DOKUZ EYLÜL UNIVERSITY ENGINEERING FACULTY DEPARTMENT OF COMPUTER ENGINEERING

CME 2210

Object Oriented Analysis and Design

CAFE MANAGEMENT SYSTEM

by

2019510086 Yavuz Yılmaz

2019510047 Dilara Karacan

CHAPTER ONE

INTRODUCTION

1.1 What the Problem is

As far as we have examined and observed, most cafes today use outdated and useless systems. Therefore, businesses have problems in daily life.Cafes have difficulties especially in terms of income-expenditure calculation and employee productivity. We thought that such a system was a necessity for the bosses and employees to work happier and more comfortablyWe know that with better planning and better software intelligence, every problem has a better solution.We have prepared such a project to solve all these problems we see.

1.2Goals for the Project

The aim of the project is to create a new generation café management system and to enable the businesses in the sector to work more efficiently. Many businesses today are experiencing difficulties in this regard but it is ignored.As a team looking from the outside, we see this problem and set out to find a solution for this situation. This system we will create will set an example in the sector.

1.3Stakeholders

Several different types of stakeholders can be noted when it comes to our software. The first thing that comes to our mind is the part that demands this system, namely bosses and employees. We noted critical details such as what the company wanted and how it wanted, and we steered the project on how it would benefit both them and their customers.

We want to instantly inform the bosses about their employees and the efficiency of the cafe.

Thus, we aim for bosses to direct their employees more efficiently.

In addition, we want the bosses and employees to realize how efficient and useful the system we will create is. In the simplest terms, we want to make sure that businesses using our software will stay one step ahead.

1.4 Motivation

As we explained above, the main purpose and motivation of the project is to provide profit and to keep boss and employee satisfaction at the highest level. We have made it our duty to embark on this path for cafes and employees who need a better system. Our systems are flexible and user-friendly, with the goals and observations of our team members. Considering all these, we know that the system we have created will meet a great need.

1.5 Process Flow Preview

For our process flow, we plan on taking the iterative route, as we find communication essential throughout the development process. We planned the project in the most effective way. We tried to pass the planning phase as efficiently as possible. However, no matter how good the planning phase passed, we encountered problems that we did not notice in the planning part during the design and creation phase. We overcame these problems by reviewing and adjusting our plans.We have created a software that can be used as continuously and for as long as possible.

As a result, we came up with the closest project to perfection, thanks to more than enough analysis and thinking at every stage.

CHAPTER TWO

REQUIREMENTS

2.1 PLAN FOR REQUIREMENTS

ENGINEERING INCEPTION TASK

The goal for the beginning is to clarify the business rationale that stakeholders need. We want to analyze the cafe operation to understand how often the software will be used. For example, menu change, personnel and customer circulation.Having a command of the process is important for the project flow. That's why we attach importance to this stage.Perspective will ensure the flow of our project.

We learn the expectations and perspectives of employees, bosses and customers, and create functions and features accordingly. We thought of some basic questions for this;

*What are the basic functions?

*Can you describe your customer base to us?

*What are your expectations from this project?

*Can you show the usage area of the project?

*Are there any other people you suggest we ask these questions to?

*Is there anything else you want to add?

ELICITATION TASK:

Our aim here is to identify the problem and reach its root. We planned to achieve this by talking among ourselves and with our stakeholders. We exchange ideas by meeting with the software team and stakeholders. our plan is how to combine the need in the sector with software how to adapt to the sector. we now have our list of requirements with these stages and stakeholder and actor profiles. technical descriptions and usage requirements are explained in detail.

ELABORATION TASK:

A grouping process is made with the data we collected in the first two parts, so that the project becomes open to task sharing and understanding. It supported by models showing all conditions. piece by piece scenario of each moment should be considered to reduce errors. for example how waiters will interact with the software. Which functions the necessary functions will be nested

NEGOTIATION TASK:

Team and stakeholders exchange ideas on any problem that is not clear and needs a solution. If there are too many requirements, the issues are ranked in order of importance. Of course, these priorities include saving time and money. non-essential problems can be overlooked by time and money

SPECIFICATION TASK:

We plan to jot down everything we will do during this task. We plan to identify all the necessary information and software in detail and focus on these issues.

Then we will determine all the qualifications and privileges that the project should have before starting the project. There will be some requirements that we will realize during the project. We will review all these requirements and create a roadmap accordingly.

VALIDATION TASK:

:

In this part, almost many questions about the project should have been answered. It should be clear what many validation projects will do, what functions they will have, and for whom they are created. All stakeholders should be identified and clearly articulated. All the requirements should be congruent with the overall objectives and can be easily understood. Any hard-to-understand phrases should be rewritten and discussed over with again with the stakeholders and team members.

REQUIREMENTS MANAGEMENT

Some requirements that may occur during the project phase should be adequately discussed and what needs to be done should be determined. Changes and improvements to be made should be shared with both the software team and the customer. All stakeholders should have sufficient knowledge and be familiar with the changes in the product.

FUNCTIONAL NONFUNCTIONAL REQUIREMENTS

CLASS NAME:

HUMAN

- getName()
- getSurname()
- getAge()
- getPhone()
- getAddress()
- getSex()
- setName()
- setSurname()
- setAge()
- setPhone()
- setAddress()
- setSex()

BOSS: -extend human

- editMenu()
- ShowIncomeOutcome()
  -return integer
- setShareAmount()
- hireWaiter()
  -get parameter(waiter)
- fireWaiter()
  -get parameter(waiter)
- cancelOrder()
- -get parameter(order)

- messagetoWaiter()
  -get parameter(waiter)
  -return string message
- enterExpense()
- showIncomeExpense()
  -return int
- changeNumberofTable()
  -parameter number of table
- changeMenu()
  get parameter(menu)
- startDay()
- finishDay()

WAITER:-extends human class

- Createorder
  -return order type
- Show_table_information
- Show_message
- getSalary
- setSalary
- cancelOrder
  -get parameter order type
- takeOrder
- getPaid
  -get parameter table type

- getSalary()
- setSalary()

BARISTA:extends human class

- getSalary()
- setSalary()

CUSTOMER:

- Fill_out_survey

TABLE:

- setNumberofTable()
- getNumberofTable()
- isAvailableTable(Boolean,returns true or false)
- calculateBill(return int)
- getIsFull()
- setIsFull()
- getPaid()

LOGIN:

- verification()

MENU:

- getProducts()
- setProducts()
- showMenu()

DATE:

- getDay():parameters day(int
- getMonth()
- getYear()
- setYear()
- getExpense()
- getIncome()

- setDay()
- setYear()
- setExpense()
- setIncome()

SURVEY:

- Edit_survey_questions()
  -get parameter survey information

- Reader()

Non-functional process

- invoice the customer
- print out the bill
- The response time must be less than 1 second
- The server must be available 24 hours a day

USECASE01:

Add Order

Use Case: Order Description: This use case allows cafeteria's waiter to add new order.Actor: waiter

Actor Action	system response
1)login to the system	2)verify the login Name and password 3)redirect main menu
4)choose table	
	5)redirect table user interface
6)select table	
7)click order	
	8)Validate order inform.
9)enter order information	
10)click confirm button	
	11)send successful message 12)redirect back

Alternative flows (other scenario)

Step 2: If system gets the incorrect login name and password.

System prompts for error message "incorrect login name and password!"

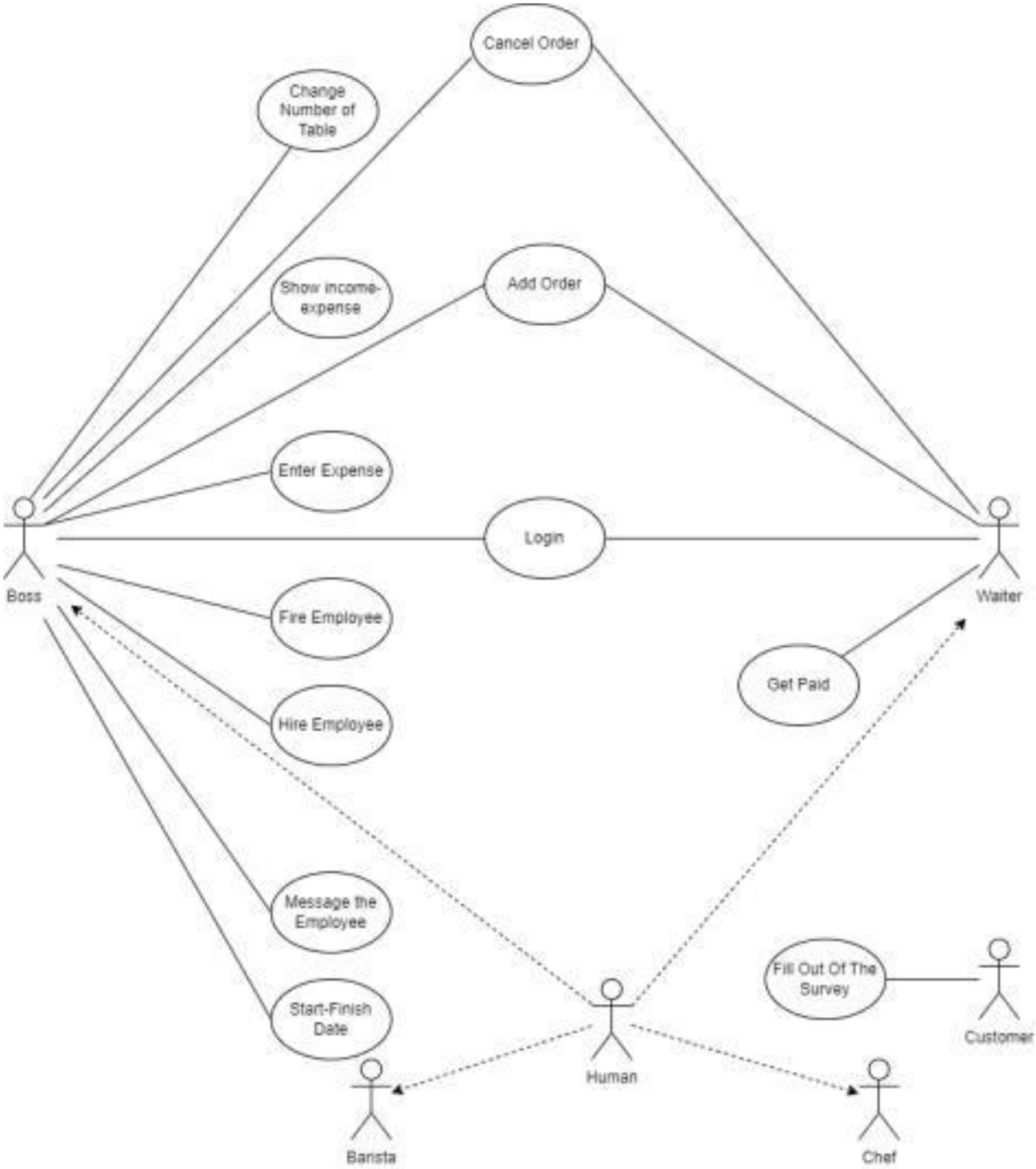
Step 6: If selected table is assigned

Order button will be enabled.

Step 9: If system found invalid format

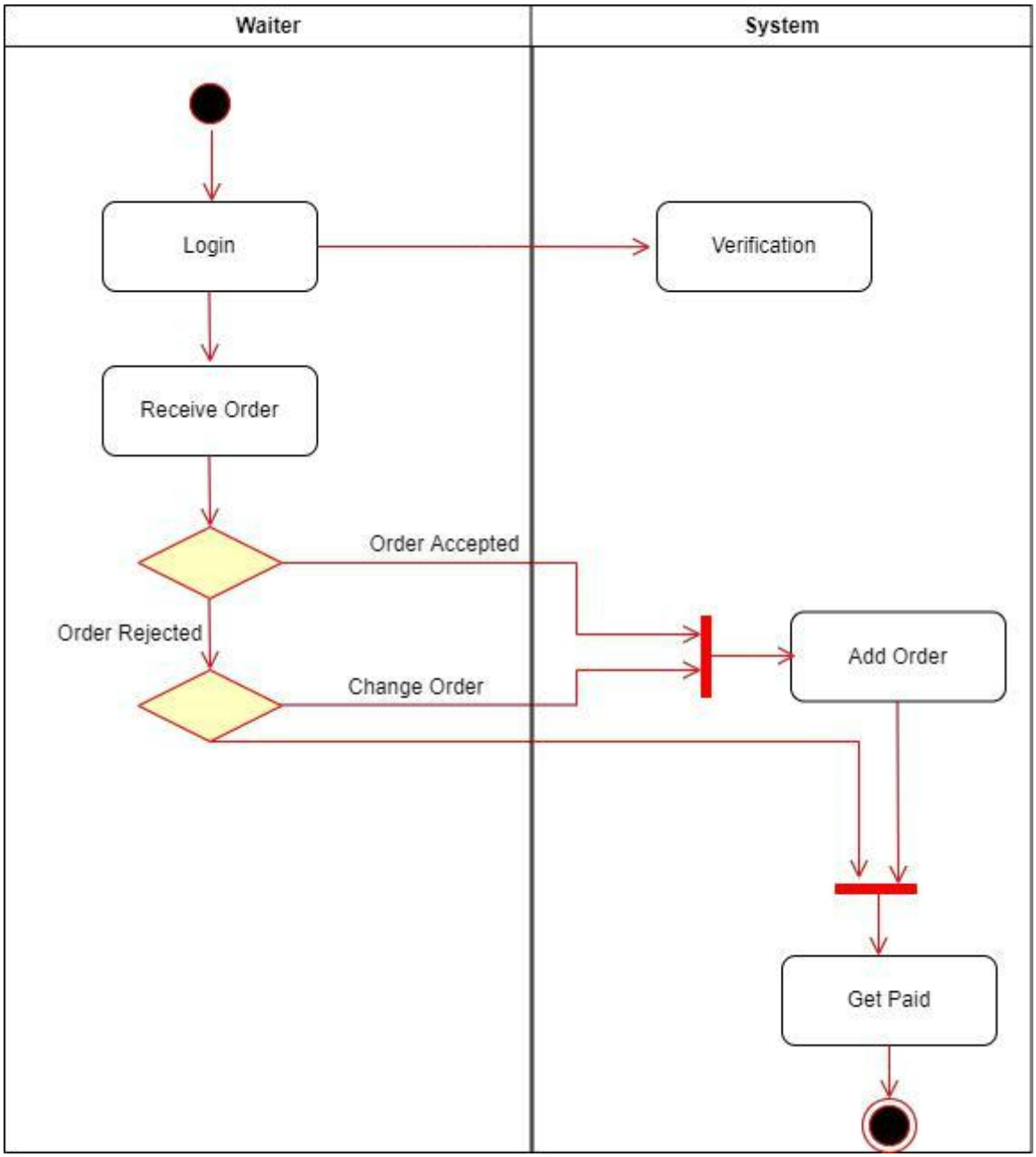
information-System display error with message

Use Case Diagram



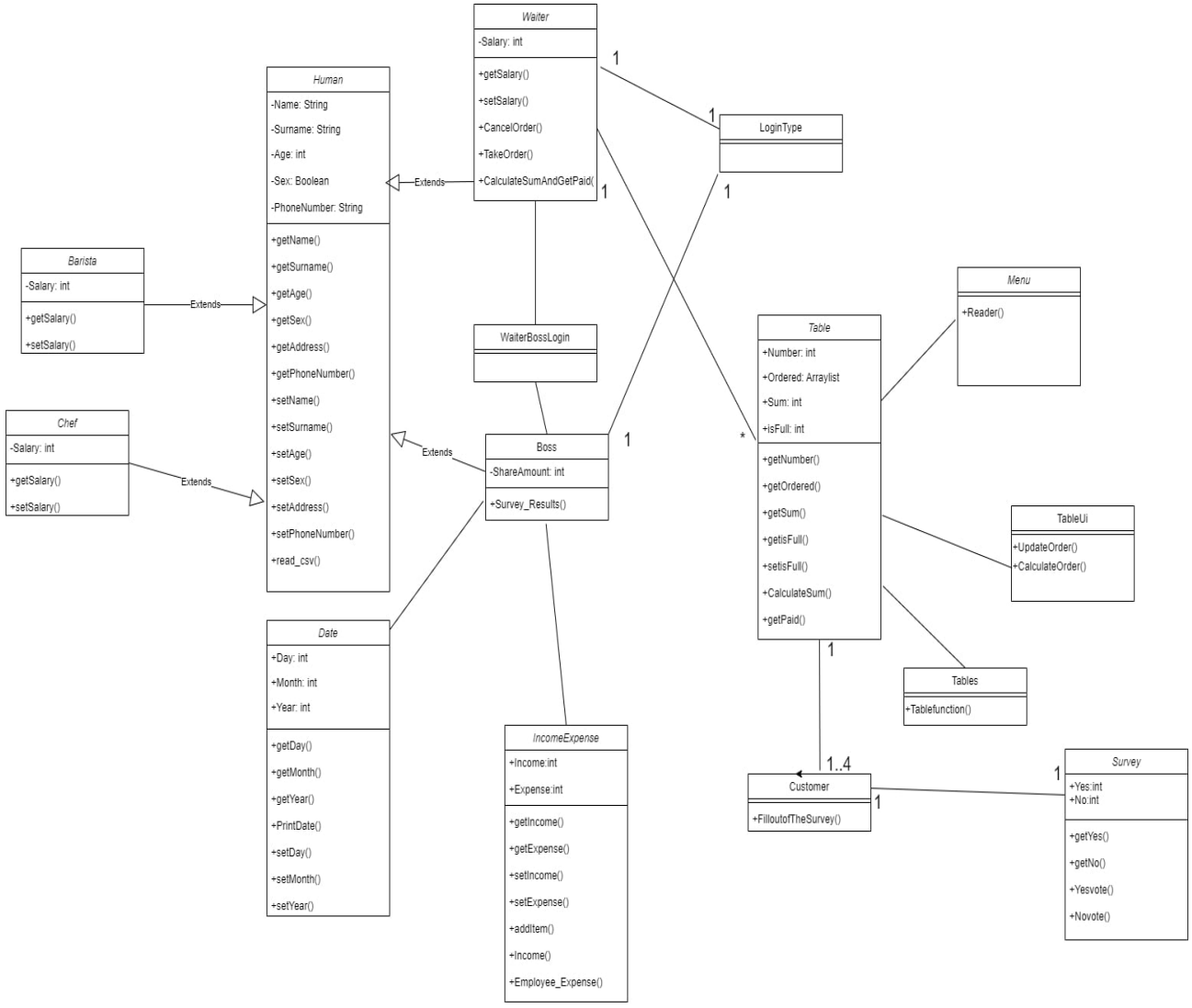
Use Case Diagram:As seen in the use case diagram, we have shown the actors and operations at hand and modeled which actor can use which function of the system.

Activity Diagram



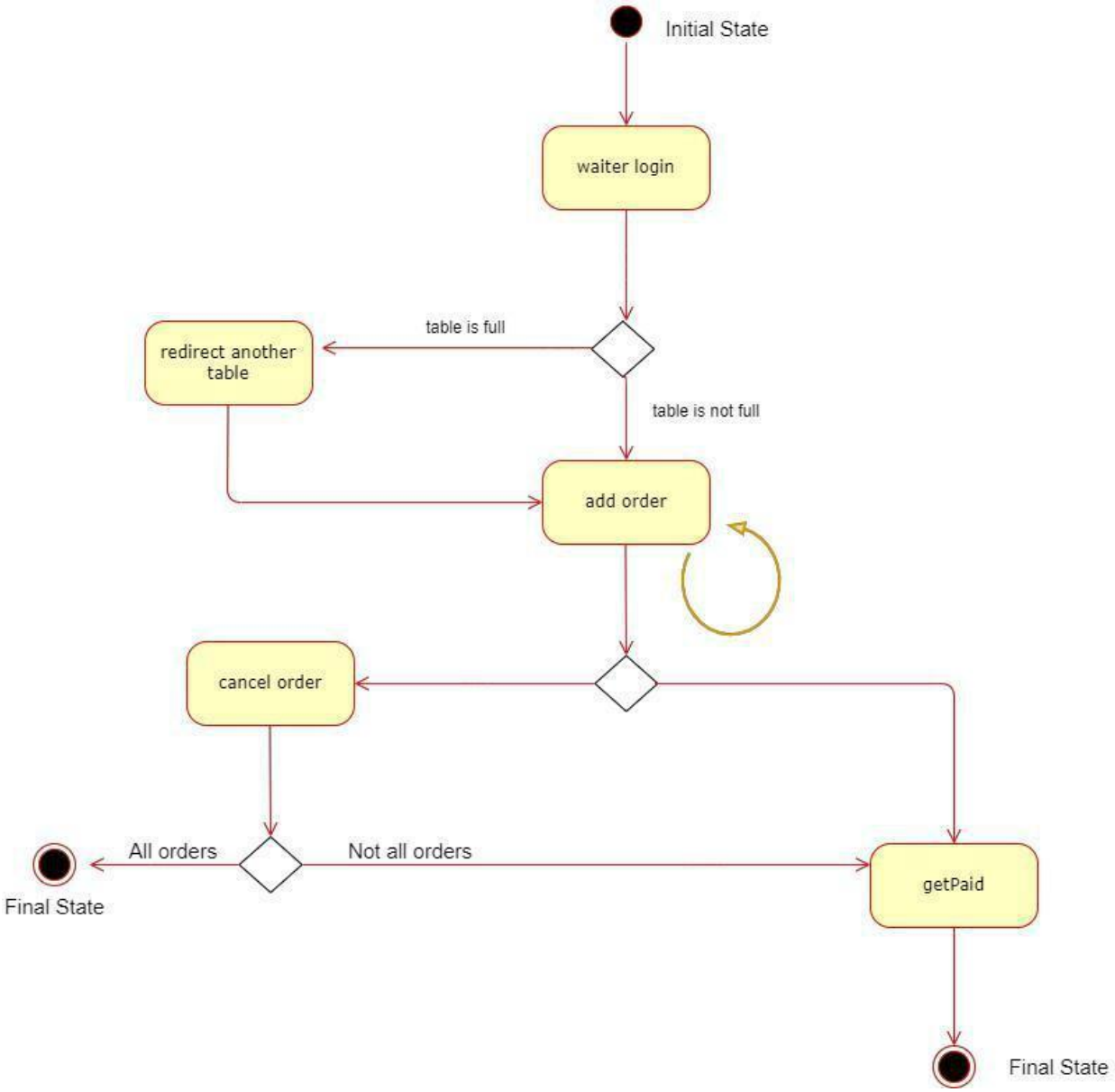
Activity Diagram:We modeled the order taking function and possible situations in the activity diagram. We showed the working logic of the system with simple actions and decisions.

Class Diagram



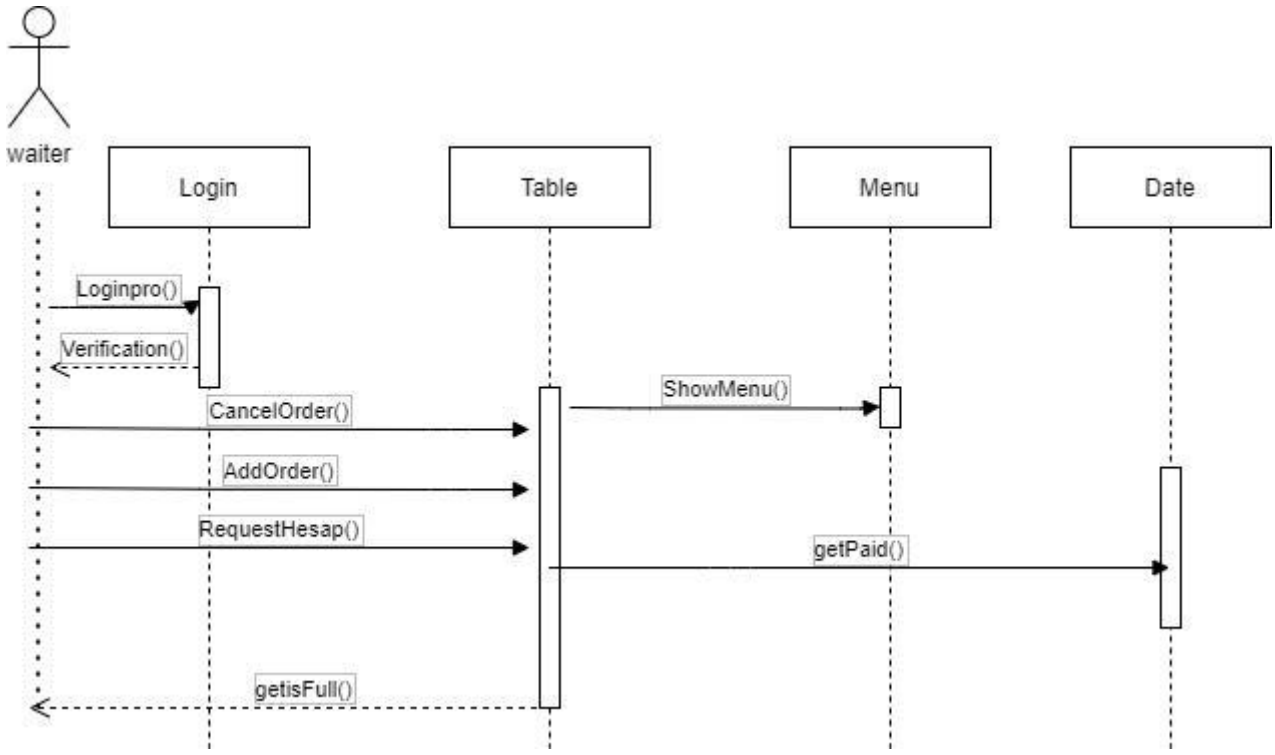
Class Diagram :We showed all our classes in the class diagram and talked about the relationships between them in the simplest way. When we look at it in general, it is clear that there is a cafe management system. A system that goes through different types of employees over bosses. The order taking part is recorded on the tables and transactions are made on the tables.

State Diagram



State Diagram :We have shown the basic function in the state diagram. We have listed the situations that can occur and the situations that can be done while taking an order, and we have shown them on the diagram. We have clearly modeled the situations that can occur.

Sequence Diagram



Sequence Diagram :In the sequence diagram, we modeled a simple order taking application. Our actor here is the waiter. The waiter communicates with the customer live and adds the necessary information to the system. First of all, of course, a login process must be made, and then, according to the customer's request, transactions such as taking an order, canceling an order or requesting an account can be performed.

Testing

In the testing part, we tried to run the code as different as possible by thinking of ourselves as users and to find our mistakes and shortcomings.We also showed some people who have nothing to do with this business to use our code and structure.In the testing part, we encountered problems that we never expected. We received very different reactions and returns from users. In addition, at the end of this test part, we received information from the Users about the missing and excess parts, and we developed and changed our project accordingly.At the end of the test part, we reconsidered our project and added the necessary changes to our project. We realized how useful the testing part really is. The test part is an essential part for a project. If there was no test part, we would probably face very different problems.

Conclusion

5.1 The Problem and Solution

As a result, we have prepared a nice cafe management system. As a problem, we chose for ourselves that the cafes do not have sufficient systems. Moving forward in the context of this problem, we solved the problem at hand with creative ideas and original solutions.

We had a good research process around the problem at hand. By talking to the cafe owners and waiters, we got information about the parts that they found missing and excessive problems they saw in the systems.

After the research process, we made an introduction to the design and analysis part, and this was the longest part. The test part of the work, where we spent this part very well and in detail, was much more enjoyable.

We got most of the returns and actions that we expect from the user. After going through a good process, we finally came up with a project that would really satisfy all of us.

We had an enjoyable project process by going step by step throughout the process. We made sure that the users would be absolutely satisfied and we decided we were done.

5.2 The Team and the SE Process

The Software Engineering process we used was the spiral method. In this method, we start in the middle of the model, and spiral outward, allowing all departments working on the software to be an active part of every aspect of the engineering. Each individual department will be able to work and test during the concept development, system development, system enhancement, and system maintenance phases of development.

5.3 Engagement of Umbrella Activities

Four of the main Umbrella activities we used were as follows:

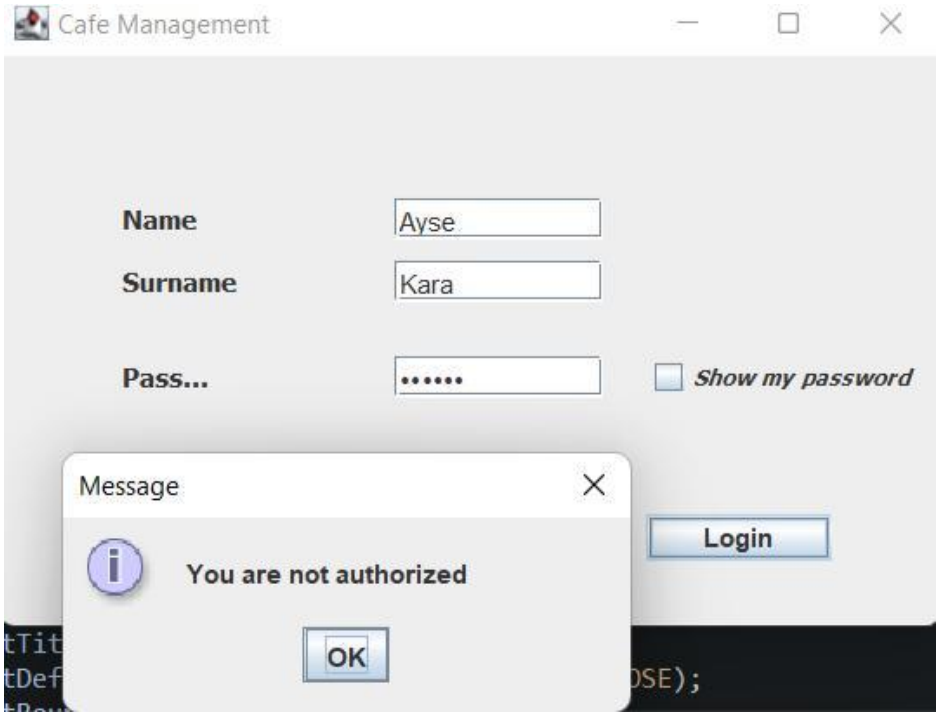
1. Software Project Management - Which was used to lead the project and ensure that the project was controlled, monitored, and on schedule.

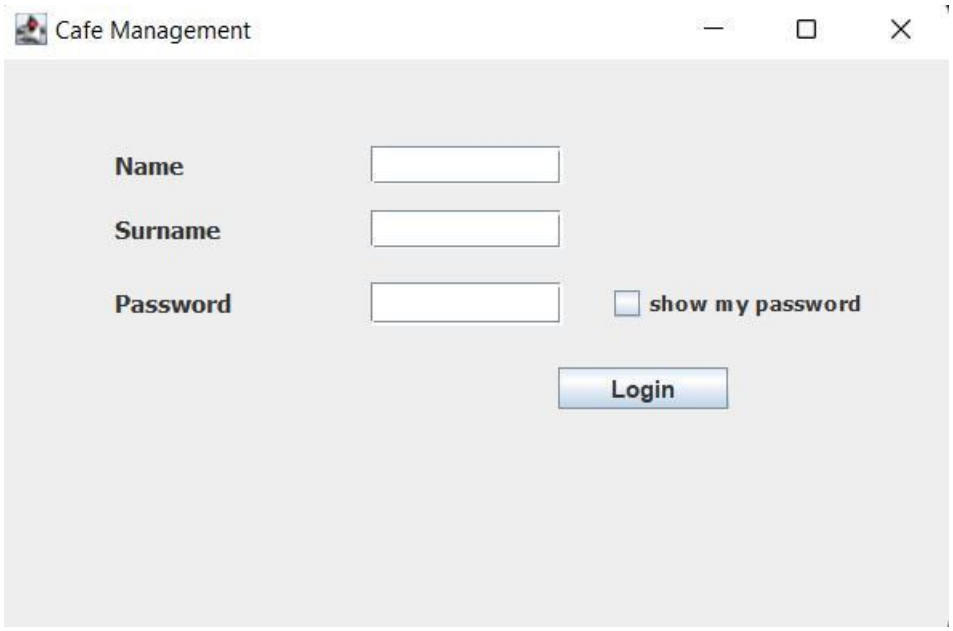
2. Formal technical Reviews - This activity was essentially implemented for peer review. Having new and fresh eyes to view code and ensure that everything met the requirements.

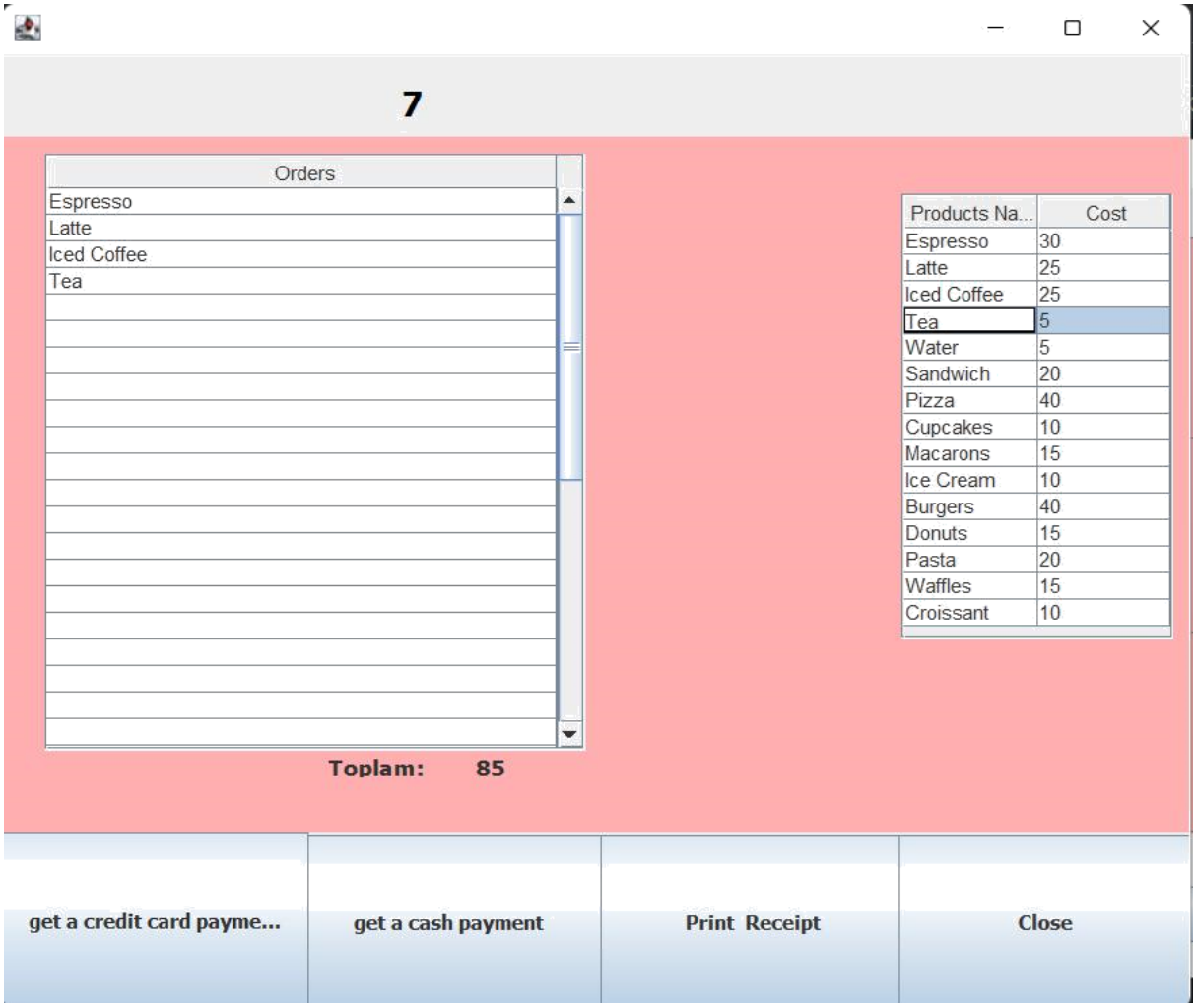
3. Reusability Management - This activity was used to help us create flexible and generic assets that may be reused for future projects or for this project in other regions. This would cut down on cost and help with consistency.

4. Risk Management - This activity was used to assess and identify potential risks with creating the software such as assuring that not too much money be spent in assets on the Project

Project UI


Cafe Management
Name
Ayse
Surname
Kara
Pass...
Show my password
Message
You are not authorized
OK
Login


Cafe Management
Name
Surname
Password
show my password
Login



References

https://online.deu.edu.tr/portal/site/57ad002a-8a6f-4e5a-b4a2-6b5f850b027f/tool/b7d98892-78a1-4263-847e-64786489e35a