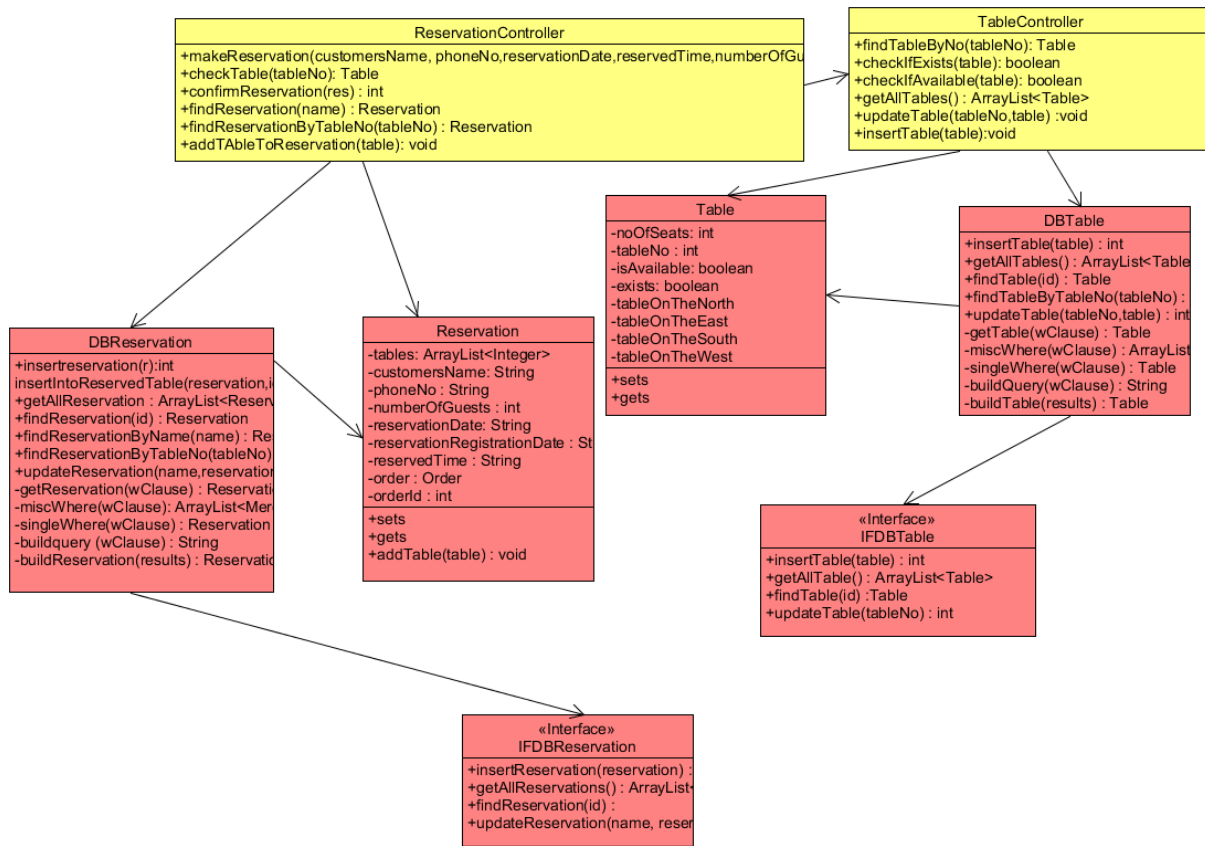
Use Case :	Get supply statistics
	Manager retrieves information from the system in order to trace supply statistics. Manager puts period information and system shows it.

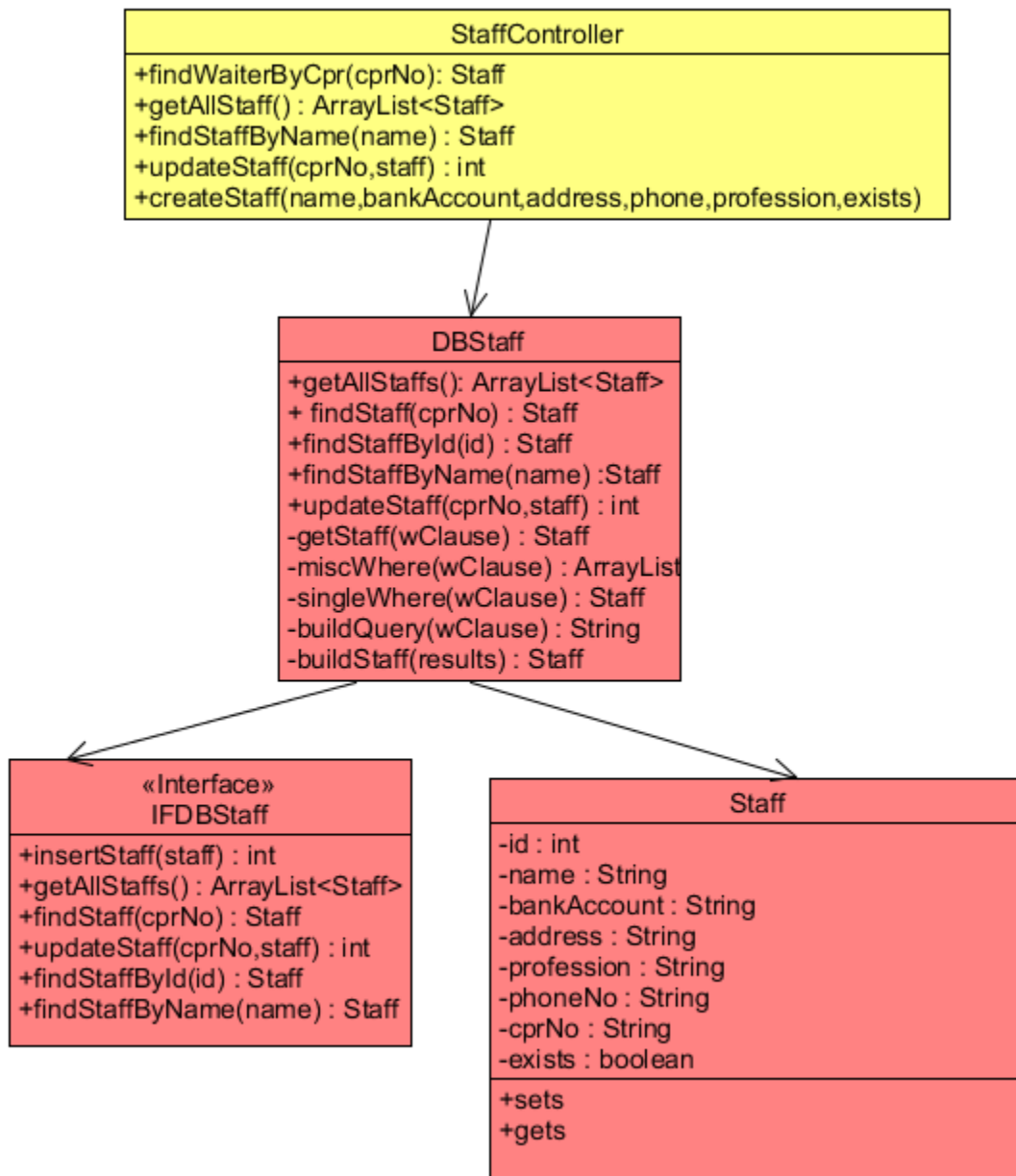
Use Case :	Send notification to chef
	Waiter(employer) puts information to the system about odred information and system sends appropriate notification to the chef (employer) what course should be made.

Use Case :	Process notification
	After waiter(employer) brought all courses to the customer , that should be confirmed by putting information to the system. System saves confirmation

9.9 Design class diagram







9.10 Relational model

