TravelDream Project

Requirements Analysis Specification Document

Riccardo B. Desantis - matr. 765106

Contents

Chapter 1

Introduction

1.1 Purpose

This document introduces and defines the general functionalities of the e-commerce system, called **TravelDream**, that is being developed for the TravelDream company to support its sale process. The **users** of this system are:

- The TravelDream **employees**: they add/delete/modify the basic products that can compose a travel. For the moment the company is offering as basic products only flights, hotels, and pre-planned excursions but in the future it plans to include also other transportation means (e.g., trains, car rental, ...). Moreover, they create predefined travel packages to be sold to customers.

- The TravelDream **customers**: after registering with the system, they can browse and search through the travel packages, select one of them, and personalize it; for instance, adding the departure and return date and modifying hotels and pre-planned excursions.

This is a **customer-driven**, **problem-driven**, **greenfield** and **single-product** project, with a **bespoke** development process.

Some **risks** that should be taken into account in this development process are:

- Working on the project alone, there could be some problem in meeting the deadlines.

  - Probability of occurrence: 0.4;
  - Severity: 9/10.

- The requests of the project are clear and understandable, but still there could be some hidden knowledge that could lead to some misunderstanding.

  - Probability of occurrence: 0.2;
  - Severity: 5/10.

- The system developed could have some performance issues.

  - Probability of occurrence: 0.4;
  - Severity: 2/10.

The **audience** of this document are:

- the **users**, to check if their goals are met;

- the **testers**, to check if the system works as expected, according to the requirements and goals;

- the **project managers** at the company, for checking if their requirements are understood and met;

- the **developer** that will take part at the developing process.

1.2   Scope

The **world** we are dealing with is made of hotels, flights, excursions and everything you can think of doing in a holiday trip. In this world, customers and employees are the agents we're interested in.

The **machine** in our system is the web application that we are going to develop. It is going to give to the customers the possibility to personalize the travel packages, operation that it was only accessible to the employees before.

We could define those goals for our project:

- **G1**: Allow people to search for packages.

- **G2**: People can join the system as customers.

- **G3**: Employees can join the system.

- **G4**: Employees can add/remove/modify basic components that can compose a travel, which are flights, other transportation means (e.g., trains, car rental, . . .), hotels, and pre-planned excursions.

- **G5**: Employees can add/remove/modify travel packages, creating them starting from the basic components.

- **G6**: Customer can customize packages. They should be able to browse and search through the travel packages, select one of them, and personalize it, for instance, adding the departure and return date and modifying hotels and pre-planned excursions.

Using the Jackson-Zave approach, we could draw this graph:
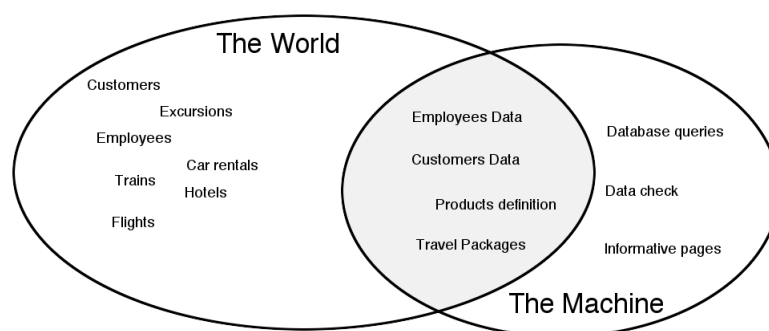
The **stakeholders** for this project are:

- the customers and the employees that will use the system;

- the user support staff that will help the customers understanding the system, and the call center operators that will be helping the users in need;

- the head of the company, that is paying for the project.

1.3 Definitions, acronyms, abbreviations

1.3.1 Definitions

Here are the definitions of the terms used:

- **Customer**: a customer of the company, a person willing to buy a travel package.

- **Registered Customer**: a registered customer in our system, that is a person which entered his/her data in our system.

- **Employee**: an employee of the TravelDream company.

- **Basic Products**: one of these elements:

  - a flight;
  - a car rental;
  - a train;
  - a hotel;
  - an excursion.

- **Travel Package**: a package composed of some basic products, created by an employee. We can assume that it can only be created or removed.

- **Customized Travel Package**: a travel package modified by a customer, with a specific flight (e.g., with a departing and returning date) etc.

1.3.2 Acronyms and abbreviations

The acronyms and abbreviations used are:

- **XML**: Extensible Markup Language.

- **RASD**: Requirements Analysis and Specification Document.

- **NFR**: Non-functional Requirements.

- **QA**: Quality Attributes.

- **FR**: Functional Requirement.

- **G**: Goal.

- **DBMS**: Database Management System.

- **AS**: Application Server.

- **JEE**: Java Enterprise Edition.

1.4 Reference documents

The documents used for the writing of this RASD are:

1. IEEE Recommended Practice for Software Requirements Specifications: `http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5841`;

2. Alloy model file: TravelDream.als.

1.5   Overview

This document is divided among four sections:

1. **Introduction**: provides a synopsis of the software product to be developed.

2. **Overall description**: describes the general factors that affect the software product and its requirements.

3. **Specific requirements**: contains the artifacts generated by the analysis. It describes all of the software requirements to a level of detail sufficient to be externally perceivable.

4. **Appendixes**: provides supporting information about how the alloy model contributed to the requirement analysis and analysis model.

Chapter 2

Overall Description

This section does not describe specific requirements, but puts the product into perspective and provides a background for specifying concrete requirements in the next section of this document.
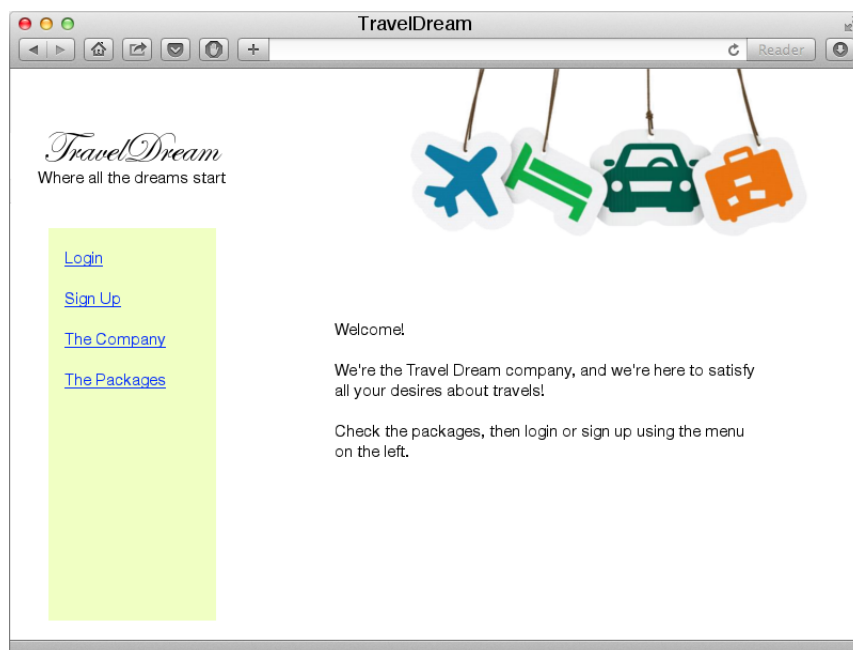
2.1 Product perspective

The software product is a complete self-contained system and it is not part of any other larger system.

2.1.1 System interfaces

The software product does not provide any external interface.
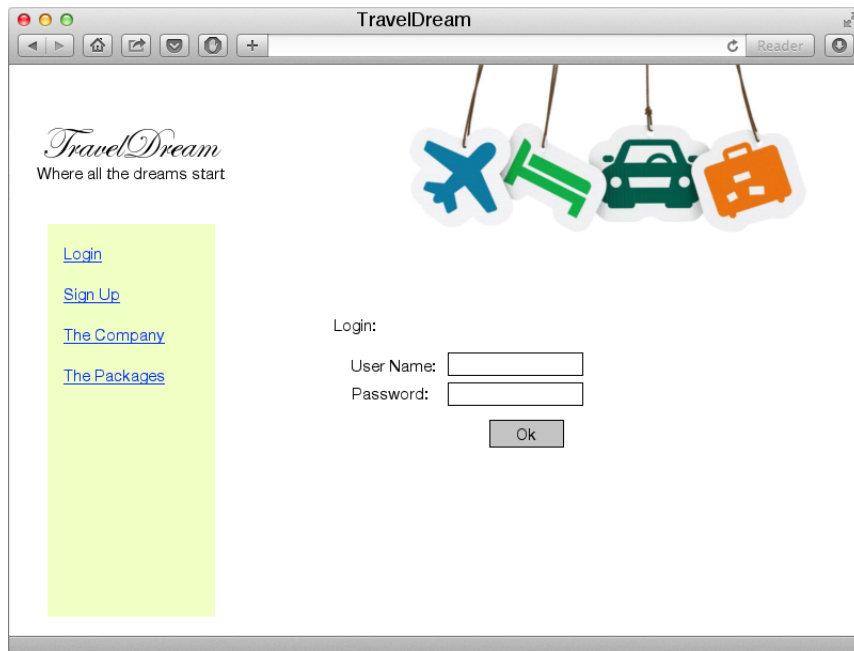
2.1.2 User interfaces

The software product will present the following page layouts as the user interface. First of all, there's the **landing page**:

from where the user can either:

- **Log in**:

where both the customers and the employee will use their data to log into the system. If it is a customer, this is the page that will be offered:
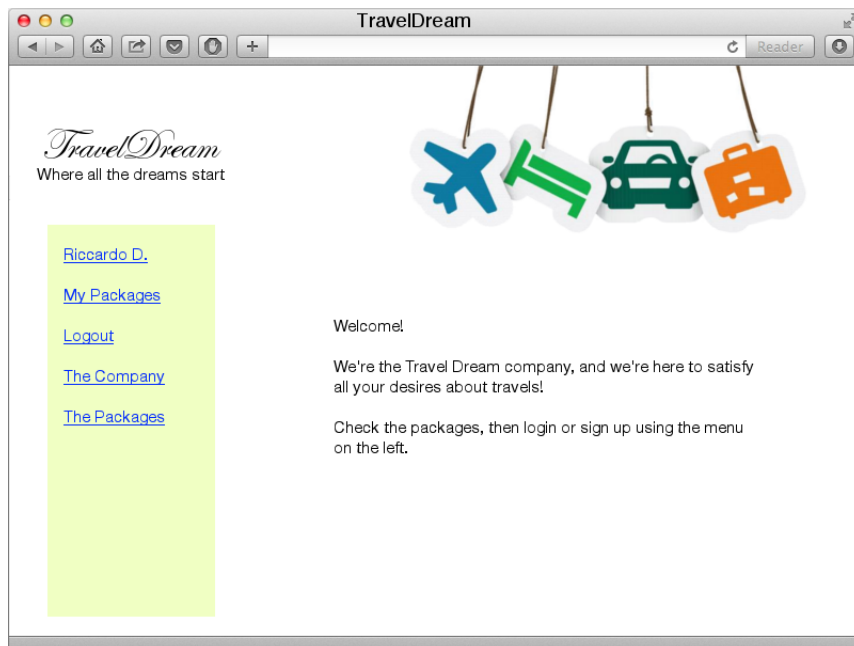
while if it is an employee:
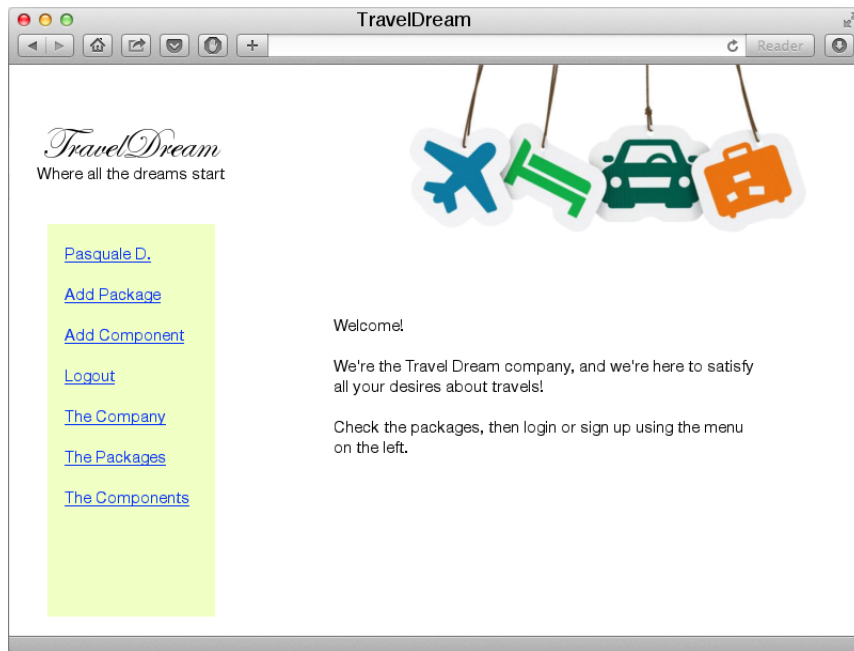
- **sign up**:
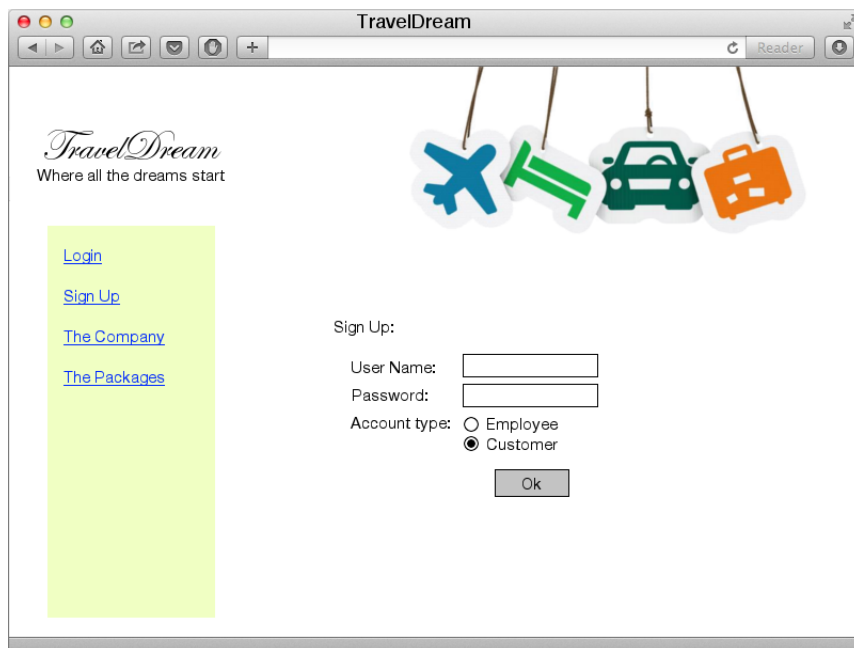
where in this first solution of the problem both the employees and the customers may join the system; a future upgrade could add an administration page where an "admin" will be able to manage the user and the employees. After signing up, we suppose that the user has still to log into the system.

- **list all the packages**:
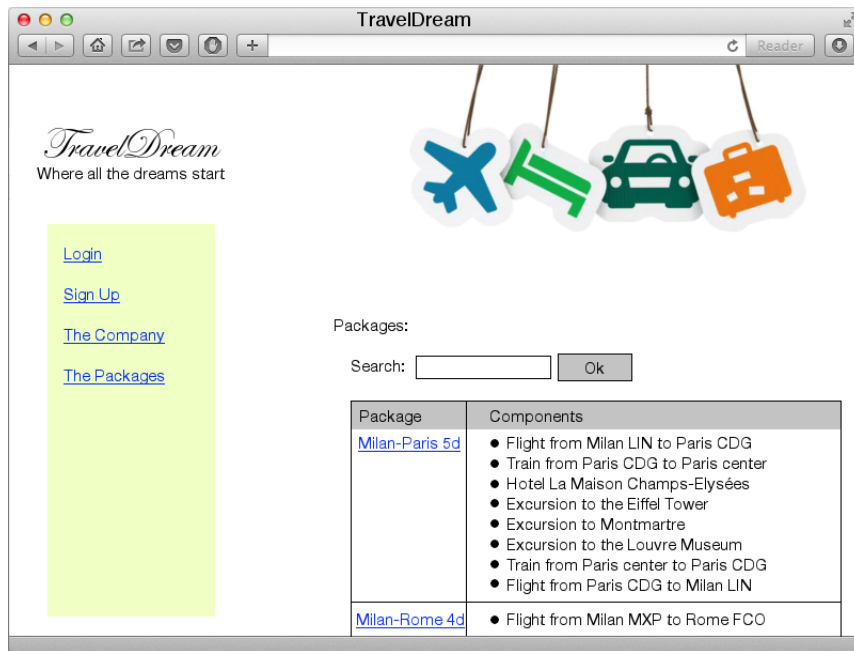
where all the packages are listed; searching for a specific page (indexed by the name and by the components) will only show the packages satisfying the search query. Clicking on the name of a package will show the details of such package:
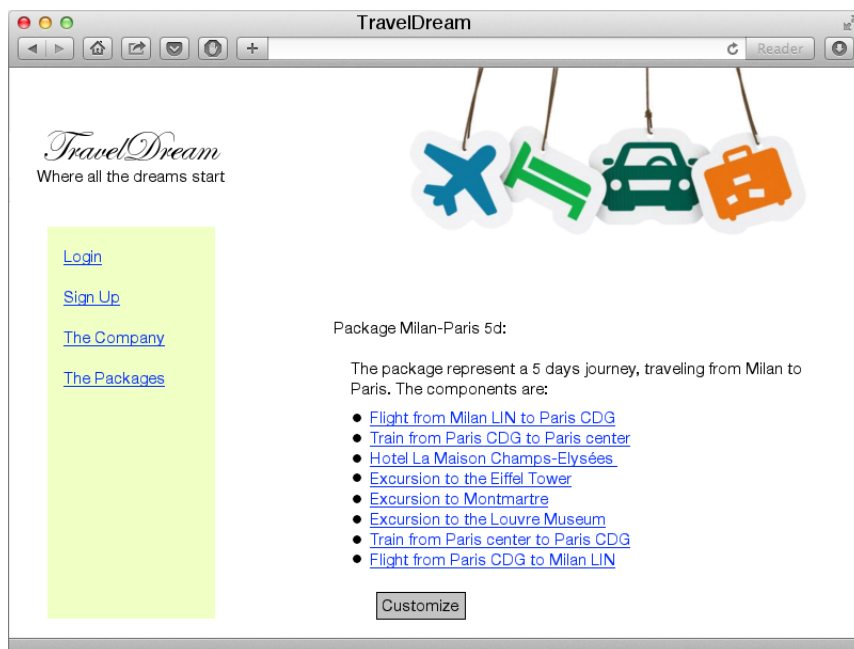
where the user could click on any of the elements to have a description, for example:

If the user wants to customize the package, it will first be asked to log into the system if it isn't already logged in; two situation arises:

– if the system recognize him/her as a customer, it will ask him/her to choose some details about the package:
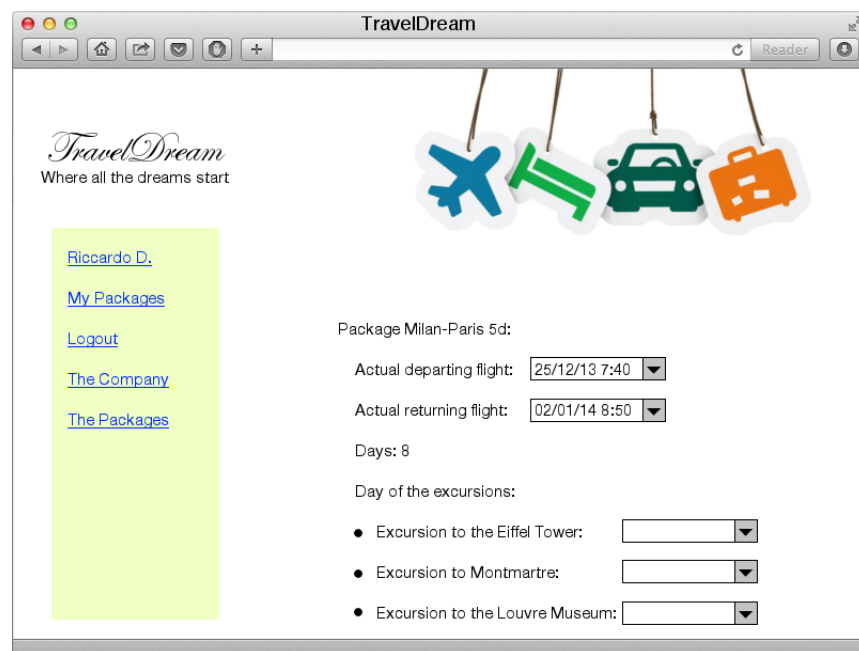
– if the system recognize him/her as an employee, it will give the possibility to modify the elements of the package, showing the same page given when he/she is adding a package, but already filled with the details of the current package (see below for more).
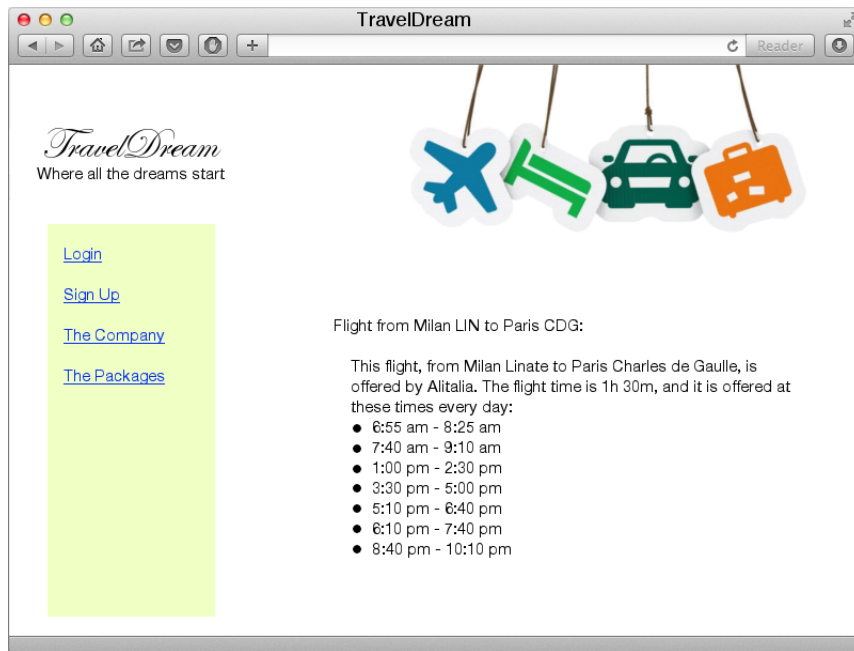
• If the user is a logged in, a link with his/her name will be showed on the menu; clicking on that voice will show the profile of the user. Also, if he/she is logged in, a **logout** voice menu is available to log out.

• If the user is logged as a customer, he/she will be able to show the packages he/she has modified, and further customize them. If the departing time is passed, the travel package isn't modifiable anymore, and it is tagged as "completed". The list of his/her packages is

exactly like the full list of packages, only that they're only the packages he/she has already customized.

- If the user is logged as an employee, he/she will be able to:

    – Add a new package:

    where we suppose that all the needed component are already present in the system.

    – Add a new component:

    In this first approximation, a component isn't modifiable after it is inserted in the system. The data requested change as soon as the type of component is selected, and in the example we've chosen to add a hotel. It could be useful to support images in the database for hotel and such, but it isn't needed for now.

- Check some details about the company.

2.1.3 Hardware interfaces

The software product does not provide any hardware interface.

2.1.4 Software interfaces

To develop the system we've chosen to use:

- **DBMS**:

  - Name: MySQL
  - Mnemonic: MySQL
  - Specification number: Community Server
  - Version number: 5.6.14
  - Source: http://dev.mysql.com/downloads/mysql/

- **AS**:

  - Name: GlassFish Server
  - Mnemonic: GlassFish
  - Specification number: Open Source Edition
  - Version number: 4.0
  - Source: https://glassfish.java.net/download.html

- **Operating System**: The software product will run in any operating that supports the Java virtual machine and the DBMS and application server described above.

2.1.5 Communications interfaces

Protocol	Port	Service
TCP	80	TCP
TCP	3306	MySQL

For the first release of the software product we will assume that the DBMS and the AS run both on the same physical server.

2.1.6 Memory

The minimum memory requirements are:

- primary memory: 2 GB;
- secondary memory: 40 GB.

2.1.7 Operations

A user can interact with the system as a functional user (anonymous or unregistered, registered customer or employee). For all the users, their functional operations are described in the product functions section.

2.1.8   Site adaptation requirements

The software product requires the following in order to run:

- Java virtual machine;
- AS;
- DBMS;
- primary memory required space;
- secondary memory required space.

Users are required to have installed any of the following web browsers:

- Internet Explorer 10.0+;
- Firefox 24+;
- Chrome 30+;
- Safari 6.0+.

The behavior on previous web browser isn't well tested and can't be assessed.

2.2   Product functions

This subsection provides a summary of the major functions of the software product.

2.2.1   General requirements

We have identified 4 main general requirements:

- managing users;

- managing components;

- managing packages;

- managing customized packages.

The functional and non-functional requirements are defined and explained in detail in the following subsections.

Managing users

- **Functional requirements**:

  - **FR1**: Register to the system;
  - **FR2**: Login;
  - **FR3**: Logout;
  - **FR4**: Modify password;
  - **FR5**: Recover password.

- **Non-functional requirements**:

  - **NFR1**: User password must be stored securely;
  - **NFR2**: System must support high number of users.

Managing components

- **Functional requirements**:

    - **FR1**: Add a component;
    - **FR2**: Remove a component;
    - **FR3**: Modify a component.
    - **FR4**: View a component.

- **Non-functional requirements**:

    - **NFR1**: Only the employees can add, remove or modify components;
    - **NFR2**: System must support high number of components.

Managing packages

- **Functional requirements**:

    - **FR1**: Add a package;
    - **FR2**: Remove a package;
    - **FR3**: Modify a package;
    - **FR4**: View a package;
    - **FR5**: Search packages.

- **Non-functional requirements**:

    - **NFR1**: Only the employees can add, remove or modify packages;
    - **NFR2**: System must support high number of packages.

Managing customized packages

- **Functional requirements**:

    - **FR1**: Customize a package;
    - **FR2**: Remove a customized package;
    - **FR3**: Update a customized package;
    - **FR4**: Search among customized packages.

- **Non-functional requirements**:

    - **NFR1**: Only the customers can customize a package;
    - **NFR2**: A customized package of a customer is modifiable only by that single user, not even by the employees;
    - **NFR3**: Travel packages in the past (that is: where the departing date is before the actual date) must be tagged as "completed";
    - **NFR4**: System must support high number of customized packages.

2.3 User characteristics

Intended users are divided among two groups:

- the customers, that are any kind of user that is willing to travel;
- the employees, that are actually employees of the company.

Both of these two classes of users must still have:

- knowledge in using a browser.

2.4   Constraints

The following constraints apply to the software product:

- **Regulatory policies**: the software product does not have to meet any regulatory policies.

- **Hardware limitations**: the software product does not have any hardware limitations.

- **Interfaces to other applications**: the software product does not interface to other applications.

- **Parallel operation**: the software product must support the operation of simultaneous users specially when working with the adding of packages, modifications of these, and so on.

- **Audit functions**: the software product does not perform any audit.

- **Control functions**: the software product does not control any device or any other system.

- **Higher-order language requirements**: the software product requires basic knowledge of HTML, Java and JEE technologies.

- **Signal handshake protocols**: the software product does not manage any handshake protocol.

- **Reliability requirements**: the software product does not require any specific requirements to perform and maintain its functions under normal operation.

- **Criticality of the application**: the software product requires proper support for concurrent users.

- **Safety and security considerations**: the software product does not require any safety and security considerations (only the users passwords must be saved in a safely way if possible).

2.5   Assumptions and Dependencies

The requirements in this document are grounded on the following assumptions:

1. The Java virtual machine is already installed on the OS.
2. Users have a decent and acceptable Internet connection.
3. The software product provides one employee user by default.
4. The software product supports any number of employees and customers.

2.6   Apportioning of requirements

Future releases of the software product may provide support for:

1. Friendship among users.
2. Sharing of packages among users
3. An administrator to handle the errors, with access to everything.
4. Logging and auditiong of operations.

Chapter 3

Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface is the website described in the previous chapter.

3.1.2 Hardware Interfaces

The software product does not provide any hardware interfaces.

3.1.3 Software Interfaces

The software product does not provide any software interfaces.

3.1.4 Communication Interfaces

The software product does not provide any communications interfaces.

3.2   Functional Requirements

3.2.1   Scenarios

Unregistered user search for a package

Unregistered user search for a package	
Code	SCS001
Description	Describing how the unregistered user search for a package
Goal	**G1**: Allow people to search for packages
Assumption	User is not registered

Alfonso is willing to spend the Christmas holiday abroad in Paris, and has just heard about this new company called TravelDream. He then decides to check the database of such company and search for an interesting travel package.

He then:

1. He connects to the Internet, and opens a web browser installed on his computer.
2. He then inserts the URL of the website of the company.
3. As suggested by the landing page, he goes to the packages list.
4. He then search for "Paris", and checks the "Milan-Paris 5d" travel package.
5. Interested by the offering and the price, goes on and joins the website.

Customer registration in the system

Customer registration in the system	
Code	SCS002
Description	Describing how a customer registers in the system
Goal	**G2**: People can join the system as customers
Assumption	User is not registered The user is already on the landing page of the website There's a database with the names of the employees

Alfonso has discovered this cool website of this great travel agency called TravelDream. He wants to join it to be able to buy some travel package.

He then:

1. Follows the link on the left menu which says "Sign Up".
2. He then chooses a user name and a password, and then insert his real name, the email address and his credit card information. He also choose to be registered as a customer. He could choose to be registered as an employee, but we could check that his name isn't in our database.
3. He then saves his data clicking on the button "Save".

Employee registration in the system

Employee registration in the system	
Code	SCS003
Description	Describing how an employee registers in the system
Goal	**G3**: Employees can join the system
Assumption	User is not registered The user is already on the landing page of the website There's a database with the names of the employees

Bianca has just been hired to work on the TravelDream company, and they've requested her to create her own profile on the website of the company.

She then:

1. Follows the link on the left menu which says "Sign Up".
2. She then chooses a user name and a password, and then insert her real name and the email address, and chooses to be registered as an employee.
3. She then saves his data clicking on the button "Save".

User logs into the system

User logs into the system	
Code	SCS004
Description	Describing how an user logs into the system
Goal	**G2**: People can join the system as customers **G3**: Employees can join the system
Assumption	User is registered The user is already on the landing page of the website

Alfonso wants to log into the system to check its personal information.

He then:

1. Follows the link on the left menu which says "Login".
2. He then write his user name and password in the form, or he can decide to recover his password if he doesn't remember it.
3. As he log into the system, he has now the possibility to check his personal information and his travel packages.

User change his/her data

User change his/her data	
Code	SCS005
Description	Describing how an user can change his/her data on the system
Goal	**G2**: People can join the system as customers **G3**: Employees can join the system
Assumption	User is registered and logged The user is already on the landing page of the website

Alfonso wants to change his password and email on the system.

He then:

1. Follows the link on the left menu which says his name, "Alfonso B.".
2. He then modify the informations about himself he wants to modify, and presses the save button.

Employee adds a component

Employee adds a component	
Code	SCS006
Description	Describing how an employee can add a new component to the system
Goal	**G4**: Employees can add/remove/modify basic components
Assumption	Employee is registered and logged The user is already on the landing page of the website

Bianca wants to add a new hotel in Paris that has just accepted to be part of the TravelDream company system.
She then:

1. Follows the link on the left menu which says "Add Component".
2. She insert all the informations about the hotel: the location, the description, the price per night and person, and so on. She then presses the save button.

Employee modify a component

Employee modify a component	
Code	SCS007
Description	Describing how an employee can modify a component
Goal	**G4**: Employees can add/remove/modify basic components
Assumption	Employee is registered and logged The user is already on the landing page of the website

Bianca wants to modify the informations about the hotel "Albergo Miramare" in Rimini.
She then:

1. Follows the link on the left menu which says "The Components".
2. She search for the component writing into the search bar "Miramare", and clicks on search.
3. Among the results, she clicks on the element she was looking for.
4. She modify the data of that component and presses the "Save" button afterwards, or she could even remove it if needed.

Employee adds a package

Employee adds a package	
Code	SCS008
Description	Describing how an employee can add a new package to the system
Goal	**G5**: Employees can add/remove/modify travel packages
Assumption	Employee is registered and logged
	The user is already on the landing page of the website

Bianca wants to add a new travel package with some excursions and travels means to Paris. She then:

1. Follows the link on the left menu which says "Add Package".
2. She insert all the informations about the package: the name, the days, and so on.
3. Then she adds all the components of the package: the flights, the excursions, the hotel, and so on. The price will be determined by the sum of all the prices of the components, plus a percentage defined by her for the company profit.
4. She then presses the save button.

Employee modify a package

Employee modify a package	
Code	SCS009
Description	Describing how an employee can modify a package
Goal	**G5**: Employees can add/remove/modify travel packages
Assumption	Employee is registered and logged
	The user is already on the landing page of the website

Bianca wants to modify the informations about a travel package for Rimini. She then:

1. Follows the link on the left menu which says "The Packages".
2. She search for the component writing into the search bar "Rimini", and clicks on search.
3. Among the results, she clicks on the element she was looking for.
4. She modify the data of that package and presses the "Save" button, or she could even remove it if needed.

Customer customize a package

Customer customize a package	
Code	SCS010
Description	Describing how a customer can customize a package
Goal	**G6**: Customer can customize packages
Assumption	Customer is registered and logged The user is already on the landing page of the website

Alfonso has just finished registering in the website and is now able to look for the travel package to "Paris" we wants to buy.
He then:

1. Follows the link on the left menu which says "The Packages".
2. He search for the package writing into the search bar "Paris", and clicks on search.
3. Among the results, he clicks on the element he was looking for.
4. In the page with the information about the package, he could either check the information about the single component of the package (by pressing on the name of such element), or he could go on with the customization process pressing the "Customize" button.
5. In the customization page, he can now choose when we wants to leave and return, and when to go to the excursions. The price will be updated with each modification he does.
6. He modify the data of that package and presses the "Save" button.

Customer modify a customized package

Customer modify a customized package	
Code	SCS011
Description	Describing how a customer can modify a customized package
Goal	**G6**: Customer can customize packages
Assumption	Customer is registered and logged The user is already on the landing page of the website

Alfonso has just discovered that he cannot return the 3rd of January because he has to go back working that same day. He then decides to modify the travel package.
He then:

1. Follows the link on the left menu which says "My Packages".
2. He clicks on his only package bought, the travel package to Paris.
3. In the page with the information about the package, he modifies the returning date. A fee is applied for the modification, but he also will get a refund for the days he's not spending anymore in the hotel.
4. He modify the data of that package and presses the "Save" button.

3.2.2   Analysis model

The analysis model represents the core concepts; the following diagram introduces the conceptual classes that we have decided to include in the software product.
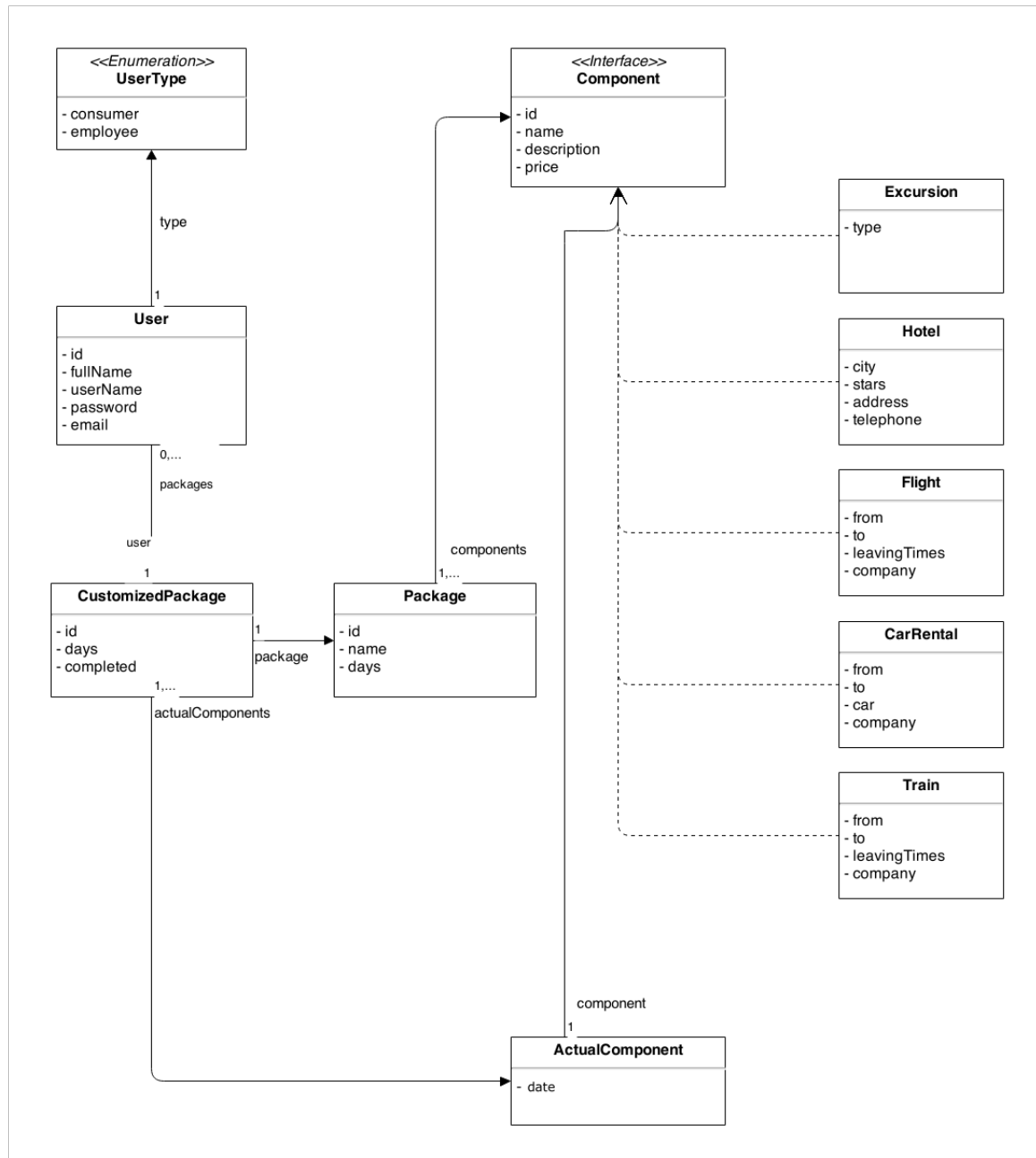
3.2.3   Activity model

Since the main goal of the software product is to create a customized package, below we introduce the activity diagram for doing so.
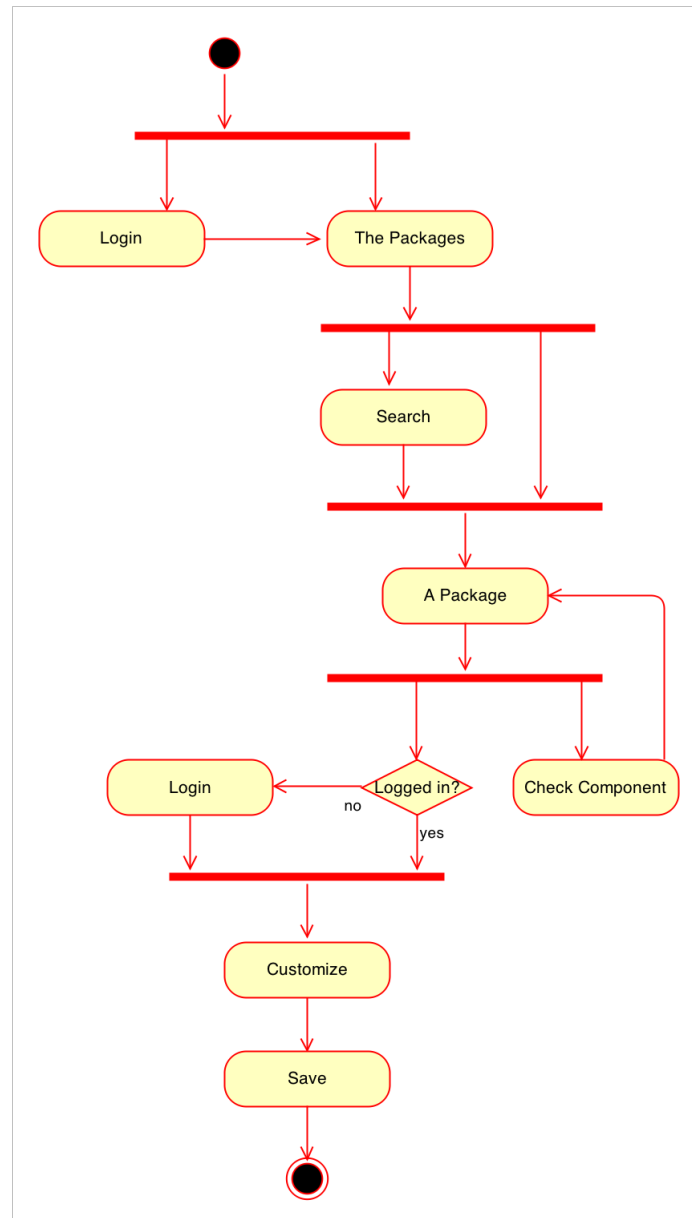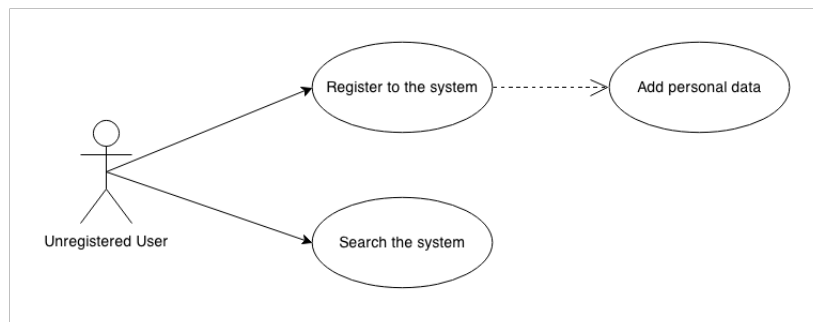
3.2.4 Use case model

The use cases presented are divided by **actors**, that in our system are:

- **unregistered users**;
- **customers**;
- **employees**.

Unregistered user

The unregistered user can search the entire network without being registered or can register into the system.

- Searching:

Searching	
Code	USC001
Description	Unregistered user searches for a package
Goal	**G1**: Allow people to search for packages
Assumptions	
Actors	Unregistered user
Entry condition	The unregistered user navigates to the homepage of TravelDream
Exit condition	The list compliant to the search is displayed to the unregistered user
Flow of events	
1. The unregistered user navigates to the homepage of TravelDream 2. The unregistered user inputs in the search box the package he/she is looking for 3. The system provides him/her with the search results matching with the travel packages with the name containing the query, or with a component with the name containing the query. 4. The unregistered user clicks on a result, and reads the details about the offering.	
Exceptions	
Special requirements	
Nonfunctional requirements	

Customers

The customers can search the entire network, log into the system, customize a package, and modify a customized package.
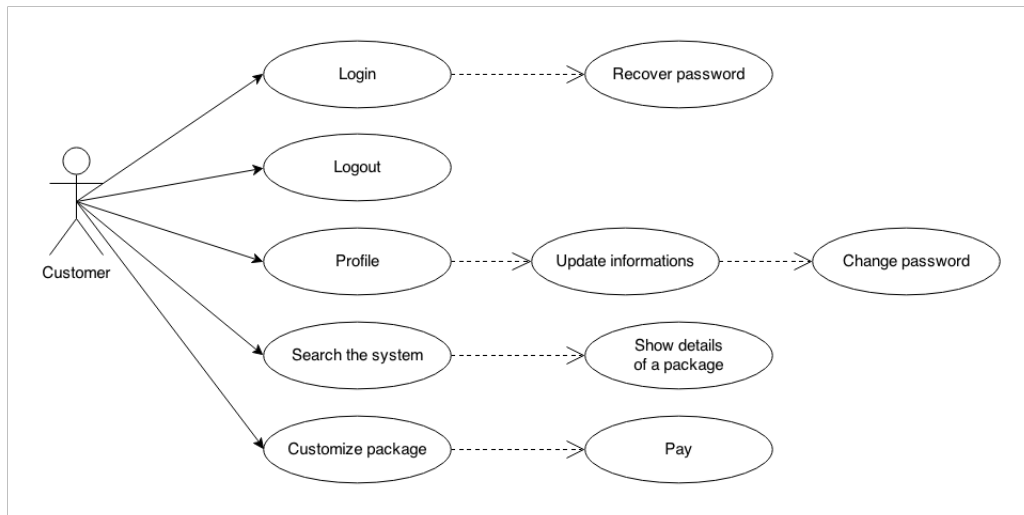
Employees

The employees can search the entire network, log into the system, add/remove/modify a package, and add/remove/modify a component.

3.3   Performance Requirements

The software product requires that every web page shall download in 15 seconds or less.

3.4   Design Constraints

The software product must be designed and implemented with JEE technologies, in particular EJBs for the business logic.

3.5 Software System Attributes

3.5.1 Reliability

The software product does not any reliability factors because in case of malfunction it will cause minor inconveniences.

3.5.2 Availability

The system shall be available 24 hours per day, 365 days per year.

3.5.3 Security

The software product must provide secure storage of the passwords of its users. This can be achieved by using any cryptographic techniques.

3.5.4 Maintainability

The database must be backed up periodically, so that new connection and skill data is not lost in case of malfunction.

3.5.5 Portability

The software product can be installed in any operating system that supports the Java virtual machine and its dependent components.

3.6 Other Requirements

The software product must provide understandable messages in text form in the event of errors, and instruct the user on what to do.