

# I. Preface: Assessment sheets

## Assessment OO analysis – case study COS 2015-2016

Fill in (0, 1 or 2) de yellow zones in the table below

0= dit not cooperate in that assignment

1= passive cooperation (no active cooperation in discussions, modelling). Only read solutions of other members.

2= active cooperation in discussions, modelling, use of Visual Paradigm

Student names (family name + first name)

Group: 2TX1

ST1: name student 1: Ruth Reynders

ST2: name student 2: Cathlene Van Moorlegghem

ST3: name student 3: Liesbeth Vandevenne

ST4: name student 4: .....

ST5: name student 5: .....

|                                      |  | Max score | Score | ST1       | ST2       | ST3       | ST4 | ST5 |
|--------------------------------------|--|-----------|-------|-----------|-----------|-----------|-----|-----|
| <b>Assignment 1</b>                  | Scope/Problem/business analysis        | 5         |       | 2         | 2         | 2         |     |     |
| <b>Assignment 2</b>                  | Requirement analysis/context           | 5         |       | 2         | 2         | 2         |     |     |
| <b>Assignment 3</b>                  | Logical analysis/Use case description  | 5         |       | 2         | 2         | 2         |     |     |
| <b>Assignment 4</b>                  | Technical analysis/Use caserealisation | 5         |       | 2         | 2         | 2         |     |     |
| <b>Total group score</b>             |  | <b>20</b> |       |           |           |           |     |     |
| <b>% time allocation per student</b> |  |           |       | <b>33</b> | <b>33</b> | <b>33</b> |     |     |
| <b>Student score</b>                 |  | <b>20</b> |       |           |           |           |     |     |

Space for lector!!!

Remarks

|  |
|--|
| <b>Assignment1</b>   |
| <b>Finishing date assignment:</b>  |
| <b>Tasks performed</b>   |
| <b>ST1: Ruth Reynders</b>  |
| <ul style="list-style-type: none"> <li>-Helped with composing the product Vision board.</li> <li>-Helped with composing the problem statement matrix.</li> <li>-Wrote problem description based on pieces.</li> <li>-Helped with composing the business use case diagram.</li> <li>-Helped with composing the overall activity diagram.</li> <li>-Drew this diagram in visual visual paradigm.</li> <li>-Helped with composing the business class diagram.</li> <li>-Drew this diagram in visual visual paradigm.</li> <li>-Review pieces and diagrams of other students</li> </ul>  |
| <b>ST2: Cathlene Van Moorleghe</b>   |
| <ul style="list-style-type: none"> <li>-Helped with composing the product Vision board.</li> <li>-Helped with composing the problem statement matrix.</li> <li>-Wrote problem description based on pieces.</li> <li>-Helped with composing the business use case diagram.</li> <li>-Drew this diagram in visual visual paradigm.</li> <li>-Helped with composing the overall activity diagram.</li> <li>-Helped with composing the business class diagram.</li> <li>-Drew this diagram in visual visual paradigm.</li> <li>-Review pieces and diagrams of other students.</li> </ul> |
| <b>ST3: Liesbeth Vandenvenne</b>   |
| <ul style="list-style-type: none"> <li>-Helped with composing the product Vision board.</li> <li>-Helped with composing the problem statement matrix.</li> <li>-Wrote problem description based on pieces.</li> <li>-Helped with composing the business use case diagram.</li> <li>-Review pieces and diagrams of other student.</li> </ul>  |
| <b>ST4</b>   |
| <b>ST5</b>   |
| <b>Remarks (which parts of the assignment are not finished or partly finished)</b>   |
|  |

|  |
|--|
| <b>Assignment2</b>   |
| <b>Finishing date assignment:</b>  |
| <b>Tasks performed</b>   |
| <b>ST1</b>   |
| <ul style="list-style-type: none"> <li>-Helped with composing the event response list.</li> <li>-Created a system usecase diagram in visual paradigm: main diagram in package diagram and detailed use case diagram.</li> <li>-Review diagram and lists other students</li> </ul>  |
| <b>ST2</b>   |
| <ul style="list-style-type: none"> <li>-Helped with composing the event response list.</li> <li>- Drew the context diagram in visual paradigm.</li> <li>- Composed a description of the data structures.</li> <li>- Review diagram and lists other student</li> </ul>  |
| <b>ST3</b>   |
| <ul style="list-style-type: none"> <li>-Helped with composing the event response list.</li> <li>- made a list of the stakeholders and actors and gave a description of their involvement with the project.</li> <li>-Made a list of relevant business rules.</li> <li>-Made a list of non functional requirements.</li> <li>-Made a ranking for the relevant use cases</li> <li>-Review diagrams other students</li> </ul> |
| <b>ST4</b>   |
|  |
| <b>ST5</b>   |
|  |
| <b>Remarks (which parts of the assignment are not finished or partly finished)</b>   |
|  |

|   |
|---|
| <b>Assignment3</b>  |
| <b>Finishing date assignment:</b>   |
| <b>Tasks performed</b>  |
| <b>ST1</b>  |
| <ul style="list-style-type: none"> <li>-Helped with composing the description of the flow of events.</li> <li>-Discusses , elaborated and designed the user stories.</li> <li>-Drew an activity diagram for the use case description in visual paradigm.</li> <li>- Review diagrams other students</li> </ul> |
| <b>ST2</b>  |
| <ul style="list-style-type: none"> <li>-Helped with composing the description of the flow of events.</li> <li>-Drew a domain class diagram in visual paradigm.</li> <li>- Drew a statechart diagram for the domain class 'Order' in visual paradigm.</li> <li>- Review diagrams others students</li> </ul>    |
| <b>ST3</b>  |
| <ul style="list-style-type: none"> <li>-Helped with composing the description of the flow of events.</li> <li>-Defined the user stories.</li> <li>-Drew a CRUD matrix.</li> <li>- Review diagrams other students</li> </ul>   |
| <b>ST4</b>  |
|   |
| <b>ST5</b>  |
|   |
| <b>Remarks (which parts of the assignment are not finished or partly finished)</b>  |
|   |
|   |

|  |
|--|
| <b>Assignment4</b>   |
| <b>Finishing date assignment:</b>  |
| <b>Tasks performed</b>   |
| <b>ST1</b>   |
| <ul style="list-style-type: none"> <li>- Drew a system sequence diagram for the use case 'order meal from cafeteria ' in visual paradigm.</li> <li>-Drew a MVC sequence diagram in visual paradigm.</li> <li>-Drew a communication diagram in visual paradigm.</li> <li>-Drew ER model in visual paradigm.</li> <li>-Review of other diagrams</li> </ul> |
| <b>ST2</b>   |
| <ul style="list-style-type: none"> <li>-Drew the design class diagram in visual paradigm.</li> <li>-Extensive review of other diagrams</li> </ul>  |
| <b>ST3</b>   |
| <ul style="list-style-type: none"> <li>-Wrote the operation contracts.</li> <li>-Composed written version of ER model.</li> <li>-Review of other diagrams</li> </ul>   |
| <b>ST4</b>   |
|  |
| <b>ST5</b>   |
|  |
| <b>Remarks (which parts of the assignment are not finished or partly finished)</b>   |
|  |

## Inhoud

|      |  |    |
|------|--|----|
| I.   | Preface: Assessment sheets .....   | 1  |
| II.  | Assignment 1: Scope/Problem analysis – business analysis .....                     | 7  |
| 1.   | Product vision .....   | 7  |
| 2.   | Problem statement table .....  | 8  |
| 3.   | Problem descriptions according to PIECES .....                                     | 10 |
| 4.   | Business use case diagram .....  | 12 |
| 5.   | Activity diagram of business use cases.....  | 13 |
| A3   | 13   |    |
| 6.   | Business class diagram .....   | 14 |
| III. | Assignment 2: Requirement analysis/Context .....                                   | 15 |
| 1.   | Event response list .....  | 15 |
| 2.   | Context diagram .....  | 17 |
| 3.   | Data flows.....  | 18 |
| 4.   | Stakeholders / actors .....  | 20 |
| 5.   | Business rules .....   | 21 |
| 6.   | Quality attributes.....  | 23 |
|      | Reliability.....   | 23 |
|      | Maintainability .....  | 24 |
|      | Portability of the system .....  | 24 |
| 7.   | Use cases with ranking .....   | 25 |
| 8.   | Main use cases as package diagram .....  | 27 |
| 9.   | System use case diagram for package “Order meal transactions” .....                | 28 |
| IV.  | Assignment 3: Logical analysis – use case description .....                        | 29 |
| 1.   | Flow of events for use case “order meal from cafeteria” .....                      | 29 |
| 2.   | User stories for use case “order meal from cafeteria” .....                        | 33 |
| 3.   | Activity diagram for use case “order meal from cafeteria” .....                    | 42 |
| 4.   | Domain class diagrams for use “order meal from cafeteria” .....                    | 43 |
| 5.   | CRUD matrix .....  | 44 |
| 6.   | State chart diagram for domain class “Order” .....                                 | 44 |
| V.   | Assignment 4: Technical analysis – use case realization.....                       | 45 |
| 1.   | System sequence diagram for main success scenario “order meal from cafeteria” .... | 45 |
| 2.   | Operation contracts .....  | 46 |
| 3.   | MVC Sequence diagram for main success scenario “order meal from cafeteria” .....   | 48 |
| 4.   | Communication diagram for main success scenario “order meal from cafeteria” .....  | 49 |
| 5.   | Design class diagram for main success scenario “order meal from cafeteria” .....   | 50 |
| 6.   | Normalized data model.....   | 51 |

## II. Assignment 1: Scope/Problem analysis – business analysis

### 1. Product vision

| <b>Vision statement</b> <ul style="list-style-type: none"> <li>• Making the workflow around lunchtime more efficient (meal orders, process payments, and trigger delivery of the prepared meals )</li> <li>• Order food more efficiently</li> <li>• Reduce operating costs of cafeteria staff</li> <li>• Less food wastage</li> <li>• The product needs to be a multiplatform application that can be used by a group or individual.</li> </ul>                    |  |  |  |
|--|--|--|--|
| <b>Target group</b>  | <b>Needs</b>   | <b>Product</b>   | <b>Value</b>   |
| <p><i>Market segment</i></p> <p>Own company:</p> <ul style="list-style-type: none"> <li>- employees who want to purchase food</li> <li>- internal cafeteria</li> </ul> <p>Local catering industry / hospitality:</p> <ul style="list-style-type: none"> <li>- Local restaurants that want to participate</li> </ul> <p><i>Targets and customers</i></p> <ul style="list-style-type: none"> <li>- Process Impact: employees</li> <li>- Local restaurants</li> </ul> | <p><i>Problem</i></p> <ul style="list-style-type: none"> <li>- Employees can save time if they order their food items in advance</li> <li>- The product reduces the cost of cafeteria food wastage.</li> <li>- The product reduces the cafeteria operating costs.</li> </ul> <p><i>Benefit</i></p> <ul style="list-style-type: none"> <li>- The product increases the change of employees getting the items they prefer</li> </ul> | <p><i>Product</i></p> <p>Cafeteria Ordering System (COS): an internet-based and smart-phone application.</p> <p><i>What makes it desirable and special?</i></p> <ul style="list-style-type: none"> <li>- The product has many advantages for both employees and the cafeteria business. Employees will save time and will probably get the items they prefer. The cafeteria business will save costs due to the decrease cafeteria food wastage and will also save on operating costs.</li> <li>- It will be easy to use, easy accessible</li> <li>- The product will be multi-platform</li> </ul> <p><i>Is it feasible to develop the product?</i></p> <ul style="list-style-type: none"> <li>- The use of the product in the cafeteria is feasible but for the local restaurants, it might be too ambitious for the moment.</li> <li>- There are a few risks that need to be followed up. Maybe a trial</li> </ul> | <p><i>How is the product going to benefit the company?</i></p> <ul style="list-style-type: none"> <li>-The cafeteria staff will be able to work more efficiently.</li> <li>- Reduction of food wastage results in more profit.</li> <li>-A happy employee is a productive employee ;). The product will lead workflow augmentation.</li> </ul> <p><i>What are the business goals?</i></p> <ul style="list-style-type: none"> <li>- Reduce the costs of cafeteria food wastage by 40% within 6 months following initial release.</li> <li>- Reduce cafeteria</li> </ul> |

|  |  |  |   |
|--|--|--|---|
|  |  | <p>period with a simpler version of the program is needed.</p> <ul style="list-style-type: none"> <li>- Cost of development is carried by the company itself.</li> </ul> | <p>operating costs by 15% within 12 months following initial release.</p><br><ul style="list-style-type: none"> <li>- Increase average effective work time by 15 minutes per cafeteria-using employee per day within 6 months following initial release.</li> </ul> |
|--|--|--|---|

## 2. Problem statement table

| Brief Statements of Problem, Opportunity, or Directive   | Urgency  | Visibility     | Priority or Rank | Proposed Solution  |
|--|--|----------------|------------------|--|
| <p>P: Too much food is thrown away</p> <p>O: A opportunity to create a system that allows for more efficient management of food stock (purchase, sales and dispose) by getting the right information about lunch preferences.</p> <p>D: Reduce the cost of cafeteria food wastage by 40%</p> | Needs to be realized in 6 months following initial release of product  | High           | 1                | A cafeteria ordering system that allows staff to assemble menus so that the employees can choose their menu in advance and the staff can manage their stock more easily and efficiently. |
| <p>P: High cafeteria operating costs and inefficient work flow of the staff .</p> <p>O: A opportunity to create a system that allows for a more efficient work flow by spreading workload and define more specific tasks.</p> <p>D: Reduce cafeteria operating costs by 15%</p>              | Needs to be realized in 12 months following initial release of product | Medium to high | 2                | A cafeteria ordering system that allows staff to see when or where they are needed. The system will result in a process / work flow where everybody has his / her specific tasks.        |



|   |   |        |   |   |
|---|---|--------|---|---|
|   |   |        |   |   |
| <p>P: Time wastage of the employees during lunch hours.</p> <p>O: A opportunity to create a system that allows for delivery of food or a system for take away at specific time slots.</p> <p>D: Increase average effective work time by 15 minutes per cafeteria-using employee per day</p> | Needs to be realized in 6 months following initial release of product | High   | 1 | A cafeteria ordering system that allows employees to pre-order their lunch so that this lunch can be delivered or can be taken away quick from the cafeteria. |
| <p>P: Preferred menus are sold out more quickly.</p> <p>O: A opportunity to create a system that allows for employees to pre-order their preferred menus.</p> <p>D: Employees can choose their preferred dish 70% of the time.</p>  | Needs to be realized in 6 months following initial release of product | High   | 3 | A cafeteria ordering system that allows employees to pre-order their preferred lunch out of the assembled menu.   |
| <p>P: Less options in cafeteria than in other local places.</p> <p>O: A opportunity to create a system that allows for employees to order menus from the local restaurants at a fair price.</p> <p>D: 60% of the local restaurants are willing to deliver their food to the company.</p>    | Fully implemented by 3th release                                      | Medium | 2 | A cafeteria ordering system that allows employees to order their dishes from local restaurant. The restaurant food will be delivered at a discount price.     |

### 3. Problem descriptions according to PIECES

#### 3.1. *To much food is thrown away - Ruth Reynders*

**Problem:**

A first problem that can be derived from the meetings with the management of 'Proces Impact' is the fact the cafeteria wastes a significant quantity of food. This problem is covered by two categories of the PIECES model, namely *efficiency* and *economics*.

**Cause:** There are a few possible causes that can explain the wastage of food within the cafeteria: 1. It's difficult for the stock manager to adequately estimate the correct quantity of needed provision, because he doesn't know what customers want to eat.

2. The ratio between price and quality could be out of balance. Patrons think the cost of the offered food is too high and prefer not to purchase the food at the cafeteria.

3. The offered menu isn't adapted to the preferences of the patron. Patrons have too little choice in what they are going to eat, so they prefer homemade lunches. Therefore a significant quantity of the offered food is not purchased.

**Effect:** First of all, the expenses of the cafeteria are higher than the gains, because prepared food isn't purchased. Next, the problem leads to extra expenses: the means for disposal of the redundant food, the extra work hours of the members of the cafeteria etc. the members of the cafeteria. Finally, the customers aren't satisfied with the offered meals on the menu and with the fact that their preferred choice of meal is often sold out. So they usually prefer to eat homemade meals.

**Opportunity:** There is an opportunity to create a system that allows for a more efficient management of food stock. This can be realised by getting the right information about food preferences of customers in advance. The system will also allow for a larger choice of meals and a better ratio between price and quality.

**Benefit:** A good system will lead to the reduction of costs of food wastage, a more efficient management of the food stock and more satisfied customers.

**Constraint:** The development of a cafeteria ordering system can firstly be hindered by the expenses of making the system. The usability and accessibility of the system can also be a constraint. It is important that both the cafeteria staff as the patrons can efficiently use the system.

**Possible improvements:** The development of a cafeteria ordering system that allows the cafeteria staff to assemble menus so that patrons can choose their menu in advance. The system will allow the cafeteria to manage their stock more easily and efficiently. So, allowing patrons to order their food online in advance, will lead to less food wastage.

#### 3.2. *High cafeteria operating costs and inefficient work flow of the staff - Liesbeth Vandevenne*

**Problem:**

This statement can be traced down to two core problems.

*Efficiency problem.* The workload of the cafeteria staff is not spread among working hours. During peak hours, the effort required of the staff is excessive. Off-peak hours it seems like the staff is wasting time, waiting for an order to come in.

*Economic problem.* Operating costs of the cafeteria are too high.

**Cause:** Due to the current ordering system it is not possible to create a decent schedule for the cafeteria staff. The human resources department can only estimate how many workers are to be scheduled in the cafeteria.

**Effect:** Unhappy cafeteria workers due to the workload problems. Unhappy cafeteria customers due to the long queues because of the excessive workload.

**Opportunity:** By creating an efficient ordering system, cafeteria staff can be scheduled properly and customers will be served faster.

**Benefit:** Both cafeteria workers and customers will spend less time in the process of lunch break.

**Constraint:** The ordering system has to be build in a way that there is no threshold for both customers and cafeteria staff in using the new system.

### ***3.3. Time wastage of the employees during lunch hours. - Cathlene Van Moorleghem***

**Problem:**

This problem is an efficiency problem. Time wastage of the employee during lunch hours is quite a big problem. The cafeteria-using employees could make more effective use of their work time if they didn't have to wait in a line until someone takes their order.

**Cause:** Employees have to order meals verbally and at rush hour it is crowded at the cafeteria so they will have to wait long before it is their turn to order.

**Effect:** Due to the time lost waiting for their turn to order, the employees increase the time they are off-site so they work less. It is also possible that employees are very busy and thus have too little time to order something. Due to this the cafeteria will have less customers. This all results in profit loss for both the company and cafeteria business.

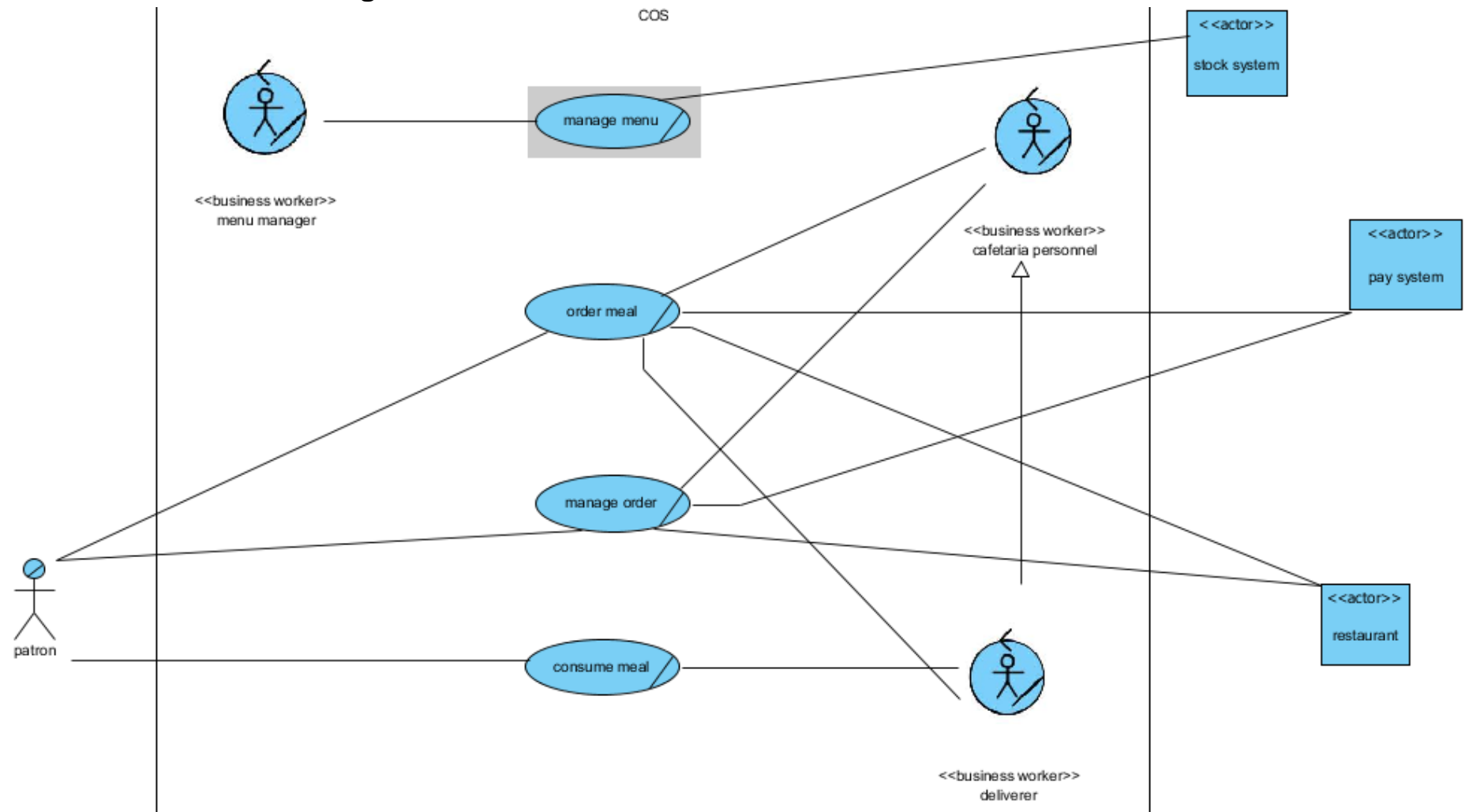
**Opportunity:** An opportunity to create a system that allows for delivery of food or a system for take away at specific time slots..

**Benefit:** The system would solve the problem of time wastage and it would increase the time employees can work and be productive.

**Constraint:** The ordering system has to be build in a way that both customers and cafeteria staff can easily make use of the new system.

**Possible improvements:** The system can provide an easy online ordering platform, so the employees don't have to wait in a line anymore.

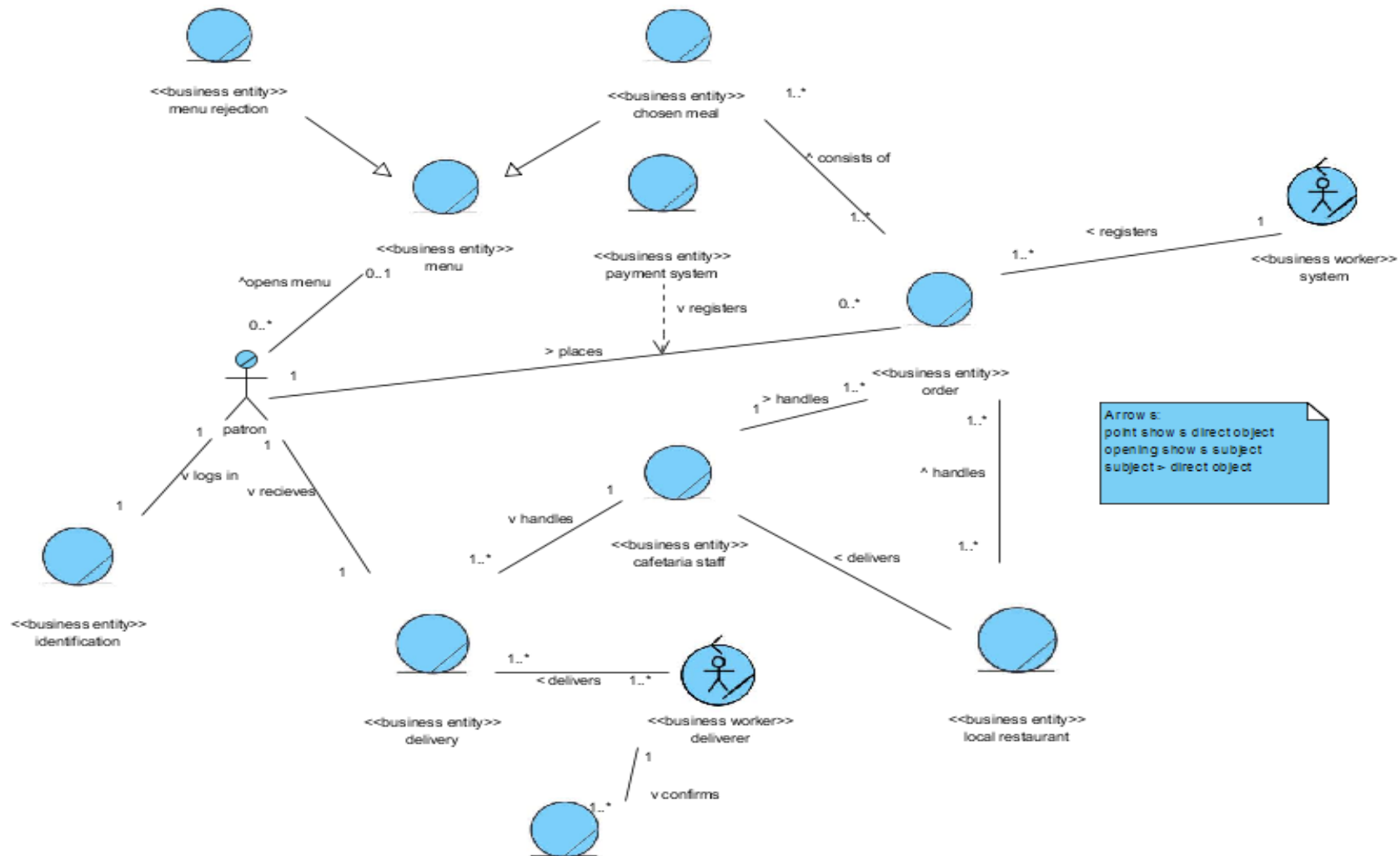
#### 4. Business use case diagram



## **5. Activity diagram of business use cases**

**A3**

## 6. Business class diagram



### III. Assignment 2: Requirement analysis/Context

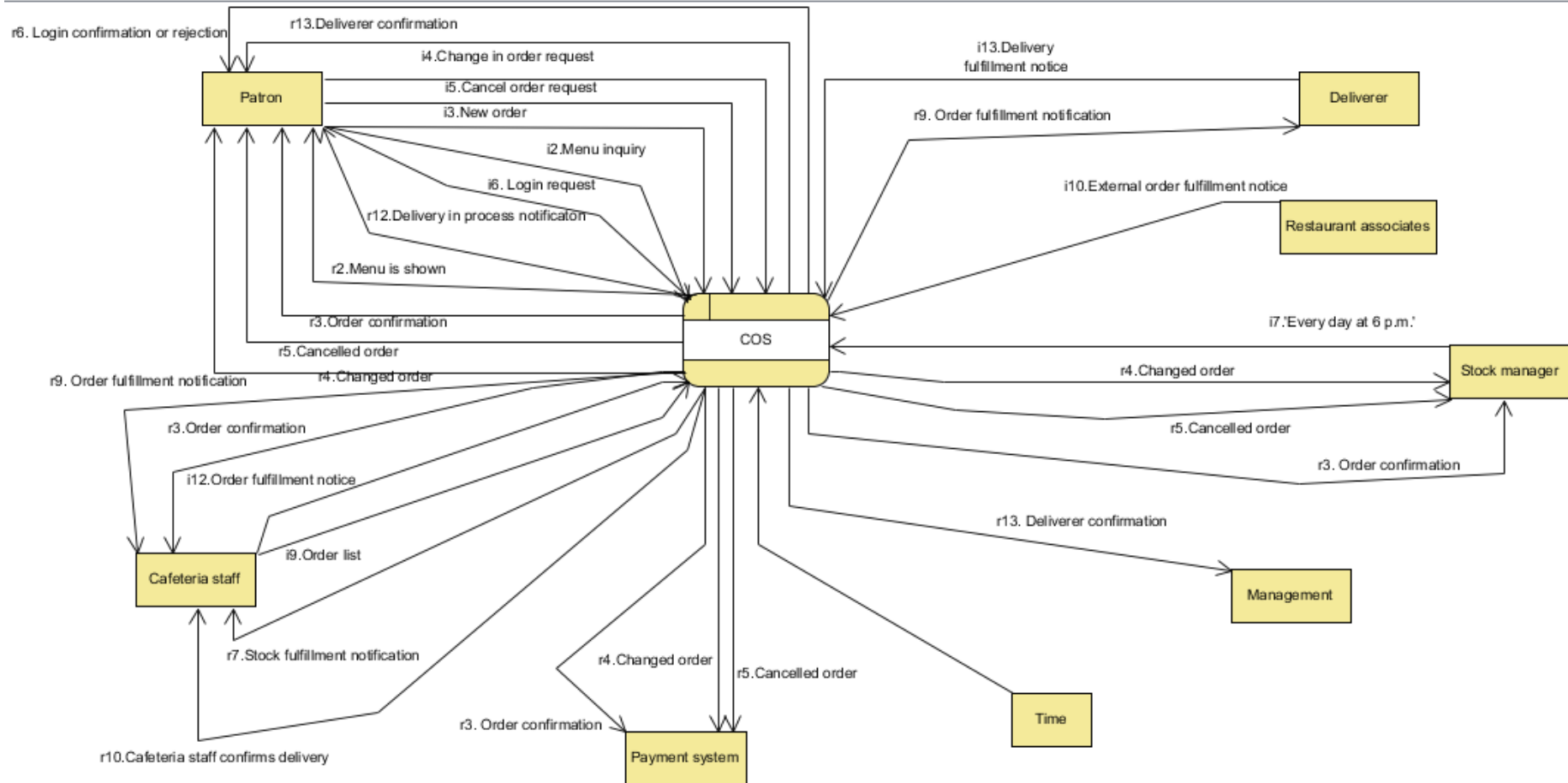
#### 1. Event response list

| Cafeteria ordering system event table    |                         |               |                           |   |  |
|--|-------------------------|---------------|---------------------------|---|--|
| Event                                    | Trigger                 | Source        | Use case                  | Response  | Destination  |
| 1. Menu manager manages menu (temporal)  | 'Beginning of the week' |               | Manage menu               | Menu fulfillment notice - menu is created                       | Stock manager<br>Patron<br>Cafeteria                         |
| 2. Patron checks menu (external)         | Menu inquiry            | Patron        | View menu                 | Menu is shown   | Patron   |
| 3. Patron places order (external)        | New order               | Patron        | Order meal from cafeteria | Order confirmation  | Stock Manager<br>Cafeteria Staff<br>Patron<br>Payment system |
| 4. Patron changes order (external)       | Change in order request | Patron        | Update order              | Changed order added in list, old version of order removed       | Stock Manager<br>Patron<br>Payment system                    |
| 5. Patron cancels order (external)       | Cancel order request    | Patron        | Cancel order              | Cancelled order removed from list, old version of order removed | Stock Manager<br>Patron<br>Payment system                    |
| 6. Patron logs in into system (external) | Login request           | Patron        | Authenticate              | Login confirmation or rejection                                 | Patron   |
| 6. Time to check requirements (temporal) | 'Every day at 5 p.m.'   |               | Check requirements        | Requirement list  | Stock manager  |
| 7. Time to manage the stock (internal)   | 'Every day at 6 p.m.'   | Stock manager | Manage stock              | Stock fulfillment notification                                  | Cafeteria staff  |

|  |                                   |                       |                          |   |                                |
|--|-----------------------------------|-----------------------|--------------------------|---|--------------------------------|
| 8. Time to check orders (temporal)                                     | 'Every morning'                   |                       | Check orders             | List with expected orders                             | Cafeteria staff                |
| 9. Cafeteria staff handles order (internal)                            | Order list                        | Cafeteria staff       | Handle orders            | Order fulfillment notification                        | Cafeteria staff<br>Deliverer   |
| 10. Restaurant associates deliver orders to cafeteria staff (internal) | External order fulfillment notice | Restaurant associates | Accept external delivery | Cafeteria staff confirms delivery                     | Cafeteria staff                |
| 11. Time to generate route plan (temporal)                             | 'every day at 9.30 a.m.'          |                       | Generate route plan      | Route plan  | Deliverer                      |
| 12. Deliverer starts delivery round (internal)                         | Order fulfillment notice          | Cafeteria staff       | Start delivery           | Delivery in process notification                      | Patron                         |
| 13. Deliverer delivers order (internal)                                | Delivery fulfillment notice       | Deliverer             | Deliver order            | Deliverer confirmation                                | Patron<br>Management           |
| 14. System summary reports are produced (temporal)                     | 'End of the week, month'          |                       | Produce system reports   | Monthly summary reports<br><br>Weekly summary reports | Management<br><br>Menu manager |
| 15. Time to settle wages (temporal)                                    | 'End of month'                    |                       | Settle wages             | Summary reports                                       | Management                     |



## 2. Context diagram



### 3. Data flows

|   |  |
|---|--|
| <p><b>Order</b> = order id + timeSlot + patron data<br/>=patron + delivery location = location +<br/>(billing address = address )+ quantity<br/>ordered + 1{meal data = meal}n +total price</p> <p><b>Location</b>= building number+ floor number<br/>+ room number</p> <p><b>Address</b> = (post office box number ) + street<br/>address + city + (country)+ postal code</p> <p><b>Meal</b> = id + description + price + ingredient</p> <p><b>Patron</b> = personal patron company number<br/>+ name + surname + address = address +<br/>(order history)</p>  | <p><b>Order confirmation</b> = confirmation id +<br/>order data = order</p> <p><b>Order</b> = order id + timeSlot + patron data<br/>=patron + delivery location = location +<br/>(billing address = address )+ quantity<br/>ordered + 1{meal data = meal}n +total price</p> <p><b>Location</b>= building number+ floor number<br/>+ room number</p> <p><b>Address</b> = (post office box number ) + street<br/>address + city + (country)+ postal code</p> <p><b>Meal</b> = id + description + price + ingredient</p> <p><b>Patron</b> = personal patron company number<br/>+ name + surname + address = address +<br/>(order history)</p>                             |
| <p><b>Delivery</b> = delivery id + timeSlot +date +<br/>order data = order</p> <p><b>Order</b> = order id + timeSlot + patron data<br/>=patron + delivery location = location +<br/>(billing address = address )+ quantity<br/>ordered + 1{meal data = meal}n +total price</p> <p><b>Location</b>= building number+ floor number<br/>+ room number</p> <p><b>Address</b> = (post office box number ) + street<br/>address + city + (country)+ postal code</p> <p><b>Meal</b> = id + description + price + ingredient</p> <p><b>Patron</b> = personal patron company number<br/>+ name + surname + address = address +<br/>(order history)</p> | <p><b>Delivery fulfillment notice</b> = notice id +<br/>order data =order +delivery id + time +date</p> <p><b>Order</b> = order id + timeSlot + patron data<br/>=patron + delivery location = location +<br/>(billing address = address )+ quantity<br/>ordered + 1{meal data = meal}n +total price</p> <p><b>Location</b>= building number+ floor number<br/>+ room number</p> <p><b>Address</b> = (post office box number ) + street<br/>address + city + (country)+ postal code</p> <p><b>Meal</b> = id + description + price + ingredient</p> <p><b>Patron</b> = personal patron company number<br/>+ name + surname + address = address +<br/>(order history)</p> |
| <p><b>Order fulfillment notification</b> = notification<br/>id + order data =order</p> <p><b>Order</b> = order id + timeSlot + patron data<br/>=patron + delivery location = location +<br/>(billing address = address )+ quantity<br/>ordered + 1{meal data = meal}n +total price</p> <p><b>Location</b>= building number+ floor number<br/>+ room number</p>  | <p><b>Deliverer confirmation</b> = confirmation id +<br/>order data = order + delivery id +(delivery<br/>quote) + confirmation time +confirmation<br/>date</p> <p><b>Order</b> = order id + timeSlot + patron data<br/>=patron + delivery location = location +<br/>(billing address = address )+ quantity<br/>ordered + 1{meal data = meal}n +total price</p> <p><b>Location</b>= building number+ floor number</p>   |

|  |  |
|--|--|
| <p><b>Address</b> = (post office box number ) + street address + city + (country)+ postal code</p> <p><b>Meal</b> = id + description + price + ingredient</p> <p><b>Patron</b> = personal patron company number + name + surname + address = address + (order history)</p> | <p>+ room number</p> <p><b>Address</b> = (post office box number ) + street address + city + (country)+ postal code</p> <p><b>Meal</b> = id + description + price + ingredient</p> <p><b>Patron</b> = personal patron company number + name + surname + address = address + (order history)</p>  |
| <p><b>Stock fulfillment notification</b> = notification id + 1{product in stock}n + total number of products in stock + time +date</p>   | <p><b>Order fulfillment notice</b> = notice id +order data =order</p> <p><b>Order</b> = order id + timeSlot + patron data =patron + delivery location = location + (billing address = address )+ quantity ordered + 1{meal data = meal}n +total price</p> <p><b>Location</b>= building number+ floor number + room number</p> <p><b>Address</b> = (post office box number ) + street address + city + (country)+ postal code</p> <p><b>Meal</b> = id + description + price + ingredient</p> <p><b>Patron</b> = personal patron company number + name + surname + address = address + (order history)</p> |

#### 4. Stakeholders / actors

| Stakeholder          | Major Value   | Attitudes  | Major interests  | Constraints   |
|----------------------|---|--|--|---|
| Corporate Management | Improved employee productivity; cost savings for cafeteria                        | Strong commitment through release 2; support for release 3 contingent on earlier results   | Cost and employee time savings must exceed development and usage costs   | None identified   |
| Cafeteria Staff      | More efficient use of staff time throughout the day; higher customer satisfaction | Concern about union relationships and possible downsizing; otherwise receptive   | Job preservation   | Training for staff in Internet usage needed; delivery staff and vehicles needed |
| Patrons              | Better food selection; time savings; convenience                                  | Strong enthusiasm, but might not use it as much as expected because of social value of eating lunches in cafeteria and restaurants | Simplicity of use; reliability of delivery; availability of food choices | Corporate intranet access, Internet access, or a mobile device is needed        |
| Payroll Department   | No benefit; needs to set up payroll deduction registration scheme                 | Not happy about the software work needed, but recognizes the value to the company and employees                                    | Minimal changes in current payroll applications                          | No resources yet committed to make software changes                             |

## 5. Business rules

| ID    | Rule Definition   | Type of Rule | Static or Dynamic | Source                              |
|-------|---|--------------|-------------------|-------------------------------------|
| BR-1  | Delivery time windows are 15 minutes, beginning on each quarter hour.   | Fact         | Dynamic           | Cafeteria Manager                   |
| BR-2  | Deliveries must be completed between 10:00 A.M. and 2:00 P.M. local time, inclusive.  | Constraint   | Dynamic           | Cafeteria Manager                   |
| BR-3  | All meals in a single order must be delivered to the same location.   | Constraint   | Static            | Cafeteria Manager                   |
| BR-4  | All meals in a single order must be paid for by using the same payment method.  | Constraint   | Static            | Cafeteria Manager                   |
| BR-11 | If an order is to be delivered, the patron must pay by payroll deduction.   | Constraint   | Dynamic           | Cafeteria Manager                   |
| BR-12 | Order price is calculated as the sum of each food item price times the quantity of that food item ordered, plus applicable sales tax, plus a delivery charge if a meal is delivered outside the free delivery zone. | Computation  | Dynamic           | cafeteria policy;<br>state tax code |
| BR-24 | Only cafeteria employees who are designated as Menu Managers by the Cafeteria Manager can create, modify, or delete cafeteria menus.  | Constraint   | Static            | cafeteria policy                    |
| BR-33 | Network transmissions that involve financial information or personally identifiable information require 256-bit encryption.   | Constraint   | Static            | corporate security policy           |
| BR-86 | Only regular employees can register for payroll deduction for any company purchase.   | Constraint   | Static            | Corporate Accounting Manager        |

|        |  |             |         |  |
|--------|--|-------------|---------|--|
| BR-88  | An employee can register for payroll deduction payment of cafeteria meals if no more than 40 percent of his gross pay is currently being deducted for other reasons. | Constraint  | Dynamic | Corporate Accounting Manager                     |
| BR-100 | Only a registered employee can order a meal  | Constraint  | Static  | Corporate Accounting Manager                     |
| BR-101 | A registered employee can place an order.  | Fact        | Static  | Cafeteria policy                                 |
| BR-102 | An order (for the next day) must be placed before 5p.m.  | Fact        | Static  | Cafeteria policy                                 |
| BR-103 | A list of ingredients needed will be supplied at the stock manager   | Computation | Dynamic | Cafeteria Manager<br>Stock Manager               |
| BR-104 | The menu for the upcoming week is published every Friday at 10a.m.   | Constraint  | Dynamic | Menu Managers<br>Cafeteria Manager               |
| BR-105 | Everybody can view the menu  | Fact        | Dynamic | Corporate policy                                 |
| BR-106 | An order that has been cancelled after 5p.m. has to be paid by the patron.   | Fact        | Static  | Corporate Accounting Manager<br>Cafeteria policy |
| BR-107 | If a delivery time slot is full, a user will not be able to choose that time slot  | Constraint  | Dynamic | Application service                              |
| BR-108 | A user can change his order (for the next day) until 5p.m.   | Constraint  | Static  | Cafeteria policy                                 |
| BR-109 | A user can refuse a payment when the delivery is more than 30 minutes too late   | Constraint  | Static  | Payroll manager<br>Cafeteria manager             |
| BR-110 | The standard delivery location will be remembered by the application   | Computation | Static  | Application service                              |
| BR-111 | Discounts are not possible   | Constraint  | Static  | Company Policy                                   |
| BR-112 | Beverage is never included in an order   | Fact        | Static  | Cafeteria manager                                |

## 6. Quality attributes

### Reliability

**Reliability in operation.** The cafeteria ordering system should be available as much as possible. It is expected that most orders will be carried out between 15:00 and 17:00. Therefore, maintenance should be scheduled outside this time slot. Maintenance outside this period should be announced in advance.

This application is a handy tool to save costs, but does not belong to the core business of the company. The economic loss suffered by downtime of this app is so limited that a monthly downtime of up to 24 hours is acceptable.

**Robustness.** Even when the stock management system or the payment system is down, a user should be able to place an order.

The credentials used to log in to the system are the same as the credentials used to log in to the users company computer. Therefore the login portal of the company is used. When this system suffers downtime users will not be able to log in.

**Restorability.** When logged in correctly a user should be able to place, change and cancel an order. Trying to perform an action after 17:00 for the day after, can never resolve into an error. Only a notification will be send.

Data necessary for handling the payments should be correct at all times. When finding doubtful data, the settlement will not be executed. In that case the Cafeteria Manager and the HR Department payments will review the payments manually and approve them.

### Usability

**Intelligibility.** An experienced Internet user - which can easily choose what he wants to eat - would have to take less than five minutes to place an order for the next day. When using this application for the first time it should not take longer than 10 minutes to place a first order.

**Learn-ability.** The system should be built in a way that ordinary users can work without training.

The menu manager and cafeteria manager should be able to carry out all administrative tasks with this tool with a training of two hours.

**Ease of use.** The system is usable on all kinds of devices. Buttons should be clearly defined.

**Attractive application.** This application is only semi-commercial. In the development a clean design is taken into account. However, the appearance of the application is not a priority. More importantly, the application is quick and easy to use.

### Efficiency

**Speed.** Using the company wireless or wired network the application should load in 90% of the time within 2 seconds. When using a mobile connection, load time depends on the connection speed of the mobile device.

Taking the growth of the staff into account and to avoid investments for this application due to this growth, the minimum number of simultaneous users is set at 700.

**Storage.** 10 GB of active storage should be sufficient. A backup storage of 50 GB will be equipped.

Since the application has a low priority within the company and the application must be easily loaded via a mobile device, the allowed network traffic is restricted to 1Mb / s.

## **Functionality**

**Correctness.** The monthly statement of the orders must be correct for 99%.

**Security.** Users authenticate themselves always with their login and password from the company. This authentication must be done over an SSH connection to prevent interception by third parties. Only in this way there can be no discussion about fraudulent orders and it is insured that passwords meet company policy.

Every action that takes place will be saved.

## **Link-ability.**

The system has to be linked with:

- login portal / ldap server
- stock management system
- payment system

## **Maintainability**

**Possibility to make changes.** The content of the menu and the price of a meal will be easily changed by the menu manager.

Delivery locations can be added by users. They will have to be confirmed by the cafeteria manager.

Time slots can only be changed or added by an administrator.

**Possibility to analyze the system.** Every change will be saved. So it is possible to pull all kinds of reports from the system.

**Stability.** Errors concerning payment issues have got to be solved within 4 hours.

Other errors can take up to 24 hours for solving.

**Testability.** Changes to the system associated with payments must be evaluated within four hours after release.

For further releases, there a margin of 24 hours is provided.

## **Portability of the system**

**Adaptability.** The software API has to be written in a platform independent programming language.

The front end is adapted for each device.

**Installation.** Server side the software needs to be installed by a system administrator.

Client side the user can easily install the application himself.

**Togetherness.** It should be possible to install the system on any server, together with any application. As long as there are enough resources.

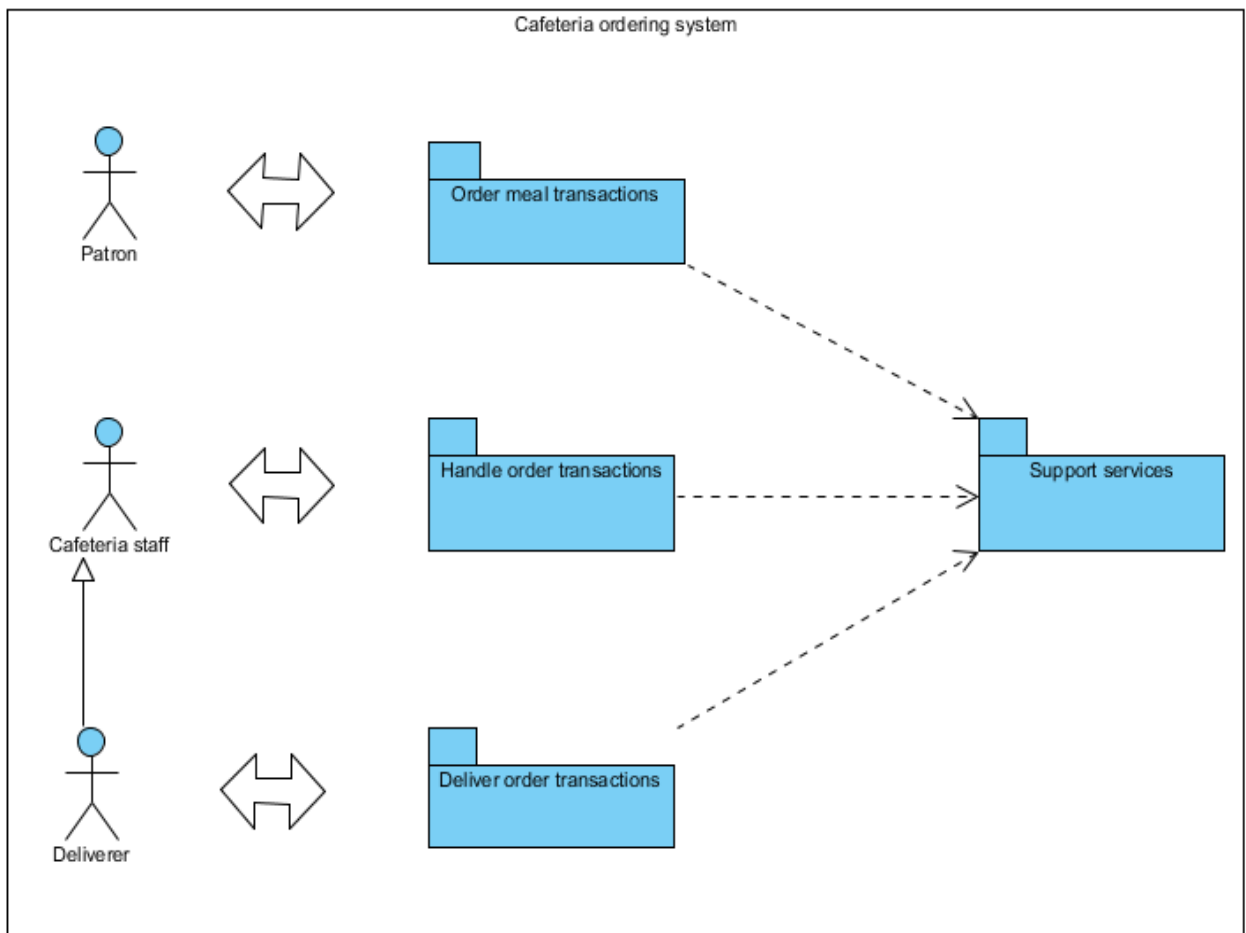


## 7. Use cases with ranking

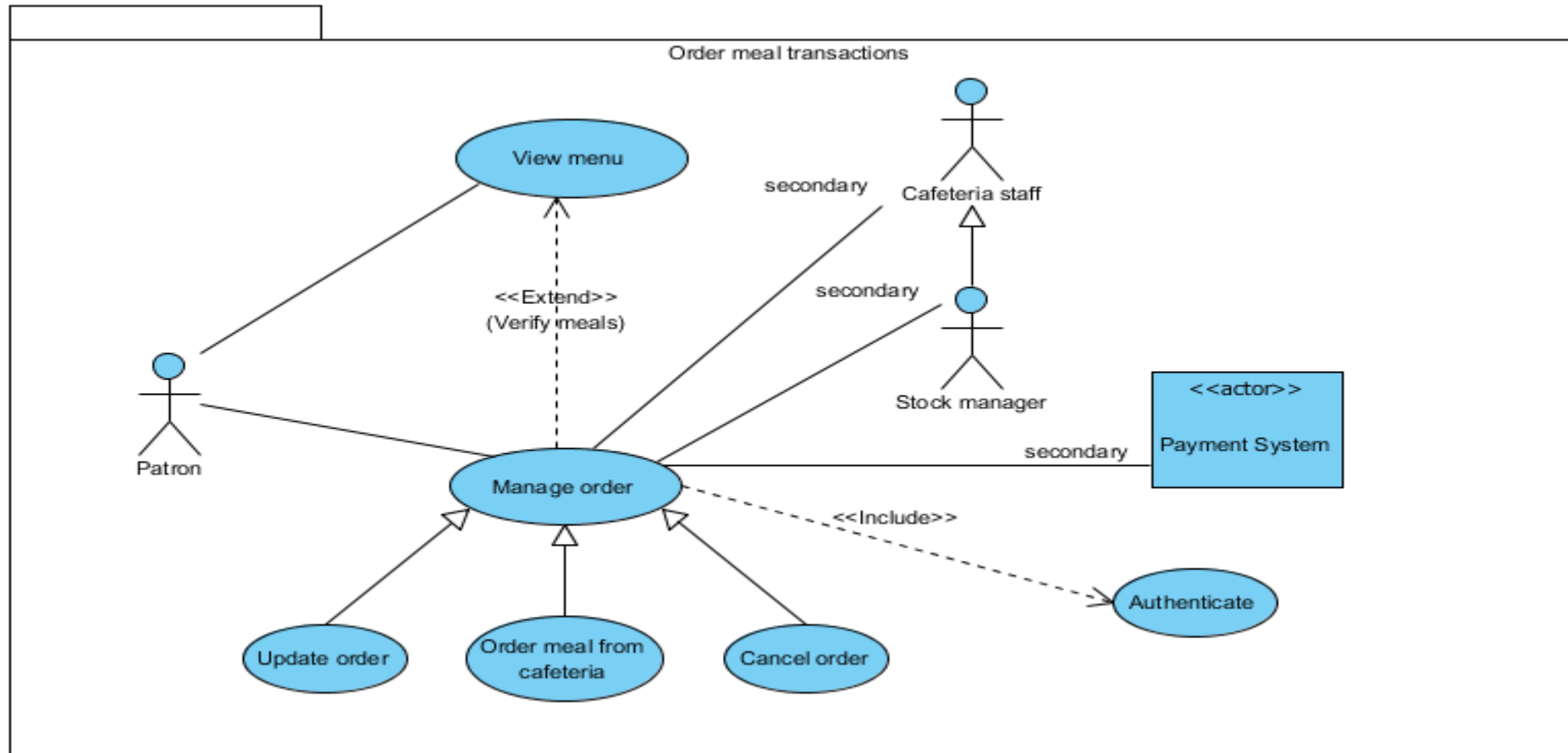
| Use case                  | Priority (M,S,C,W) | Release (1,2 of 3) | Motivate priority  |
|---------------------------|--------------------|--------------------|--|
| Manage menu               | M                  | 1                  | Without a menu it is impossible to order a meal thus the app is useless.   |
| View menu                 | M                  | 1                  | It is impossible to place an order without viewing the menu.   |
| Order meal from cafeteria | M                  | 1                  | The core business of the application.  |
| Update order              | S                  | 2                  | The application is usable without this functionality. But the possibility of updating an order is a customer friendly feature.       |
| Cancel order              | S                  | 2                  | The application is usable without this functionality. But the possibility of cancelling an order is a customer friendly feature.     |
| Authenticate              | M                  | 1                  | Identifying users is impossible without a authentication feature. It is important to know who ordered what.                          |
| Check requirements        | S                  | 2                  | To save money an automated list to check the requirements is a handy tool. Without this tool this task has to be performed manually. |
| Manage stock              | C                  | 2                  | This could be a feature of the ordering system. But for now the stock management system is outsourced.                               |
| Check orders              | M                  | 1                  | No order can be handled without this tool.   |

| Use case                 | Priority<br>(M,S,C,W) | Release (1,2 of 3) | Motivate priority  |
|--------------------------|-----------------------|--------------------|--|
| Handle orders            | C                     | 2                  | A handy feature for the deliverers is knowing when the orders are ready to be delivered. But this task can also be performed manually. |
| Accept external delivery | S                     | 3                  | Accepting external deliveries is a practical use case. Not necessary.  |
| Generate routeplan       | C                     | 3                  | This is a very expensive feature. We could ask ourselves if the cost out lift the benefit.   |
| Start delivery           | S                     | 2                  | Register which orders have left the kitchen.   |
| Deliver order            | S                     | 2                  | Register which order has been delivered.   |
| Produce system reports   | S                     | 3                  | Important for the business analysts. Only useful when every other feature is implemented.  |
| Settle wages             | M                     | 1                  | Important feature. Without this, the cafeteria won't earn any money.   |

## 8. Main use cases as package diagram



## 9. System use case diagram for package “Order meal transactions”



## IV. Assignment 3: Logical analysis – use case description

### 1. Flow of events for use case “order meal from cafeteria”

#### 1.1. Flow of events Word

|   |
|---|
| <b>Use case ID + Use case name:</b> “Order meal from cafeteria”   |
| <b>Primary actors:</b> Patron<br><b>Secondary actors:</b> Stock manager, Cafeteria Staff, Payment system  |
| <b>Short description:</b> This use case allows patrons to order a meal from the cafeteria by using the online / desktop application. The application registers the order and keeps the stock manager, cafeteria staff and payment system in the loop.   |
| <b>Preconditions:</b> <ul style="list-style-type: none"><li>• Application needs to be available.</li><li>• Menu needs to be prepared and placed on the application.</li><li>• Patron has to be registered.</li></ul>  |
| <b>Postconditions:</b> <ul style="list-style-type: none"><li>• The order is confirmed to the stock manager, cafeteria staff, patron and payment system.</li><li>• The order is added to the list with orders.</li></ul>   |
| <b>Basic flow:</b> <ol style="list-style-type: none"><li>1. The use case begins when the patron opens the menu on the application</li><li>2. The system prompts the available menu on the user screen.</li><li>3. The patron selects a preferred meal(s) from the menu and goes to order basket.</li><li>4. The system prompts for a login name and password.</li><li>5. The patron enters credentials.<br/><br/>A1: The patron enters false credentials<br/>E1: Login system not accessible</li><li>6. The systems shows options for delivery (e.g. standard location or other location; standard time slot or other time slot).</li><li>7. The patron selects standard location and standard time slot<br/><br/>A2: Patron selects other location<br/>A3: Patron selects other time slot<br/>A4: Patron cancels order.</li><li>8. The patron sends his request to the system.</li><li>9. The system prompts a confirmation screen</li><li>10. The patron confirm his order(s).<br/><br/>A5: Patron changes his preference for time slot and/or location<br/>A6: Patron cancels his order.</li></ol> |

11. The system adds order to the ordering list.

E2: System failure, internal database not accessible.

12. The system sends confirmations to involved actors.

13. This use case ends.

#### **Alternative flows:**

A1 The patron enters false credentials

1. System displays a message that credentials are invalid.
2. Patron re-enters his credentials.
3. If the login is successful, the flow returns to step 6.
4. If the patron cancels the login, the flow returns to step 2.
5. If the login is unsuccessful, the flow returns to alternative flow A1, step 1.

A2: Patron selects other location.

1. The patron indicates he wants another location for delivery.
2. The system shows options for delivery location.
3. If the patron selects another location, the flow returns to step 7.

A3: Patron selects other time slots.

1. The patron indicates he wants another time slot for delivery.
2. The system shows options for time slots.
3. If the patron selects another time slot, the flow returns to step 7.

A4: Patron cancels order. (Patron has the possibility to cancel his order before the confirmation screen is shown)

1. If the patron chooses to cancel the order, the flow returns to step 2.

A5. Patron changes his preference for time slot and/or location

1. Patron selects other options for location and time slot of delivery.
2. If patron confirms his order, the flow returns to step 11.

A6: Patron cancels his order. (Patron has the possibility to cancel his order during the time the confirmation screen is shown.)

1. If the patron chooses to cancel the order, the flow returns to step 2.

#### **Error flows:**

E1: Login system not accessible.

1. The system displays a message that the login system is not available.
2. The flow returns to step 2.

E2: System failure, internal database not accessible.

1. The system displays a message that the database is not accessible.
2. The flow returns to step 2.

## 1.2. Flow of events Visual Paradigm

|                          |  |
|--------------------------|--|
| <b>Super Use Case</b>    | Manage order   |
| <b>Author</b>            | Ruth   |
| <b>Date</b>              | 13-dec-2015 22:25:18   |
| <b>Brief Description</b> | This use case allow patrons to order a meal from the cafeteria by using the online / desktop application. The application registers the order and keeps the stock manager, cafeteria staff and payment system in the loop.<br>Application needs to be available. |
| <b>Preconditions</b>     | Menu needs to be prepared and placed on the application. (BR-104, BR-105)<br>Patron has to be registered. (BR-100, BR-101)   |
| <b>Post-conditions</b>   | The order is confirmed to the stock manager, cafeteria staff, patron and payment system (BR-103, BR - 106, BR - 12, BR - 4).<br>The order is added to the list with orders.  |

|                       | <b>Actor Input</b>  | <b>System Response</b>   |
|-----------------------|---|--|
| <b>Flow of Events</b> | 1 Patron opens menu on the application (BR-105)                                       |  |
|                       | 2   | The system prompts available menu on user screen. (BR - 104)   |
|                       | 3 Patron selects preferred meal(s) from the menu and goes to order basket.            |  |
|                       | 4   | The system prompts for a login name and password. (BR-100, BR-101)   |
|                       | 5 The patron enters credentials.(BR - 33)   |  |
|                       | 6   | The system shows options for delivery (e.g. standard location or other location; standard time slot or other time slot) (BR-2, BR-3, BR - 107) |
|                       | 7 The patron selects standard location and standard time slot. (BR-2, BR-3, BR - 107) |  |
|                       | 8 The patron sends his request to the system. (BR-100, BR-101, BR-102)                |  |
|                       | 9   | The system prompts a confirmation screen.  |
|                       | 10 The patron confirms his order(s). (BR-100, BR-101, BR-102)                         |  |
|                       | 11  | The system adds order to the ordering list. (BR - 108)   |
|                       | 12  | The system sends confirmations to involved actors. (BR-103, BR-106, BR-111, BR - 112, BR -4, BR-12)  |

|  |                    |                                |
|--|--------------------|--------------------------------|
| <b>Alternative A1:<br/>The patron enters false</b> | <b>Actor Input</b> | <b>System Response</b>         |
|  | 1                  | System displays a message that |

|  |                    |   |
|--|--------------------|---|
| <b>credentials</b>   |                    | credentials are invalid.  |
|  | 2                  | Patron re-enters his credentials.   |
|  | 3                  | If login is successful the flow returns to step 6.                            |
|  | 4                  | If the patron cancels the login, the flow returns to step 2.                  |
|  | 5                  | If the login is unsuccessful, the flow returns to alternative flow A1, step1. |
| <b>Alternative A2:<br/>Patron selects other location</b>                               | <b>Actor Input</b> |   |
|  | 1                  | The patron indicates he wants another location for delivery.                  |
|  | 2                  | The system shows options for delivery location.                               |
|  | 3                  | If the patron select another location, the flow returns to step 7.            |
| <b>Alternative A3:<br/>Patron selects other time slot</b>                              | <b>Actor Input</b> |   |
|  | 1                  | The patron indicates he wants another time slot for delivery.                 |
|  | 2                  | The system shows options for time slots.                                      |
|  | 3                  | If the patron selects another time slot, the flow returns to step 7.          |
| <b>Alternative A4:<br/>Patron cancels order</b>  | <b>Actor Input</b> |   |
|  | 1                  | If the patron chooses to cancel the order, the flow returns to step 2.        |
| <b>Alternative A5:<br/>Patron changes his preference for time slot and/or location</b> | <b>Actor Input</b> |   |
|  | 1                  | The patron selects other options for location and/or time slot of delivery.   |
|  | 2                  | If the patron confirms his order, the flow returns to step 11.                |
| <b>Alternative A6:<br/>Patron cancels his order</b>                                    | <b>Actor Input</b> |   |
|  | 1                  | If the patron chooses to cancel his order, the flow returns to step 2.        |
| <b>Error E1: Login system not accessible</b>   | <b>Actor Input</b> |   |
|  | 1                  | The system displays a message that the login system is not available          |
|  | 2                  | If the patron has read the message, the flow returns to step 2.               |
| <b>Error E2:<br/>System failure, internal database not accessible</b>                  | <b>Actor Input</b> |   |
|  | 1                  | The system displays a message that the database is not accessible.            |
|  | 2                  | If the patron has read the message, the flow returns to step2                 |



## 2. User stories for use case “order meal from cafeteria”

### 2.1. Phase 1:



#### Use Case Statement

Identify actors and use cases from simple statements

|                    |  |  |   |
|--------------------|--|--|---|
| As a patron        | , I want to place an order for one meal              | so that I can have lunch tomorrow  | . |
| As a patron        | , I want to place an order for multiple sandwiches   | so that I can offer my business associates a sandwich during the meeting on Friday | . |
| As a patron        | , I want to place an order for a different location  | so that I can eat my lunch at my colleague's office                                | . |
| As a patron        | , I want to place an order for a different time slot | so that I can finish my deadline before eating lunch                               | . |
| As a patron        | , I want to place multiple orders                    | so that I am sure that I have a meal every day this week                           | . |
| As a stock manager | , I want to get a list of all the ordered items      | so that I can make sure that all the right ingredients are in stock.               | . |


### 2.2. Phase 2:

- User story 1:

 **A patron can place an order for a single meal.**  US004

Wim Maes, employee of the IT department wants to place an order for a single meal. It's Monday, December 14th, 2 p.m. and Wim wants to place an order for Tuesday.

Conversation | Confirmation (7/7) | Scenario (1) | Storyboard | Design | References



- Allow patron to order single meal on following conditions:
  - All systems are up and running
  - Patron is registered in the system
  - Patron gives valid credentials
  - If order is placed before 5 p.m.: patron can still choose lunch for the next day.
  - If order is placed after 5 p.m.: patron can't choose lunch for the next day.
- Fields required
  - Add to order button
  - Quantity field
  - Button: go to order basket

Test if patron is registered in the system.

Test if error message is given with false username

Test if error message is given with false password

Test all fields and buttons

Test if invalid value in quantity field gives error message

Test if order contains single meal

- User story 2:

## A user can place an order for multiple meals.

US003 Pending

Johan Bosmans, CEO, can place an order for multiple meals. It's Wednesday, 6 p.m. and the meeting is planned on Friday at 12 p.m.. Johan Bosmans wants to order a meal for himself and sandwiches for his business associates. The meeting takes place on Friday in the conference room.

Conversation

Confirmation (0/7)

Scenario (1)

Storyboard

Design

References

- Allow patron to order multiples meals on following conditions:
  - All systems are up and running
  - Patron is registered in the system
  - Patron gives valid credentials
  - If order is placed before 5 p.m.: patron can still choose lunch for the next day.
  - If order is placed after 5 p.m.: patron can't choose lunch for the next day.
- Fields required
  - Add to order button
  - Quantity field
  - Button: go to order basket

Test if patron is registered in the system

Test if error message is given with false username

Test if error message is given with false password

Test all fields and buttons

Test if invalid value in quantity field gives error message

Test if order contains multiple meals

- User story 3:

## A user can make an order for a different time slot than his usual time slot.

US002

Chandler Bing, Junior copywriter, needs to make his deadline of 1 p.m. So, he would like to place an order for a different time slot than usual.

Conversation

Confirmation (0/8)

Scenario (1)

Storyboard

Design

References

- Allow patron to change his time slot on following conditions:
  - All systems are up and running
  - Patron is registered in the system
  - Patron gives valid credentials
  - If order is placed before 5 p.m.: patron can still choose lunch for the next day.
  - If order is placed after 5 p.m.: patron can't choose lunch for the next day.
  - Meals are chosen
- Fields en buttons required
  - Checkbox: standard time slot
  - Checkbox + drop down menu: non standard time slot
  - Button: continue
  - Button: cancel
  - Button: go to menu

Test if patron is registered in the system

Test if error message is given with false username

Test if error message is given with false password

Test all fields and buttons


Test if chosen meals are being remembered

Test if invalid value in time slot field gives error message

Test if unavailable time slots aren't shown in drop down menu

Test if time slot in order object has been changed

- User story 4:













## A user can make an order for a different location than his standard location.

US00

Anja Marx, cleaning lady, wants to order lunch for a different location. Normally she eats lunch at the cafeteria, but tomorrow she wants to eat lunch at office 02.45.


Conversation
Confirmation (0/7)
Scenario (1)
Storyboard
Design
References

- Allow patron to change his location on following conditions:
  - All systems are up and running
  - Patron is registered in the system
  - Patron gives valid credentials
  - If order is placed before 5 p.m.: patron can still choose lunch for the next day.
  - If order is placed after 5 p.m.: patron can't choose lunch for the next day.
  - Meals are chosen
- Fields en buttons required
  - Checkbox: standard location
  - Checkbox + drop down menu: non standard location
  - Button: continue
  - Button: cancel
  - Button: go to menu

- Test if patron is registered in the system
- Test if error messages is given with false username
- Test if error message is given with false password
- Test all fields and buttons
- Test if chosen meal are being remembered by the system
- Test if invalid value in location field gives error message
- Test if location in order object has been changed

- User story 5:













## A patron can make multiple orders.

US008
Pending

Alain Van Damme, accountant, is a practical man. He likes to order multiple meals for the rest of the week, so that he is sure that lunch every day and that he doesn't need to open the application on a daily basis.


Conversation
Confirmation (0/7)
Scenario (1)
Storyboard
Design
References

- Allow patron to make multiple orders on following conditions:
  - All systems are up and running
  - Patron is registered in the system
  - Patron gives valid credentials
  - If order is placed before 5 p.m.: patron can still choose lunch for the next day.
  - If order is placed after 5 p.m.: patron can't choose lunch for the next day.
- Fields required
  - Add to order button
  - Quantity field
  - Button: go to order basket
  - Tab buttons, depending of actual day and time: Monday, Tuesday, Wednesday, Thursday, Friday

- Test if patron is registered in the system
- Test if error message is given with false username
- Test if error message is given with false username
- Test all fields and buttons
- Test if invalid value in quantity field gives error message
- Test if multiple order objects have been made
- Test if multiple order objects have been added to the list of ordered meals

- User story 6:



# A stock manager can get a list of all the ordered items

US009 Pending

Monica Geller, cafeteria and stock manager, needs to be able to check the list with ordered items.

Conversation






Confirmation (0/5)

Scenario (1)

Storyboard

Design

References

- Allow stock manager to get a list of ordered items on following conditions:
  - All systems are up and running
  - Stock manager is registered in the system
  - Stock manager gives valid credentials
  - If list is queried before 5 p.m.: list may not be completed.
  - If list is queried after 5 p.m.: patron can't choose lunch for the next day.
- Fields required
  - Button: login
  - Button: generate list

Test if stock manager is register in the system

Test if error message is given with false username

Test if error message is given with false password

Test all fields and buttons

Test if complete list is generated

### 2.3. Phase 3:

Scenario (1)

Storyboard

Design

References

USER STORY 1

- Patron opens menu
- System prompts menu
- Patron selects menu for next day (before 5 p.m.)
- Patron goes to order basket
- System prompts a login screen
- Patron gives correct credentials
- System show screen for order location and time slot.
- Patron selects standard time slot and standard location
- System shows confirmation screen for order.
- Patron confirms order

Extension:


- Alternative: Patron gives false credentials
  - System displays a message that credentials are invalid.
  - Patron re-enters his credentials.
  - If the login is successful, the flow returns to step 6.
  - If the patron cancels the login, the flow returns to step 2
  - If the login is unsuccessful, the flow returns to alternative flow 6a, step 1.

Scenario (1)

Storyboard

Design

USER STORY 2



1. Patron opens menu
2. System prompts menu
3. Patron selects meals for Friday (before 5 p.m. on Thursday)
4. Patron goes to order basket
5. System prompts a login screen
6. Patron gives correct credentials
7. System show screen for order location and time slot.
8. Patron selects standard time slot and non standard location: conference room
9. System shows confirmation screen for order.
10. Patron confirms order

Extension:

+

6.a. Alternative: Patron gives false credentials

Scenario (1) Storyboard Design References

**USER STORY 3**

1. Patron has chosen preferred meal(s) and has entered his correct credentials
2. System show screen for order location and time slot.
3. Patron selects standard location and non standard time slot: 1:30 p.m. - 2 p.m.
4. Patron click 'continue'
5. System shows confirmation screen for order

Scenario (1) Storyboard Design References

**USER STORY 4**

1. Patron has chosen preferred meal(s) and has entered his correct credentials
2. System show screen for order location and time slot.
3. Patron selects standard time slot and non standard location: office 02.45
4. Patron click 'continue'
5. System shows confirmation screen for order

Scenario (1) Storyboard Design References

**USER STORY 5**

1. Patron opens menu
2. System prompts menu
3. Patron selects meal for every day of the week (before Sunday, 5 p.m.)
4. Patron goes to order basket
5. System prompts a login screen
6. Patron gives correct credentials
7. System show screen for order location and time slot.
8. Patron selects standard time slot and non standard location: conference room
9. System shows confirmation screen for orders.
10. Patron confirms orders

**Extension:**

- 6.a. Alternative: Patron gives false credentials
1. System displays a message that credentials are invalid.
  2. Patron re-enters his credentials.
  3. If the login is successful, the flow returns to step 6.
  4. If the patron cancels the login, the flow returns to step 2

Scenario (1) Storyboard Design References

**USER STORY 6**

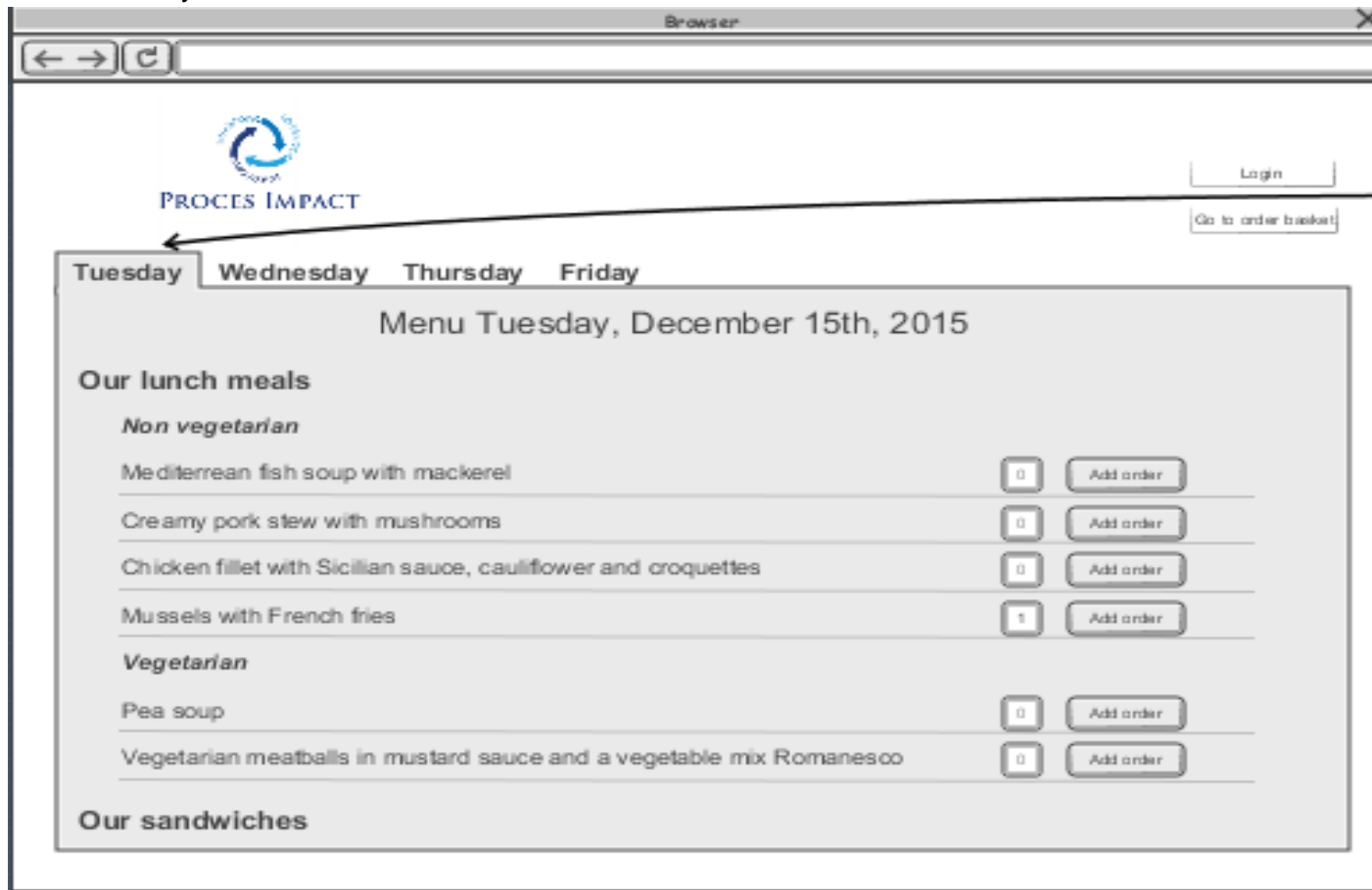
1. Stock manager opens application
2. System prompts opening screen
3. Stock manager clicks 'go to login' button.
4. System prompts a login screen
5. Patron gives correct credentials.
6. System shows screen with placed orders.
7. Stock manager chooses correct day and clicks 'generate list' button.
8. System prompts printable lists with ordered items.

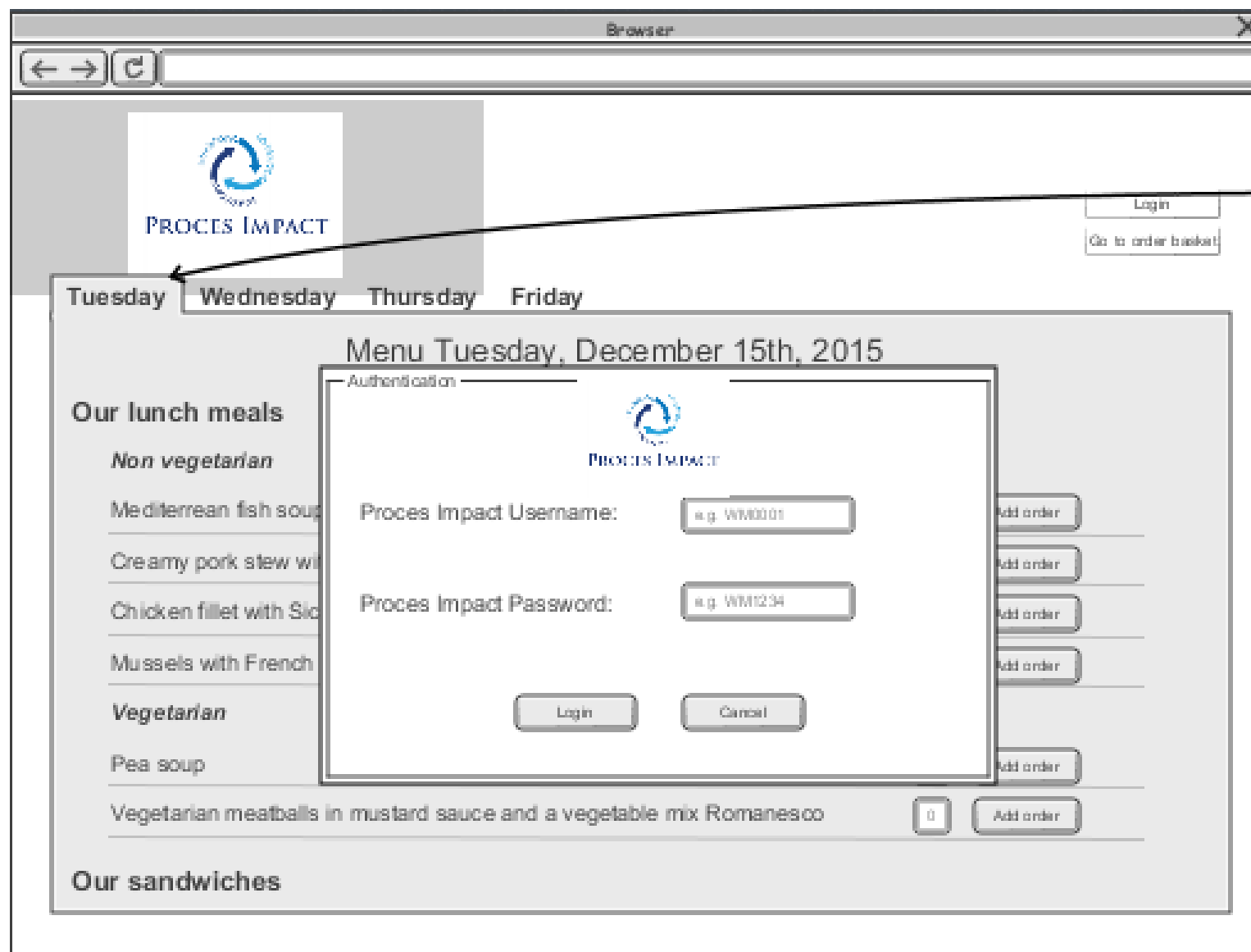
**Extension:**

- 5.a. Alternative: Patron gives false credentials
1. System displays a message that credentials are invalid.
  2. Patron re-enters his credentials.
  3. If the login is successful, the flow returns to step 6.
  4. If the patron cancels the login, the flow returns to step 2
  5. If the login is unsuccessful, the flow returns to alternative flow 6a, step 1.

## 2.4. Mock ups


- User Story 5:





Browser

← → ↻



PROCES IMPACT

Welcome, Wim Mass  
[Go to menu](#)

Choose meals

Choose location and time

Confirmation

Choose your location and time slot for delivery

Tuesday

Thursday

Location

☒ Standard location: office 2.45

☐ Non standard location

Time slot

☒ Standard time slot: 12:45 p.m. - 1 p.m.

☐ Non standard time slot


Continue

Cancel



Browser

← → ↻



PROCES IMPACT

Welcome, Wim Maas

Go to menu

Choose meals

Choose location and time

Confirmation

Confirm your order

Order for Tuesday, December 15, 2015

Meals:

Mussels with French fries (2x)

Sandwiches

Parini mozzarella and tomatoes (1x)

Location:

Office 2.45

Time slot:

12:45 p.m. - 1 p.m.

Order for Thursday, December 17, 2015

Meals:

Spaghetti (2x)

Sandwiches

Parini mozzarella and tomatoes (2x)

Location:

Office 2.45

Confirm

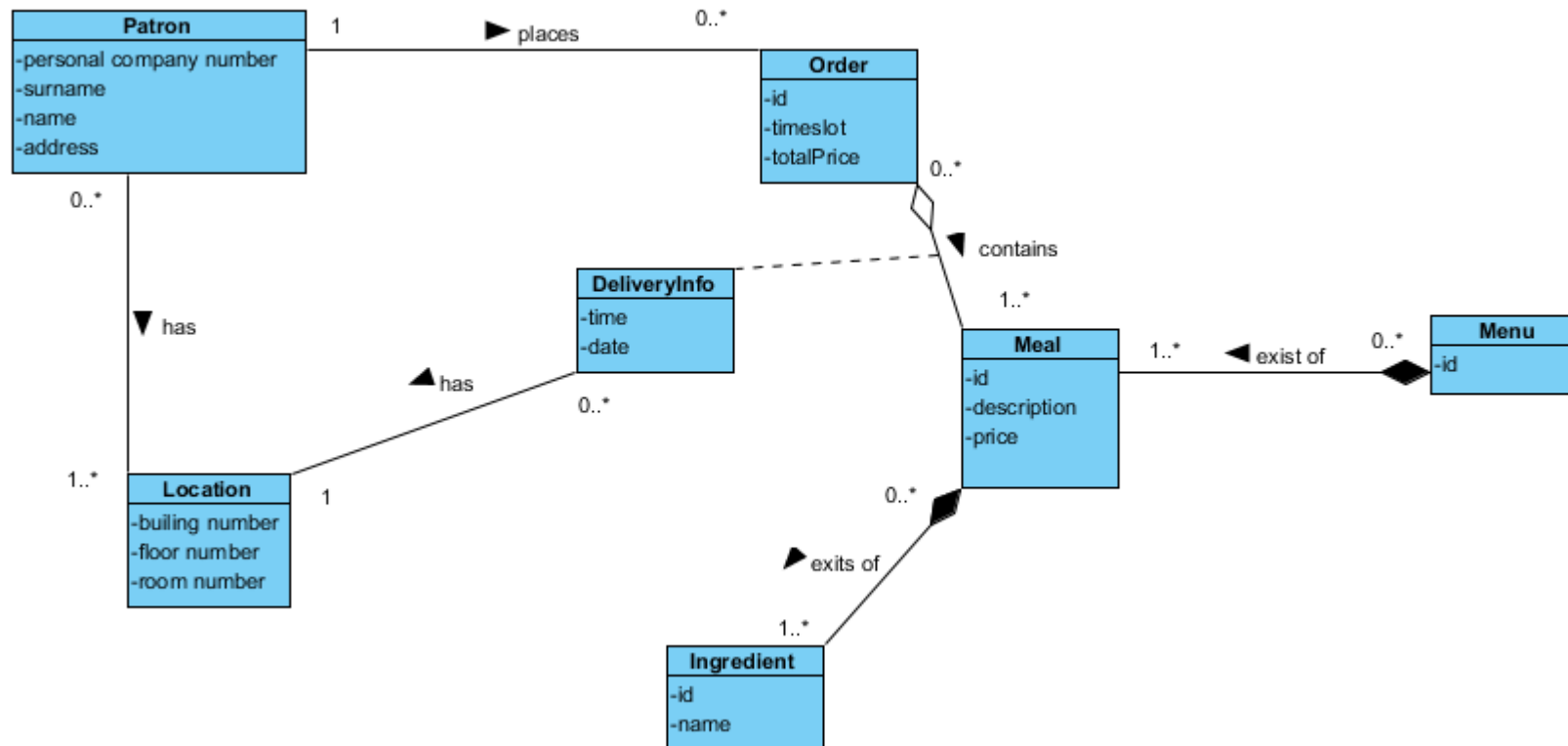
Cancel

41

### **3. Activity diagram for use case “order meal from cafeteria”**

A3

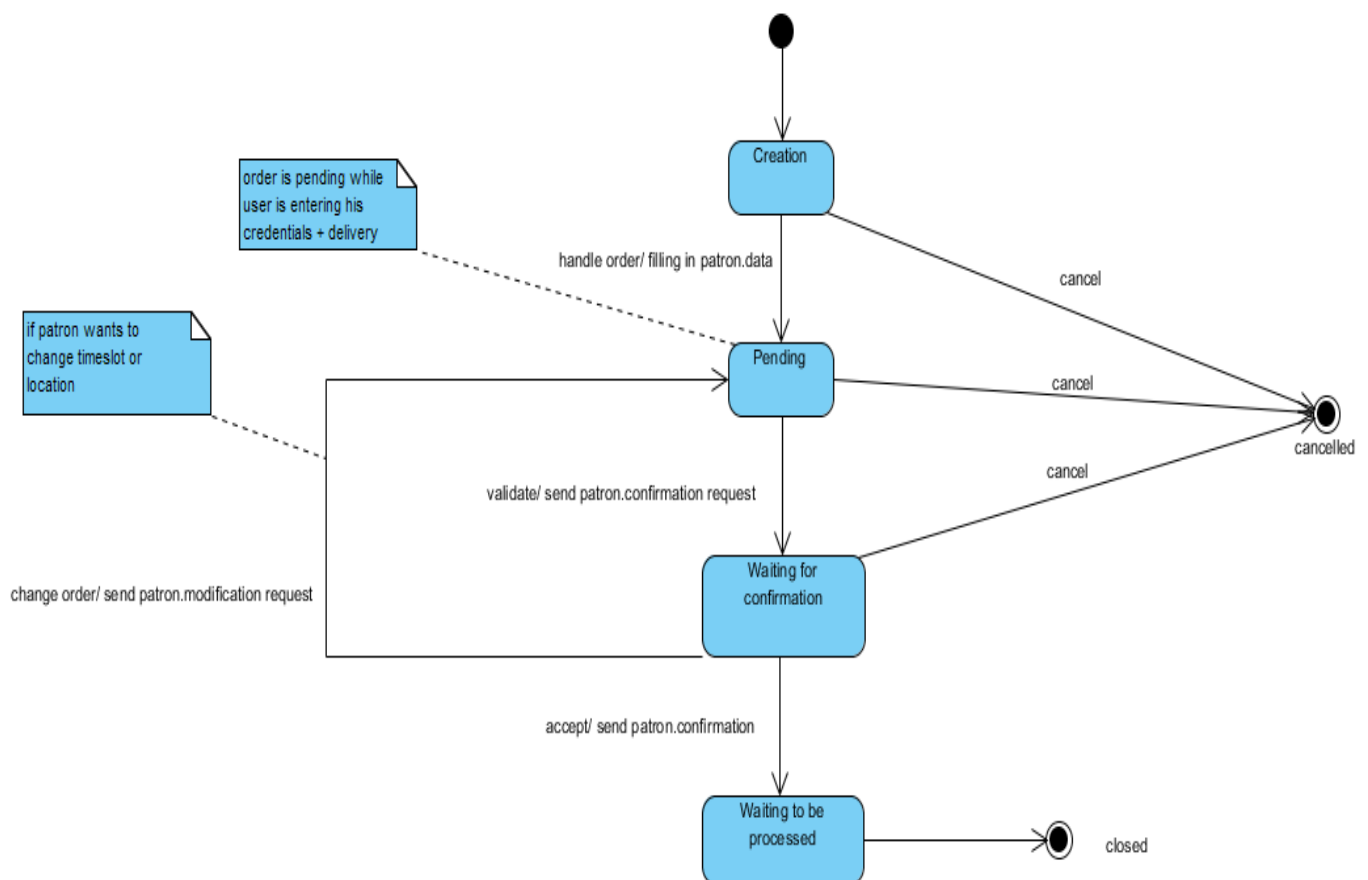
#### 4. Domain class diagrams for use “order meal from cafeteria”



## 5. CRUD matrix

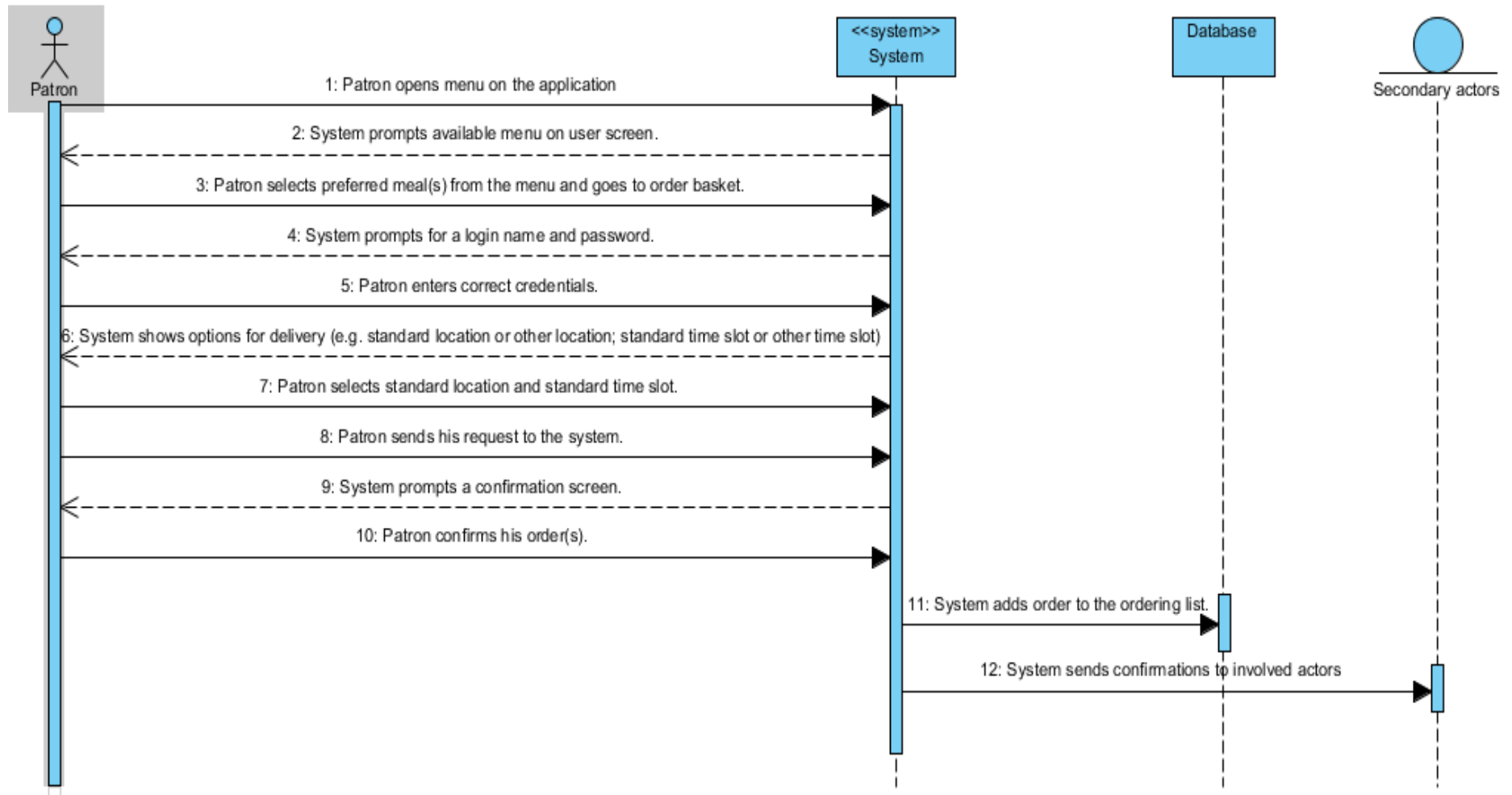
| Nr | User Action          | Create        | Read                      | Update        | Delete |
|----|----------------------|---------------|---------------------------|---------------|--------|
| 1  | Log in to the system |               | Users                     |               |        |
| 2  | Show menu            |               | Menu                      |               |        |
| 3  | Place order          | Order         | Menu<br>Location          |               |        |
| 4  | Change order         |               | Order<br>Menu<br>Location | Order         |        |
| 5  | Cancel order         |               | Order                     |               | Order  |
| 6  | Pay for order        | Order<br>User |                           | Order<br>User |        |
| 7  | Create menu          | Menu          | Meals                     |               |        |
| 8  | Create Meal          | Meal          | Ingredients               |               |        |

## 6. State chart diagram for domain class “Order”



## V. Assignment 4: Technical analysis – use case realization

### 1. System sequence diagram for main success scenario “order meal from cafeteria”

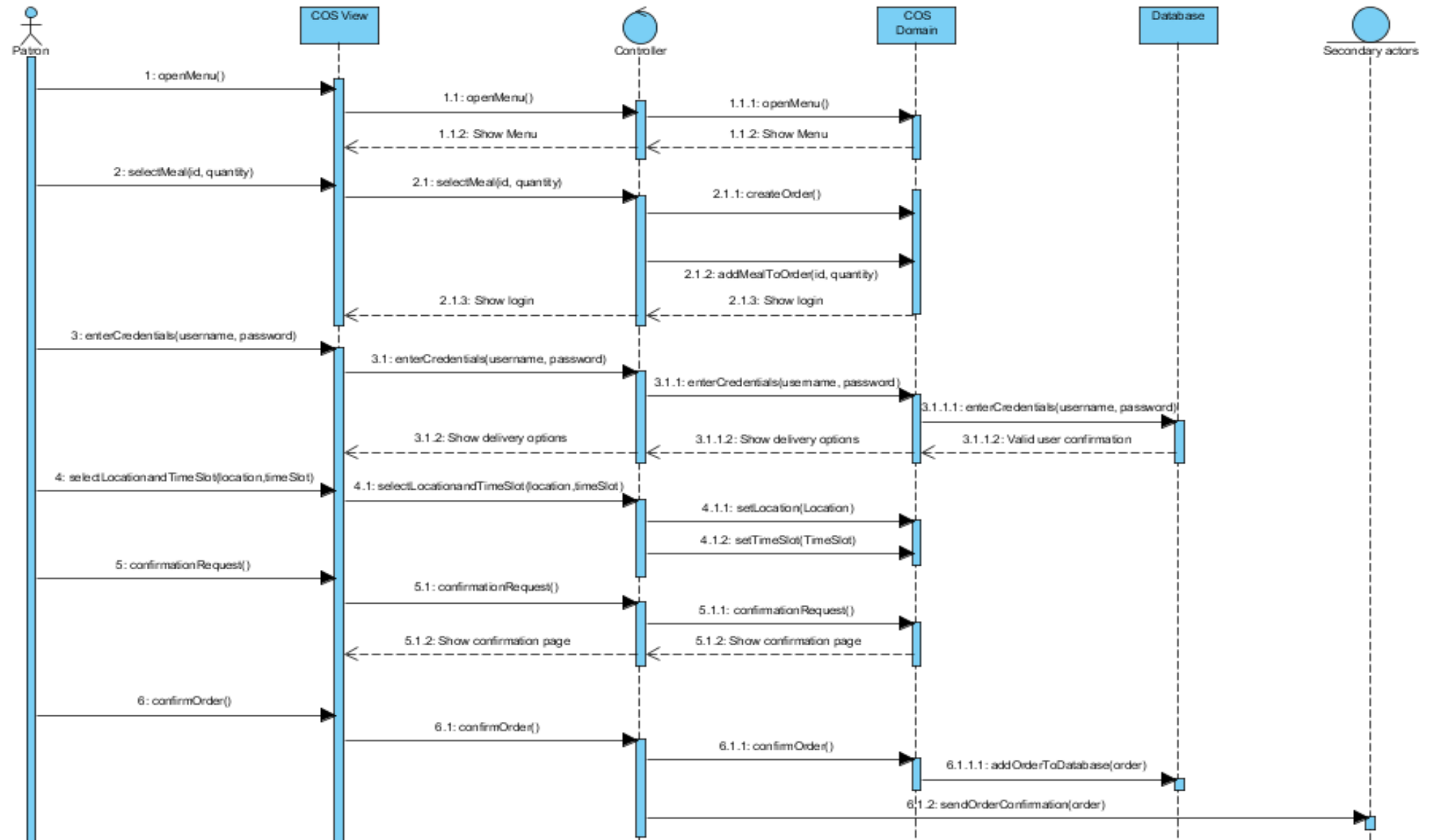


## 2. Operation contracts

| Name             | Responsibilities  | References              | Preconditions  | PostconditionsOrd  |
|------------------|---|-------------------------|--|--|
| openMenu         | List al Meals concerning the Menu of the week                             | Menu of the week exists | - Meals are created<br>- Meals are linked with Menu with a certain weeknumber                                  | Menu is shown  |
| enterCredentials | Create a session with the server to identify the user of the application. | Authenticate use case   | - Patron needs to have credentials for the company portal  | Patron is identified   |
| orderMeal        | A specified Meal is ordered for a specified location and time-slot.       | Order meal use case     | - Meals are created<br>- Meals are linked Menu with a certain weeknumber<br>- Patron is logged into the system | - A new order is created<br>- A Location and TimeSlot are defined in order<br>- The created order is added to the order list |
| cancelOrder      | A specified Order is cancelled.   | Cancel order use case   | - Order (identified by ID) is placed<br>- Patron is logged in (is identified)                                  | Order is cancelled.<br>When the cancellation time is too late, a payement will be noted.                                     |
| changeOrder      | A specified Order is changed.   | Change order use case   | - Order (identified by ID) is placed<br>- Patron is logged in (is identified)                                  | Only if the time slot allows a change, the order will be changed. Otherwise the order will keep his state.                   |
| checkOrders      | The list with orders is supplied.   | Check orders use case   | - Patron (→ cafeteria staff) is logged in<br>- Orders are created and are added to the orderlist               | Patron gets a list of al the orders for the next day.  |
| prepareOrders    | Orders which are prepared get the state "prepared"                        | Prepare order use case  | - Patron (→ cafeteria staff) is logged in<br>- Orders are created and are added to the orderlist               | Orderstate changes to prepared.  |
| deliverOrders    | Orders ready to be  | Deliver order use       | - Patron (→ cafeteria staff) is  | Orderstat changes to inDelivery  |

| Name                  | Responsibilities   | References       | Preconditions  | PostconditionsOrd   |
|-----------------------|--|------------------|--|---|
|                       | delivered get the state "inDelivery"   | case             | logged in<br>- Orders are created and are added to the orderlist<br>- Orders are prepared  |   |
| orderIsDelivered      | Orders who are delivered get the state "delivered".                                    | Payment use case | - Patron (→ deliverer) is logged in<br>- Orders are created and are added to the orderlist<br>- Order are prepared<br>- Order is delivered | - Order is delivered<br>- Payment is noted                  |
| externalOrdersCheckIn | Meals ordered at a local restaurant are delivered to be divided by the cafeteria staff | /                | - Order is placed  | - Order is accepted by the cafeteria and will be delivered. |

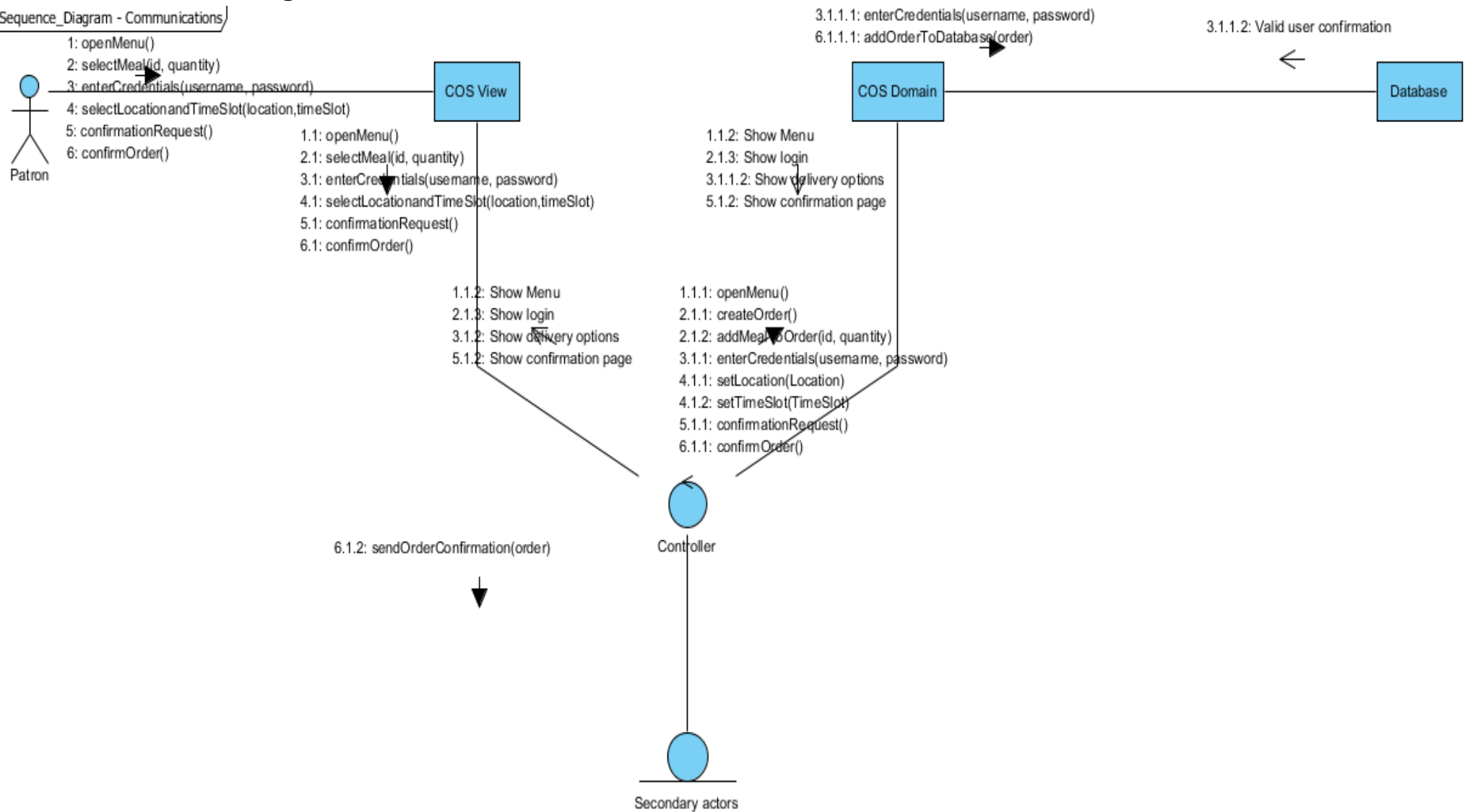
### 3. MVC Sequence diagram for main success scenario “order meal from cafeteria”



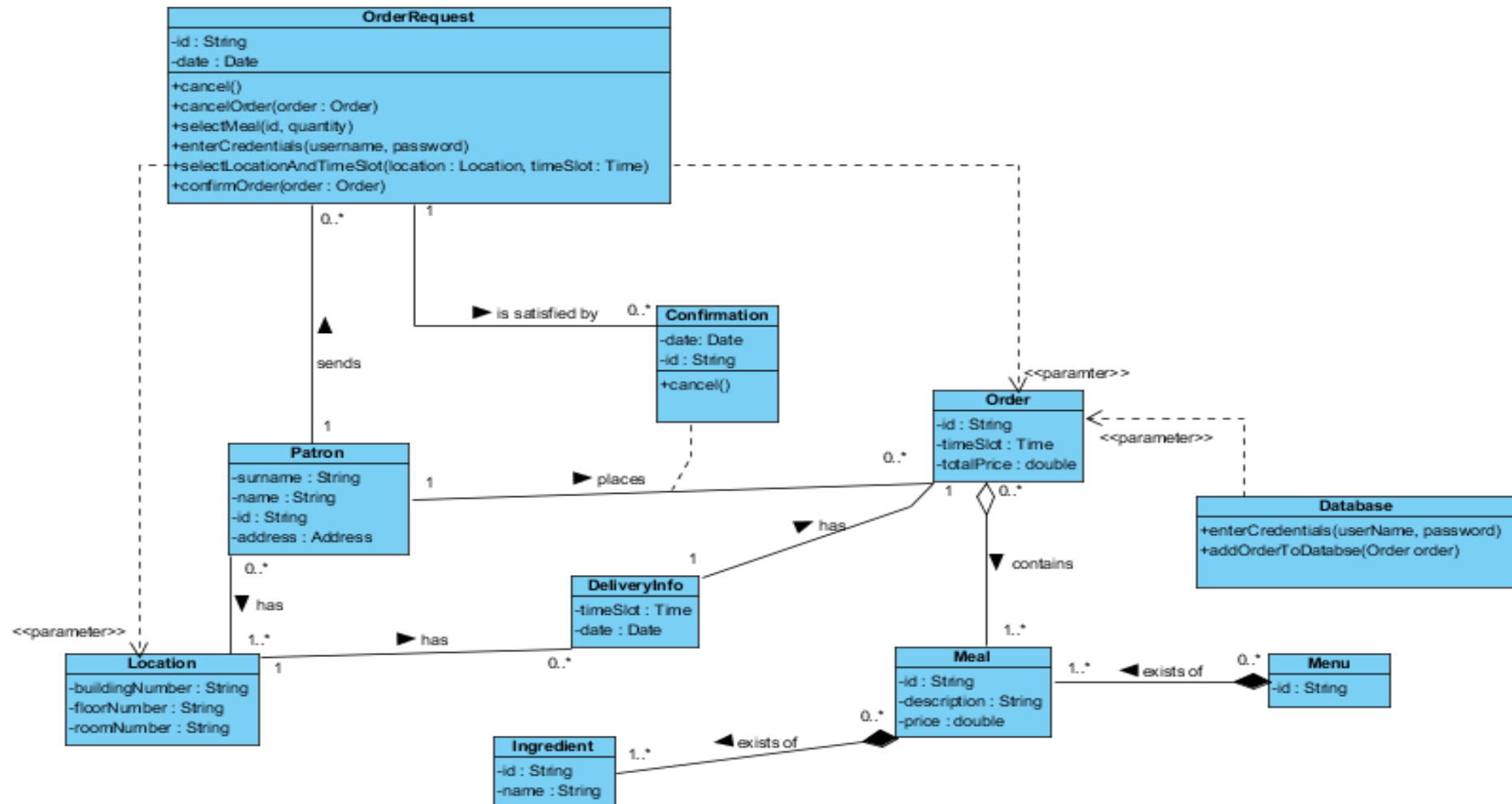


#### 4. Communication diagram for main success scenario “order meal from cafeteria”

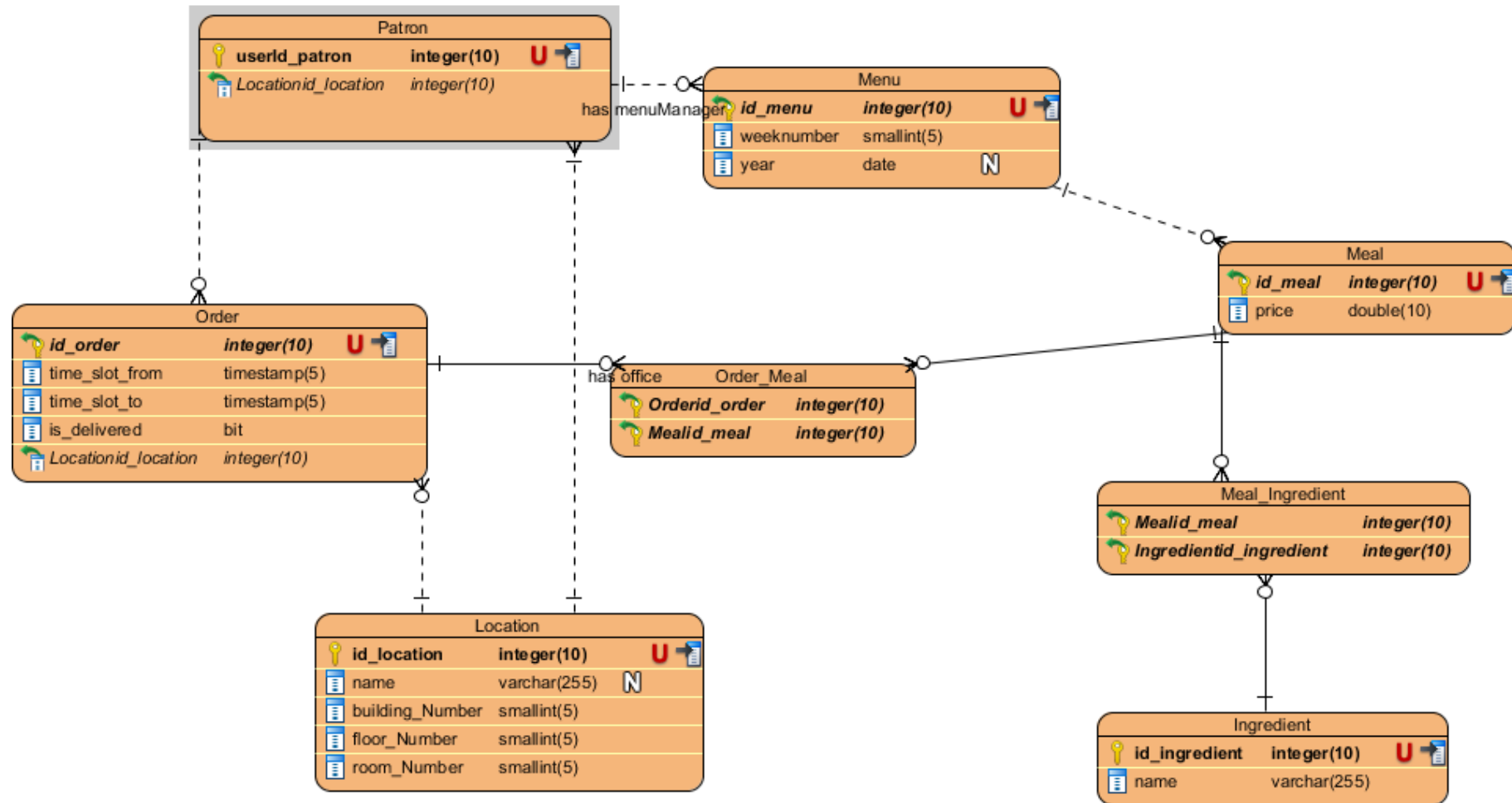
Sequence\_Diagram - Communications



## 5. Design class diagram for main success scenario “order meal from cafeteria”



## 6. Normalized data model



### ***Written version of ERD***

USER → extern ldap server with a login portal

- id\_location → heeft 1

#### **ORDER**

- id\_order (serial)
- id\_location → heeft 1
- time\_slot\_from (timestamp)
- time\_slot\_to (timestamp)
- is\_delivered (boolean)

#### **MENU**

- id\_menu (serial)
- id\_meal -> heeft meerdere
- weeknumber (smallint)
- year (date)
- id\_user (serial) -> heeft 1 (menu manager)

#### **ORDER\_MEAL**

- id\_order → behoort tot 1 tot veel
- id\_meal → heeft 1 tot meerdere

#### **MEAL**

- id\_meal (serial)
- price (double)

#### **MEAL\_INGREDIENT**

- id\_meal → behoort tot 1 tot veel
- id\_ingredient → heeft 1 tot veel

#### **INGREDIENT**

- id\_ingredient (serial)
- name (varchar)

#### **LOCATION**

- id\_location (serial)
- name (varchar)
- building\_Number (smallint)
- floor\_Number (smallint)
- room\_Number