

# Model

*Package in package "*

Model  
Version Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

# Activity

*Package in package 'Model'*

Activity  
Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

# Activity diagram

*Activity diagram in package 'Activity'*

Activity  
Version 1.0  
student created on 13.11.2023. Last modified 13.11.2023

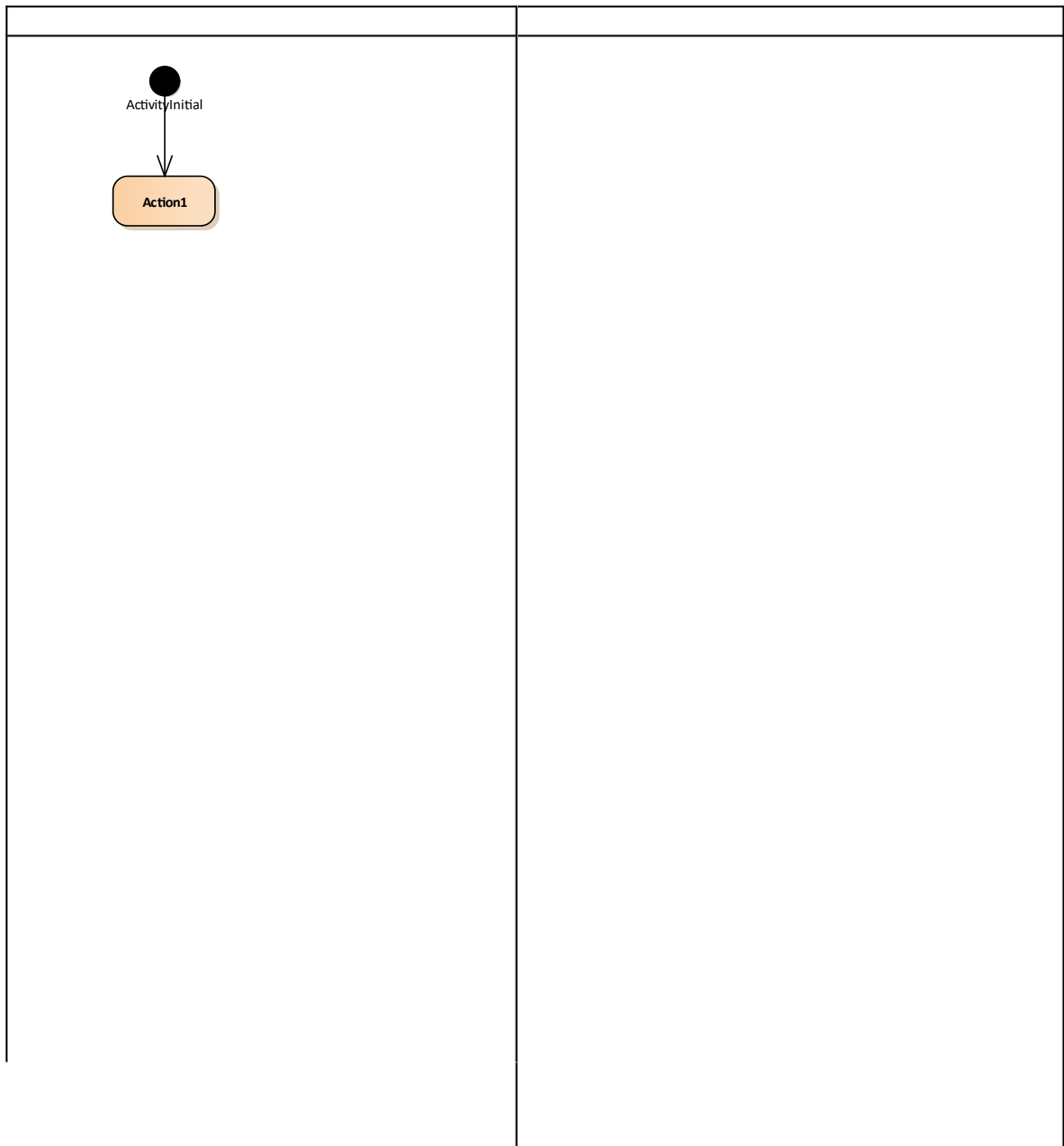




Figure 1: Activity

## ActivityPartition

*ActivityPartition in package 'Activity'*

ActivityPartition  
Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023


ELEMENTS OWNED BY

**ELEMENTS OWNED BY** Action1 : Action ActivityInitial : ActivityInitial

## Action1


Action in package 'Activity'

Action1  
Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

**INCOMING BEHAVIORAL RELATIONSHIPS** ControlFlow from ActivityInitial to Action1

## ActivityInitial

ActivityInitial in package 'Activity'

**OUTGOING BEHAVIORAL RELATIONSHIPS** ControlFlow from ActivityInitial to Action1

## ActivityPartition

ActivityPartition in package 'Activity'

ActivityPartition  
Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

## Requirements

*Package in package 'Model'*

Requirements  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

## Functional req

*Package in package 'Requirements'*

Functional req  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

## User Management

*Package in package 'Functional req'*

User Management  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

## User Management diagram

*Requirements diagram in package 'User Management'*

User Management  
Version 1.0  
student created on 09.10.2023. Last modified 09.10.2023

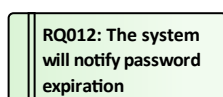


Figure 2: User Management

## RQ012: The system will notify password expiration

*Requirement «Functional» in package 'User Management'*

System will display an error message, when user password expiration deadline is out of limit.

RQ012: The system will notify password expiration  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

## Invoice Management

Package in package 'Functional req'

Invoice Management  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

### Invoice Management diagram

Requirements diagram in package 'Invoice Management'

Invoice Management  
Version 1.0  
student created on 09.10.2023. Last modified 09.10.2023

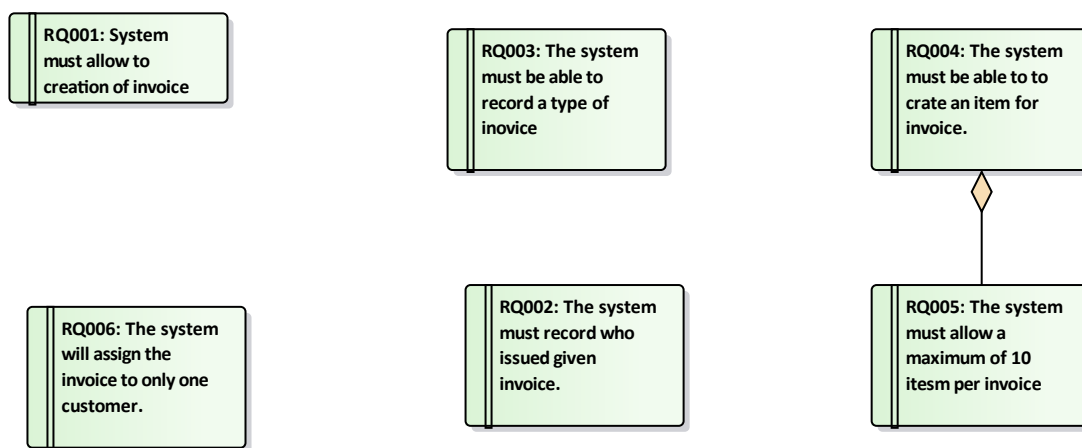


Figure 3: Invoice Management

### RQ001: System must allow to creation of invoice

Requirement «Functional» in package 'Invoice Management'

Unique invoice number will exit for each invoice. System must also record the date of issue.

RQ001: System must allow to creation of invoice  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

### RQ002: The system must record who issued given invoice.

Requirement «Functional» in package 'Invoice Management'

For each invoice, within the parameters, system will record and employee id who is issuing the invoice.

RQ002: The system must record who issued given invoice.  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

### RQ003: The system must be able to record a type of invoice

*Requirement «Functional» in package 'Invoice Management'*

The system will recognize two types of invoices

- 1) received invoices
- 2) issued invoices

RQ003: The system must be able to record a type of invoice  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

**RQ004: The system must be able to create an item for invoice.***Requirement «Functional» in package 'Invoice Management'*

For each invoice an item will be described as:

item number, name, item price, units, number of units.

RQ004: The system must be able to create an item for invoice.  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

**INCOMING STRUCTURAL RELATIONSHIPS**

⇒ Aggregation from «Functional» RQ005: The system must allow a maximum of 10 items per invoice to «Functional»  
RQ004: The system must be able to create an item for invoice.

[ Direction is 'Source -> Destination'. ]

**RQ005: The system must allow a maximum of 10 items per invoice***Requirement «Functional» in package 'Invoice Management'*

In internal directions, there is a limitation of 10 items per issued invoice.

RQ005: The system must allow a maximum of 10 items per invoice  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

**OUTGOING STRUCTURAL RELATIONSHIPS**

⇐ Aggregation from «Functional» RQ005: The system must allow a maximum of 10 items per invoice to «Functional»  
RQ004: The system must be able to create an item for invoice.

[ Direction is 'Source -> Destination'. ]

**RQ006: The system will assign the invoice to only one customer.***Requirement «Functional» in package 'Invoice Management'*

Customer association to invoice is unique and only one customer will be selected for each issued invoice. Customer needs to be in company address book.

RQ006: The system will assign the invoice to only one customer.

Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

## Customer Management

Package in package 'Functional req'

Customer Management  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

### Customer Management diagram

Requirements diagram in package 'Customer Management'

Customer Management  
Version 1.0  
student created on 09.10.2023. Last modified 09.10.2023

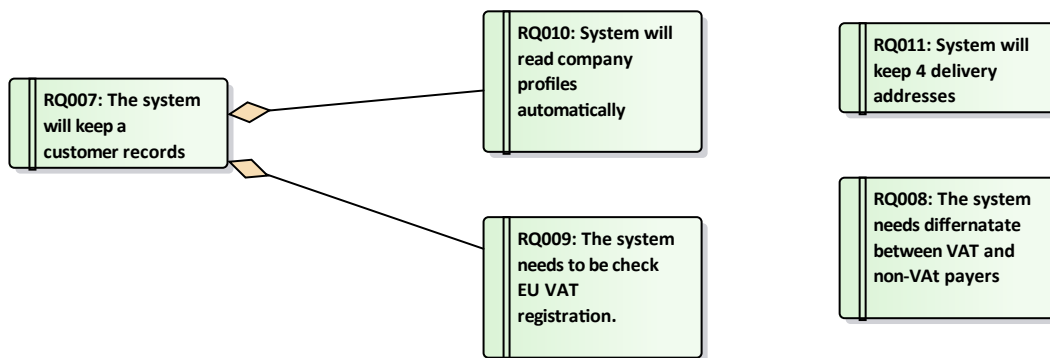


Figure 4: Customer Management

### RQ007: The system will keep a customer records

Requirement «Functional» in package 'Customer Management'

For each customer system will be able to record a legal name, address, identification number, value added tax identification number.

RQ007: The system will keep a customer records  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

#### INCOMING STRUCTURAL RELATIONSHIPS

⇒ Aggregation from «Functional» RQ009: The system needs to be check EU VAT registration. to «Functional» RQ007: The system will keep a customer records

[ Direction is 'Source -> Destination'. ]

⇒ Aggregation from «Functional» RQ010: System will read company profiles automatically to «Functional» RQ007: The system will keep a customer records

[ Direction is 'Source -> Destination'. ]

### RQ008: The system needs diffrenatate between VAT and non-VAt payers

Requirement «Functional» in package 'Customer Management'



It is need to record a VAT id for customers which VAT obligation and for non-VAT payers.

RQ008: The system needs diffrenatate between VAT and non-VAt payers  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

### RQ009: The system needs to be check EU VAT registration.

*Requirement «Functional» in package 'Customer Management'*

System will check a VAT registration using EU VAT ID Checking API.

RQ009: The system needs to be check EU VAT registration.  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

#### OUTGOING STRUCTURAL RELATIONSHIPS

← Aggregation from «Functional» RQ009: The system needs to be check EU VAT registration. to «Functional» RQ007: The system will keep a customer records

[ Direction is 'Source -> Destination'. ]

### RQ010: System will read company profiles automatically

*Requirement «Functional» in package 'Customer Management'*

For a customers, whitin a CZ systems needs to be able to read their profiles - legal names, addresses (business) from CZ national business register.

RQ010: System will read company profiles automatically  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

#### OUTGOING STRUCTURAL RELATIONSHIPS

← Aggregation from «Functional» RQ010: System will read company profiles automatically to «Functional» RQ007: The system will keep a customer records

[ Direction is 'Source -> Destination'. ]

### RQ011: System will keep 4 delivery addresses

*Requirement «Functional» in package 'Customer Management'*

For each customer profile the system will be able to keep up to 4 addresses for delivering. One of them will be set as primary.

RQ011: System will keep 4 delivery addresses  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

## Non-functional req

*Package in package 'Requirements'*

Non-functional req  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

## Use Case

Package in package 'Model'

Use Case  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

## Actors

Package in package 'Use Case'

Actors  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

## Actors diagram

Use Case diagram in package 'Actors'

Actors  
Version 1.0  
student created on 09.10.2023. Last modified 09.10.2023

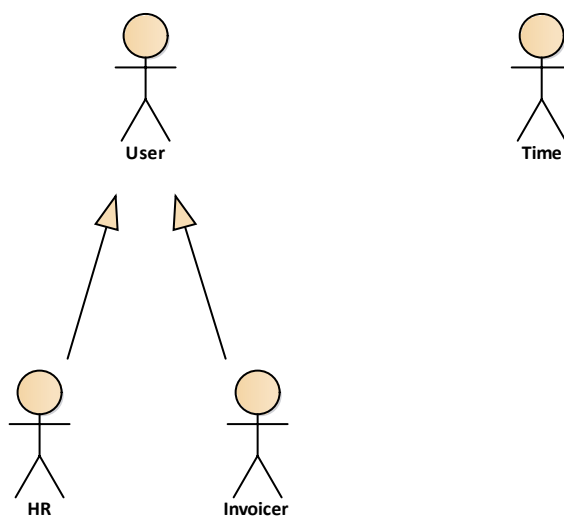



Figure 5: Actors

## HR

Actor in package 'Actors'

HR  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

**OUTGOING STRUCTURAL RELATIONSHIPS**
 Generalization from HR to User

[ Direction is 'Source -&gt; Destination'. ]

**ASSOCIATIONS**
 Association (direction: Unspecified)

Source: Public (Actor) HR


Target: Public (UseCase) Create a user

**Invoicer***Actor in package 'Actors'*


Invoicer  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

**OUTGOING BEHAVIORAL RELATIONSHIPS**

Name: issueInvoice()

 Sequence from Invoicer to Object
**OUTGOING STRUCTURAL RELATIONSHIPS**
 Generalization from Invoicer to User

[ Direction is 'Source -&gt; Destination'. ]

**INCOMING BEHAVIORAL RELATIONSHIPS**
 Sequence from Object to Invoicer
**ASSOCIATIONS**
 Association (direction: Unspecified)

Source: Public (Actor) Invoicer

Target: Public (UseCase) Create a customer entry

 Association (direction: Unspecified)

Source: Public (Actor) Invoicer

Target: Public (UseCase) Issue an invoice

 Association (direction: Unspecified)

Source: Public (Actor) Invoicer

Target: Public (UseCase) Add item to invoice

## Time

Actor in package 'Actors'

Time  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

### ASSOCIATIONS



Association (direction: Unspecified)

Source: Public (Actor) Time

Target: Public (UseCase) Generate a password notification

## User

Actor in package 'Actors'

User  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

### INCOMING STRUCTURAL RELATIONSHIPS



Generalization from Invoicer to User

[ Direction is 'Source -> Destination'. ]



Generalization from HR to User

[ Direction is 'Source -> Destination'. ]

### ASSOCIATIONS



Association (direction: Unspecified)

Source: Public (Actor) User

Target: Public (UseCase) Change a password

## Basic Use Case Model

Package in package 'Use Case'

Basic Use Case Model  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

### Basic Use Case Model diagram

Use Case diagram in package 'Basic Use Case Model'

Basic Use Case Model  
Version 1.0  
student created on 09.10.2023. Last modified 16.10.2023

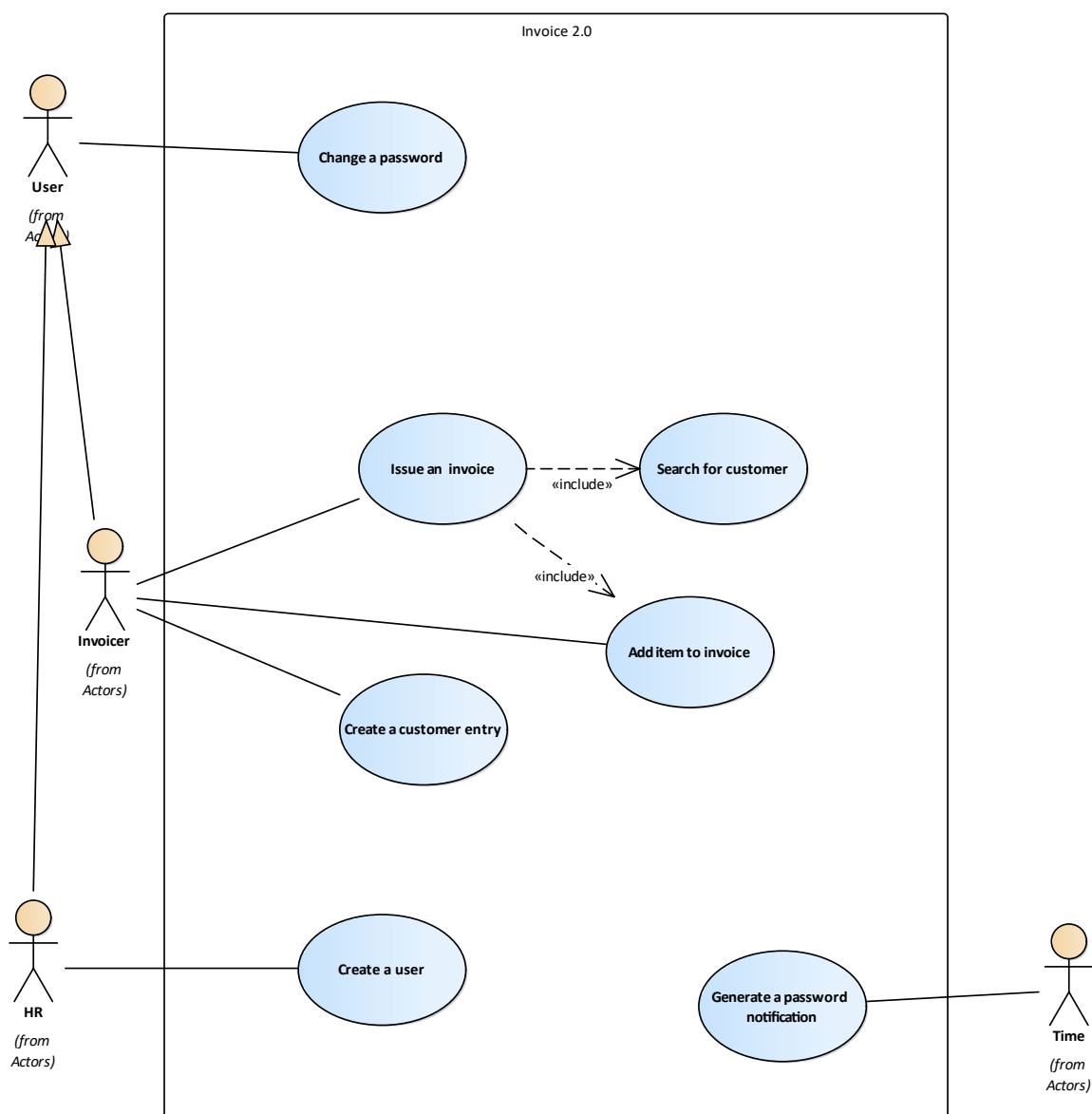


Figure 6: Basic Use Case Model

## HR

Actor in package 'Actors'

HR  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

**OUTGOING STRUCTURAL RELATIONSHIPS**

← Generalization from HR to User

[ Direction is 'Source -> Destination'. ]

**ASSOCIATIONS**

✎ Association (direction: Unspecified)

Source: Public (Actor) HR

Target: Public (UseCase) Create a user

## Invoicer

Actor in package 'Actors'

Invoicer  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

**OUTGOING BEHAVIORAL RELATIONSHIPS**

Name: issueInvoice()

⇨ Sequence from Invoicer to Object

**OUTGOING STRUCTURAL RELATIONSHIPS**

← Generalization from Invoicer to User

[ Direction is 'Source -> Destination'. ]

**INCOMING BEHAVIORAL RELATIONSHIPS**

⇨ Sequence from Object to Invoicer

**ASSOCIATIONS**

✎ Association (direction: Unspecified)

Source: Public (Actor) Invoicer

Target: Public (UseCase) Create a customer entry

✎ Association (direction: Unspecified)

Source: Public (Actor) Invoicer

Target: Public (UseCase) Issue an invoice

✎ Association (direction: Unspecified)

**ASSOCIATIONS**

Source: Public (Actor) Invoicer

Target: Public (UseCase) Add item to invoice

## Time

*Actor in package 'Actors'*

Time

Version 1.0 Phase 1.0 Proposed

student created on 09.10.2023. Last modified 09.10.2023

**ASSOCIATIONS** Association (direction: Unspecified)

Source: Public (Actor) Time

Target: Public (UseCase) Generate a password notification


## User

*Actor in package 'Actors'*


User

Version 1.0 Phase 1.0 Proposed

student created on 09.10.2023. Last modified 09.10.2023

**INCOMING STRUCTURAL RELATIONSHIPS** Generalization from Invoicer to User

[ Direction is 'Source -&gt; Destination'. ]

 Generalization from HR to User

[ Direction is 'Source -&gt; Destination'. ]

**ASSOCIATIONS** Association (direction: Unspecified)

Source: Public (Actor) User

Target: Public (UseCase) Change a password

## Invoice 2.0

*Boundary in package 'Basic Use Case Model'*

Invoice 2.0

Version 1.0 Phase 1.0 Proposed

student created on 09.10.2023. Last modified 09.10.2023

Extends




## Add item to invoice


UseCase in package 'Basic Use Case Model'

Add item to invoice  
Version 1.0 Phase 1.0 Proposed  
student created on 16.10.2023. Last modified 16.10.2023

### ELEMENTS OWNED BY Add item to invoice

 Add item to invoice\_ActivityGraph : Activity

### CONNECTORS

 **Include** «include» Source -> Destination  
From: Issue an invoice : UseCase, Public  
To: Add item to invoice : UseCase, Public

### ASSOCIATIONS

 Association (direction: Unspecified)

Source: Public (Actor) Invoicer

Target: Public (UseCase) Add item to invoice

## Add item to invoice\_ActivityGraph

Activity owned by 'Add item to invoice', in package 'Basic Use Case Model'

Add item to invoice\_ActivityGraph  
Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

## Create a customer entry

UseCase in package 'Basic Use Case Model'

Create a customer entry  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

### ASSOCIATIONS

 Association (direction: Unspecified)

Source: Public (Actor) Invoicer

Target: Public (UseCase) Create a customer entry

## Create a user

UseCase in package 'Basic Use Case Model'

Create a user  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

### ASSOCIATIONS



Association (direction: Unspecified)

Source: Public (Actor) HR

Target: Public (UseCase) Create a user

## Generate a password notification

UseCase in package 'Basic Use Case Model'

Generate a password notification  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

### ASSOCIATIONS



Association (direction: Unspecified)

Source: Public (Actor) Time

Target: Public (UseCase) Generate a password notification

## Change a password

UseCase in package 'Basic Use Case Model'

Change a password  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023

### ASSOCIATIONS



Association (direction: Unspecified)

Source: Public (Actor) User

Target: Public (UseCase) Change a password

## Issue an invoice

UseCase in package 'Basic Use Case Model'


Issue an invoice  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 13.11.2023

### ELEMENTS OWNED BY Issue an invoice


**ELEMENTS OWNED BY Issue an invoice**

 Issue an invoice\_ActivityGraphWithAction : Activity

**SCENARIOS**


 Basic Path. Basic Path


1. The use case starts when invoicer select a new invoice option.
2. The system generates an invoice number and displays the form new invoice.
3. The invoicer selects a customer, to whoam the invoice will be issued for.
4. <include> Search for customer
5. The invoicer will add a text - what is invoiced, variable symbol, paemtn due date, tax limits.  
Alternate: 5a. Invoice with items
6. Use case ends with saving the invoice.

 Alternate. Invoice with items

1. The invoicer selectrs the options to add an item.
2. <include> Add item to invoice
3. System will peform updates in invoice overview and perform recalculation of the prices.

**CONNECTORS**

 **Include** «include» Source -> Destination  
From: Issue an invoice : UseCase, Public  
To: Add item to invoice : UseCase, Public

 **Include** «include» Source -> Destination  
From: Issue an invoice : UseCase, Public  
To: Search for customer : UseCase, Public

**ASSOCIATIONS**

 Association (direction: Unspecified)

Source: Public (Actor) Invoicer


Target: Public (UseCase) Issue an invoice





**Issue an invoice\_ActivityGraphWithAction**

*Activity owned by 'Issue an invoice', in package 'Basic Use Case Model'*

Issue an invoice\_ActivityGraphWithAction  
Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

**ELEMENTS OWNED BY Issue an invoice\_ActivityGraphWithAction**

 <include> Add item to invoice : CallBehaviorAction

ELEMENTS OWNED BY Issue an invoice_ActivityGraphWithAction
 <include> Search for customer : CallBehaviorAction
 System will perform updates in invoice overview and perform recalculation of the prices. : Action
 The invoicer selects the options to add an item. : Action
 The invoicer selects a customer, to whom the invoice will be issued for. : Action
 The invoicer will add a text - what is invoiced, variable symbol, payment due date, tax limits. : Action
 The system generates an invoice number and displays the form new invoice. : Action
 The use case starts when invoicer select a new invoice option. : Action
 Use case ends with saving the invoice. : Action
 Alternate1 : DecisionNode
 End : ActivityFinal
 Start : ActivityInitial

### ***Issue an invoice\_ActivityGraphWithAction diagram***

*Activity diagram in package 'Basic Use Case Model'*

Issue an invoice\_ActivityGraphWithAction  
Version 1.0  
student created on 13.11.2023. Last modified 13.11.2023

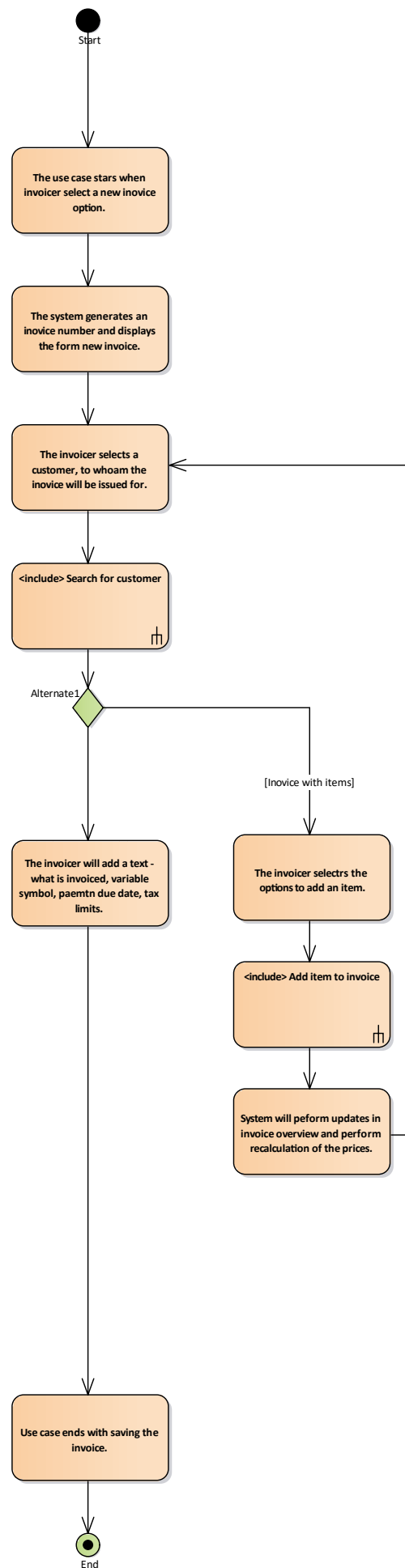


Figure 7: Issue an invoice\_ActivityGraphWithAction

**<include> Add item to invoice**

CallBehaviorAction owned by 'Issue an invoice\_ActivityGraphWithAction', in package 'Basic Use Case Model'

<include> Add item to invoice

Version 1.0 Phase 1.0 Proposed

student created on 13.11.2023. Last modified 13.11.2023

**OUTGOING BEHAVIORAL RELATIONSHIPS**

ControlFlow from <include> Add item to invoice to System will perform updates in invoice overview and perform recalculation of the prices.

**INCOMING BEHAVIORAL RELATIONSHIPS**

ControlFlow from The invoicer selectrs the options to add an item. to <include> Add item to invoice

**<include> Search for customer**

CallBehaviorAction owned by 'Issue an invoice\_ActivityGraphWithAction', in package 'Basic Use Case Model'

<include> Search for customer

Version 1.0 Phase 1.0 Proposed

student created on 13.11.2023. Last modified 13.11.2023

**OUTGOING BEHAVIORAL RELATIONSHIPS**

ControlFlow from <include> Search for customer to Alternate1

**INCOMING BEHAVIORAL RELATIONSHIPS**

ControlFlow from The invoicer selects a customer, to whom the invoice will be issued for. to <include> Search for customer

**System will perform updates in invoice overview and perform recalculation of the prices.**

Action owned by 'Issue an invoice\_ActivityGraphWithAction', in package 'Basic Use Case Model'

System will perform updates in invoice overview and perform recalculation of the prices.

Version 1.0 Phase 1.0 Proposed

student created on 13.11.2023. Last modified 13.11.2023

**OUTGOING BEHAVIORAL RELATIONSHIPS**

ControlFlow from System will perform updates in invoice overview and perform recalculation of the prices. to The invoicer selects a customer, to whom the invoice will be issued for.

**INCOMING BEHAVIORAL RELATIONSHIPS**

ControlFlow from <include> Add item to invoice to System will perform updates in invoice overview and perform recalculation of the prices.

***The invoicer selectrs the options to add an item.***

Action owned by 'Issue an invoice\_ActivityGraphWithAction', in package 'Basic Use Case Model'

The invoicer selectrs the options to add an item.  
Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

**OUTGOING BEHAVIORAL RELATIONSHIPS**

ControlFlow from The invoicer selectrs the options to add an item. to <include> Add item to invoice

**INCOMING BEHAVIORAL RELATIONSHIPS**

ControlFlow from Alternate1 to The invoicer selectrs the options to add an item.  
Guard: Invoice with items

***The invoicer selects a customer, to whoam the invoice will be issued for.***

Action owned by 'Issue an invoice\_ActivityGraphWithAction', in package 'Basic Use Case Model'

The invoicer selects a customer, to whoam the invoice will be issued for.  
Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

**OUTGOING BEHAVIORAL RELATIONSHIPS**

ControlFlow from The invoicer selects a customer, to whoam the invoice will be issued for. to <include> Search for customer

**INCOMING BEHAVIORAL RELATIONSHIPS**

ControlFlow from System will perform updates in invoice overview and perform recalculation of the prices. to The invoicer selects a customer, to whoam the invoice will be issued for.

ControlFlow from The system generates an invoice number and displays the form new invoice. to The invoicer selects a customer, to whoam the invoice will be issued for.

***The invoicer will add a text - what is invoiced, variable symbol, paemtn due date, tax limits.***

Action owned by 'Issue an invoice\_ActivityGraphWithAction', in package 'Basic Use Case Model'

The invoicer will add a text - what is invoiced, variable symbol, paemtn due date, tax limits.

Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

**OUTGOING BEHAVIORAL RELATIONSHIPS**

ControlFlow from The invoicer will add a text - what is invoiced, variable symbol, paemtn due date, tax limits. to Use case ends with saving the invoice.

**INCOMING BEHAVIORAL RELATIONSHIPS**

ControlFlow from Alternatel to The invoicer will add a text - what is invoiced, variable symbol, paemtn due date, tax limits.

***The system generates an invoice number and displays the form new invoice.***

Action owned by 'Issue an invoice\_ActivityGraphWithAction', in package 'Basic Use Case Model'

The system generates an invoice number and displays the form new invoice.

Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

**OUTGOING BEHAVIORAL RELATIONSHIPS**

ControlFlow from The system generates an invoice number and displays the form new invoice. to The invoicer selects a customer, to whoam the invoice will be issued for.

**INCOMING BEHAVIORAL RELATIONSHIPS**

ControlFlow from The use case stars when invoicer select a new invoice option. to The system generates an invoice number and displays the form new invoice.

***The use case stars when invoicer select a new invoice option.***

Action owned by 'Issue an invoice\_ActivityGraphWithAction', in package 'Basic Use Case Model'

The use case stars when invoicer select a new invoice option.

Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

**OUTGOING BEHAVIORAL RELATIONSHIPS**

ControlFlow from The use case stars when invoicer select a new invoice option. to The system generates an invoice number and displays the form new invoice.

**INCOMING BEHAVIORAL RELATIONSHIPS**

ControlFlow from Start to The use case stars when invoicer select a new invoice option.



**Use case ends with saving the invoice.**

Action owned by 'Issue an invoice\_ActivityGraphWithAction', in package 'Basic Use Case Model'

Use case ends with saving the invoice.

Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

**OUTGOING BEHAVIORAL RELATIONSHIPS**

ControlFlow from Use case ends with saving the invoice. to End

**INCOMING BEHAVIORAL RELATIONSHIPS**

ControlFlow from The invoicer will add a text - what is invoiced, variable symbol, paemtn due date, tax limits. to Use case ends with saving the invoice.

**Alternate1**

DecisionNode owned by 'Issue an invoice\_ActivityGraphWithAction', in package 'Basic Use Case Model'

**OUTGOING BEHAVIORAL RELATIONSHIPS**

ControlFlow from Alternate1 to The invoicer will add a text - what is invoiced, variable symbol, paemtn due date, tax limits.

ControlFlow from Alternate1 to The invoicer selectrs the options to add an item.  
Guard: Invoice with items

**INCOMING BEHAVIORAL RELATIONSHIPS**

ControlFlow from <include> Search for customer to Alternate1

**End**

ActivityFinal owned by 'Issue an invoice\_ActivityGraphWithAction', in package 'Basic Use Case Model'

**INCOMING BEHAVIORAL RELATIONSHIPS**

ControlFlow from Use case ends with saving the invoice. to End

**Start**

ActivityInitial owned by 'Issue an invoice\_ActivityGraphWithAction', in package 'Basic Use Case Model'

**OUTGOING BEHAVIORAL RELATIONSHIPS**

ControlFlow from Start to The use case stars when invoicer select a new invoice option.


**OUTGOING BEHAVIORAL RELATIONSHIPS**

## Search for customer


UseCase in package 'Basic Use Case Model'

Search for customer  
Version 1.0 Phase 1.0 Proposed  
student created on 16.10.2023. Last modified 16.10.2023

**ELEMENTS OWNED BY Search for customer**

 Search for customer\_ActivityGraph : Activity

**CONNECTORS**

 **Include** «include» Source -> Destination  
From: Issue an invoice : UseCase, Public  
To: Search for customer : UseCase, Public

## Search for customer\_ActivityGraph

Activity owned by 'Search for customer', in package 'Basic Use Case Model'

Search for customer\_ActivityGraph  
Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

## Classes

Package in package 'Model'

Classes

Version 1.0 Phase 1.0 Proposed

student created on 30.10.2023. Last modified 30.10.2023

## Classes diagram

Class diagram in package 'Classes'

Classes

Version 1.0

student created on 30.10.2023. Last modified 30.10.2023

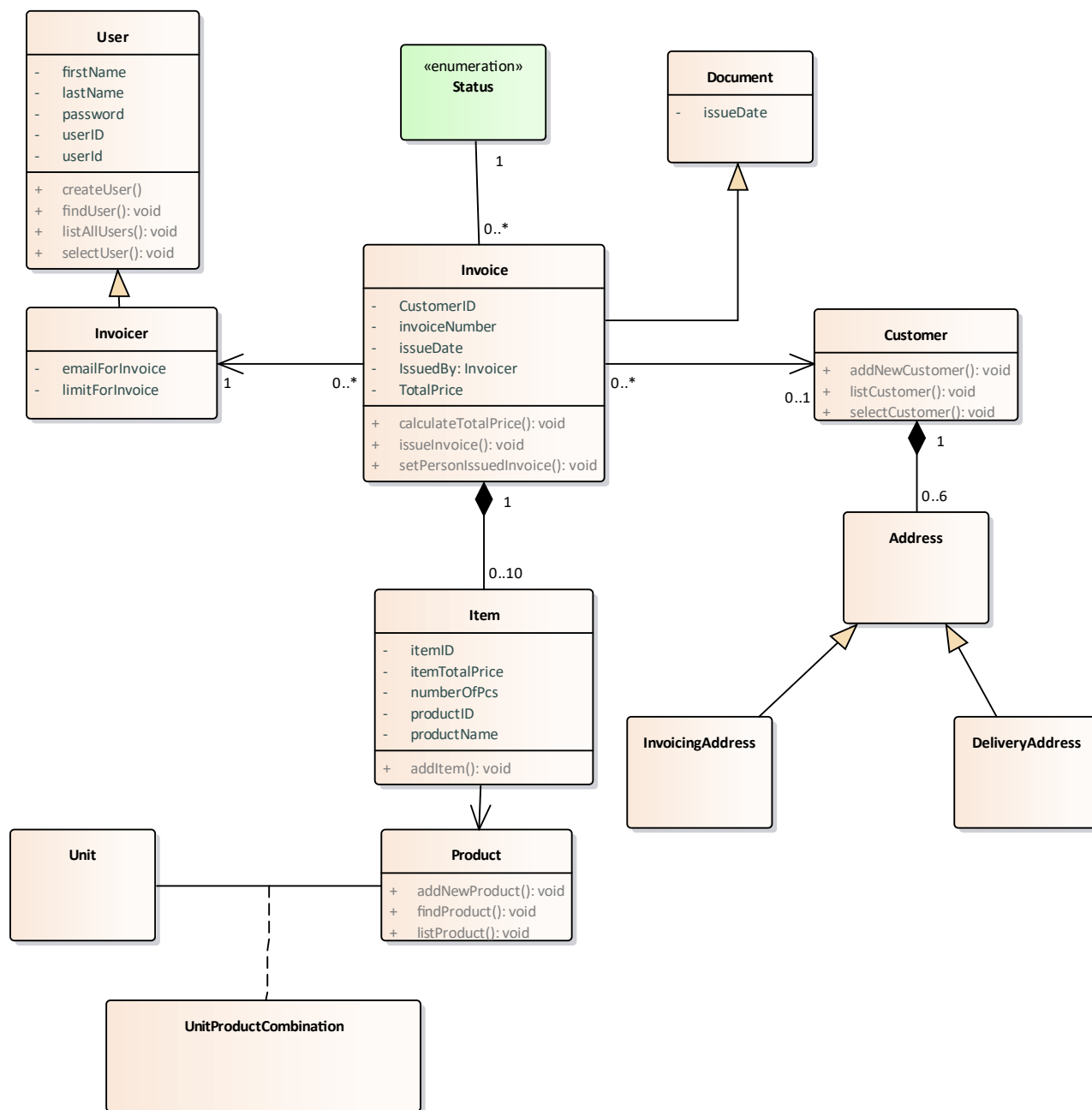


Figure 8: Classes

## UnitProductCombination

AssociationClass in package 'Classes'

UnitProductCombination  
Version 1.0 Phase 1.0 Proposed  
student created on 30.10.2023. Last modified 30.10.2023

## Address

Class in package 'Classes'

Address  
Version 1.0 Phase 1.0 Proposed  
student created on 30.10.2023. Last modified 30.10.2023

### INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Generalization from DeliveryAddress to Address [ Direction is 'Source -> Destination'. ]
- ⇒ Generalization from InvoicingAddress to Address [ Direction is 'Source -> Destination'. ]

### ASSOCIATIONS



-  Association (direction: Unspecified)
- Source: Public (Class) Address  
Cardinality: [0..6]
- Target: Public (Class) Customer  
Cardinality: [1]

## Customer

Class in package 'Classes'

Customer  
Version 1.0 Phase 1.0 Proposed  
student created on 30.10.2023. Last modified 30.10.2023

### ASSOCIATIONS

-  Association (direction: Source -> Destination)
- Source: Public (Class) Invoice  
Cardinality: [0..\*]
- Target: Public (Class) Customer  
Cardinality: [0..1]
-  Association (direction: Unspecified)
- Source: Public (Class) Address  
Cardinality: [0..6]
- Target: Public (Class) Customer  
Cardinality: [1]

## ASSOCIATIONS

## OPERATIONS

addNewCustomer () : void Public  
 [ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

listCustomer () : void Public  
 Properties:  
 ea\_guid = {DCA3AC31-411A-472a-B89A-E9A69A2EDE17}  
 [ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

selectCustomer () : void Public  
 Properties:  
 ea\_guid = {FA4DFC2E-A4D6-43c1-9311-9AD4F88BD727}  
 [ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

## DeliveryAddress

Class in package 'Classes'

DeliveryAddress  
 Version 1.0 Phase 1.0 Proposed  
 student created on 30.10.2023. Last modified 30.10.2023  
 Extends Address

## OUTGOING STRUCTURAL RELATIONSHIPS

Generalization from DeliveryAddress to Address  
 [ Direction is 'Source -> Destination'. ]

## Document

Class in package 'Classes'

Document  
 Version 1.0 Phase 1.0 Proposed  
 student created on 30.10.2023. Last modified 30.10.2023

## INCOMING STRUCTURAL RELATIONSHIPS

Generalization from Invoice to Document  
 [ Direction is 'Source -> Destination'. ]

## ATTRIBUTES

issueDate : Private  
 [ Is static True. Containment is Not Specified. ]

## ATTRIBUTES

## Invoice

*Class in package 'Classes'*

Invoice  
Version 1.0 Phase 1.0 Proposed  
student created on 30.10.2023. Last modified 30.10.2023  
Extends Document

## OUTGOING STRUCTURAL RELATIONSHIPS

← Generalization from Invoice to Document  
[ Direction is 'Source -> Destination'. ]

## INCOMING STRUCTURAL RELATIONSHIPS

⇒ Aggregation from Item to Invoice  
[ Direction is 'Source -> Destination'. ]

## ATTRIBUTES

◆ CustomerID : Private  
[ Is static True. Containment is Not Specified. ]

◆ invoiceNumber : Private  
[ Is static True. Containment is Not Specified. ]

◆ issueDate : Private  
[ Is static True. Containment is Not Specified. ]

◆ IssuedBy : Invoicer Private  
[ Is static True. Containment is Not Specified. ]

◆ TotalPrice : Private  
[ Is static True. Containment is Not Specified. ]

## ASSOCIATIONS


✎ Association (direction: Unspecified)  
Source: Public (Class) Invoice  
Cardinality: [0..\*]  
Target: Public (Enumeration) Status  
Cardinality: [1]

✎ Association (direction: Source -> Destination)  
Source: Public (Class) Invoice  
Target: Public (Class) Customer

**ASSOCIATIONS**

Cardinality: [0..\*]

Cardinality: [0..1]

 Association (direction: Source -> Destination)Source: Public (Class) Invoice  
Cardinality: [0..\*]Target: Public (Class) Invoicer  
Cardinality: [1]**OPERATIONS** calculateTotalPrice () : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ] issueInvoice () : void Public  
  
Properties:  
  ea\_guid = {DF04BCA7-0DE5-4789-858B-9F78335BDF7D}  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ] setPersonIssuedInvoice () : void Public  
[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

## Invoicer

*Class in package 'Classes'*Invoicer  
Version 1.0 Phase 1.0 Proposed  
student created on 30.10.2023. Last modified 30.10.2023  
Extends User**OUTGOING STRUCTURAL RELATIONSHIPS** Generalization from Invoicer to User  
[ Direction is 'Source -> Destination'. ]**ATTRIBUTES** emailForInvoice : Private  
[ Is static True. Containment is Not Specified. ] limitForInvoice : Private  
[ Is static True. Containment is Not Specified. ]**ASSOCIATIONS** Association (direction: Source -> Destination)

Source: Public (Class) Invoice

Target: Public (Class) Invoicer

## ASSOCIATIONS

Cardinality: [0..\*]

Cardinality: [1]

## InvoicingAddress

*Class in package 'Classes'*


InvoicingAddress

Version 1.0 Phase 1.0 Proposed

student created on 30.10.2023. Last modified 30.10.2023

Extends Address

## OUTGOING STRUCTURAL RELATIONSHIPS

 Generalization from InvoicingAddress to Address

[ Direction is 'Source -&gt; Destination'. ]

## Item

*Class in package 'Classes'*

Item

Version 1.0 Phase 1.0 Proposed


student created on 30.10.2023. Last modified 30.10.2023

## OUTGOING STRUCTURAL RELATIONSHIPS

 Aggregation from Item to Invoice

[ Direction is 'Source -&gt; Destination'. ]

## ATTRIBUTES

 itemID : Private

[ Is static True. Containment is Not Specified. ]

 itemTotalPrice : Private

[ Is static True. Containment is Not Specified. ]

 numberOfPcs : Private

[ Is static True. Containment is Not Specified. ]

 productID : Private

[ Is static True. Containment is Not Specified. ]

 productName : Private

[ Is static True. Containment is Not Specified. ]

## ASSOCIATIONS




**ASSOCIATIONS**

 Association (direction: Destination -> Source)

Source: Public (Class) Product

Target: Public (Class) Item

**OPERATIONS**

 addItem () : void Public

Properties:

ea\_guid = {2145AAA9-6932-4ff5-9012-61B85EB0548C}

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

**Product**

*Class in package 'Classes'*


Product  
Version 1.0 Phase 1.0 Proposed  
student created on 30.10.2023. Last modified 30.10.2023

**ASSOCIATIONS**

 Association (direction: Destination -> Source)

Source: Public (Class) Product

Target: Public (Class) Item

 AssociationClass (direction: Unspecified)


Source: Public (Class) Unit

Target: Public (Class) Product

**OPERATIONS**

 addNewProduct () : void Public


[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

 findProduct () : void Public

Properties:

ea\_guid = {04782DE3-D67A-40e3-AA83-304F00E60A93}

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

 listProduct () : void Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

**Status**

*Class in package 'Classes'*


Status  
Version 1.0 Phase 1.0 Proposed  
student created on 30.10.2023. Last modified 30.10.2023

## Unit

*Class in package 'Classes'*

Unit  
Version 1.0 Phase 1.0 Proposed  
student created on 30.10.2023. Last modified 30.10.2023

### ASSOCIATIONS

 AssociationClass (direction: Unspecified)

Source: Public (Class) Unit

Target: Public (Class) Product

## UnitProductCombination

*AssociationClass in package 'Classes'*


UnitProductCombination  
Version 1.0 Phase 1.0 Proposed  
student created on 30.10.2023. Last modified 30.10.2023

## User

*Class in package 'Classes'*


User  
Version 1.0 Phase 1.0 Proposed  
student created on 30.10.2023. Last modified 30.10.2023

### INCOMING STRUCTURAL RELATIONSHIPS


 Generalization from Invoicer to User

[ Direction is 'Source -> Destination'. ]


### ATTRIBUTES

 firstName : Private

[ Is static True. Containment is Not Specified. ]

 lastName : Private

[ Is static True. Containment is Not Specified. ]

 password : Private

[ Is static True. Containment is Not Specified. ]

**ATTRIBUTES**

userID : Private

[ Is static True. Containment is Not Specified. ]



userId : Private

[ Is static True. Containment is Not Specified. ]

**OPERATIONS**

createUser () : Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]



findUser () : void Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]



listAllUsers () : void Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]



selectUser () : void Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

**Status***Enumeration in package 'Classes'*

Status  
Version 1.0 Phase 1.0 Proposed  
student created on 30.10.2023. Last modified 30.10.2023

**ASSOCIATIONS**

Association (direction: Unspecified)

Source: Public (Class) Invoice  
Cardinality: [0..\*]

Target: Public (Enumeration) Status  
Cardinality: [1]

## Sequence Model

*Package in package 'Model'*

Sequence Model  
Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

## Sequence Model diagram

*Interaction diagram in package 'Sequence Model'*

Sequence Model  
Version 1.0  
student created on 13.11.2023. Last modified 13.11.2023

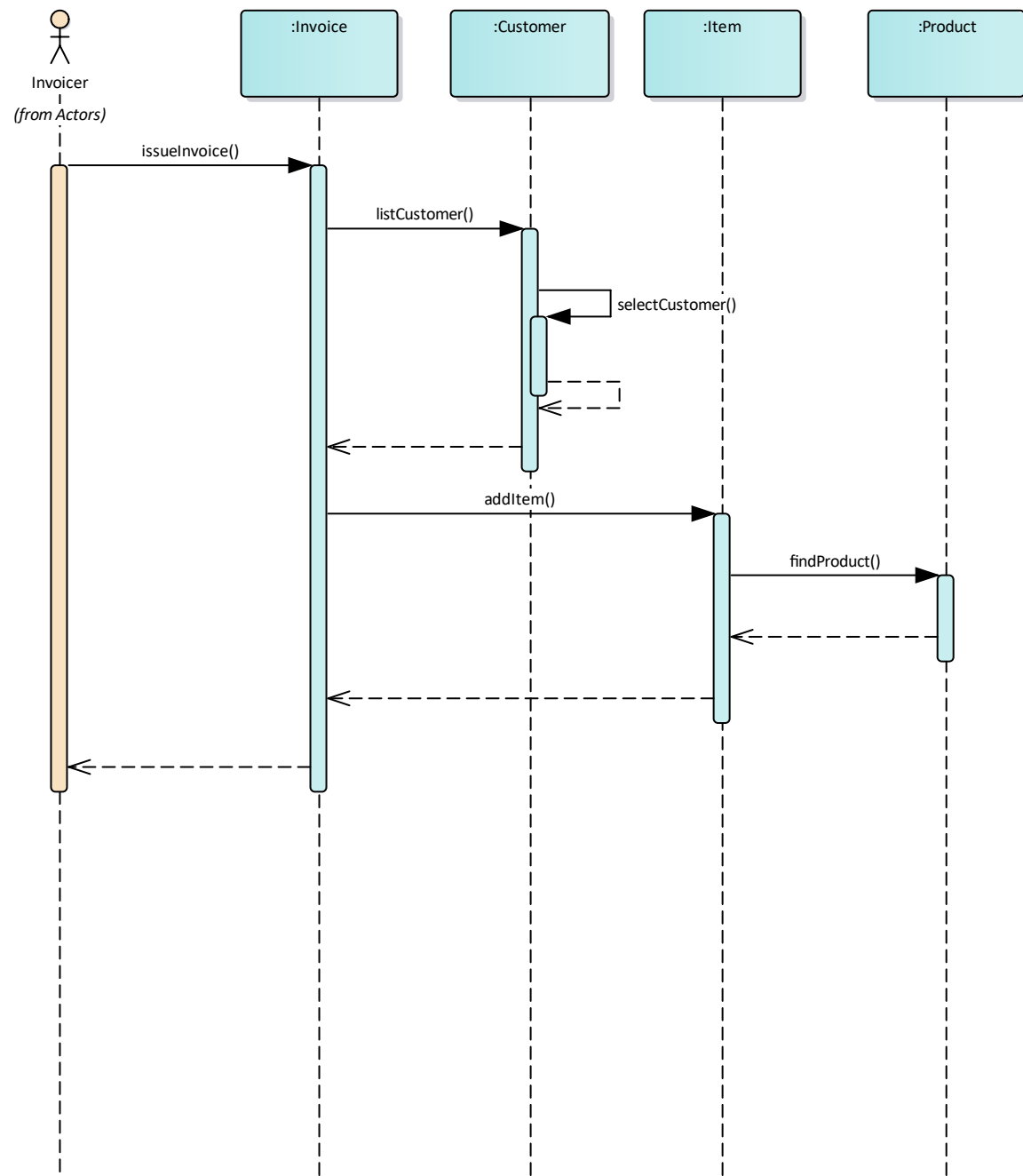


Figure 9: Sequence Model

## Invoicer

Actor in package 'Actors'

Invoicer  
Version 1.0 Phase 1.0 Proposed  
student created on 09.10.2023. Last modified 09.10.2023


### OUTGOING BEHAVIORAL RELATIONSHIPS

Name: `issueInvoice()`

Sequence from Invoicer to Object

**OUTGOING STRUCTURAL RELATIONSHIPS** Generalization from Invoicer to User

[ Direction is 'Source -&gt; Destination'. ]

**INCOMING BEHAVIORAL RELATIONSHIPS** Sequence from Object to Invoicer**ASSOCIATIONS** Association (direction: Unspecified)

Source: Public (Actor) Invoicer

Target: Public (UseCase) Create a customer entry

 Association (direction: Unspecified)


Source: Public (Actor) Invoicer

Target: Public (UseCase) Issue an invoice

 Association (direction: Unspecified)


Source: Public (Actor) Invoicer

Target: Public (UseCase) Add item to invoice

**INTERACTION MESSAGES** **1.0 'issueInvoice()'** from 'Invoicer ' sent to ':Invoice'.


Synchronous Call.

[ Return is False. Iteration is False. New group is False. ]

 **1.1 'listCustomer()'** from ':Invoice' sent to ':Customer'.

Synchronous Call. Returns void.

[ Return is False. Iteration is False. New group is False. ]

 **1.2 'selectCustomer()'** from ':Customer' sent to ':Customer'.





Synchronous Call. Returns void.

[ Return is False. Iteration is False. New group is False. ]

 **1.3 ''** from ':Customer' sent to ':Customer'.

Synchronous Call. Returns void.

[ Return is True. Iteration is False. New group is False. ]

 <b>1.4 "</b> from ':Customer' sent to ':Invoice'.
Synchronous Call. Returns void.
[ Return is True. Iteration is False. New group is False. ]
 <b>1.5 'addItem()'</b> from ':Invoice' sent to ':Item'.
Synchronous Call. Returns void.
[ Return is False. Iteration is False. New group is False. ]
 <b>1.6 'findProduct()'</b> from ':Item' sent to ':Product'.
Synchronous Call. Returns void.
[ Return is False. Iteration is False. New group is False. ]
 <b>1.7 "</b> from ':Product' sent to ':Item'.
Synchronous Call. Returns void.
[ Return is True. Iteration is False. New group is False. ]
 <b>1.8 "</b> from ':Item' sent to ':Invoice'.
Synchronous Call. Returns void.
[ Return is True. Iteration is False. New group is False. ]
 <b>1.9 "</b> from ':Invoice' sent to 'Invoicer '.
Synchronous Call. Returns void.
[ Return is True. Iteration is False. New group is False. ]

## Object

*Object in package 'Sequence Model'*

Object  
Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

## Object

*Object in package 'Sequence Model'*

Object  
Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

## Object

*Object in package 'Sequence Model'*

Object  
Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023

## Object

*Object in package 'Sequence Model'*

Object  
Version 1.0 Phase 1.0 Proposed  
student created on 13.11.2023. Last modified 13.11.2023