# RASD - TravelDream

29th November 2013

## Contents

# 1 Introduction

## 1.1 Purpose

The document introduces the general functionalities of TravelDream, the project for the course "Software Engineering 2" at Politecnico di Milano. TravelDream's management, all of the developers involved and project testers are the intended audience for this document.

## 1.2 Scope

The main software component to be delivered by the development team is a web application for the TravelDream travel agency. The main objective of this web application is to provide a product showcase and a very basic bi-directional interaction with the customers. This piece of software has the following goals:

**[G1]** allow a customer to browse packages

**[G2]** allow a customer to search packages

**[G3]** allow a customer to log into the system

**[G4]** allow a customer to reserve a package

**[G5]** allow a customer to personalize a package

**[G6]** allow a customer to easily share details about a package with friends that could also be interested in reserving the same package

**[G7]** allow an unregistered user to register to the site

**[G8]** allow employees to create predefined packages of products

This piece of software will NOT provide the following features:

- delivery system for travel packages sharing
- validation of personalized packages
- payment system

Note: the term "*package*" is used here informally and only to convey the general idea. From now on it will be used as described in table 1.

## 1.3 Definitions, acronyms, and abbreviations

Table 1: Definitions

| Keyword | Definition |
|---|---|
| Product | a flight, hotel or excursion that TravelDream offers to the customers to purchase |
| Package | a list of products |
| Final product | a concrete product, in which details about a possible reservation are specified (e.g. departure time for a flight) |
| Final package | a package that can contain final products. It can be considered finalized if all of the products it contains are final. It is reservable by a customer. |
| Customer | a person who is already identified by the system, and that is about to reserve or has reserved a package from TravelDream |
| To reserve | to specify the intention of reserving a package, through channels not provided by the application |
| Registered user | a customer or an employee that is able to be authenticated by the system |
| Requirement | a functional requirement |
| Quality | a non functional requirement |

Table 2: Acronyms

| Acronym | Meaning |
|---|---|
| OS | Operating System |
| JRE | Java Runtime Environment |
| DBMS | Data Base Management System |
| TCP | Transport Control Protocol |

### 1.4 References

- IEEE-803-R2009 standard
- Project assignment document "Progetto AA 2013-2014 (TravelDream)"

### 1.5 Overview

This document is divided in three main sections: the first one is a general introduction to the document and specifies a common terminology on which to build the remaining two. The second part analyzes the system that is to be built and the relationship that it will have with the environment in which it will be inserted. The final section provides a detailed description about specific behaviours of the system to be developed and goes deeper into analyzing non trivial needs of the customer.

## 2 Overall description

### 2.1 Product perspective

This product is totally self-contained. It is not part of a larger system, nor it interacts with other external ones.

#### 2.1.1 System interfaces

No system interfaces present.

#### 2.1.2 User interfaces

A web interface accessible through a compliant browser will be the main channel of communication with users. Further specifications will be provided in the design document.

#### 2.1.3 Hardware interfaces

No hardware interfaces present.

### 2.1.4 Software interfaces

Requirements that are not in the scope of this document, impose the use of some software products. See table 3.

Table 3: Software interfaces

| Component | Name | Mnemonic | Specification number | Version number | Source |
|---|---|---|---|---|---|
| DBMS | MySQL | MySQL | Community Server | 5.6.14 | http://www.mysql.com/ |
| Web Server | Glassfish 4 | Glassfish | Glassfish Server Open Source Edition | 4.0 | http://glassfish.java.net |
| Java Virtual Machine | Oracle's Java Runtime Environment | JRE | Java<TM> SE Runtime Environment | 1.7.0_21 | http://www.oracle.com |
| Operating System | Any OS that is able to run the specified Java Virtual Machine. | | | | |

### 2.1.5 Communication interfaces

No communication interfaces relevant at this stage.

### 2.1.6 Memory constraints

No details were provided to give a good estimate about memory constraints.

### 2.1.7 Operations

The system should provide operations that can be started by customers and employees of TravelDream. No system-initiated operations are planned to be developed.

### 2.1.8 Site adaptation requirements

No particular adaptations are necessary.

## 2.2 Product functions

This section provides a hierarchy of functions offered by the product.

### 2.2.1 Main functions

The main functions the system will support are:

- User management
- Package administration
- Packages information delivery
- Package usage

**2.2.1.1 User management** This area deals with anything that involves identification, authentication and any information about customers and employees. In particular:

**Requirements**

**[R1]** Registration of a new customer

**[R2]** Log in of a customer

**[R3]** Log in of an employee

**[R4]** Log out of a customer

**[R5]** Log out of an employee

**[R6]** Update information about a customer

**[R7]** Register a new employee

**Qualities**

**[Q1]** Registered user's passwords must be stored securely

**[Q2]** Registration must not require more than two easy steps (don't scare new customers)

**2.2.1.2 Package administration** These functions are meant for employees only.

**Requirements**

**[R1]** Create a package

**[R2]** Modify a package

**[R3]** Delete a package

**[R4]** Modify a final package

**[R5]** Add a product

**[R6]** Modify a product

**[R7]** Delete product

**[R8]** Modify final product

**Qualities**

**[Q1]** It's required for these functions to be well protected from unauthorized accesses.

**2.2.1.3 Package information delivery** These functions deal with everything that has to do with viewing items and users.

**Requirements**

**[R1]** View package

**[R2]** Browse package

**[R3]** Search package

**[R4]** View final package

**[R5]** Browse final packages (performed by a customer on its owned final packages)

**[R6]** Search final package (performed by an employee to retrieve a specific final package present in the system)

**Qualities**

**[Q1]** The system should be responsive enough to satisfy requests in a timely manner.

**2.2.1.4 Package usage** These functions provide all of the tools that allow a customer to manage, and perform operations on final packages.

**Requirements**

**[R1]** Reserve a package (only if at least one and only final products are contained in it)

**[R2]** Customize a package

**[R3]** Create a final package (only from a preexisting package)

**[R4]** Share a final package (provide a unique identifier for every final package)

**Qualities**

**[Q1]** The unique identifiers should be easily shareable

## 2.3 User characteristics

Two main types of users are the target of the system: customers and employees.

**Customer** The typical customer has a discrete level of technological knowledge, in that he will at least be required to know how to operate a web browser.

**Employee** Employees of TravelDream are trained to use the system, so will be slightly more experienced than an average customer.

## 2.4 Constraints

See table 3 for constraints about software usage. No other constraints were given, except for development time.

## 2.5 Assumptions and dependencies

The software components selected are required to be compatible with each other.

## 2.6 Apportioning of requirements

These requirements may be extended or implemented more extensively in the future:

- the system only handles hotels, flights and excursions as products
- a payment service is not provided, and will have to be managed through an external channel between the customer and TravelDream
- the channel through which a final package is shared is not provided by the system at the moment

# 3 Specific requirements

## 3.1 External interfaces

No external interfaces are required for the functioning of the system at the moment. They may be required in the future though (payment service, social network integration...).

## 3.2 Functions

This section specifies the fundamental actions the software is going to support.

### 3.2.1 Function details

This subsection provides details about functional requirements specified previously.

**3.2.1.1 Final package creation** Final packages can only be created starting from a package already present in the system. That means that right after creation, the final package should only have those products contained in a package, no more no less.

**3.2.1.2 Final package modification** Final packages can only be modified in these ways:

- substitute a hotel with another one from a selection provided by the system
- dates and times of flights
- add excursions from a selection provided by the system
- remove excursions already present in the final package
- modify excursion dates

Note: the predefined selections of hotels and excursions will be stored by employees in the package at the moment of creation.

**3.2.1.3 New employee registration** New employees can only be created by an already registered employee, thus in the first deployment of the system a default employee will be already present.

### 3.2.2 Scenarios

#### 3.2.2.1 Scenario 1

| | |
|---|---|
| **Informal description** | Logging in to the system and browsing the packages |
| **Goals covered** | G1, G3 |
| **Assumptions** | • Mary is registered to the TravelDream website. |
| Mary has a couple of weeks off from work and wants to organize a trip. To get an idea of the possible destinations she has opened the TravelDream website. The homepage offers her the possibility of logging in. She inputs her username and her password and the website authenticates her, consequently redirecting her to the default browsing page. The page presents a list of packages from which she can select the ones she is interested in viewing details about. The details for a package are presented in a dedicated page. She spends a couple of minutes doing so, successively closing the browser. | |

#### 3.2.2.2 Scenario 2

| | |
|---|---|
| **Informal description** | Searching the packages |
| **Goals covered** | G1, G2, G3 |
| **Assumptions** | • Christian is registered to the TravelDream website. |
| Christian wants to spend his Christmas vacations in Egypt. He opens the TravelDream website and logs in, filling in his username and password. The website authenticates him e provides him a selection of the current available packages. Since he doesn't find anything that satisfies his preferences in this selection, he types "Sharm el-Sheikh" in the search box and presses the search button. The system then presents him with all of the packages that contain the searched string in the title or description. He selects a package named "Christmas on the Nile" and views it. | |

#### 3.2.2.3 Scenario 3

| | |
|---|---|
| **Informal description** | Package customization and reservation |
| **Goals covered** | G4, G5 |
| **Assumptions** | • Christian is registered to the TravelDream website;<br>• Christian is already logged in;<br>• Christian is viewing the package "Christmas on the Nile". |
| Christian believes that the hotel offered by the package is not elegant enough, so he decides to change it. He clicks on "Customize" transforming it into a final package (he will be able to retrieve it even after logging out), and selects a better one from a list of proposed alternatives. He fills out all of the remaining details needed to reserve the final package and reserves it. The site also provides him with an URL from which the final package is reachable for viewing. | |

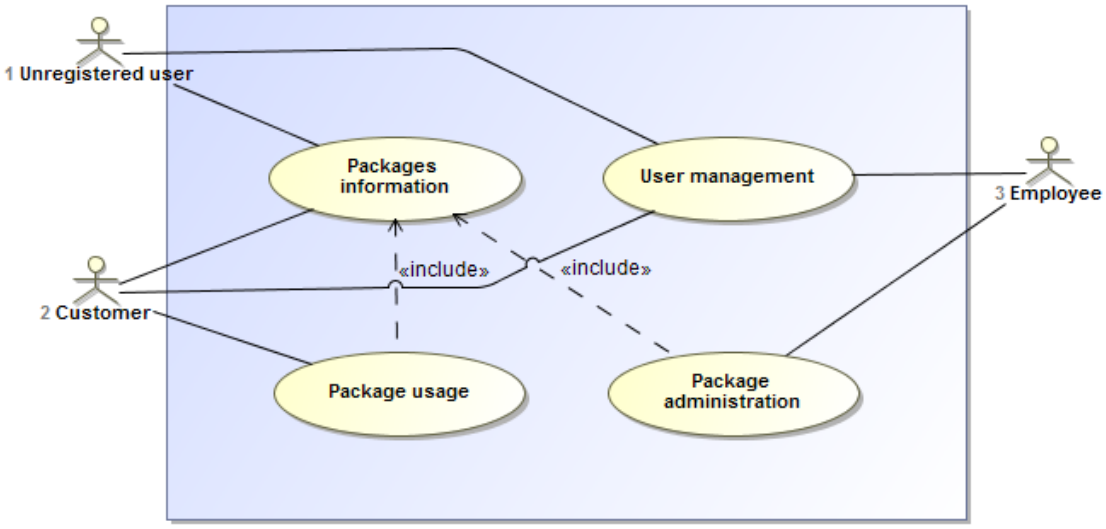
#### 3.2.2.4 Scenario 4

| | |
|---|---|
| **Informal description** | Viewing of customized package and registration to the website |
| **Goals covered** | G6, G7 |
| **Assumptions** | • Massimo is not registered to the TravelDream website, but he is friends with Christian, who is;<br>• Christian has already customized and reserved a final package. |
| Massimo receives an email from Christian, who asks him if he wants to spend the Christmas vacations with him. An URL which points to the page providing details of the travel is present in the email. Massimo clicks it and views the page. He is very interested since he has always dreamed to see Egypt, the destination of the trip. He clicks "Reserve" and the website asks him for some personal information needed for the registration of a new account, then confirms the reservation after providing all of the details needed. | |

#### 3.2.2.5 Scenario 5

| | |
|---|---|
| **Informal description** | Creating a package |
| **Goals covered** | G8 |
| **Assumptions** | • Germano is an employee of TravelDream. |
| Germano who works at TravelDream is asked by his boss to add a special offer for the Easter period. He logs into the system through the main website, and is presented a special administrative page. He then proceeds to a page where he can select different single products and combine them in a new package that will be inserted in the system. After picking the products and a list of alternatives for each default one, he inserts the title of the package, "Easter in Love", and enters a brief description of it. He then proceeds to save it in the database. | |

Figure 1: General use cases




### 3.2.3 Use Case Models

In the following subsections use cases for the software product will be presented. For every section a set of sequence and activity diagrams may be included to further specify behaviours of the system.

**3.2.3.1 Main model** This use case diagram [figure 1] represents the most high level view of the operations that the system will support. Four main areas were identified and will be presented in detail in the following subsections. Also a general overview of the classes of entities that contribute to the functioning of the system is presented as a UML class diagram [figure 2].

Figure 2: General class diagram

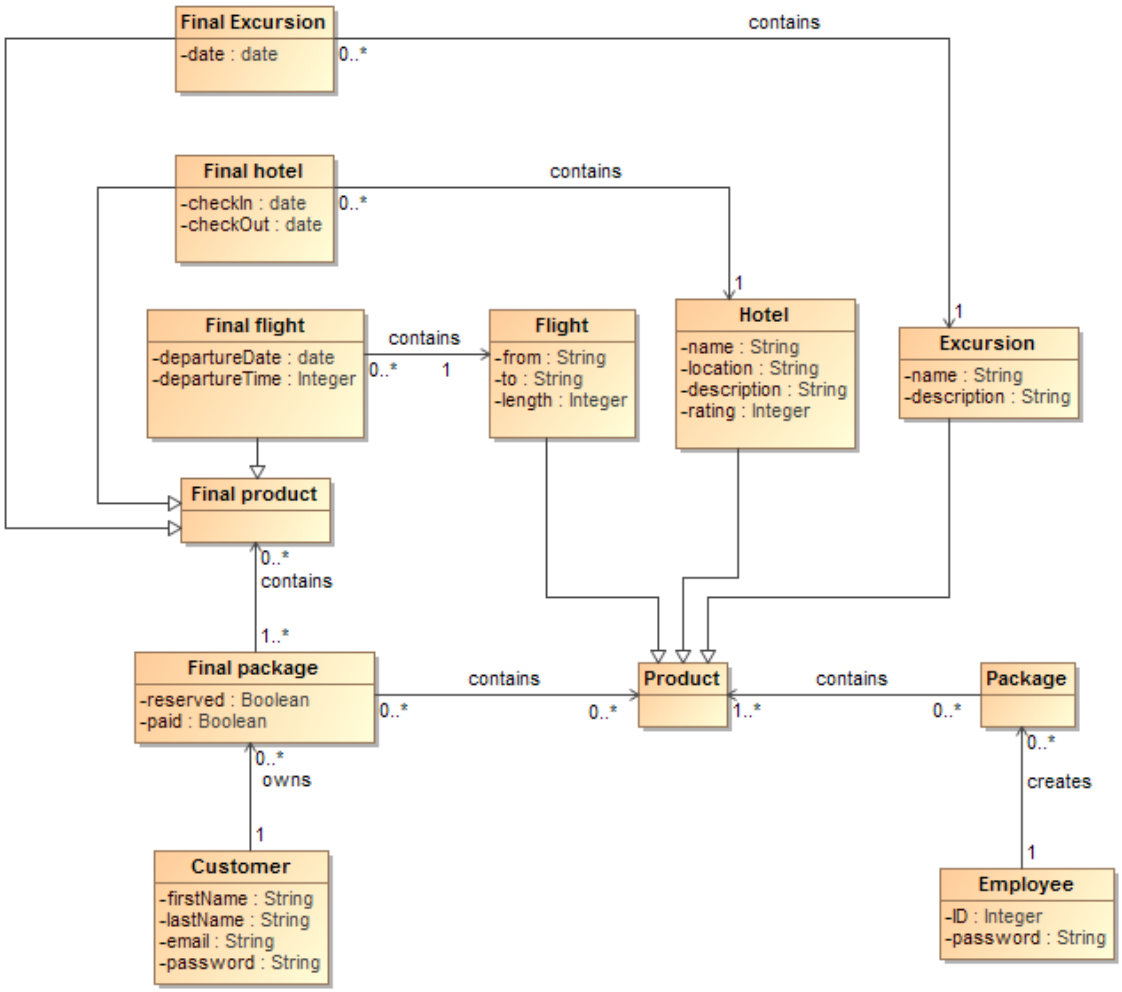
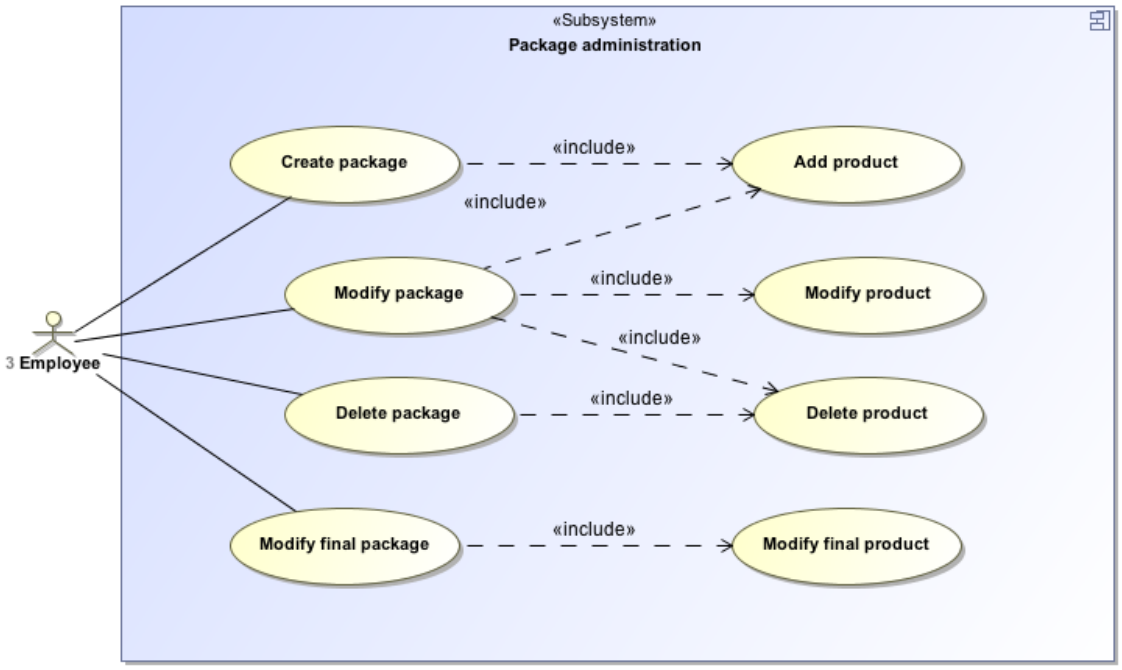


Figure 3: Package administration use cases




**3.2.3.2 Package administration model** These use cases [figure 3] describe the portion of the system that allows employees to manage packages, final packages and products.

Figure 4: Create package sequence diagram.

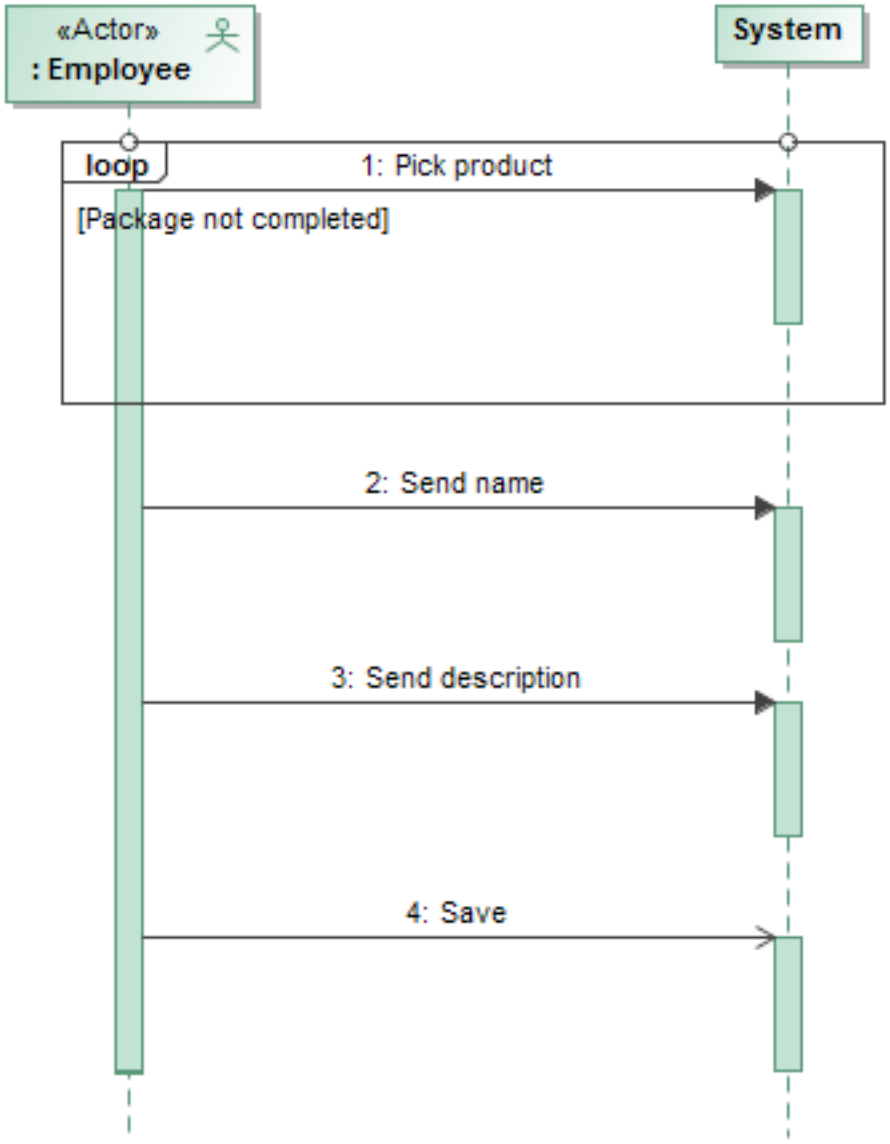
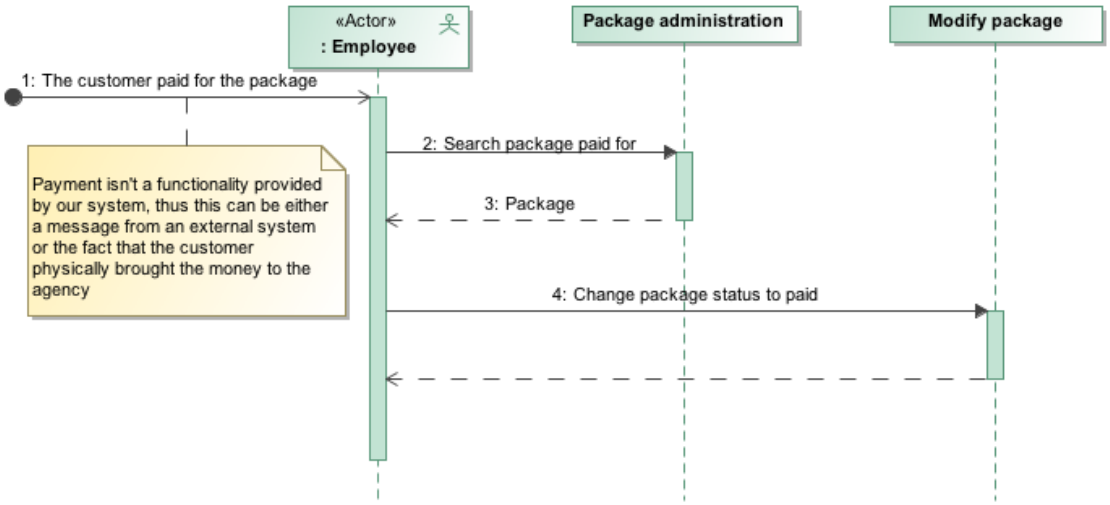


Figure 5: Payment registration sequence diagram. Notice how the activation of the procedure is triggered by an external event.



**3.2.3.3 Package information model** The main purpose of these use cases [figure 6] is to deliver information about the packages offered by TravelDream. Only registered users are able to perform advanced searches and browsing, limiting unregistered users to view single packages shared by customers. Notice how all of these use cases do not modify the state of the system.

Figure 6: Package information use cases

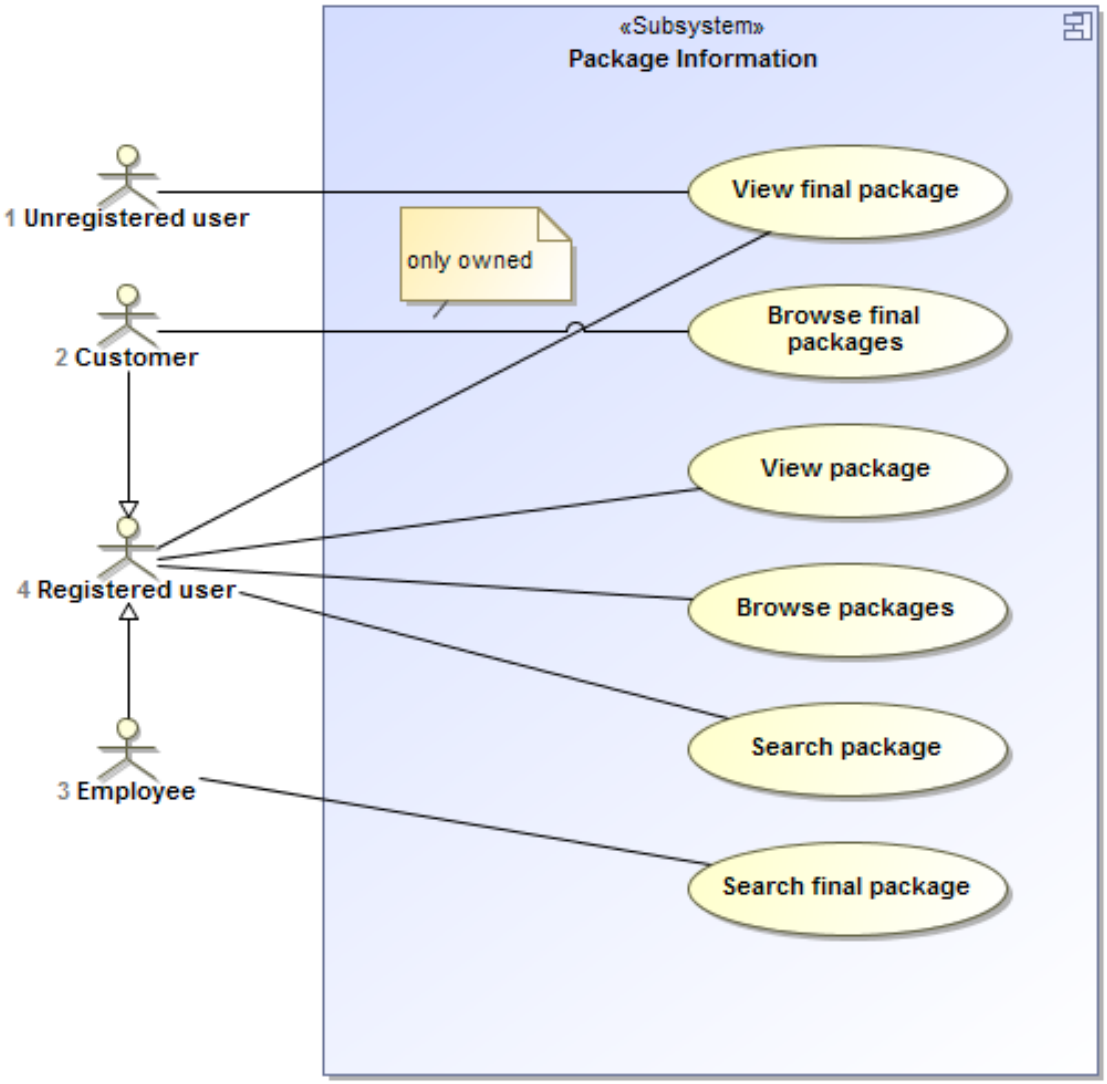
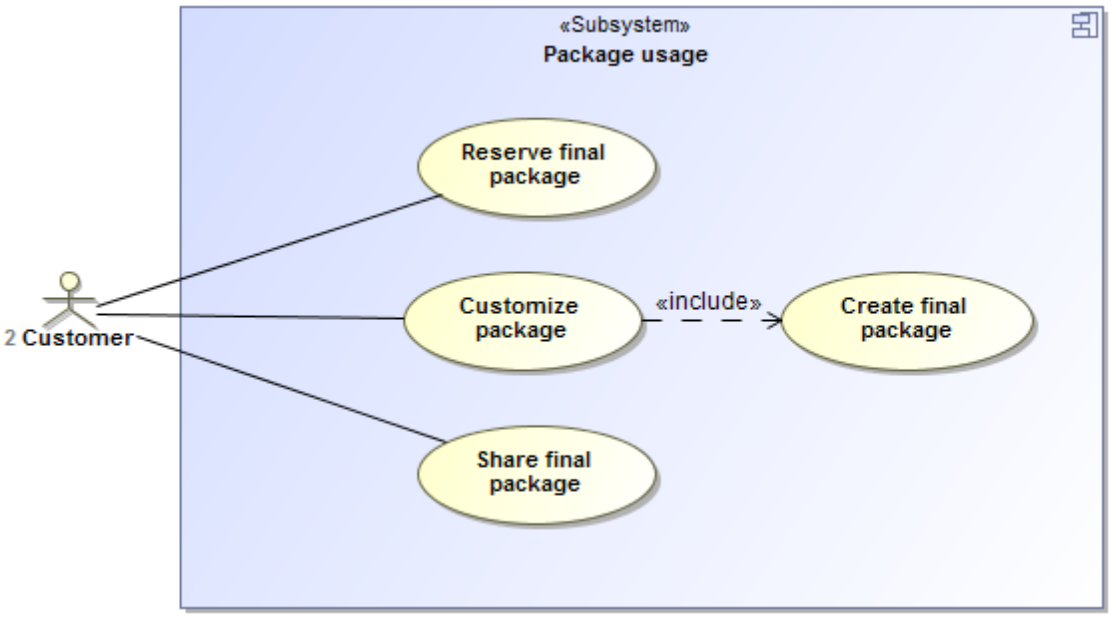


Figure 7: Package usage use cases




**3.2.3.4 Package usage model** These use cases [figure 7] describe the interaction of a customer with its own packages. In a sense this is the core of the business functionality.

Figure 8: Show package activity diagram. Even though this activity diagram includes a use case not present in the subsystem, it is more relevant to this section than any other.

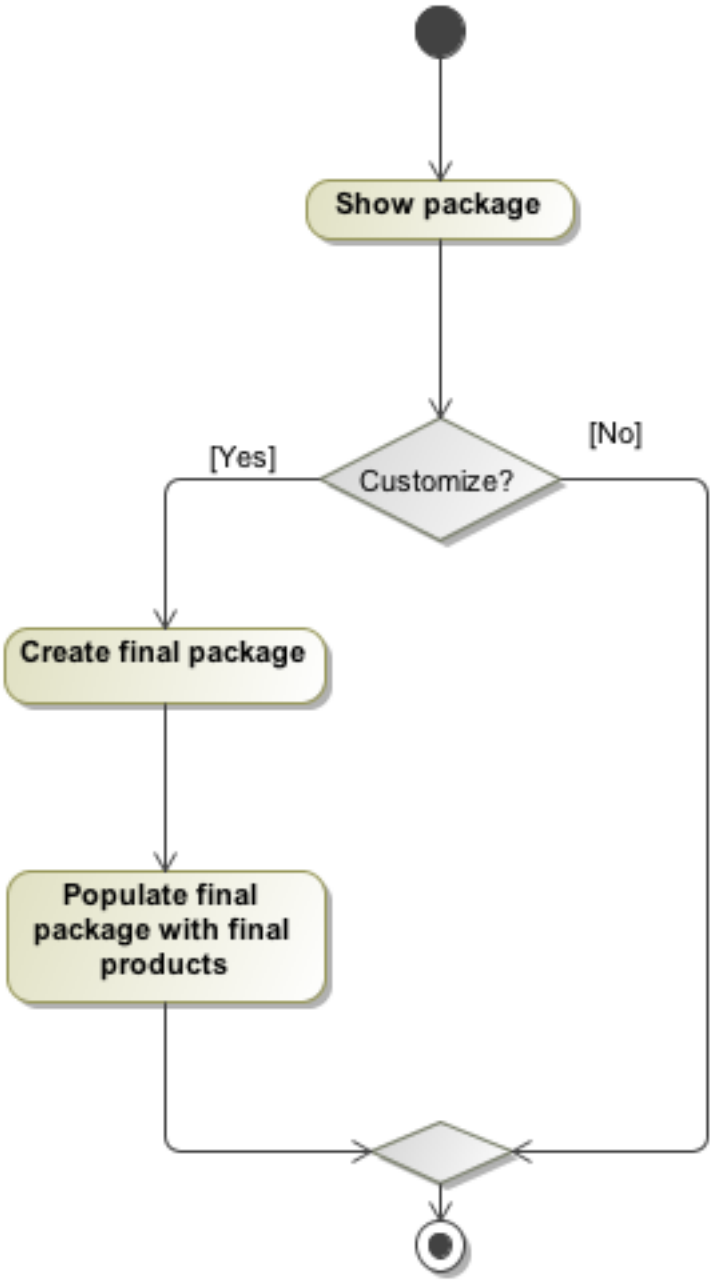


Figure 9: Reserve package sequence diagram

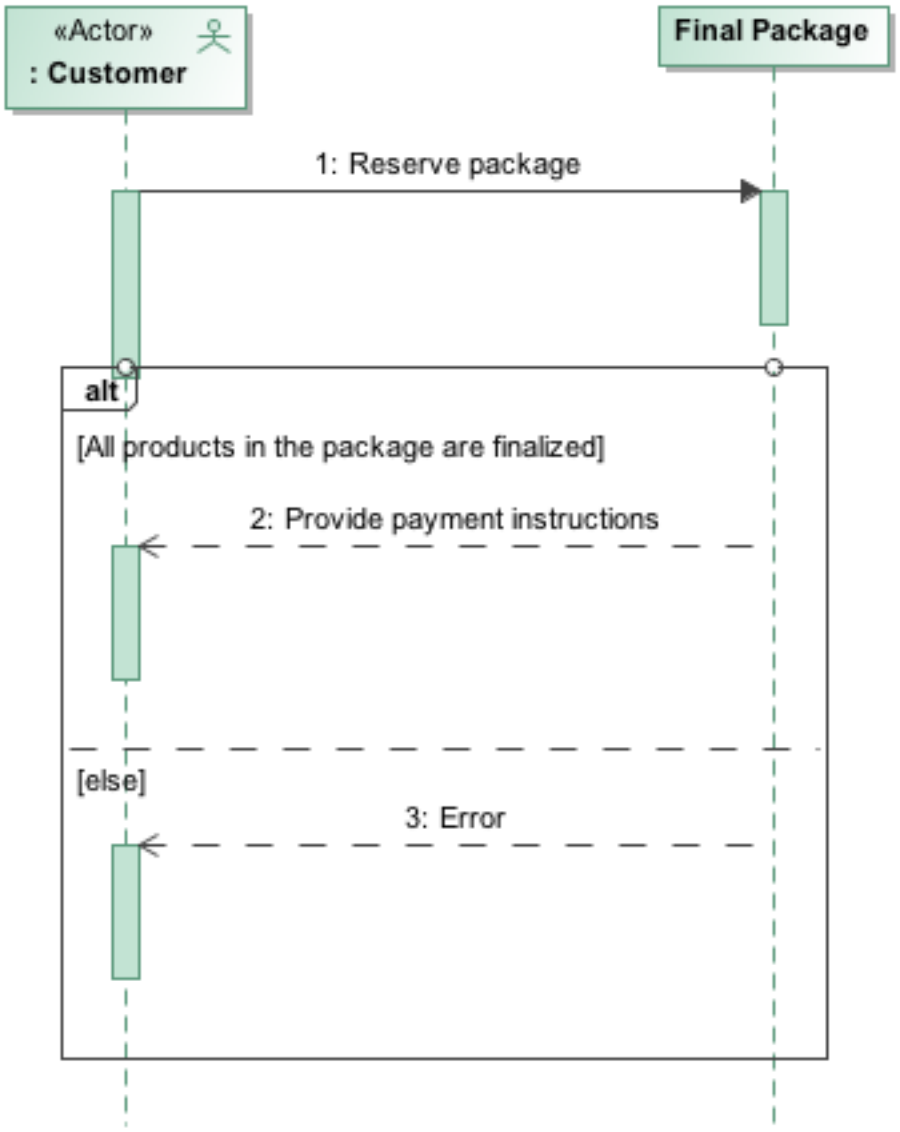


Figure 10: Reserve package check activity diagram

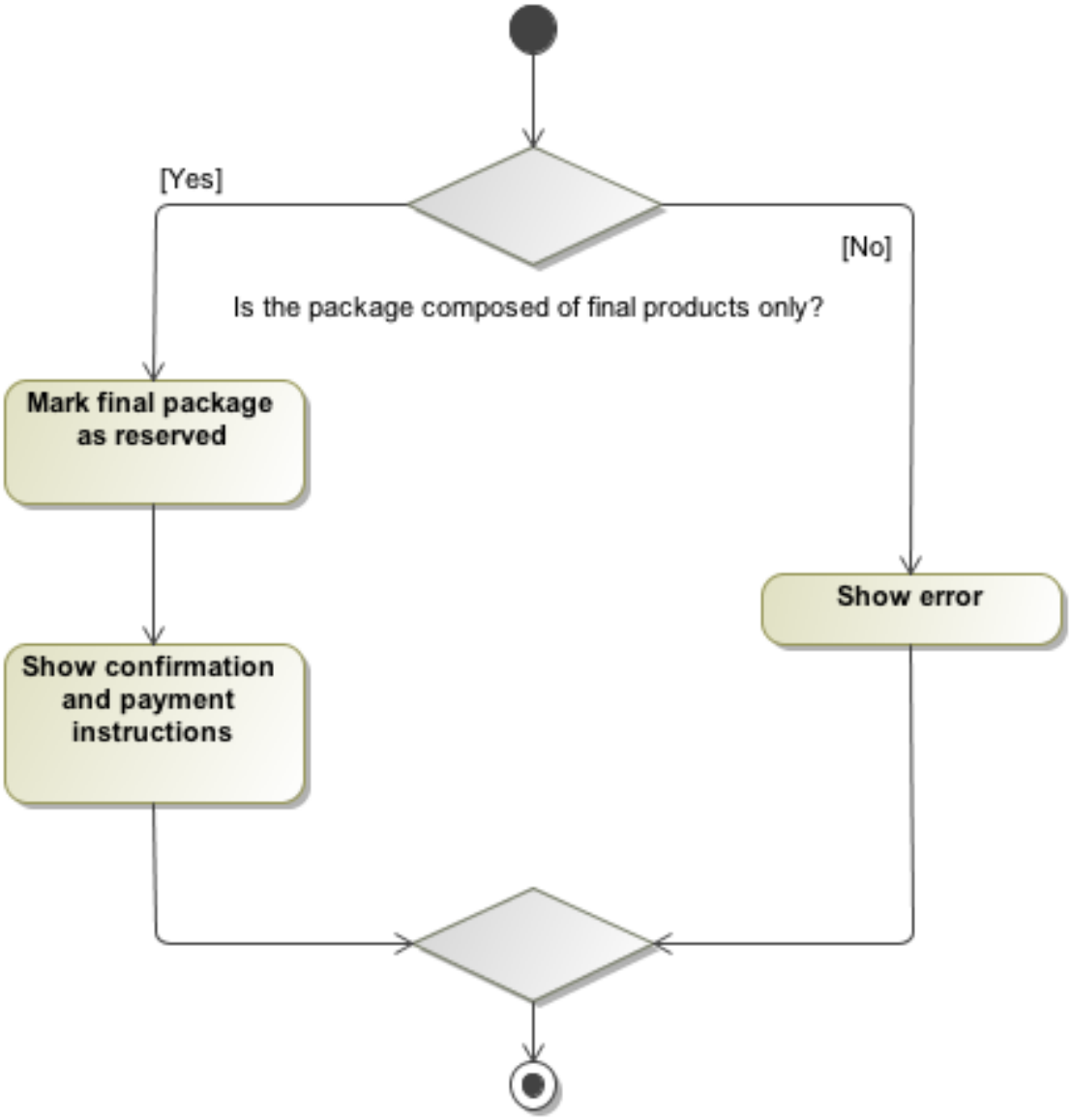


Figure 11: Customize package sequence diagram

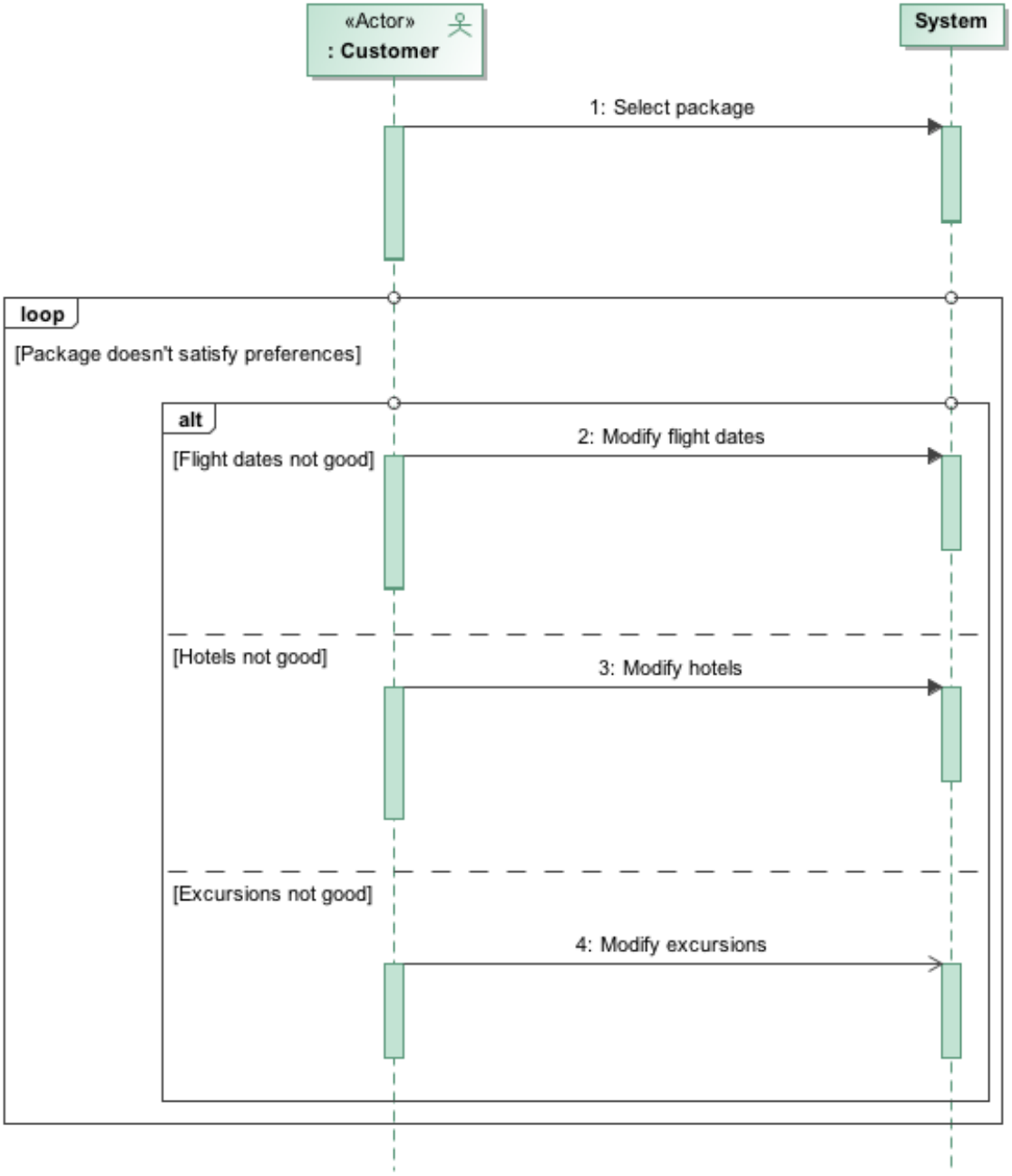
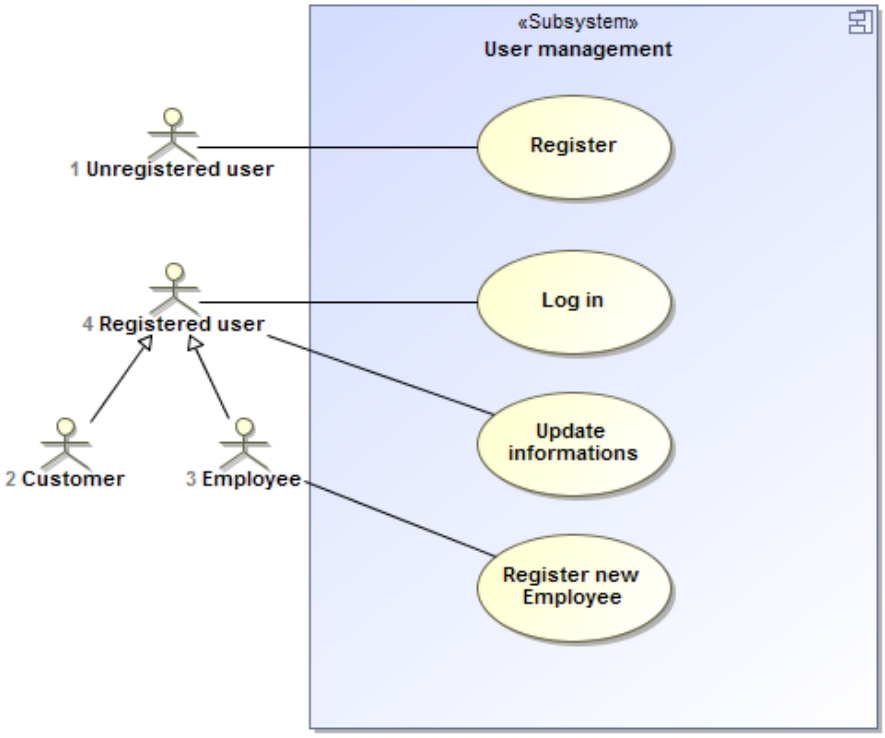


Figure 12: User management use case




**3.2.3.5 User management model** These use cases [figure 12] describe how users should be identified and authenticated by the system.

Figure 13: Login sequence diagram

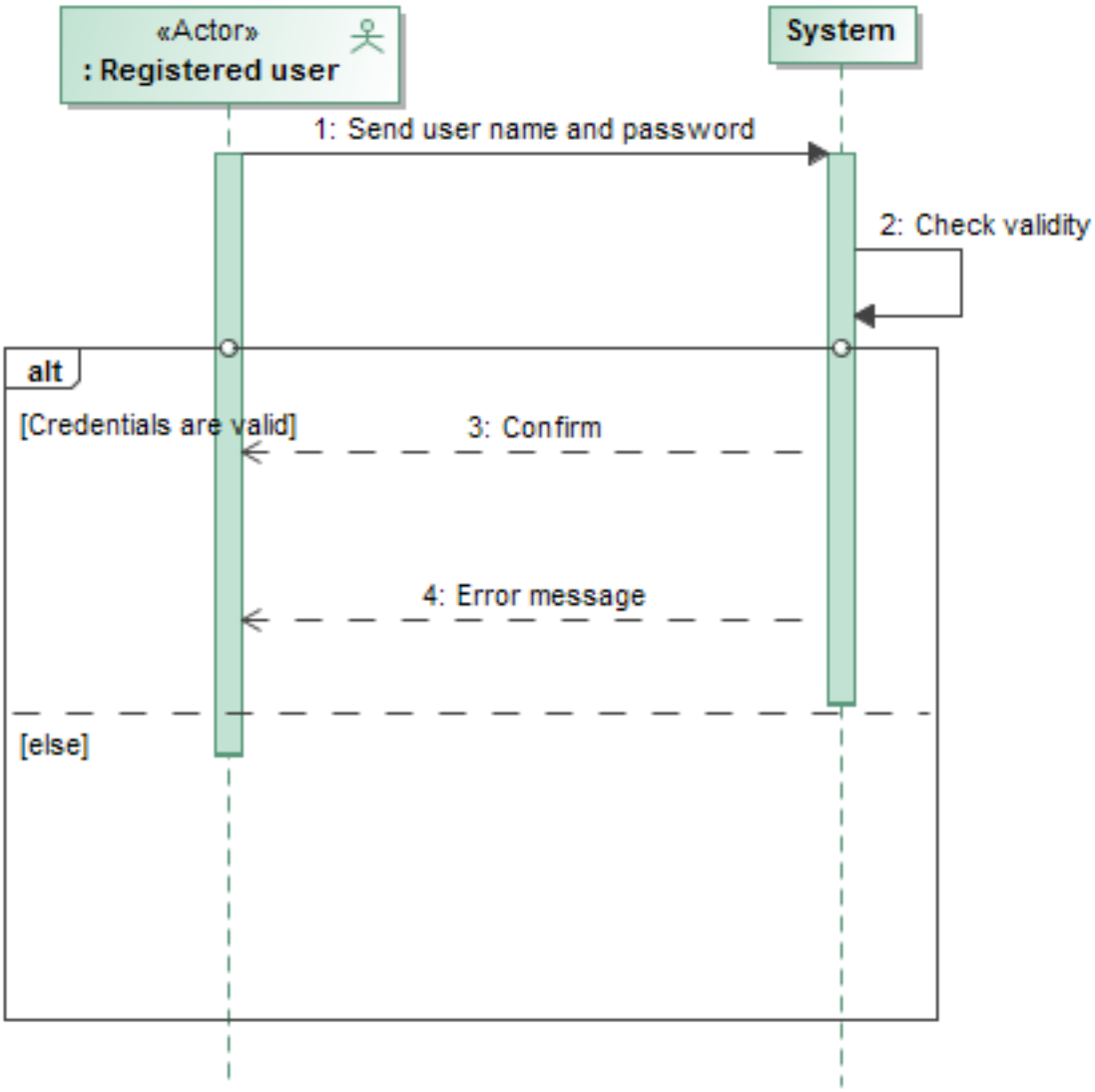


Figure 14: Register sequence diagram

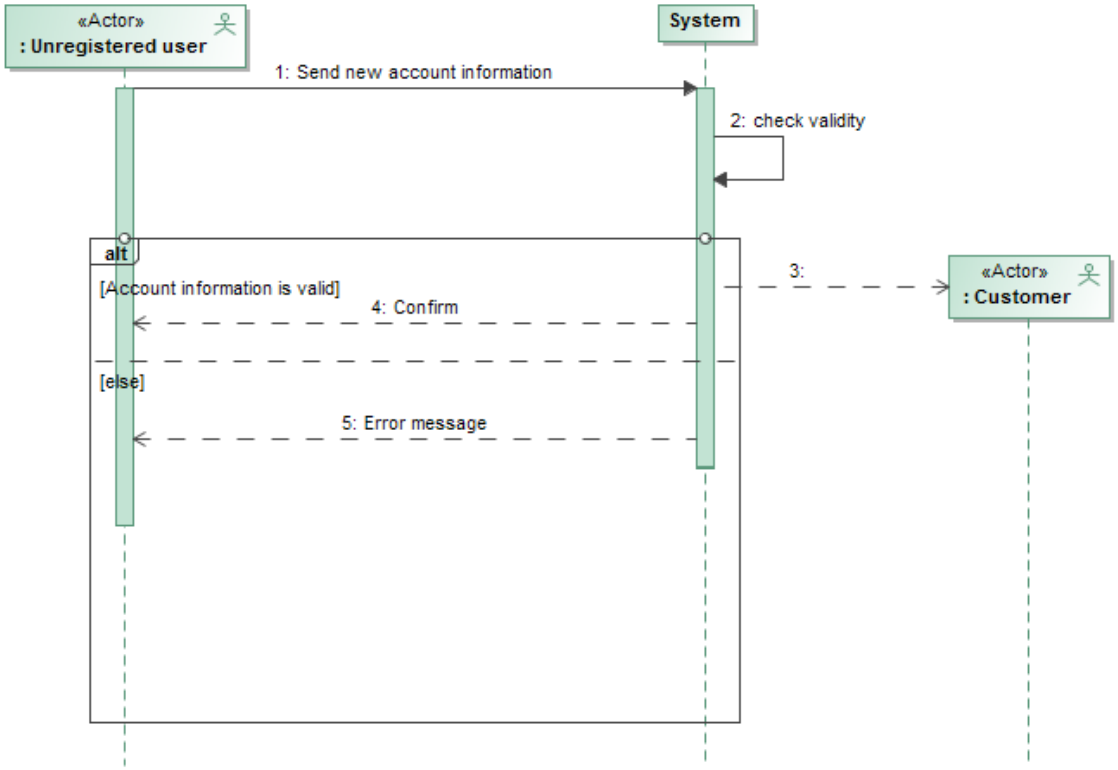
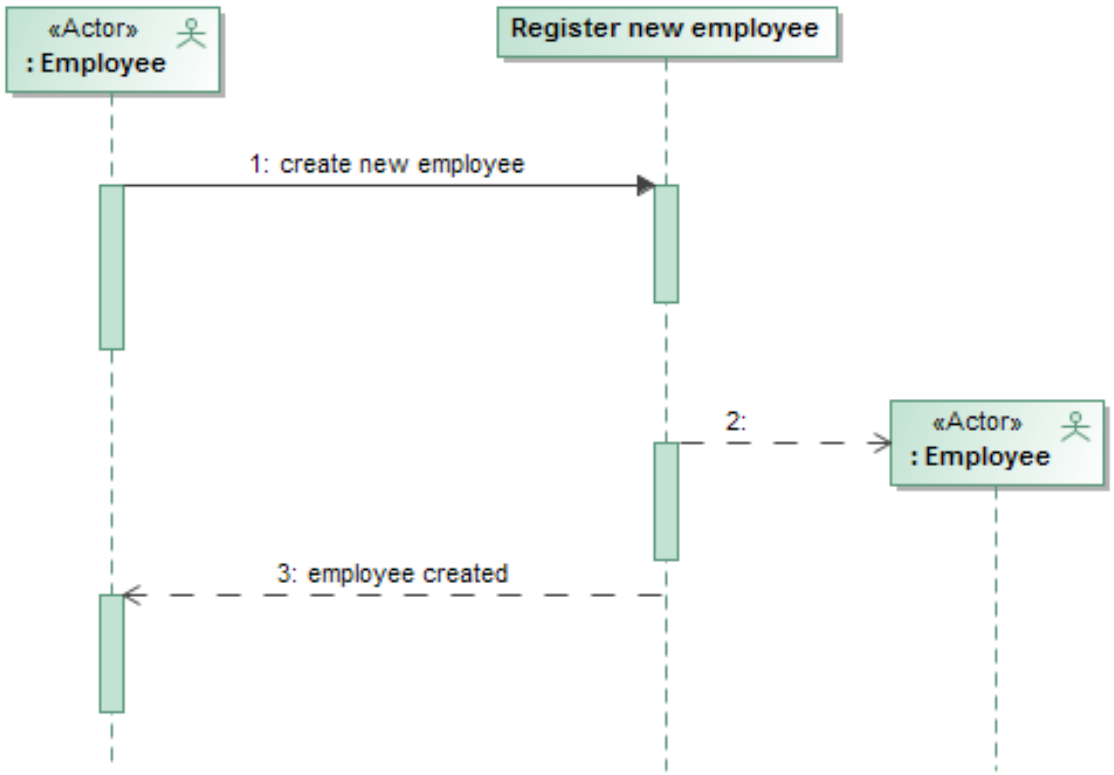


Figure 15: Register new employee sequence diagram




## 3.3 Performance requirements

The software product is required to offer a smooth browsing experience to every user.

## 3.4 Logical database requirements

Every information added to the system is required to be able to be represented in a MySQL database.

## 3.5 Design constraints

The software must be developed using Java EE technologies.

### 3.6 Software system attributes

#### 3.6.1 Security

The software must store passwords in a secure way. This can be achieved using cryptography.

# 4 Supporting Information

## 4.1 Appendixes

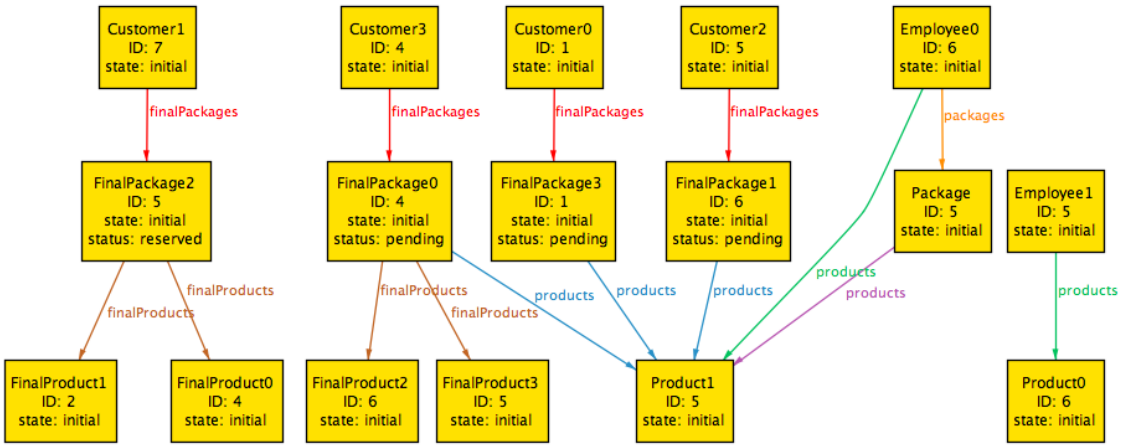
### 4.1.1 Alloy

We modeled the system using the Alloy language. Graphs showing particular system states are given in the following paragraphs. The main concept behind our model is a 2-state machine. The *initial* state represents the system before any operation. The *afterOp* state represents the system after an operation. Only one operation is admitted between states. Every entity in the model has an ID as its primary key, consenting us to distinguish it between states. Also we guarantee sequential coherence (e.g. operation correctness or enforcing same IDs between states for same entities) on top of inter-state coherence (e.g. no duplicate IDs in a state). The included file "alloy.als" contains our model. For more easily viewing models we included a custom theme name "customTheme.thm".

#### 4.1.1.1 Static model

**Static model: time not involved** This static model is actually a subset of the 2-state machine, representing only an initial state. It mostly shows inter-state correctness. See figure 16.

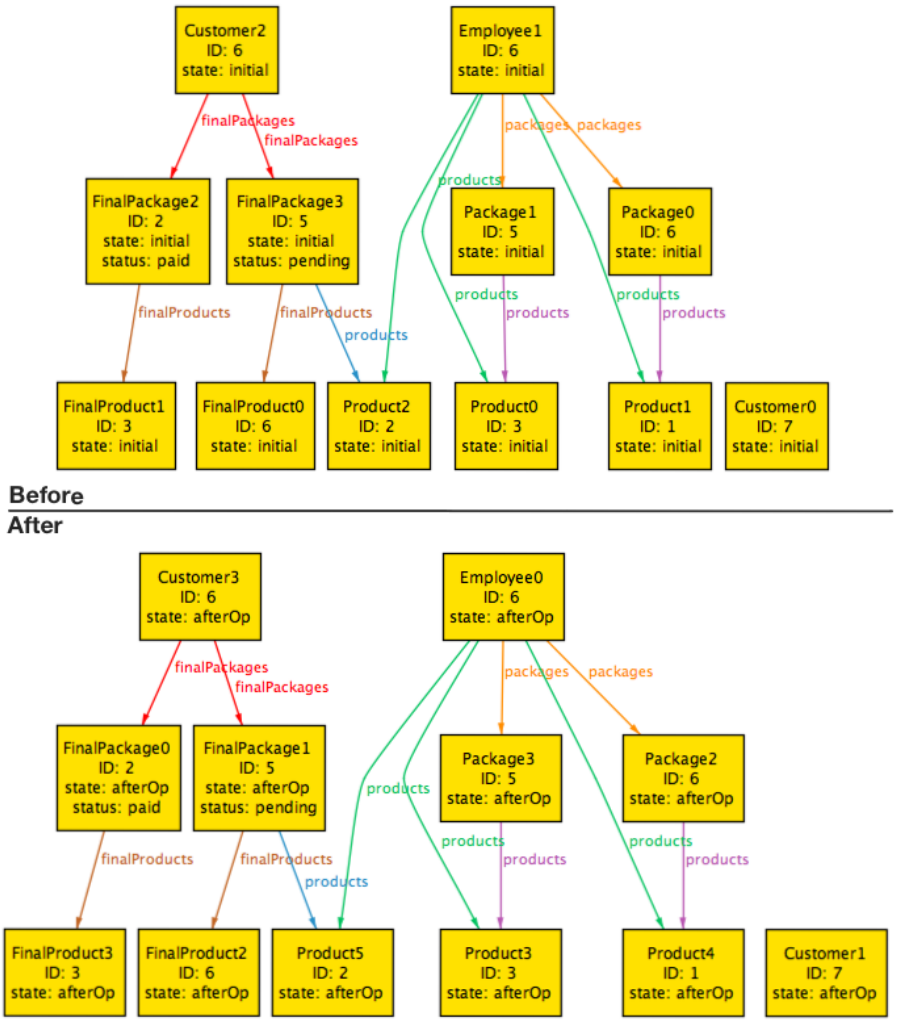
Figure 16: An Alloy model representing the system in a possible initial state.



#### 4.1.1.2 Dynamic model (operations)

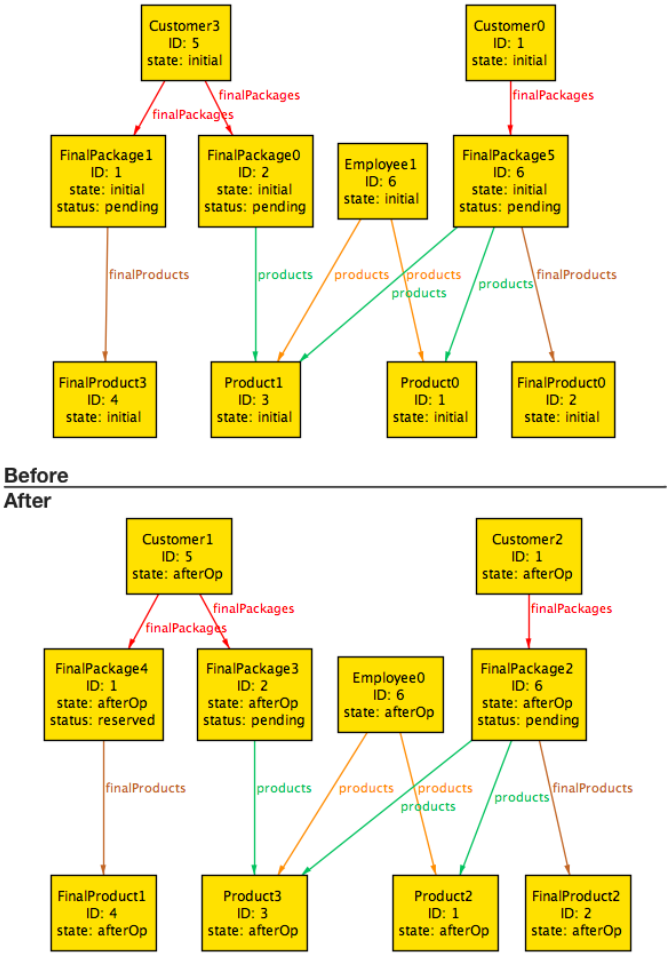
**No change model** This model represents a "no operation" transition between states. As you can see in figure 17 the system is duplicated, only changing its state attribute.

Figure 17: An Alloy model representing two successive states in which no operation was performed.



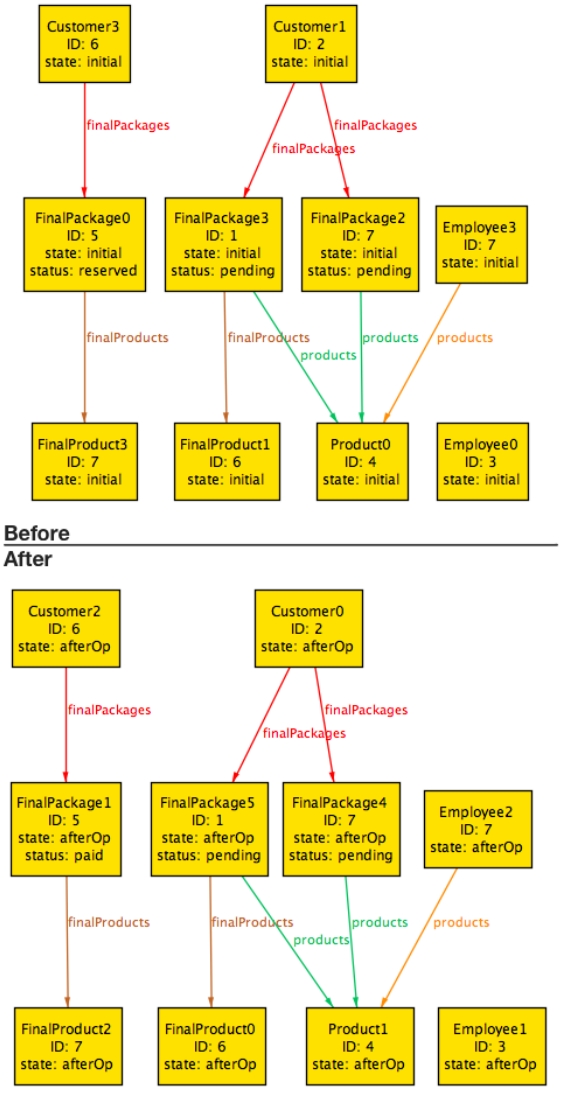
**Reserve final package model** This model represents the operation of a customer reserving one of its final packages. See figure 18.

Figure 18: An Alloy model representing the reservation of a final package. As you can see the customer with ID 5 reserved its final package with ID 1.



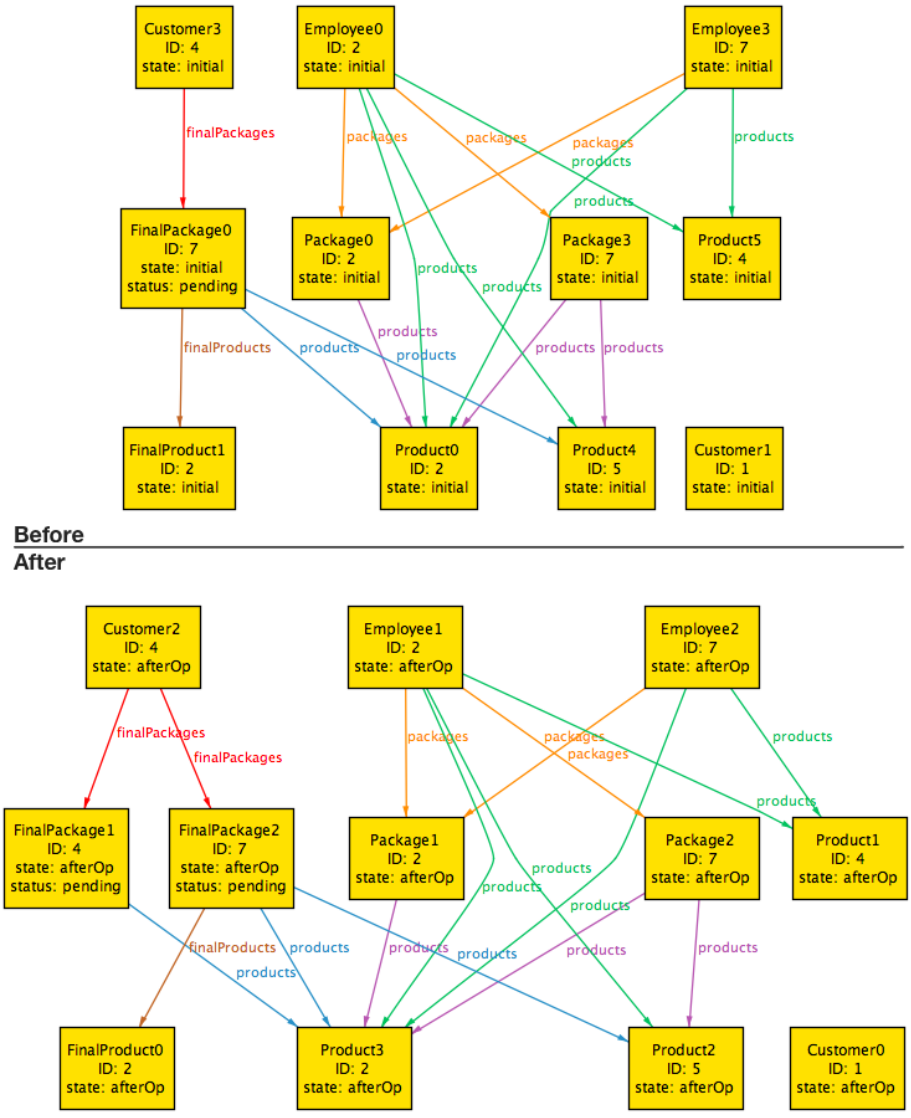
**Payment registration model** This model represents a subset of the general operation of modifying a final package. The employee is notified by an external system that the customer has paid for the final package, so he updates manually the information in the final package. See figure 19.

Figure 19: An Alloy model representing the payment registration of a final package. The payment of customer with ID 6 was registered for the final package with ID 5.



**Create final package model (select basic package)** This model represents a customer that has selected a package to customize, so a final package with all the products present in the package selected is added to the final packages belonging to the customer. See figure 20.

Figure 20: An Alloy model representing the creation of a final package. Notice how the package created (ID 4) has only the products belonging to package with ID 2.



# List of Figures



# List of Tables