# Computer Science Program

2th Semester 2015

**Semester Project**

**Implementation of IT system to “Kad Zebras Deg” Bar**

|  |  |
| --- | --- |
| **The report has been prepared by :** | **Supervisor:** |
| Ionut Danci-Bumbea |  |
| Janis Indrans | Henrik Munk Hvarregaard |
| Leonard Hirja | Date of submission: |
| Nikita Cimaskevic | 02-06-2015 |
| Richard Retezi |  |



# Table of Content

[Computer Science Program 1](#_Toc420407327)

[Table of Content 2](#_Toc420407328)

[1. Introduction 4](#_Toc420407329)

[Preliminary investigation 4](#_Toc420407330)

[Inception phase 4](#_Toc420407331)

[Elaboration phase 5](#_Toc420407332)

[Construction phase 5](#_Toc420407333)

[2. Preliminary investigation 5](#_Toc420407334)

[2.1 Company Description 5](#_Toc420407335)

[3 Inception phase 6](#_Toc420407336)

[3.1 Business analysis and feasibility study 6](#_Toc420407337)

[Style of management 6](#_Toc420407338)

[Motivation 7](#_Toc420407339)

[Culture 9](#_Toc420407340)

[Problem Statement 10](#_Toc420407341)

[Problem description 10](#_Toc420407342)

[SWOT analysis 11](#_Toc420407343)

[Porters Five Forces 12](#_Toc420407344)

[Stakeholder Analysis 14](#_Toc420407345)

[Mission / vision 15](#_Toc420407346)

[Business case for the system 15](#_Toc420407347)

[Payback projection 16](#_Toc420407348)

[Business case conclusion 17](#_Toc420407349)

[Activity diagram on business level 18](#_Toc420407350)

[Event tables 18](#_Toc420407351)

[3.2 Functional Requirements 19](#_Toc420407352)

[Use case diagram 19](#_Toc420407353)

[Brief use cases 21](#_Toc420407354)

[Mock-ups 21](#_Toc420407355)

[4 Elaboration phase 22](#_Toc420407356)

[4.1 SCRUM 22](#_Toc420407357)

[4.2 Use case prioritization 23](#_Toc420407358)

[4.3 First sprint 24](#_Toc420407359)

[Fully dressed use case 24](#_Toc420407360)

[Candidate class 26](#_Toc420407361)

[Domain model 26](#_Toc420407362)

[System sequence diagram 30](#_Toc420407363)

[Operation contracts 30](#_Toc420407364)

[Interaction diagram 31](#_Toc420407365)

[Design class diagram 32](#_Toc420407366)

[Code 33](#_Toc420407367)

[Test 34](#_Toc420407368)

[5 Construction phase 35](#_Toc420407369)

[5.1 Second sprint 35](#_Toc420407370)

[Data base architecture 35](#_Toc420407371)

[Relational model 36](#_Toc420407372)

[Grasp patterns 36](#_Toc420407373)

[6 Group Evaluation 36](#_Toc420407374)

[7 Conclusion 36](#_Toc420407375)

[8 Appendices 36](#_Toc420407376)

# 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.

# 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 the capital city of Latvia in Riga. It’s been founded in 2014 by Renars Bents. He is the only owner of the company and he has 6 employees. One cashier, 2 waiters / bartenders, 1 chef and 2 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 witch 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 a 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 indicates 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 coasted 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 is 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 where 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 a 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 rises 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 disatisfied 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 raward.

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

Responisbility 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 experince. Young staff also might show better performance in busiy 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 resistence against the new system and the staff will share with ideas in it’s 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 analyse external and internal factors which affects achieving the goal that in turn should be specified for the business.

“Kad zebras deg” bar / restaurant

|  |  |
| --- | --- |
| **Strengths**   * Popular among young adults * New and fresh * Unique design | **Weaknesses**   * Unexperienced management * The lack of advertising * The lack of “brand loyalty” * Limited budget * The lack of an IT system |
| **Opportunities**   * Expansion * Hiring more experienced staff * Demand for organizing bigger and better events | **Threats**   * Competition in the city * Economic situation (Crisis) * The threat losing in a price competition |

Source: Self-made

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 it’s 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 propriety and transform it into a bar. Most of the client could take advantage of this. Since is very hard to predict new entrants is it better to focus on creating brand and a very high stats 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 entrants can be a threat. Existing rivals could also improve and the competition can become bigger is 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 influent 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 they appearance. The only solution is to race with them and stays 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 its 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 staff 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 4 computers in order to meet the minimum requirements for the system. 1 computer in the kitchen, 1 for the waiters, 1 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 that 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 data base 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 where examined based on the average wages and prices in Latvia [Trading economics], [X Net].

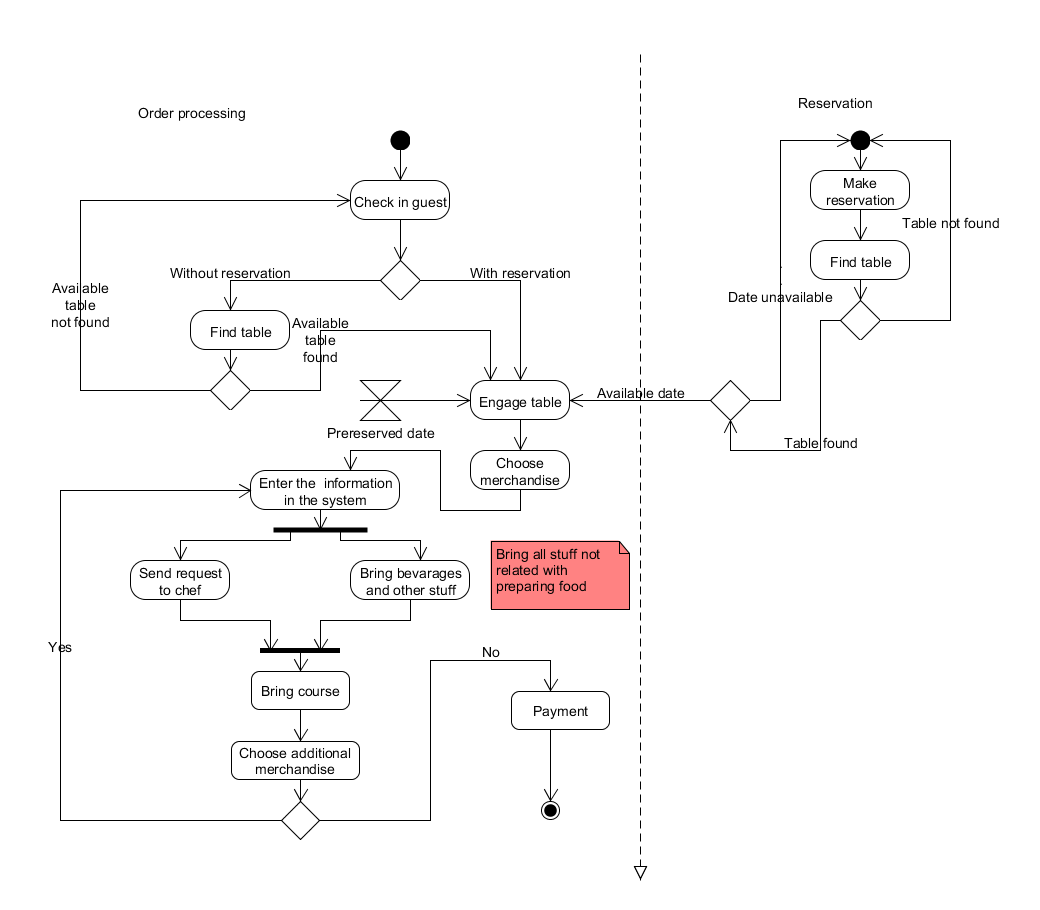
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Item** | **Year 1** | **Year 2** | **Year 3** | **Year 4** | **Year 5** |
| **Hardware purchase** | 2540 |  |  |  |  |
| **Hardware maintenance** | 1000 | 1000 | 1000 | 1000 | 1000 |
| **Software purchase** | 150 |  |  |  |  |
| **Software 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 costs** | **7310** | **17160** | **27010** | **36860** | **46710** |

### 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.



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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Event (AS IS)** | **Activity** | **Step in activity** | **Actor** |
| Order is beeing 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 the order sheet to chef | Employee / Manager |
| Table is beeing 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 the reservation list. | Employee / Manager |

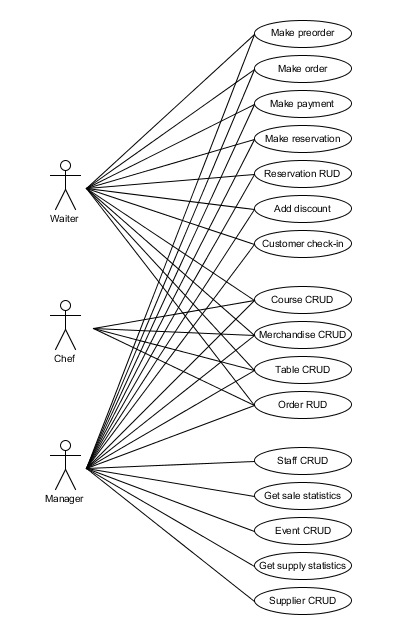
|  |  |  |  |
| --- | --- | --- | --- |
| **Event (TO BE)** | **Use Case** | **Step in use case** | **Actor** |
| Customer wants to make an order | Make order | \*Customer sits at the table and chooses courses \*Employee/Manager writes down information in the notepad \*Employee/Manager inputs customers desired course or product in the system \*Employee/Manager repeats the action until all desired courses or products are in the order \*Employee/Manager complete the order | Employee / Manager |
| 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 | 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.



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.

### Brief use cases

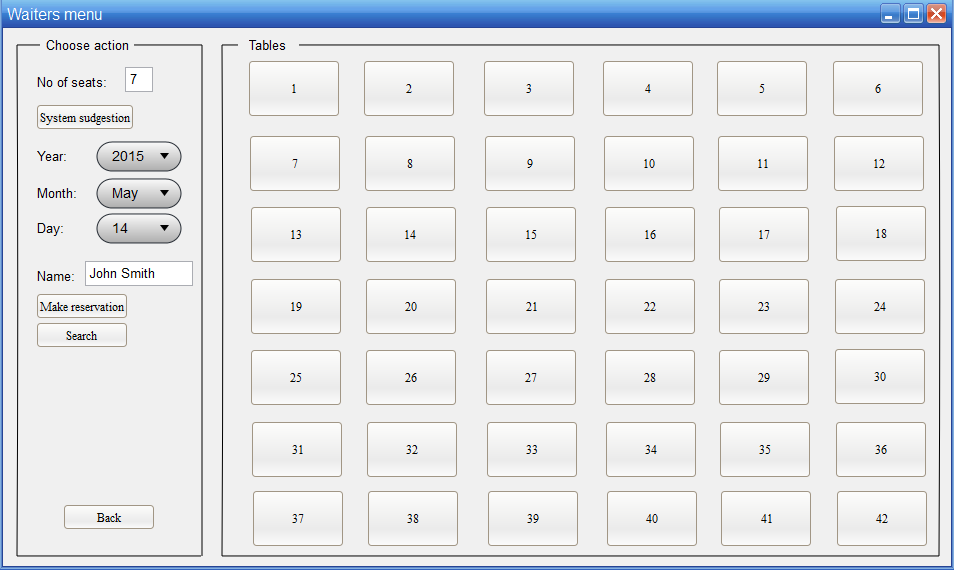
The purpose of brief use case description is shortly outline and define what system does in response of actors actions.

|  |  |
| --- | --- |
| **Use Case :** | Reservation CRUD |
|  | Employee and manager(actors) uses the system and able to do :   * Create reservation * Find reservation * Update reservation * Delete reservation   To create the reservation actors register it in the system and save it. System creates and saves the course.  To find reservation actors put information in the system. The system finds reservation .  To update reservation actors put information. System finds course and actors update it.  To delete reservation system finds it and actors delete it. |

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.



In the mock-up above is shown waiters menu where waiter can make reservation and assign guests to the appropriate table 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 need 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, organised 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 deals. We consider each iteration as sprint, product backlog as use cases and etc.



## 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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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.

|  |  |  |  |
| --- | --- | --- | --- |
| Product backlog for Reservation system | | | |
| Item | **Priority** | **Estimate** | **Planned** |
| *Make order* | 1 | 5 team days | Sprint 1,Elaboration |
| *Make reservation* | 2 | 4 team days | Sprint 1, Elaboration |
| *Create merchandise* | 3 | 3 team days | Sprint 2, Elaboration |
| *Find reservation* | 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 purposes of sprint were 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 totally described use case step by step, it was supposed to be coherent as well as specific.

|  |  |  |
| --- | --- | --- |
| **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, notification sent to chef | |
| **Frequency** | 70 times a day | |
| **Main success scenario (flow of events)** | **Actor(action)** | **System(response)** |
|
| 1.Guests come to the restaurant |  |
|
| 2.Waiter(employee)/Manager makes check-in | 3.System searching reservation and assign table |
|
| 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 |
| 11.When course is made cook press complete | 12. System informs waiter when appropriate course is made |
| 13.Waiter serve the table with the course |  |
|  | 14.Waiter asks if guests would like to order something more and repeats steps from 6 to 14 until guests are finished |  |
|  | 15. Waiter makes payment | 16.System calculate sum and prints receipt |
| **Alternative flows** | 2a. Guest come without reservation   1.Waiter makes reservation | |
|
| 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 | |

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

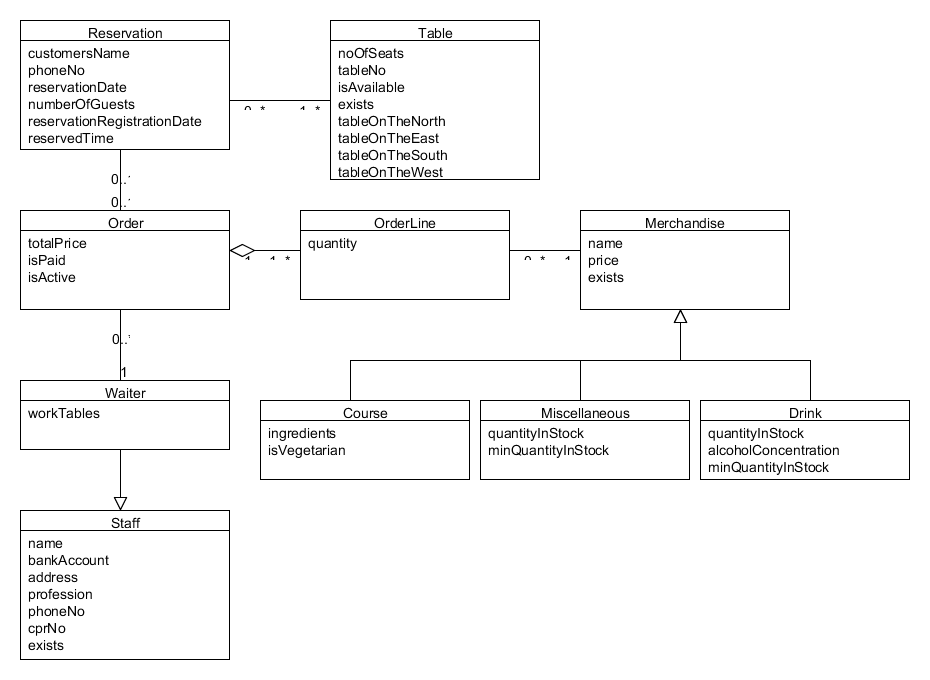
### Candidate class

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 represent 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.

|  |  |  |  |
| --- | --- | --- | --- |
| Merchandise | Table | Receipt | Reservation |
| Chef | Payment | Miscellaneous | CheckIn |
| Guest | Waiter | Course |  |
| Notification | Order | Drink |  |

### Domain model

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



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 can’t 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 can’t 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

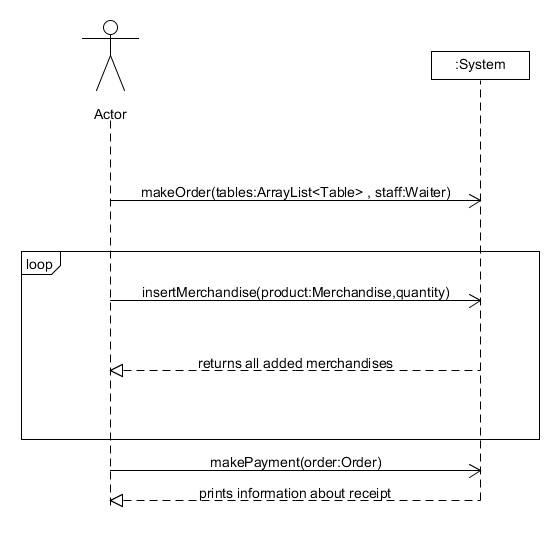
Analysis of the business workflow and its requirements for the system gave this project a good overview of the structure and functionality. This led to domain model as it is shown and described in this section. This is the core plan for the classes and their fields for the code and gave all the necessary 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 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 makes payment with specified order it also means that order was finished and system prints the receipt.



### 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.

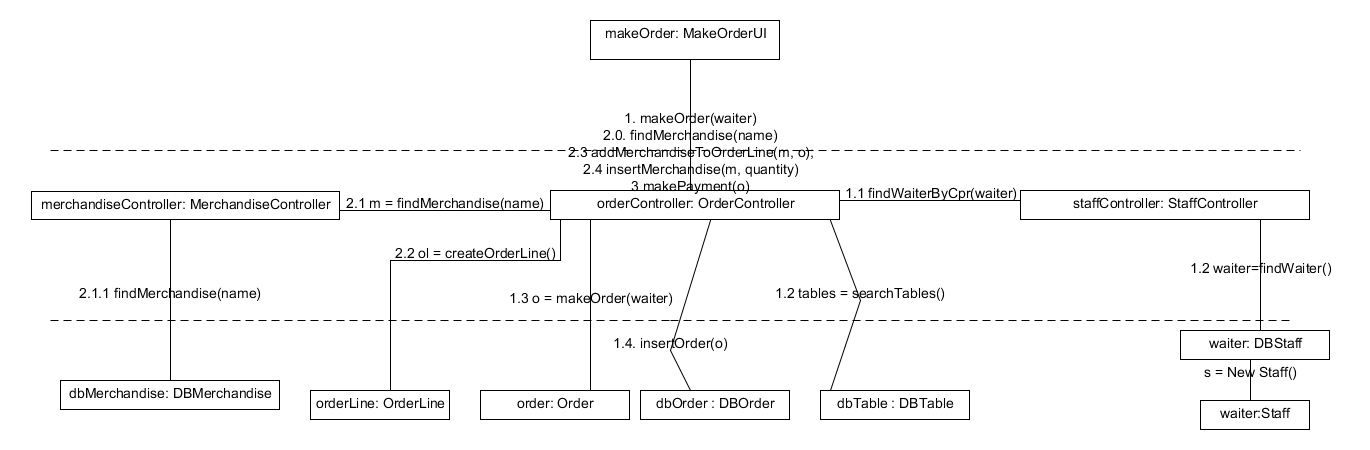
|  |  |
| --- | --- |
| Operation: | makeOrder(table:Table, waiter:Stuff) |
| Use cases: | Make order |
| Precondition: | Table is created, Stuff is created |
| Post conditions: | Order object is created and accosiated with table |
| Stuff object associated with the order object |

|  |  |
| --- | --- |
| Operation: | makeReservation(customersName,phoneNo,date,preorder) |
| 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 |
|

We described only most important methods in our use case. The table 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 classes and the layers.



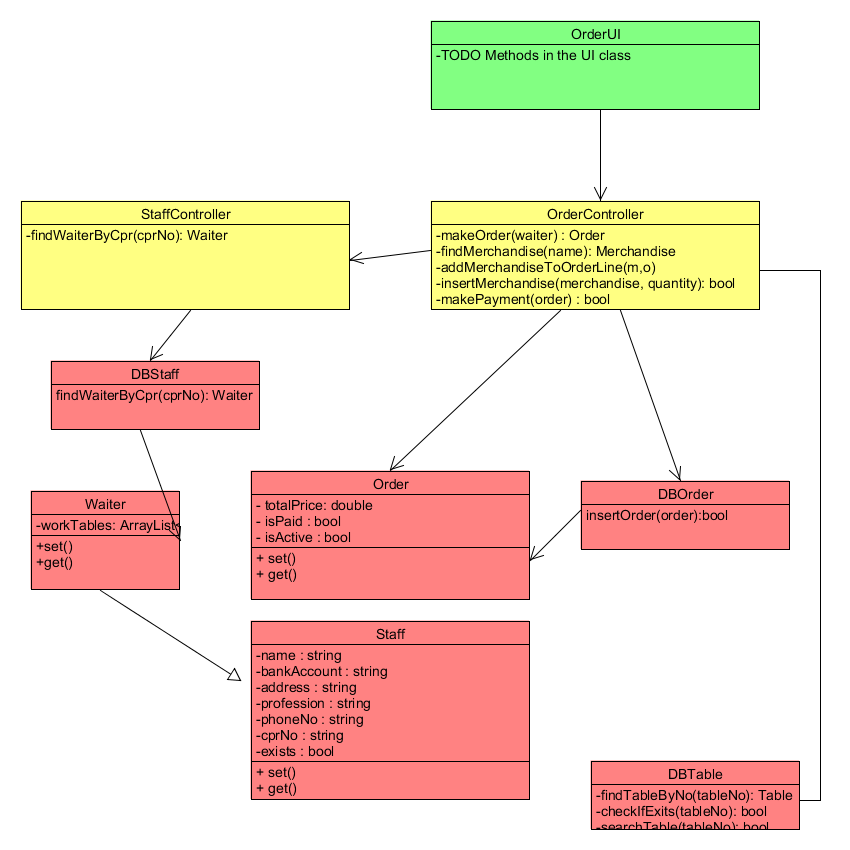
In all the cases the order of procedure, the different tasks start from the user interface (generally by an input) and the subtasks are in the control and model layer. In the make order use case the waiter starts new order, system finds waiter by cpr number in DBStaff(Database)via staff controller and searches for free tables in DBTable(Database). Through the controller system creates order instance and inserts it in database.

In order to add merchandises to the order object waiter inputs name of the merchandise and it should be retrieved from database afterwards. For adding merchandises system creates orderLine object and adds there merchandises with quantity.

The last step is makePayment with specified order. System computes total.

### 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 objects.



Design class diagram shown above describes one use case make order. Full design class diagram can be found in the appendix.

### Code

……………..

### 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

|  |  |
| --- | --- |
| **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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |
| 1. 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

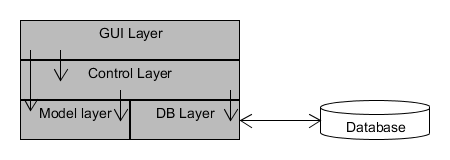
After implementing main use case through one sprint in the Elaboration phase we had a preliminary implemented program and basic architecture. To make other use cases and features of the system we had to start another sprint according to construction phase.

## 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

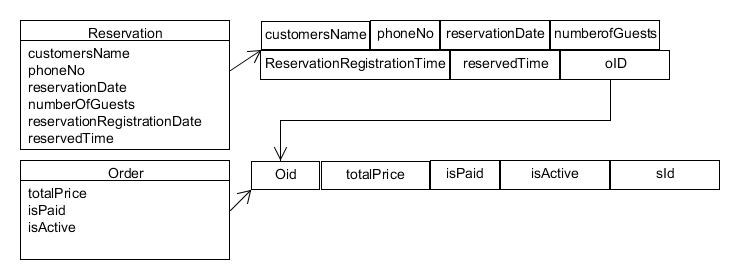
In order to store all data in the database in our software we have applied Data access layer besides GUI layer and Business Logic Layer(Control Layer).



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.



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 represent 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 Single tone has been employed from GoF collection.

# 6 Group Evaluation

According to the Insight test, our team is formed by a considerate mixture of the color green and yellow but also green and blue. After working on the first semester’s final project and the second semester’s final project (with the same team members), this combination of colors in team members proved to be an efficient one, both in work process and communication between team members.

<table with color for each member>

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. Even though we split the tasks, everyone had a contribution to everything, resulting in fully understanding of the project for every team member. Every member of the group got involved from the outset, proving strong commitment and involvement in every phase of the project.

Preparation was made in a professional way, before starting to work on everything, we made sure that everyone in the team had a full understanding on the project case and all the requirements. As we moved forward with the project, everything went smoothly. We did mistakes along the project’s evolution, but we managed to find them, understand them, and come up with a solution for it, extending our knowledge and experience for future projects.

Communication inside the group was excellent. If anyone had a question about anything, other team members tried to explain to him and if nobody knew the answer, everyone started conducting researches on that matter. Because of that, the group had a wide area of knowledge and through debates, the common and final answer was always found. Everyone in the group was open-minded, as a result everyone felt free to share his point of view or his ideas with the rest of the members. An important policy inside the group was to relax from time to time, the atmosphere inside the group being relaxing and positive. We all contributed to the well-being of the members as a group, lowering the tensions with some funny moments, like jokes and music listening.

# 7 Conclusion

# 8 Appendices