University of Trento
Information Engineering and Computer Science Department (DISI)

Master Degree in Computer Science - Requirement Engineering

Surf Easy: an online platform for surfers

Students:
Tiziano Antico
Anastasiia Karpenko

Professor:
Prof. Anna Perini

Assignment submitted in 2016

Contents

Revision History

Date	Description	Author	Comments
02.03.16	Version 1	Tiziano Antico Anastasiia Karpenko	Identify the problem, the goal of the system and the stakeholders. Page 4-7
08.03.16	Version 1.1	Tiziano Antico Anastasiia Karpenko	Software requirements added in the specification: both user stories and functionalities of the system. Page 8
16.03.16	Version 1.3	Tiziano Antico Anastasiia Karpenko	Added the first set of requirements expressed in natural language. Page 16-17
15.04.16	Version 1.4	Tiziano Antico Anastasiia Karpenko	Changed requirement matrix added. Page 26
22.04.16	Version 1.5	Tiziano Antico Anastasiia Karpenko	Actor diagram - Early requirements Goal diagram - Early requirements. Page 18-20
23.04.16	Version 1.6	Tiziano Antico Anastasiia Karpenko	Actor diagram - Late requirements Goal diagram - Late requirements. Page 21-22
24.04.16	Version 1.7	Tiziano Antico Anastasiia Karpenko	Class diagram, traceability matrices. Page 23-27
29.05.16	Version 1.8	Tiziano Antico Anastasiia Karpenko	Page description of the GO diagrams added. Page 17-18
01.06.16	Version 1.9	Tiziano Antico Anastasiia Karpenko	Risks analysis model completed. Page 28-32
02.06.16	Version 2.0	Tiziano Antico Anastasiia Karpenko	Requirements prioritization completed. Page 33-35

Date	Description	Author	Comments
02.06.16	Version 2.1	Tiziano Antico Anastasiia Karpenko	Added a new goal in LR diagram: “Use feedbacks”. Page 22

Date	Description	Author	Comments
02.06.16	Version 2.2	Tiziano Antico Anastasiia Karpenko	Added a new goal in LR diagram: “Provide the possibility to upload video/photos”. Page 22

Date	Description	Author	Comments
03.06.16	Version 2.3	Tiziano Antico Anastasiia Karpenko	Secure requirements analysis added. Page 37-40

Date	Description	Author	Comments
03.06.16	Version 2.4	Tiziano Antico Anastasiia Karpenko	Analysis of how the privacy law impact the relevant goals in the LR goal model. Page 36

Date	Description	Author	Comments
03.06.16	Version 2.5	Tiziano Antico Anastasiia Karpenko	Added an unique id for requirements. Page 16-17

Date	Description	Author	Comments
04.06.16	Version 2.6	Tiziano Antico Anastasiia Karpenko	Requirements FUN19-FUN20 added in "Changed requirements matrix". Page 26

Date	Description	Author	Comments
05.06.16	Version 2.6	Tiziano Antico Anastasiia Karpenko	Requirement FUN22 added in "Changed requirements matrix". Page 26

Date	Description	Author	Comments
06.06.16	Version 2.6	Tiziano Antico Anastasiia Karpenko	Included requirements related to law. Section "New requirements included". Page 37

Document Approval

Signature	Printed Name	Title	Date
	Tiziano Antico Anastasiia Karpenko	Requirement Engineering - students	
	Anna Perini	Instructor, 145072	

1 Introduction

The purpose of this document is to collect, analyze, and define high-level needs and features of *Surf Easy* (an on-line platform for surfers). It focuses on the capabilities needed by the stakeholders and the target users, and why these needs exist. The details of how *Surf Easy* fulfills these needs are detailed in the use-case and supplementary specifications.

2 Part I

In this first section of the document, there would be described: the purpose, scope, definitions, acronyms, abbreviations and references of our project.

2.1 Purpose

Organize surfing activities is a very time consuming stuff. Each time a person wants to go surfing, he has to organize everything needed for the purpose. Most of the time, all the information and all the stuff he needs are not in the same place. For this reason, before he organizes everything he has to look over lots of web-sites and waste lots of time. Since surfing places are distributed all over the world different surfing services providers have web-sites in their local language that is not understandible by the majority of users.

When a person decides to go surfing he needs to find a good accommodation, check the weather forecast to see if he chooses the right day for this kind of activity. For instance, if there is no wind, there will not be any waves and there are no ways a surfer would enjoy it.

Another important aspect to check, before leaving, is find a good spot. Not all the places are good for surfing and, sometime, can be difficult to find the right one without asking other surfers' opinions. Also, if you want to find an accommodation, you need to check other web-sites and see if what you have found is near to the spot or the school you are planning to go. Besides the fact that everyone is looking for the cheapest accomodation/offer of that moment and the only way for doing so is checking more than one web-site till you find the best solution which fits your disposable income. Another problem they usually face themselves is about finding a good surf school where they can learn or improve their technical skills. The choice of the surf school (especially for beginners) is an essential task. Surf school and the professional experience of a surf teacher makes a direct impact on the surfer's experience and influences many issues like safety and security of the surfer. Surfers need also to look at events related to this activity (non-professional/professional surfing competitions, surf camps, yoga classes for surfers, thematic beach parties, festivals etc). You can find them all over the Internet and it might be difficult to find the one you are looking for without wasting lots of time on the research.

While a surfer gains experience in this sport he constantly changes the type of board and needs to recycle his old one. There is a lack of online platform for second hand equipement that will connect the owners of equipement and buyers

together. For all of mentioned reasons we believe that building a system, based on a web-platform which contain all the services needed by a surfer, would be very helpful for those who are going to practice this sport. It will satisfy all the users' needs and will help them to book/buy all the stuff needed for surfing in a very easy and fast way.
We want to help surfers enjoy this great sport even before they jump on their surfboards!

2.2 Scope

Though surfing is a very popular sport in the world (some sources mention the number of 23 million people around the world), there is still no tool for planning a surf trip easily. Lots of people have troubles to organize everything needed for surfing by themselves without specialized tourism agency. Surfers have to spend lots of time on looking various web-sites where they can find all the information needed.

To solve these issues, we have decided to build an all-in-one online platform for surfers that will hold all the centralized information about surfing in the world. For example, they would be able to find and book various services, register to and participate in the surfing - related events worldwide, buy or rent the second - hand equipement and organize their surf trips/trainings. The software will help surfers to minimize the time they spend looking at various web-sites, letting them to find all the information in just one place.

Thanks to this platform, we want to simplify the following user activities:

- **search/book a surf school**: there are lots of surf schools all over the world. We want to group them in just single place. For the user would be much more easier to find the cheapest school in a specified location, without looking at web-pages which offer similar tasks. Also, each schools, usually sell various packages for surf lessons: from the beginner level to more advanced levels. Looking at the list of available schools, the surfer can find also the surf lesson which fit better for his/her needs. The choice of the surf school will be also influenced by the rating of the school made by the previous users (similarly to TripAdviser feature).

- **search/book an accomodation**: one of the main needs for a surfer is booking an accomodation near the place where he/she is going to surf. Most of the time, it happens that a person who is passionate to this sport decides to spend some time in that surfing area. For this reason, he/she is looking for a cheap accomodation but, at the same time, it has to be near not so far from the place he/she wants to surf. This platform wants to help surfers to solve also this kind of issues.

- **buy second hand stuff**: due to the high number of second hand shops, a surfer has to check more than one web-site to but the bost suitable and cheapest tool for surfing. Lots of time is needed to be spent for this activity as well. We thought he/she might need one single place where all the items sold over more second hand shops are grouped (and made available) to be bought by a single user. At the same time the surfers who would like to sell their own equipement will receive an opportunity to do that online without middlemen or second hand shops.

- **let them join surfing events**: there are lots of surfing-related events all over the world and it might be difficult, for the surfer, to identify the one which is more attractive to him. We want to group, by place, the available events, thus so the user will find (and join) the most important events to himself/herself in an easily a fast.

Most of the time, it is difficult for the user to identify the best option to choose. We are planning to give space to users who have tried on themselves some issues. For some sections of this platform we want to have a place where users can share their opinions on some particular issue. They can write feedbacks so that, it would be easy for surfers to identify the best choice they can make.

2.3 Definitions, Acronyms, and Abbreviations

In this subsection there are the definitions of all terms, acronyms, and abbreviations required to properly interpret the *Surf Easy* system.

Abbreviation	**Definition**
SE	Surf Easy
OPFS	Online Platform For Surfers

Table 1: Abbreviations

Term	**Definition**
Spot	The place where surfers can go surfing
Sign Up	Enter your personal data in a form to be stored into a database
Sign In	Process that allow the user to join the system (after he/she has entered Id and password)

Table 2: Terms

2.4 The stakeholders

There are four types of users involved in this product:

Stakeholder	**Description**
Surfers	They have to organize and book everything to go surfing. It is the person who wants to: find surf location appropriate for them, find and register to take part in surf competitions/events, book accomodations, buy or rent second hand surf equipmentgadgets and buy packages for surf lessons and surf schools. Surfers vary from beginners to experts. Everyone should be able to find services specific for their level.
Surf Event organizer	They provide and organize surf events, sport events (yoga campus, fitness, skating - all related to surfing), surfing theme/beach parties, surfing carnivals and festivals, surf photographers, surfing trip organizers. These actors are interested to get as much visitors as possible for their events and in specific locations where they are located.
Surf teacher/schools	They are surf instructors that teach surf lessons to the ones who want to learn or improve surfing skills. Either it can be individual instructors or most probably they are a part of a surf school. Each of them has several years experience of teaching , the more experienced they are the better are their services and the more reliable they are.
Equipment providers	They sell products for surfing, such as: surfboards, swimwears, etc. They can be both second hand shops and first hand shops, individuals who want to sell their second hand equipment.

Table 3: Stakeholders

2.5 Software Requirements

The first part of this section describes the identified sources of requirements and why they have been decided to be used.

The second part of the section focuses on the description of the chosen requirements elicitation techniques. It follows an explanation of how requirements elicitation has been planned.

The third and last section describes a set of requirements expressed in natural language. It follows the rules for writing *good requirements.*

2.5.1 Identified sources of requirements

For the development of this on-line platform, we have decided to find information about competitors on the market. Our purpose is building an all-in-one web-platform. For this reason, we have decided to analyze existing software systems to get inspired by them. We want to study how these systems have been designed and built. Finding out both *functional* and *non-functional* requirements to apply and improve in our *Surf Easy* platform.

In a second moment, we have decided to build a *prototype*, to help the stakeholders to understand better how the platform should look like. Using a prototype, we have been able to find other requirements also.

2.5.2 Elicitation techniques chosen

For this project, we have decided to choose and use **traditional techniques** and **collaborative techniques**. Beside lots of them, we have decided to look at *existing similar systems* and *analyzing data* provided by them. More in datails, we have compared some of the well-known web-sites to see how they have managed *functional* and *non-functional* requirements.

In a second moment, we have used the **collaborative technique** of *prototyping.* For this purpose we have used the on-line mock-up tool called *Moqups.* Using a prototype of our system we want to give to users, developers and customers a better understanding of the system requirements identified using the previous technique. It will help also the stakeholders to find new functionalities, discuss ability and establish priorities.
We want to underline that our prototype will not be the final version of our product. It might look completly different at the end.

2.5.3 Traditional techniques used

Specifically, we have analyzed the following web-sites:

- **TripAdvisor**: it is a web-site which, basically, gives the user the possibily of leaving feedbacks about a particular place visited or activities done in the selected place. So that, other users can read these feedbacks and decide where to go for their next travel and what they should avoid. By studying this web-site we have mainly focused our analysis trying to understand how the feedbacks' system work and how it has been managed.

  System analysis The system allows the user to select a specific place he/she wants to go. After that, it shows a menù where the user can choose what he/she wants to see.

  For example, we can look at hotels, activities, things to do, etc. By choosing one of them, the system shows a list of items rated by other users related to the user's research. The ranking start from the top activities (the ones which have got the highest number of positive feedbacks) to the activities which got the lowest number of positive feedbacks. It is also possible to order the results based on the number of friends who have given positive feedbacks.

  By clicking on one of the items showed in the list, you can both read a more datailed description about the item and you can look at the following sections:

  - **reviews**: you can both read all the reviews written by users and write new ones;
  - **questions & answers**: In this section, the users who want to know something about particular issues related to the selected item, can post questions and receive answers from other users;
  - **location on the map**: in this section a map of the selected item is shown.

  Non functional requirements system analysis Overall speaking, the web-pages are composed by essential elements (they have a minimalistic design). The search bars use AJAX response: while the user is typing something, the system shows some suggestions about what he/she is going to look at. The information are well-organized in sections or submenu. This implies a faster recovery of them and a less impact for the user to recognize where the information are.
  The web-site uses the same search bar at the top of each page. Doing so, the user can easily change the parameter of the research all the time he/she wants to.
  It is important to point out also that, each time you move the cursor over some elements of the web-site, it provides a feedback to the user (by underlying words, changing color of the item, etc.). In this way you are

always aware where you point the mouse.
The system uses Google Maps services to show where an item is located.

- **Booking.com**: it is a web-site that allows the user to look for an accomodation, at the cheapest price, after choosing a place where he/she wants to travel. By studying this web-site we have mainly focus on how the booking system has been projected and which functionalities it includes.

  Functional requirements system analysis The home page of the site contains a search bar where you have to type the name of the hotel you want to go or, otherwise, the name of the destination. The second step is to establish the date of check-in and the date of check-out. This is done by a date picker widget (to prevent any possible mistakes by the user while selecting a date range). Finally, you have to specify the number of rooms you wish to book, the numer of adults and childrens who are going to take part in the trip.

  When you start the research, the system shows you all the possibilities which matches you needs. It is possible to order the items in the list by: lowest price, number of positive feedbacks, distance from the city center and the ones suggested by the system itself. Each item on the list, is shown with a picture. You have also the possibility to filter the results based on some criterias (price, number of positive feedbacks given by people who have already been there, services, etc...).

  By clicking over one result, you can see a more datailed description and some pictures of it. You have also the possibility to modify the check-in date and the check-out date. There is a table that shows you the number of available rooms, the max number of people who is allowed to be there, the price, the conditions and the number of rooms you want to book.

  Once you have set everything, you can book your room, filling your personal information and your credit card information in a secure booking page (which uses the HTTPS protocol).

  Non functional requirements system analysis The main page of the site is a bit confusing. There are lots of information that prevent the user to focus in what he/she is looking.
  A yellow form is shown at the top of the page. This might help the user to perform, in an easy way, the main task of the site. The system, once you do a research, does not show you the list of all the results but it paginates them by showing fifteen items in each page. In this way, the user does not have to scroll down the page a lot and it makes less confusion in recognize the righe solution for him/her.
  The system, at the top of the page, displays a box of instagrams which shows the variation in term of day and cost of the solutions. This is done by a graphical way, thus the user can easily find out this information.
  The web-site uses some select boxes to filter the results as well.

- **Hostelworld.com**: it is another services which offer the user the possibility to look for the cheapest accomodation in a selected area. This site has a different structure compared to booking.com and, for this reason we want to compare both of them and find out the best solutuions to use in our on-line platform. This system is useful to be analyzed since our platform should provide similar tasks. In particular, it should allow to book surf lessons and accomodations at the cheapest and affordable price.

  Functional requirements system analysis At first, in the main page the user has to fill all the information like: city (or hostel name), arriving date, departing date and number of guests.

  After that, you just need to click the orange button below and start your research. The system proposes to you a list of hostels. There are three tabs: view list, map view and hostel reviews.

  In the "view list" there is the possibility select how many results we want to see in each page and there is also the possibility to sort by:

  - availability;
  - overall rating;
  - price;
  - name

  The other tab is "map view". It shows the map (in a visual way) where the result items are located. The third tab, "hostel reviews" shows the results from the top rated ones to the lowest rated ones.

  On the left side of the page, you can set some filters: currency, price range, rating, property type, room type, facilities and payment type.

  By clicking on the interessed item, you are redirected in another page where you can see photos about that item and, next to it, related information (with the possibility to "read more". In this way the page is clean and well-readable). Next-to the property information there is the overall score it got. Scrolling the page, there is a section where it is possible to modify departing and arriving days and a table shows all the possible solutions available for the selected days.

  At the end, all the facilities are written and you can book your place to sleep. You will be redirected to a "secure page" where you have to fill all the personal information and the credit card information.

  Non functional requirements system analysis The main page is very minimalistic. The user can find what he/she is looking for in a easy and efficient way. There is a search bar in the middle and some more filters to set the research.
  The page which contains the result is very clear. The results are aligned in the center of the page so that, the user mainly focus the attention in that part of the page.
  On the left side of the page, there are some filters to reduce the results of the research. It uses checkboxes and scroll menù.

The overall system has a very good balance of color. The main menù has a dark background; while the content of the site has a white background. In this way the user's brain recognize easily in which part of the site the information has to be checked.
The site is very intuitive and not difficult to be surfed. It alerts also the user if some operations are not allowed, by showing pop-ups.

- **Magicseaweed.com**: this web-site is the major weather and forecast web-site for surfing community. It gives an overview of the surf spots around the globe that are submitted by the users and provides professional weather forecast for each of them. We decided to take this web-site as an example for elicitaton of requirements because it represents the part of functions we would like our platform to perform.

  Functional requirements system analysis The main menu of the web-site provided the opportunity to choose the surf spot wherever you are interested (Continent, Region/country, name of the spot) and see the recommendations and photos of the users about the spot, describtion of the spot and weather forecast. If you choose the country or the region, on the other page there will be the list of surf spots available in this region with short wave and weather forecast about each of then. If you click on the surf spot you will see more detailed describtion of it with more detailed wave and weather forecast. You can also look at or upload photos made by local people, submit surf report for specific surf spot and location. The users can upload photos and videos for the locality and surf spot if they want. Each user can create his our account and save his favorit locations, calibrate his forecasts and save language and location preferences.

  Non functional requirements system analysis The main page is mostly minimalistic, featured by the big header image. There are two menu lines in the beginning of the page. One menu line gived the opportunity to chose fotos or videos or to go to the online surf shop. The second menu offers the opportunity to choose the location of the surf spot. Then down the page other opportunities are listed : the window for account name and features, window for the surf shop categories, winow for latest news, videos and photos.

- **Theoryandpractice.ru**: Theory and practice is a web-site in Russian language about the public events in the city, lectures, master classes etc. It provides a user an opportunity to register to an event and to see who else is going.

  Functional requirements system analysis The main valuable function of this web-site we would like to consider for our project is registration of the user account and search for events that will be in the user's location of interest. The main menu of the web-site has a button "lectures". On the opened page we can see the left - side menu with the search parameters for the events (location, topic, free of charge/paid, type of event). On the right side of the page there are all events in the calendar order. On the page of each event you can read the description of it and if you want to push the button “I will go” or/and “Add to calendar” (that will automatically add this event to Google Calendar. It is also possible to see who of other users are going to an event. The page of account registration is minimalistic, the user can register via Facebook which can be very useful and fast.

Non functional requirements system analysis The web-site is very minimalistic, clean and pleasant to look at. Every feature is presented in the clear way on the white background. The pages are not overloaded with information and data is presented ina harmonic way.

2.5.4 Collaborative techniques used

In this subsection, we are going to show the prototype we have started to build. You will find some web-pages, with their explanation, built using the tool *Moqups.*

In 1 is shown a possible main page to get access into the platform. In the case of a new user, he/she has to sign-up into the system before using the OPFS. He has to write some personal information and then process the request.

Once the user information are stored into the database’s platform, he/she needs to write the user-name and password at the top of the page. Doing so, he/she can log-in the system and have access to all its functionalities.

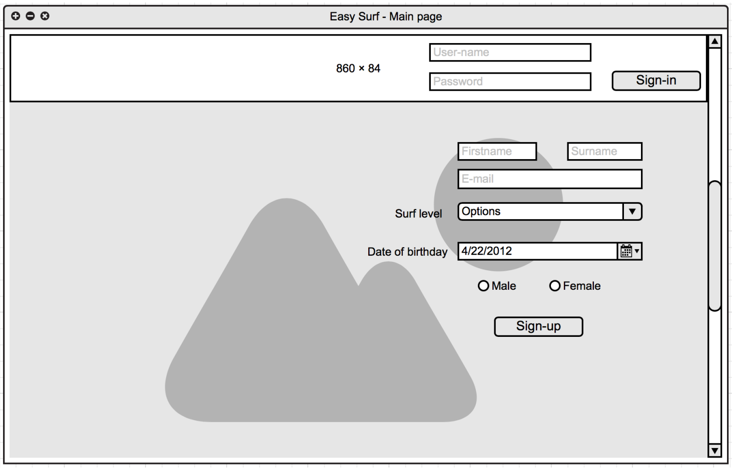


Figure 1: Main Page

At this point, the user can start searching what he/she wants. In this second web-page, there is the avatar of the user (it might be the profile picture) and the activity he/she can perform on it.
The user can fill the search text area with the place wished and, thanks to

an AJAX call the field would be automatically filled. He/she needs also to write wich kind of activity to perfom, related to the selected place. He/she can search for: surf schools, accomodations and spots. After that, the user needs to establish a range of dates and start searching (as it is shown in 2).

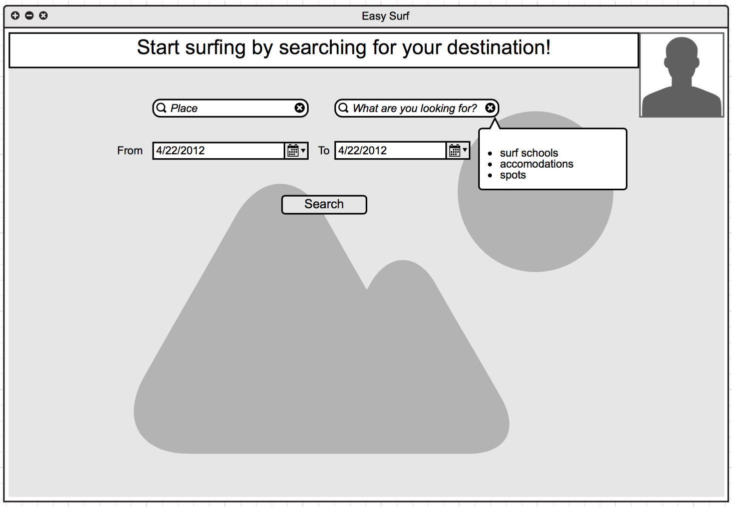


Figure 2: Searching page

At this point, the user will be redirected to another page where a list of results is available. At the top of the page, there is the user's avatar and some fields to modify what you are going to search. At the center of the page, you can see the list of results. Each of them is composed by a picture, a name of the result, a description, avarage number of positive feedback given by the users, a map which shows the location and the eventual price of the item (in the case the user looked for an accomodation).
It is possible to filter the results by setting some parameters. You can order them by price and from the most ranked one to the lowest.
The results are shown by paginate them. This means that in each page, you will have twenty items. Otherwise the user would need to scroll down a lot to look at all the results.

A scratch of the mentioned web-page can be seen in 3.

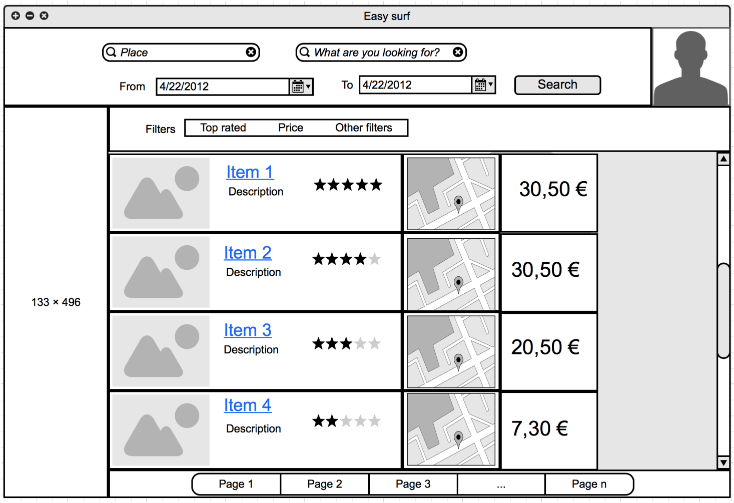


Figure 3: Result of the research

Finally, the user, by clicking on the interessed result, can see the information related to the specific item. In the case of surf school or accomodation, he will have the possibility to book the solution.
After the item description, there is a menù where the user can see feedbacks and write one. Also he can ask some questions to the community, by clicking over the other section of the menù.

The web-page is shown in 4.

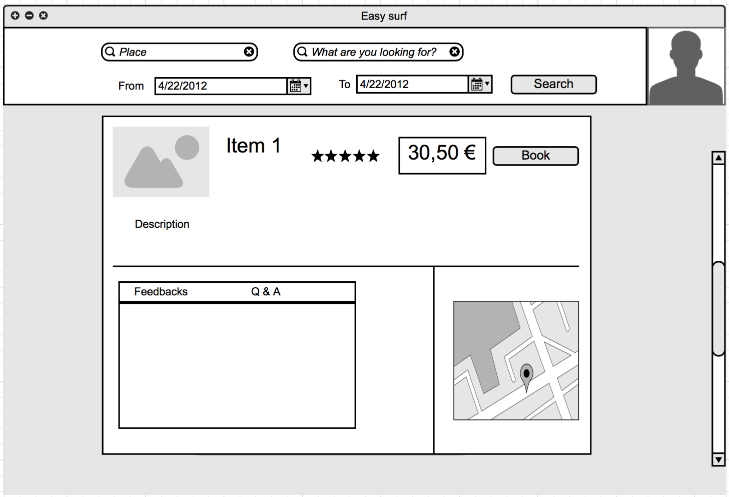


Figure 4: Selected item details

2.5.5 Set of requirements in natural language

In the table below, it is shown the first set of requirements which has been found for the system we are analyzing.

Requirement	**Type**
FUN1. The system **must** provide an authentication page, which allow the user to be identified, by other users that use the platform.	*FUNCTIONAL*
FUN2. The system **must** provide the possibility for the user to sign-up into the system, if it is the first time he/she is going to access it.	*FUNCTIONAL*
FUN3. For every new user, the system **must** get the following information: first name, last name, date of birth, username, password, e-mail address, level of surfing.	*FUNCTIONAL*
NFR1. The system **shall** store the new information about a user into a database system, within no more than 15 seconds, using INSERT queries.	*NON FUNCTIONAL*
NFR2. The system **shall** guarantee security while the user purchases an item. This is done by using the HTTPS secure protocol.	*NON FUNCTIONAL*
NFR3. The system **should** display, in less than 5 seconds, to the user the suggestions in the search bars.	*NON FUNCTIONAL*
NFR4. The system **should** use, in the search bars, AJAX requests to query the database and display the results while the user is typing, within 5 seconds.	*NON FUNCTIONAL*
FUN4. The system **must** provide a rating system which let users to give feedbacks, by using a text area and a likert scale	*FUNCTIONAL*
FUN5. The system **should** make an avarage of the ratings given to a particular item by all the users, within 3 seconds.	*FUNCTIONAL*
NFR5. The system **must** be accessible, in less than 1 minute, by all the users through an Internet connection, over the HTTP protocol.	*NON FUNCTIONAL*
FUN6. The system **shall** show to the user (once he/she has selected a result from the research) the feedbacks written by other users.	*FUNCTIONAL*
FUN7. The system **must** allow to the user to book an accomodation at the selected place.	*FUNCTIONAL*
FUN8. The system **must** allow to the user to book surfing lessons in a specified place.	*FUNCTIONAL*
FUN9. The system **must** provide to the user the possibility to check where a surf school and the accomodation are located, by using a search bar.	*FUNCTIONAL*
NFR6. The system **must** use Google API. In this way, it lets the users to check the location of surf schools and accomodations, by showing them on a world-map.	*NON FUNCTIONAL*

Requirement	**Type**
FUN10. The system **must** provide four authentication levels (common user, event organizer, surf teacher and equipment provider), based on the role of the user going to use the platform.	*FUNCTIONAL*
FUN11. The system **must** provide search menu for surf spots including 3 levels: continent, country and surf spot. This is achieved by using a drop down menu.	*FUNCTIONAL*
FUN12. The system **must** provide to the user the list of events in particular location (of user's choice) within 10 seconds.	*FUNCTIONAL*
FUN13. The system **must** provide the possibility for the user to sign-up into the system using his/her Facebook account, if it is the first time he is using the system.	*FUNCTIONAL*
FUN14. The system **must** provide a search menu for surf events and a possibility for user to sort events by location, type of events, price.	*FUNCTIONAL*
FUN15. The system must provide the user with possibility to see who else is registered for an event.	*FUNCTIONAL*
FUN16. The system **must** provide the user with the map of surf spots in requested area.	*FUNCTIONAL*
FUN17. The system **must** provide the user with the recommendation of nearby spots.	*FUNCTIONAL*
FUN18. The system **must** provide the possibility for the user to upload photos and videos related to specific surf spot by drag and drop method.	*FUNCTIONAL*
FUN19. The system **shall** show to the user a calander where are posted the surf events in the selected days.	*FUNCTIONAL*
FUN20. The system **must** allow to the user to buy/sell surf equipment.	*FUNCTIONAL*
FUN21. The system **must** allow to the user to book surf events.	*FUNCTIONAL*
FUN22. The system **shall** give recommendations to the user based on history of search, history of bookings and user's geolocation.	*FUNCTIONAL*

Table 4: Set Of Requirements

3 Part II

In this section of the document, it will be explained and illustrated the **Goal-Oriented analysis**. It will follow: the **actor diagram**, the **goal diagram** and the **class diagram**.

In the model designed, we have analyzed each both hard goals and soft goals. We can say that all the hard goals have clear criteria for deciding when they are satisfied. Turning into the soft-goals, all of them do not have any clear criteria to decide if they can be satisisfied or not.

The section will end with the definition and explanation of the **traceability matrices**.

3.1 Actor diagram analysis

We guided the actor diagram analysis by answering the following questions:

- **Which are the main stakeholders and existing systems (actors)?** Mainly, the system is composed by the following actors: *Surfers*, *Equipment providers*, *Surf school*, *Surf event organizer*;
- **What's their role in the part of the organization we are considering?**
  - *Surfer*: It is the main actor responsible for booking a surf destination where he can go to surf. He can choose to buy the equipment he needs and join the events he wants to take part of during his vacation;
  - *Equipment provider*: It is the shop (or shops) which posts all the offers available related to the equipment the surfer can think to buy. They are both first hand and second hand online shops;
  - *Surf school*: It is the place/places where the sufer can book surf lessons and can ask for an accomodation during his vacation;
  - *Surf event organizer*: It refers to the people who has the puropose of organizing surf events which can be joined by surfers.
- **What are the goals and soft goals of these actors?**
  - *Surfer* has one hard goal: *Organize a surf vacation*
  - *Equipment provider* has one hard goal: *Provide equipment*
  - *Surf school* has one hard goal: *Give lessons*
  - *Surf event organizer* has one hard goal: *Organize events*
- **How can they achieve them? (can they do it by themselves? Do they have the plan and resources needed?)** All the actors need to use a web platform. It will connect all of them, helping them to achive their goals.
- **Does an actor depend on another one to achieve its goals?** The *surfer* needs to contact the *Equipment provider* to get the equipment he wishes and to sell the used one.
  Surf school provide some locations where the surfer can go and stay over the nights. Also, it also offers lessons to the surfer.
  The *Surf event organizer* creates events which the surfer can join during his staying.

3.2 Actor dependency diagram - Early req.

In the Figure 5, it is shown the **Actor dependency diagram - Early Requirements** for the analyzed project. There are four domain actors, which are: *Surfer*, *Equipment provider*, *Surf school* and the *Surf event organizer.* All the relationships between them are shown: *goals*, *softgoal* and *resources.*

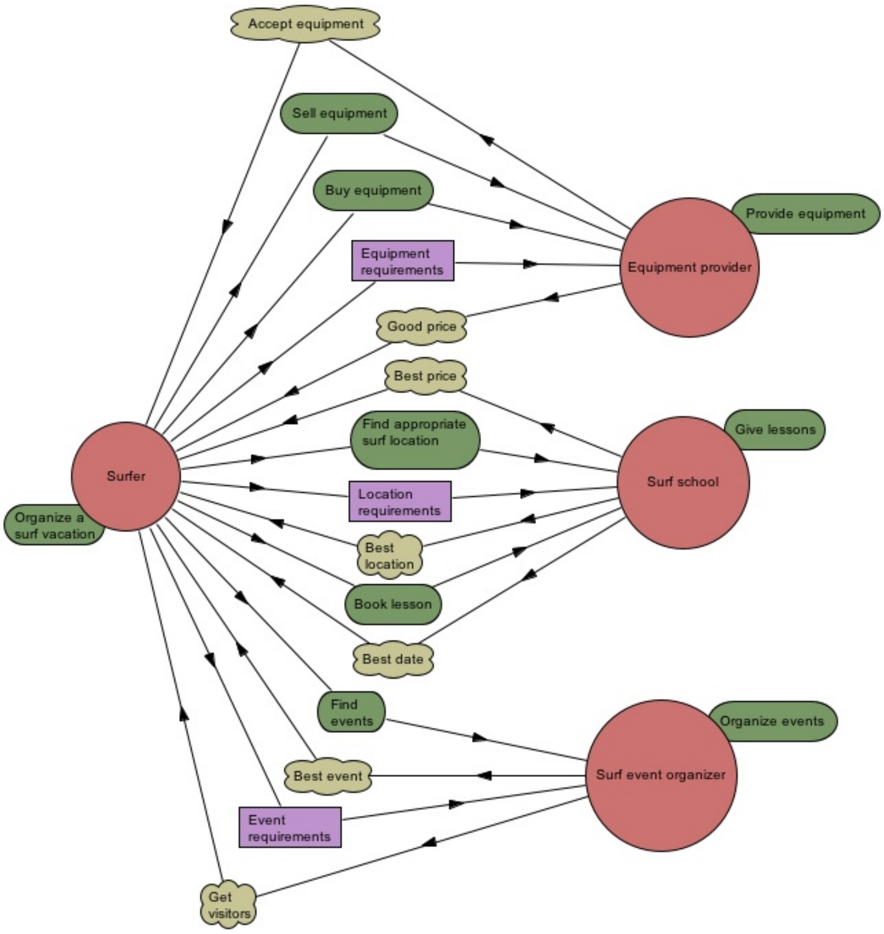


Figure 5: Actor Diagram - Early Requirements

3.3 Goal diagram analysis

We guided the goal diagram analysis by answering the following questions:

- **Can a goal/plan be decomposed into sub-goal/plan so that the root goal/plan can be achieved only once I'll have achieved all the sub-goals (plans)?** The plan of *organazing a trip* is decomposed in three sub-goals by using an *and decomposition.* Each sub-goals are decomposed thanks to both and decomposition and *or decomposition* in other sub-goals.

- **Can a goal/plan be achieved in alternative ways?** It has been used *or decompositions* to let a goal being achived in alternative ways. This is the case of the following goals: *Find equipment*, *Location requirements* and *Event requirements.*

- **What are the plans/resources/softgoals that provide means for achieving the goal?** They are: *By using a platform*, *Event requirements*, *Location requirements* and *Equipment requirements collection.*

- **Are there goals that can contribute positively or negatively in reaching the analyzed goal (plan)?** The goals which can contribute positively in reaching a goal are: *Safety payment*, *Effective organization*, *Flexibility* and *Enjoyable*.

3.4 Goal diagram - Early req.

In the picture 6, it is shown the **Goal dependency diagram - Early Requirements** for the analyzed project. The main user actor is the *Surfer*.

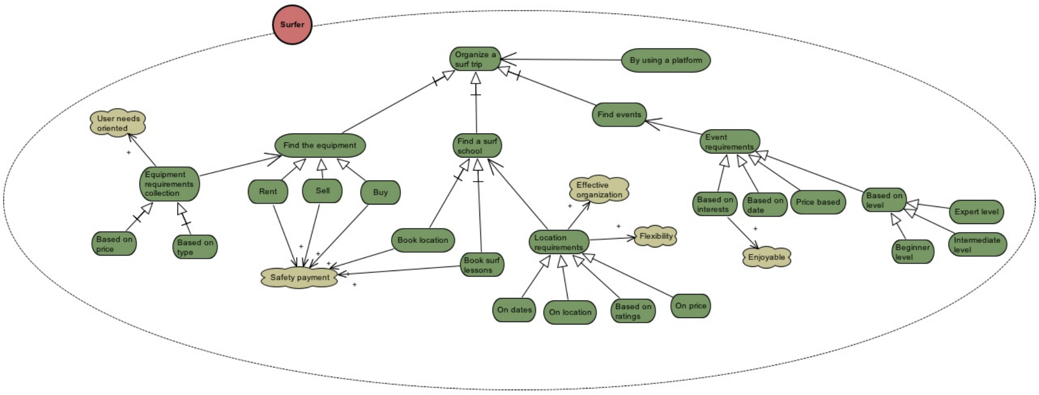


Figure 6: Goal Diagram - Early Requirements

3.5 Actor diagram - Late req.

In the picture 7, it is shown the **Actor dependency diagram - Late Requirements** for the analyzed project. There are four domain actors, which are: *Surfer*, *Equipment provider*, *Surf school* and the *Surf event organizer*.

In this actor diagram, we have introduced the **system-to-be actor**, the **Surf Easy** platform. In this way, we have analyzed the dependencies with actors in its environment identifying system's functional and non-functional requirements.

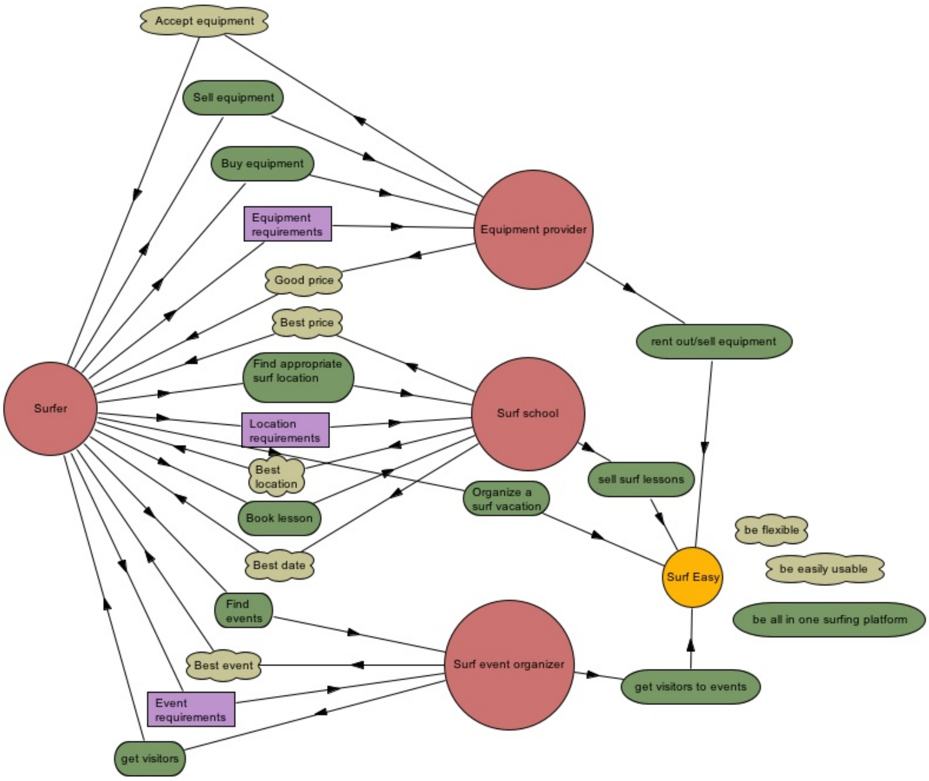


Figure 7: Actor Diagram - Late Requirements

3.6 Goal diagram of the system-to-be actor - Late req.

In the picture 8, it is shown the **Goal diagram - Late Requirements** for the analyzed project. It is analyzed the **System-to-be actor**: *Surf Easy*. In particular, the picture shows which are its goals and how to reach them by using plans, soft-goal and resources.

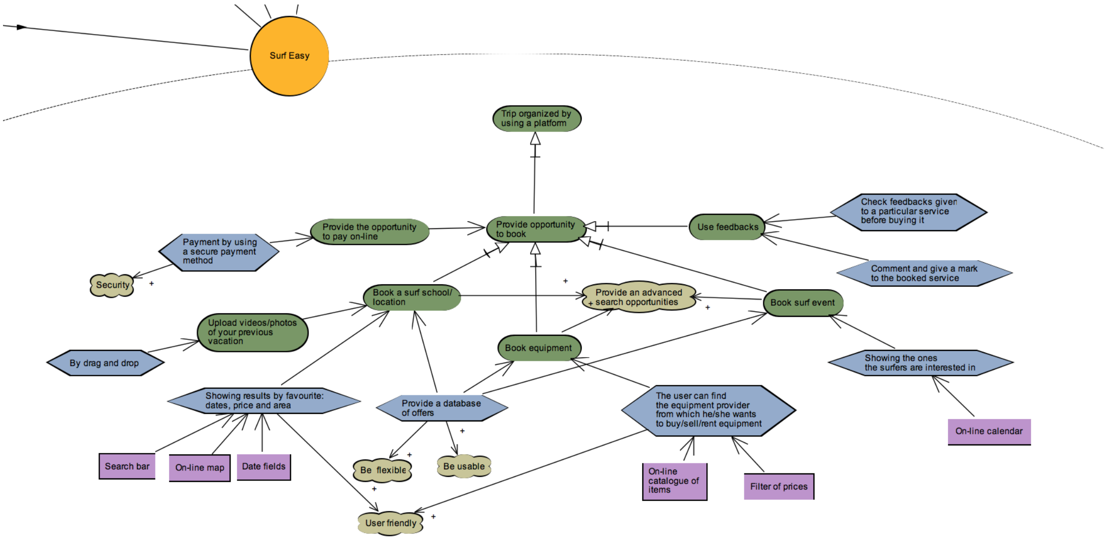

Surf Easy
Trip organized by using a platform
Provide the opportunity to pay on-line
Provide opportunity to book
Use feedbacks
Check feedbacks given to a particular service before buying it
Payment by using a secure payment method
Comment and give a mark to the booked service
Security
+
Book a surf school/ location
+
Provide an advanced search opportunities
+
Book surf event
+
Upload videos/photos of your previous vacation
Book equipment
Showing the ones the surfers are interested in
By drag and drop
Showing results by favourite: dates, price and area
Provide a database of offers
The user can find the equipment provider from which he/she wants to buy/sell/rent equipment
On-line calendar
+
+
Search bar
On-line map
Date fields
Be flexible
Be usable
On-line catalogue of items
Filter of prices
+
User friendly
+

Figure 8: Goal Diagram - Late Requirements

3.7 Class diagram

In figure 9, it is shown the class diagram for the main entities of the application domain. We can describe them briefly:

- *Surfer*: it is the one who has access to the system, after register on it, by using user-name and password. He can take part in events, can contact the surf schools (and buy packages from them). He has the possibility also to have access to the catalogue of first hand (where he can buy/rent equipment) and second hand shop (where he can buy/sell and rent equipment).
- *Event organizer*: he is responsible of creating an event (with *description*, *date*, *location* and *price*). These events are included into a calendar, accessible by the surfer.
- *Surf school*: it has packages to offer to the surfers. Packages can include: *events*, *surf lessons*, *accomodation* and *item to rent/buy* from the surfer.
- *Equipment provider*: he is responsable of the managment of a catalogue where all the items to be rent/sell/bought are listed. The items are grouped by sections. In case of **first hand shops**, they can just sell and rent out items. In the case of **second hand shops**, they can buy, sell and rent out the items.

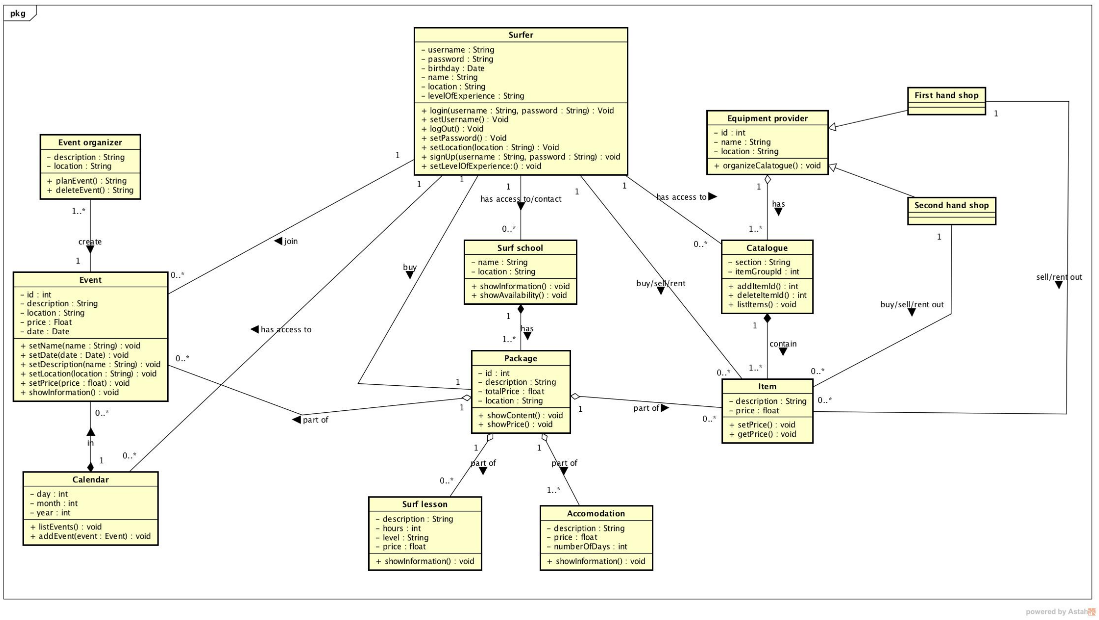


Figure 9: Class Diagram

3.8 Requirements revision

In this second part of the documentation, several requirements have been revisited. This due the analysis done till now which pointed out some possible changes usable to be made. In this way, the system might result more compact and functional.

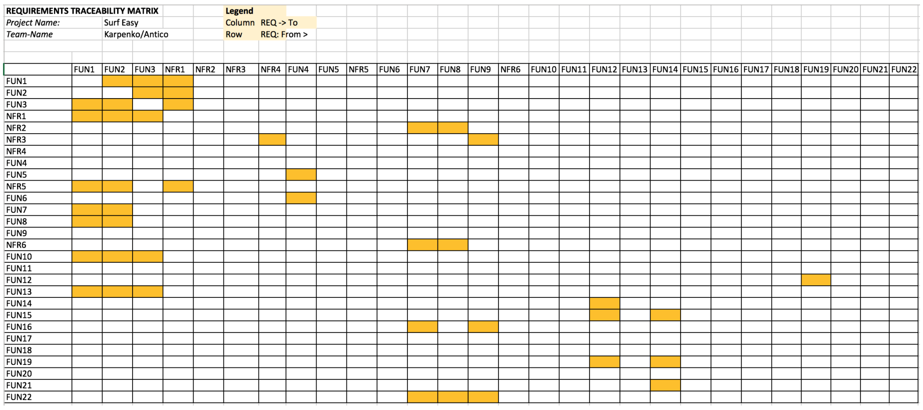
In this section, you can find the screenshots of the traciability matrices used for this assignment.

3.8.1 Traceability matrices

This subsection shows different traciability matrices which help the reader to understand better the dependencies between requirements and how the previous requirements have been modified. Finally, a traciability matrix with relations between requirements and goals is also described.

3.8.2 Dependencies between requirements

In figure 10 you can see the **Traceability matrix** which shows the dependencies between requirements. The matrix shows the relations between the requirements in the row and the ones in the columns. These dependencies mean that some functionalities of the platform cannot be performed without the presence some other ones.



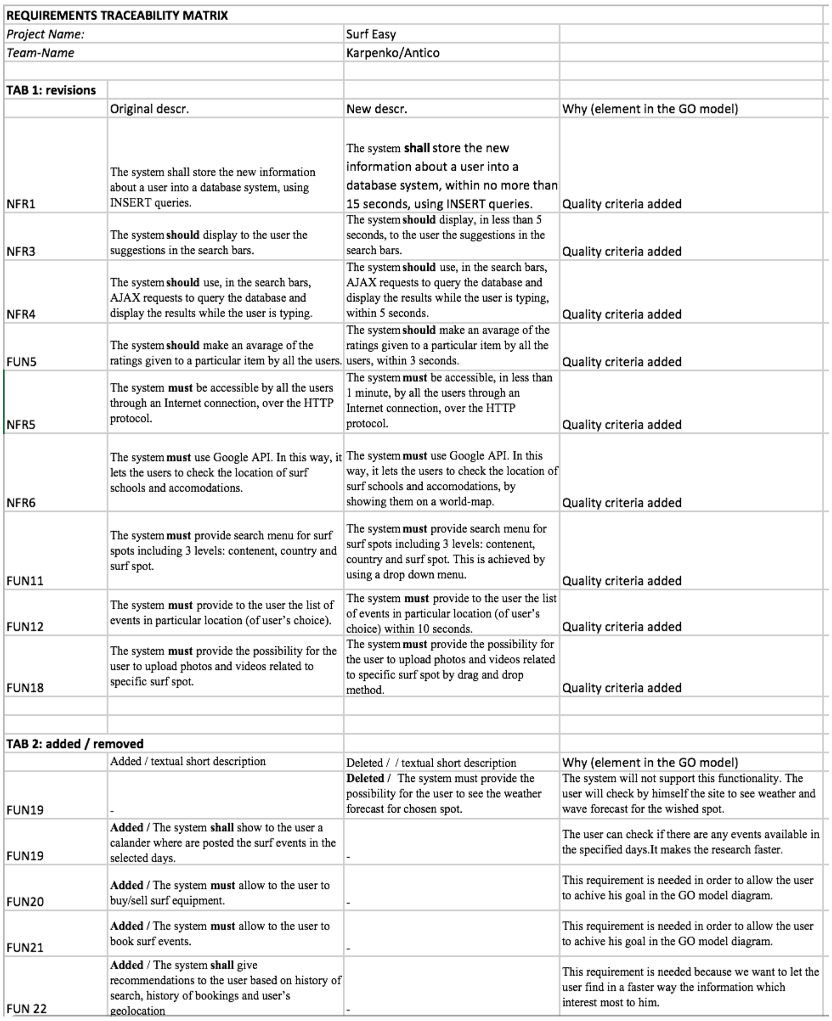
REQUIREMENTS TRACEABILITY MATRIX		**Legend**	
Project Name:	Surf Easy	Column	REQ -> To
Team-Name	Karpenko/Antico	Row	REQ: From >

	FUN1	FUN2	FUN3	NFR1	NFR2	NFR3	NFR4	FUN4	FUN5	NFR5	FUN6	FUN7	FUN8	FUN9	NFR6	FUN10	FUN11	FUN12	FUN13	FUN14	FUN15	FUN16	FUN17	FUN18	FUN19	FUN20	FUN21	FUN22
FUN1		🟧	🟧	🟧																								
FUN2			🟧	🟧																								
FUN3	🟧	🟧		🟧																								
NFR1	🟧	🟧	🟧																									
NFR2												🟧	🟧															
NFR3							🟧							🟧														
NFR4																												
FUN4																												
FUN5								🟧																				
NFR5	🟧	🟧		🟧																								
FUN6								🟧																				
FUN7	🟧	🟧																										
FUN8	🟧	🟧																										
FUN9																												
NFR6												🟧	🟧															
FUN10	🟧	🟧	🟧																									
FUN11																												
FUN12																									🟧			
FUN13	🟧	🟧	🟧																									
FUN14																		🟧										
FUN15																		🟧		🟧								
FUN16												🟧		🟧														
FUN17																												
FUN18																												
FUN19																		🟧		🟧								
FUN20																												
FUN21																				🟧								
FUN22												🟧	🟧	🟧														

Figure 10: Traceability matrix: requirements dependences

3.8.3 Changed requirements matrix

Figure 11 shows which requirements have been changed, added and deleted. An exaplanation is described as well.

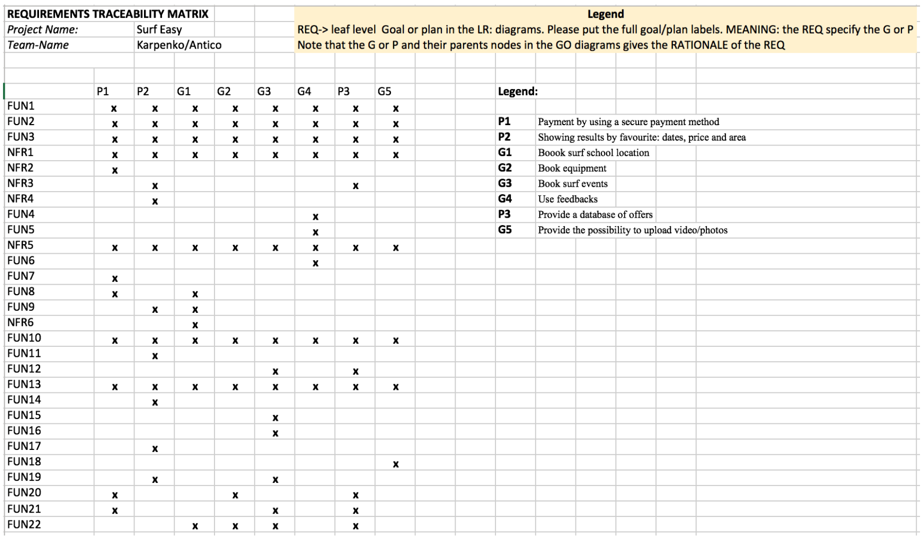


REQUIREMENTS TRACEABILITY MATRIX			
Project Name:		Surf Easy	
Team-Name		Karpenko/Antico	
TAB 1: revisions			
	Original descr.	New descr.	Why (element in the GO model)
NFR1	The system shall store the new information about a user into a database system, using INSERT queries.	The system **shall** store the new information about a user into a database system, within no more than 15 seconds, using INSERT queries.	Quality criteria added
NFR3	The system **should** display to the user the suggestions in the search bars.	The system **should** display, in less than 5 seconds, to the user the suggestions in the search bars.	Quality criteria added
NFR4	The system **should** use, in the search bars, AJAX requests to query the database and display the results while the user is typing.	The system **should** use, in the search bars, AJAX requests to query the database and display the results while the user is typing, within 5 seconds.	Quality criteria added
FUN5	The system **should** make an avarage of the ratings given to a particular item by all the users.	The system **should** make an avarage of the ratings given to a particular item by all the users, within 3 seconds.	Quality criteria added
NFR5	The system **must** be accessible by all the users through an Internet connection, over the HTTP protocol.	The system **must** be accessible, in less than 1 minute, by all the users through an Internet connection, over the HTTP protocol.	Quality criteria added
NFR6	The system **must** use Google API. In this way, it lets the users to check the location of surf schools and accomodations.	The system **must** use Google API. In this way, it lets the users to check the location of surf schools and accomodations, by showing them on a world-map.	Quality criteria added
FUN11	The system **must** provide search menu for surf spots including 3 levels: contenent, country and surf spot.	The system **must** provide search menu for surf spots including 3 levels: contenent, country and surf spot. This is achieved by using a drop down menu.	Quality criteria added
FUN12	The system **must** provide to the user the list of events in particular location (of user's choice).	The system **must** provide to the user the list of events in particular location (of user's choice) within 10 seconds.	Quality criteria added
FUN18	The system **must** provide the possibility for the user to upload photos and videos related to specific surf spot.	The system **must** provide the possibility for the user to upload photos and videos related to specific surf spot by drag and drop method.	Quality criteria added
TAB 2: added / removed			
	Added / textual short description	Deleted / / textual short description	Why (element in the GO model)
FUN19	-	**Deleted /** The system must provide the possibility for the user to see the weather forecast for chosen spot.	The system will not support this functionality. The user will check by himself the site to see weather and wave forecast for the wished spot.
FUN19	**Added** / The system **shall** show to the user a calander where are posted the surf events in the selected days.	-	The user can check if there are any events available in the specified days.It makes the research faster.
FUN20	**Added** / The system **must** allow to the user to buy/sell surf equipment.	-	This requirement is needed in order to allow the user to achive his goal in the GO model diagram.
FUN21	**Added** / The system **must** allow to the user to book surf events.	-	This requirement is needed in order to allow the user to achive his goal in the GO model diagram.
FUN 22	**Added** / The system **shall** give recommendations to the user based on history of search, history of bookings and user's geolocation	-	This requirement is needed because we want to let the user find in a faster way the information which interest most to him.

Figure 11: Traceability matrix: modifications of requirements

3.8.4 Dependencies between goals and requirements

Figure 12 shows the dependencies between the requirements identified in the platform and the main goals/plans in the GO model (late requirements). The matrix underlines with an "x" when the goal element provide a rationale for the requirement.



REQUIREMENTS TRACEABILITY MATRIX

Project Name: Surf Easy

Team-Name Karpenko/Antico

Legend

REQ-> leaf level Goal or plan in the LR: diagrams. Please put the full goal/plan labels. MEANING: the REQ specify the G or P

Note that the G or P and their parents nodes in the GO diagrams gives the RATIONALE of the REQ

	P1	P2	G1	G2	G3	G4	P3	G5
FUN1	x	x	x	x	x	x	x	x
FUN2	x	x	x	x	x	x	x	x
FUN3	x	x	x	x	x	x	x	x
NFR1	x	x	x	x	x	x	x	x
NFR2	x							
NFR3		x					x	
NFR4		x						
FUN4						x		
FUN5						x		
NFR5	x	x	x	x	x	x	x	x
FUN6						x		
FUN7	x							
FUN8	x		x					
FUN9		x	x					
NFR6			x					
FUN10	x	x	x	x	x	x	x	x
FUN11		x						
FUN12					x		x	
FUN13	x	x	x	x	x	x	x	x
FUN14		x						
FUN15					x			
FUN16					x			
FUN17		x						
FUN18								x
FUN19		x			x			
FUN20	x			x			x	
FUN21	x				x		x	
FUN22			x	x	x		x	

Legend:	
P1	Payment by using a secure payment method
P2	Showing results by favourite: dates, price and area
G1	Boook surf school location
G2	Book equipment
G3	Book surf events
G4	Use feedbacks
P3	Provide a database of offers
G5	Provide the possibility to upload video/photos

Figure 12: Traceability matrix: dependencies between goals and requirements

4 Part III

In this section of the document, it will be presented the system risks identified in the project and their model with RiskML. Then, it will be presented the requirements prioritization of a sub-set of 10 chosen requirements. It will follow a discussion about how the privacy law impact on relevant goals identified in the platform.
The section will end with the identification of security requirements in the project and how the requirements have been verified and validated.

4.1 Risk analysis

This section is about risks analysis and how the risks identified in this context impact the overall project. The main risks (and the relative indicators and possible treatments) are discussed and shown below.

1. **Customers stop booking services from the platform** (for example: booking surf schools, finding appropriate surf location, buy equipment, join events). This is a *security risk*. The RiskML model of this risk is shown in the schema 13. The results of this action will impact in a negative way all the actors involved in the system. The main situations (and related indicators) which generate the likelihood and significance of events are the following:

   a. Lack of feedbacks for a specific service or seller (this makes possible the achievement of an event).
   Indicators:

      i. Counting the number of feedbacks left by users for a specific service

   b. Unsafe payment methods (this makes critical an event).
   Indicators:

      i. Numbers of complains received by users
      ii. Statistics about attacks on web-sites

   c. Good feedbacks for a particular seller (this reduce the impact an event has on the system).
   Indicators:

      i. Percentage of good feedbacks left by users

   Treatments:

   - Send an e-mail notification to the user (to remind him to leave a feedback after he has bought a service)
   - Use secure protocols for the payment (for example HTTPS)

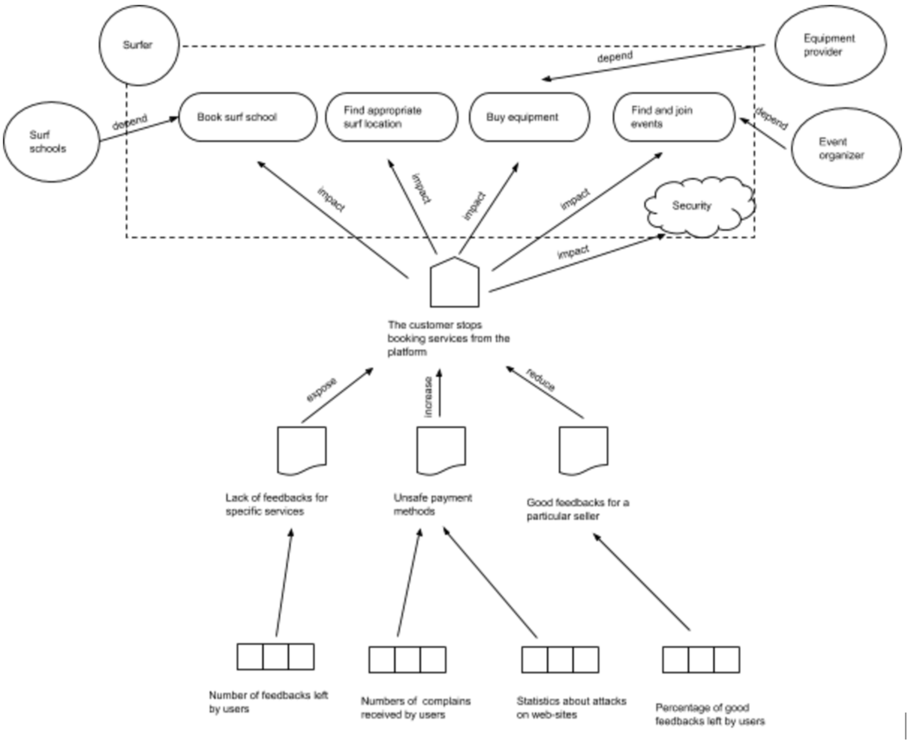

Surfer
Equipment provider
depend
Surf schools
depend
Book surf school
Find appropriate surf location
Buy equipment
Find and join events
depend
Event organizer
impact
impact
impact
impact
Security
impact
The customer stops booking services from the platform
expose
increase
reduce
Lack of feedbacks for specific services
Unsafe payment methods
Good feedbacks for a particular seller
Number of feedbacks left by users
Numbers of complains received by users
Statistics about attacks on web-sites
Percentage of good feedbacks left by users

Figure 13: Risk 1 - Customers stop booking services from the platform

2. **Problems with pages loading** (for example: slow loading time, grapichal elements not visible in all the browsers, etc.). This is a maintenance risk. The RiskML model of this risk is shown in Figure 14. The results of this action will impact in a negative way the system's user. The main situations (and related indicators) which generate the likelihood and significance of events are the following:

   a. Slow Internet connection (this makes possible the achievement of an event).
   Indicators:

      i. Check the number of users who are using the platform at the same time

   b. High quality images and lots of graphical elements (this makes critical an event).
   Indicators:

      i. Check the size of images uploaded by users in the platform
      ii. Gather information about usability and minimalist design

   c. Newest browsers do not support (or behave in a different way) some Javascript functions (this makes possible the achievement of an event).
   Indicators:

    i. Make researches to check how to implement same functionalities in different browsers
    ii. Check reports sent by users

Treatments:

- Restrict to the user and service providers the possibility to upload picture up to a certain size
- Suggest to the user which browser is better to be used
- Avoid network congestion

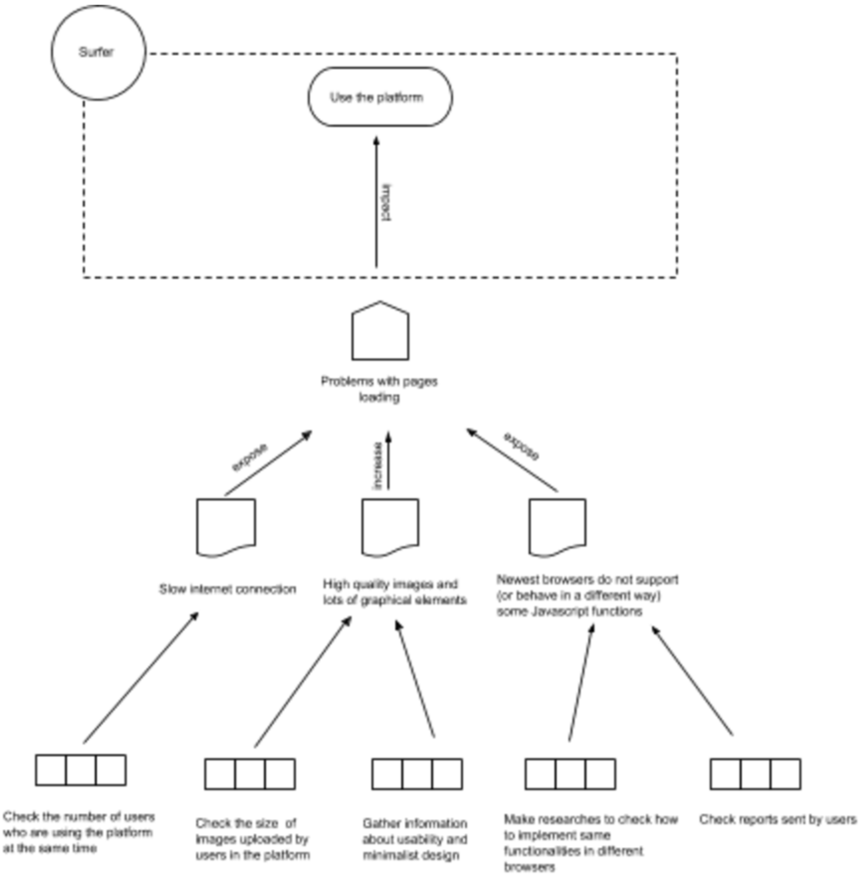

Surfer
Use the platform
impact
Problems with pages loading
expose
increase
expose
Slow internet connection
High quality images and lots of graphical elements
Newest browsers do not support (or behave in a different way) some Javascript functions
Check the number of users who are using the platform at the same time
Check the size of images uploaded by users in the platform
Gather information about usability and minimalist design
Make researches to check how to implement same functionalities in different browsers
Check reports sent by users

Figure 14: Risk 2 - Problems with pages loading

3. **The server crashes**. If this happens, the whole community will stop working and nobody will be able to do any kind of operations within the platform. This is a *maintenance risk*. The RiskML model of this risk is shown in Figure 15. The results of this action will impact in a negative way the whole community that uses the system. The main situations (and related indicators) which generate the likelihood and significance of events are the following:

    a. Part of the hardware does not work anymore (this makes critical an event).
    Indicators:

        i. Check the hardware's status periodically
        ii. Check the server's logs to determine the hardware's errors
        iii. Statistics and documentation about the most stable servers which can be used

    b. Software errors (this makes critical an event).
    Indicators:

        i. Statistics and documentation about the most stable servers which can be used
        ii. Make researches to check how to implement same functionalities in different browsers

    Treatments:

    - Use more performant servers
    - Use services which prevent software problems
    - Find employees responsible for system's maintenance

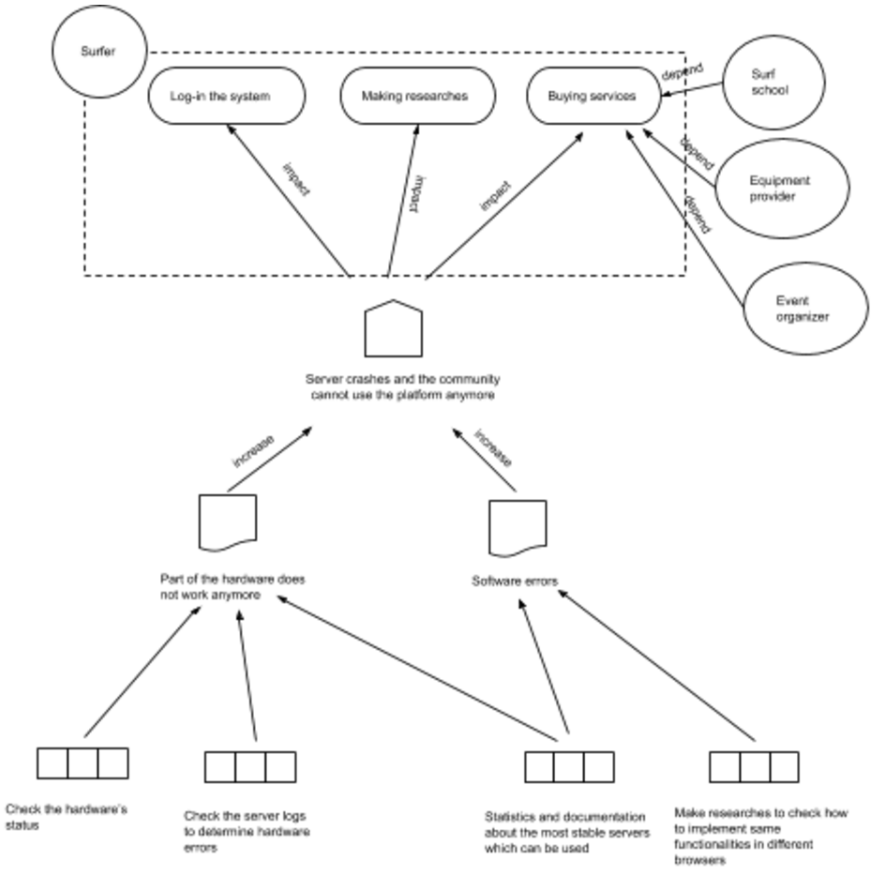

Surfer
Log-in the system
Making researches
Buying services
depend
Surf school
depend
Equipment provider
depend
Event organizer
impact
impact
impact
Server crashes and the community cannot use the platform anymore
increase
increase
Part of the hardware does not work anymore
Software errors
Check the hardware's status
Check the server logs to determine hardware errors
Statistics and documentation about the most stable servers which can be used
Make researches to check how to implement same functionalities in different browsers

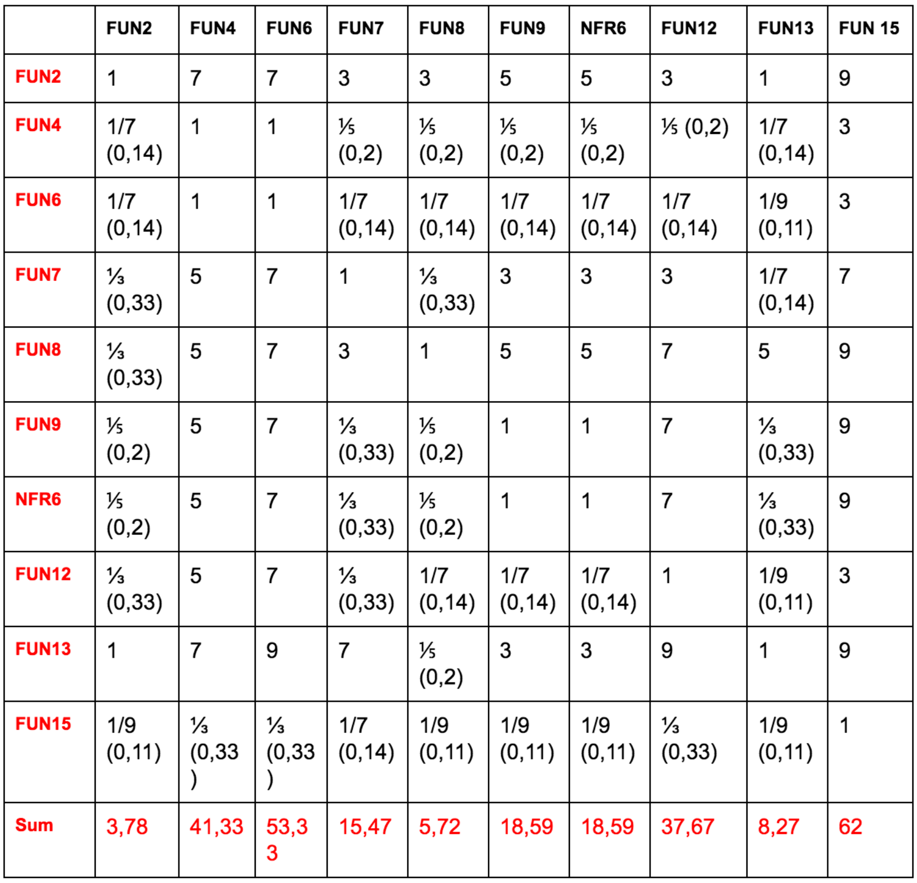
Figure 15: Risk 3 - The server crashes

From the risks analysis above, there are some new requirements which can be introduced in the system to make it more stable, secure and performant. The new requirements are the following:

1. The system **must** send an e-mail to the user who booked a specific service in order to remind him to leave a feedback. It would be helpful for the community
2. The system **must** prevent its users to upload images over a certain size
3. The system **should** show a message to the users to let him know which browser is not supported by the platform

4.2 Requirements prioritization

In this subsection we will go into the requirements prioritization. For doing so, we decided to take 10 random requirements (see Table 2.5.5) and analyse their prioritisation in terms of importance to the user and complexity of implementation for the developer. As the method we used the **AHP based pairwise comparison**. In order to implement the AHP method for both user and developer we fill in 2 tables from the different point of view.

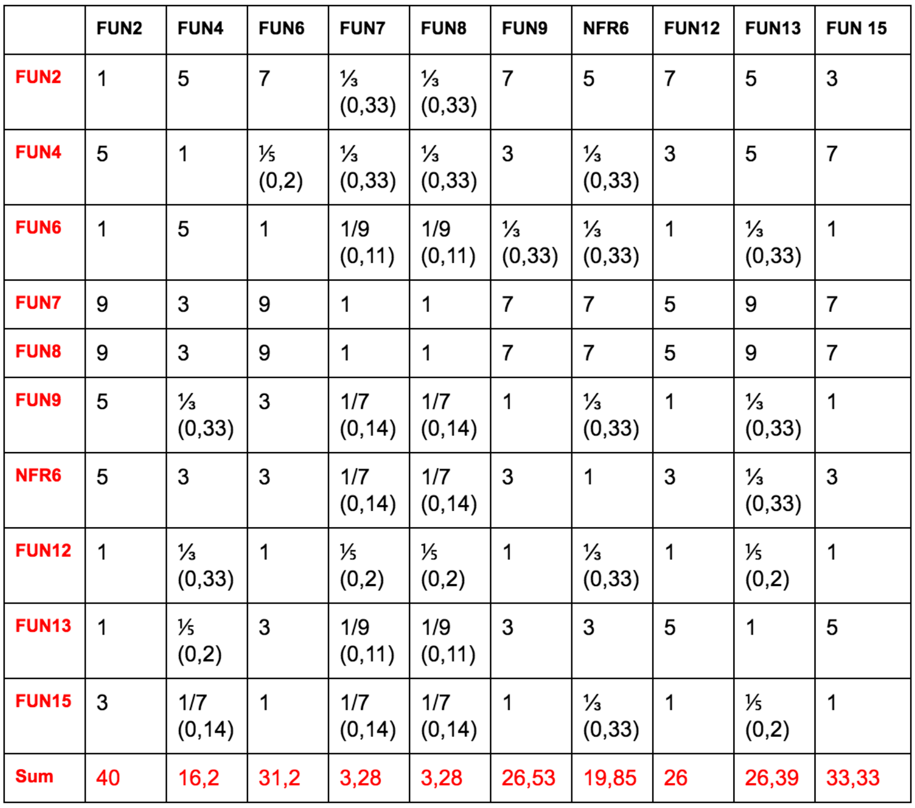


	FUN2	FUN4	FUN6	FUN7	FUN8	FUN9	NFR6	FUN12	FUN13	FUN 15
FUN2	1	7	7	3	3	5	5	3	1	9
FUN4	1/7 (0,14)	1	1	⅕ (0,2)	⅕ (0,2)	⅕ (0,2)	⅕ (0,2)	⅕ (0,2)	1/7 (0,14)	3
FUN6	1/7 (0,14)	1	1	1/7 (0,14)	1/7 (0,14)	1/7 (0,14)	1/7 (0,14)	1/7 (0,14)	1/9 (0,11)	3
FUN7	⅓ (0,33)	5	7	1	⅓ (0,33)	3	3	3	1/7 (0,14)	7
FUN8	⅓ (0,33)	5	7	3	1	5	5	7	5	9
FUN9	⅕ (0,2)	5	7	⅓ (0,33)	⅕ (0,2)	1	1	7	⅓ (0,33)	9
NFR6	⅕ (0,2)	5	7	⅓ (0,33)	⅕ (0,2)	1	1	7	⅓ (0,33)	9
FUN12	⅓ (0,33)	5	7	⅓ (0,33)	1/7 (0,14)	1/7 (0,14)	1/7 (0,14)	1	1/9 (0,11)	3
FUN13	1	7	9	7	⅕ (0,2)	3	3	9	1	9
FUN15	1/9 (0,11)	⅓ (0,33)	⅓ (0,33)	1/7 (0,14)	1/9 (0,11)	1/9 (0,11)	1/9 (0,11)	⅓ (0,33)	1/9 (0,11)	1
Sum	3,78	41,33	53,33	15,47	5,72	18,59	18,59	37,67	8,27	62

Figure 16: Requirements prioritization: importance for the user

After making the normalization of 16, the result of the AHP method showed that several requirements should be given more priority than the others: FUN8,FUN13,FUN2 because we ranked them as more important for the user or the platform.

After this first analysis, we started to focus our attention in the table 17; identifying the effort required for the developer to develop those requirements.



	FUN2	FUN4	FUN6	FUN7	FUN8	FUN9	NFR6	FUN12	FUN13	FUN 15
FUN2	1	5	7	⅓ (0,33)	⅓ (0,33)	7	5	7	5	3
FUN4	5	1	⅕ (0,2)	⅓ (0,33)	⅓ (0,33)	3	⅓ (0,33)	3	5	7
FUN6	1	5	1	1/9 (0,11)	1/9 (0,11)	⅓ (0,33)	⅓ (0,33)	1	⅓ (0,33)	1
FUN7	9	3	9	1	1	7	7	5	9	7
FUN8	9	3	9	1	1	7	7	5	9	7
FUN9	5	⅓ (0,33)	3	1/7 (0,14)	1/7 (0,14)	1	⅓ (0,33)	1	⅓ (0,33)	1
NFR6	5	3	3	1/7 (0,14)	1/7 (0,14)	3	1	3	⅓ (0,33)	3
FUN12	1	⅓ (0,33)	1	⅕ (0,2)	⅕ (0,2)	1	⅓ (0,33)	1	⅕ (0,2)	1
FUN13	1	⅕ (0,2)	3	1/9 (0,11)	1/9 (0,11)	3	3	5	1	5
FUN15	3	1/7 (0,14)	1	1/7 (0,14)	1/7 (0,14)	1	⅓ (0,33)	1	⅕ (0,2)	1
Sum	40	16,2	31,2	3,28	3,28	26,53	19,85	26	26,39	33,33

Figure 17: Requirements prioritization: complexity of implementation for the developer

Once we have normalized both table 16 and 17, we created the diagram in 18. We built a plot diagram having as "x-axis" the criterion "value for the user" (x-criterion), and as "y-axis" the development complexity (y-criterion). We plotted the requirements as (x,y) points depending on their value with respect to the x-criterion and y-criterion.

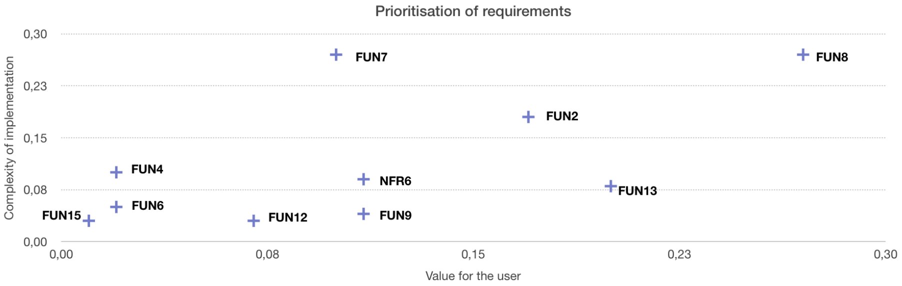


Figure 18: Prioritization of requirements

According to the diagram 18 we see that one requirement FUN8 has the most priority according to the importance for the user and complexity of development from our point of view. Beside FUN8, there are also FUN7 and FUN2 which require more effort to be implemented by developers (compared to the other requirements). All the other requirements are not so complex to develop. Overall speaking the requirements which are more important to the user are also those ones which require more implementation efforts.

4.3 Privacy law

In this subsection we discuss about how the privacy law impact on the relevant goals in our LR goal model. We then clarified how we ensure to be compliant.

More specifically, privacy law will impact the requirements for the using of Surf Easy online platform in 3 occasions: 1) registration of a user in the system (creating a new account), 2) using private user's data in order to recommend him/her other services based on user's previous bookings/orders or geolocation, 3) online payment (sending the secure private and banking information to the 3rd party payment system for making online payments when booking services on the platform).

4.3.1 Registration of the user account

Surf Easy is an online platform where the user creates an account which will work for his/her authentication and identification during the work in the platform. So in order to use the platform and create an account the user should give some private information, such as name, surname, age, address, telephone and cellular number, e-mail address, his surfing level and some other data that may be used for different purposes.

Registration data shall be stored in the user database and will be used for authentication purposes only. Surf Easy will notify the user whose data will not be disclosed and will be processed only for the purpose of reaching user's goals when using the system. When registering, the user will have to share his personal data and also will have to read and agree with terms and conditions of using the system, where the system owners will list all terms and conditions according to the current laws as well with notification of safety and non-disclosure of the personal data. User will have to check the "I agree" button in order to proceed with the registration. Also the platform will offer the user to subscribe to the system's email notifications. According to the E-Communications privacy directive 2002/58/EC as well as to the data retention directive (2006/24/EC) one of the main principles is on electronic marketing which requires organisations to obtain prior consent before sending electronic marketing messages to consumers (see Section 130). So each user will be asked to confirm his email subscription to the messages of Surf Easy platform. WIthout user's agreement Surf Easy will not send email notifications.

4.3.2 Recommendations of services to the user for booking

Online platform will notify and get consent from the user that his personal data, geolocation and history of previous choices stored in the system's database will be used for recommendation service for future user's bookings. According to Section 37 of Legislative Decree no. 196/2003 of Data Protection Code, "a data controller shall notify the processing of personal data it intends to perform exclusively if the said processing concerns any of the following:

a. genetic data, biometric data, or other data disclosing geographic location of individuals or objects by means of an electronic communications network;

d. data processed with the help of electronic means aimed at profiling the data subject and/or his/her personality, analysing consumption patterns and/or choices, or monitoring use of electronic communications services except for such processing operations as are technically indispensable to deliver said services to users".

The situation (a) may occur when geolocation will be used in order to recommend the nearest surf spots/schools/events and other services to the user to help him make the most suitable order. The situation (d) may occur when deploying the recommendation service for the surfing schools,surf spots, events for users based on their previous bookings and choices. All the data that user inserts in the platform will be used inside the platform in order to make the services more efficient and better for the user to meet his goals. The user will be previously notified that his data will be processed in the earlier mentioned occasions.

4.3.3 Sending user data when making online payment

Our system will use a third party payment service provider that will take care of the compliance and security regulations for online payments. Our platform will redirect to the partner payment service provider web-site and only will link the existing user account to the payment procedure and future transaction. Payment service provider will take care of compliance to the requirements of online payment security regulations (such as Payment Card Industry Data Security Standard, European Commission Payment Service Directive). The data about payment, credit card, card holder and bill address will be filled in by the user himself independently from our system on the redirected web-site of payment service provider or payment initiation service provider. So the user's private data will not be transferred by our system to the payment service provider because the user will have to type all the private data in the transaction page himself.

4.3.4 New requirements included

After checking how the law can impact in our system in a negative way, we have decided to introduce three new requirements. They help the platform to respect the regulations exposed previously.
The requirements which can be added are the following:

- **NF-RR1.** The system must notify the user about the usage of his personal data inside the system to reach user's goal
- **NF-RR2.** The system must allow the user register only when he agrees on terms and conditions of using the system according to the privacy law
- **NF-RR3.** The system must ask user's consent for sending e-mail notifications and online marketing e-mails

4.4 Security requirements

In this section, we describe the security requirements identified in our system and the approaches used to analyze them. We decided to use **abuse** and **misuse** case diagrams.

4.4.1 Security requirements analysis

The *Surf Easy* bases its activities and tasks on Internet services. This means that there are lots of security issues we have to take care of behind the platform. For this reason, we need to introduce some security requirements in order to handle some particular situations which can happen and which limit the security of the user.

We will not consider the task of purchasing items in our system since the payment service is not handled by our platform. This is done by redirecting the user to the chosen service provider's page.

Beside this issue, if we analyze the system in detail, we can say that there are some security requirements we have to introduce.

First of all we should manage sensible data introduced by the user. His personal information has to be safe while surfing on the system and, also, when he is offline. We know that some crooks can take advantage of the system's weaknesses and can hack the system to acquire personal users' information for bad purposes. For example, they can spam the user's e-mail by sending viruses or monitoring their activities and preferences. In this latter case, they can use this information to send to the user fake e-mails (about products which can interest them) that redirect him to a fake payment page which asks for money. In order to avoid this unpleasant sistuations we can introduce the following security requirements:

- **NF-SR 1.** The system **must** include an encryption system which prevent to expose sensible data to hackers
- **NF-SR2.** The system **must** use POST methods in order to submit information from one page to the next one
- **NF-SR3.** The system **must** use strong passwords in order to prevent hacking actions. They have to include special characters, numbers and letters

The figure 19 shows this scenario by using a **misuse diagram**.

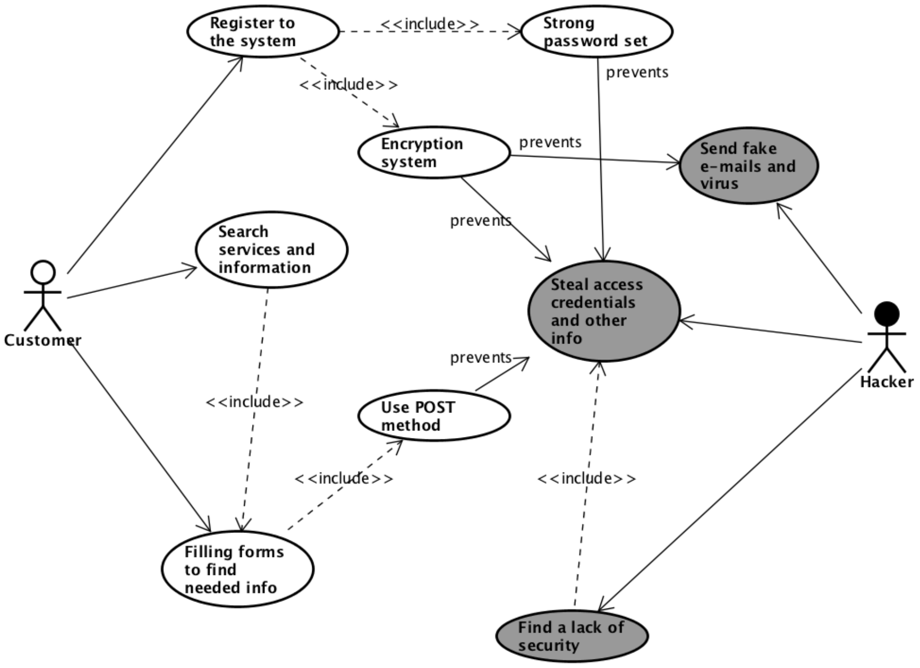


Figure 19: Misuse use case diagram

Beside the sensible data security requirements, in the system there can be found some other security requirements. For example, a user can log in the system by using someone else's username and password. He can try until he fails to get the access. Also, the service providers must not have access to the users' information, otherwise they can fake the feedback system. The platform should avoid these situations and this is possible by introducing the following requirements:

- **F-SR4.** The system **must** send a notification to the owner of the user account if he fails to log-in after five tries
- **F-SR5.** The system **can** give to the user the possibility to receive a notification each time an access to the system (with his credentials) is detected
- **F-SR6.** The system **must not** give to the service providers the possibility to have access to the database where all the users' information are memorized. Otherwise, they can give feedbacks to themselves
- **NF-SR7.** The system **must** check, at least the first 3 times the customer uses the platform, if he is a real person or a machine used for spamming and make confusion in the system. This is done by using bot detection techniques.

The Figure 20 shows this scenario by using an abuse diagram.

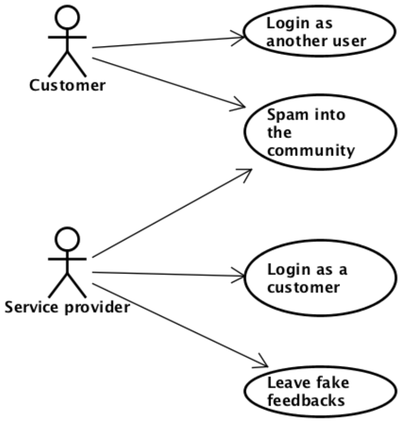


Figure 20: Abuse use case diagram

We have decided to analyze the above security requirements by using both **misuse** and **abuse** diagram.

In the first case, we used a **misuse diagram** in order to have a wider view of the security problems which can occur in our system. They include: *misuser* (in our case the *hacker*), *misuse* case (behaves which are not expected in the system) and *relations*. In this way we could introduce negative actors (which are not part of the system) and brainstorm how they can bring security problems into it, with the actions they might perfom.

In the second case, we decided to focus our attention on the security problems which can be brought by the system's actors. In this way, we identified actions (which are not part of normal situations or tasks) which can be performed by the users of the system that can bring a lack of security into it.

4.5 Requirements validation and verification

In order to know if all the requirements identified in our project make possible the realization of the right product, we need to use some validation and verification approaches. In this section are explained the techniques used and the ones we might think to use for future improvements.

4.5.1 Approaches for validatation and verification

There are several approaches for requirement validation and verification. In our case, we have focused our attention on the traciability matrix (simple checks approach). As shown in 10, we created a matrix where the rows represent our requirements and the columns the goals identified in the project. We can see that all the goal elements are satified by the requirements used. There are no goals which cannot be achived with the set of requirements we have. Beside this, we also analyzed carefully all the requirements to check their correctness, by respecting all the criteria that good requirements should have. Doing so, we avoid misunderstanding, ensuring that they all are well-written.

Other important approaches which can be applied in our scenario for validate and verify requirements can be:

- **prototyping**: we can think to build a vertical prototype which performs some of the activities the platform should be able to do. In this way, we can ask the users to try it out and leave feedbacks to us. For example, we can build a prototype which lets the user to achive the goal of searching services (surf schools, spots and equipment). It can show the list of results which fit the users' needs. We can ask the users (either through questionnaires or interviews) questions about usability and how difficut it has been for them achiving specific goals (for example, how much time they needed to make a research of a surf schools based on specific criteria)

- **typical review / inspection steps**: in order to identify some categorization problems in our requirements (such as: clarification, lack of information, conflicts and unrealistic requirements), we might think to ask the opinion of stakeholders with different backgrounds. In this case, we can use the feedbacks received in order to improve the quality of our requirements to make them more understandable and clear to others.