Web based booking system for NETS Test Center

Benjamin Bennike Aagren

Kongens Lyngby 2012
IMM-B.Eng-2012-<2>

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

Preface

This report describes the creation of a web application for the company NETS. The project was conducted from 2th of January – 10th of April 2012.

Foremost I would like to thank my supervisors Senior Scientist Finn Gustafsson and Test Engineer Kim Westergaard for their help and guidance during my project.

I would also like to thank manager Tamim Ghaussy for providing me with this project assignment, as well as the employees of NETS for their involvement in the project.

Finally I would like to thank my father Steen Bennike Mortensen his feedback and advice at the final writing process and my wife Christina Spuur Nødvig for her love and support during the whole process.

Benjamin Bennike Aagren

10th of April 2012

Summary

The Test Center in NETS handles all test systems, terminals, simulators and lines leading out of the building.

In this IT Diplom report there will be designed and developed a new web-based application for booking all components in NETS Test Center. This will make it easier for the users as well as the administrators to gain an overview of the use of these components, and simplify the booking process.

This new application will be based on data gathered from stakeholders in NETS, as well as an analysis of the users' needs and routines. Throughout the project several prototypes will be created, which will be tested together with the users, to find the most optimal solution. The application will be developed in Java, HTML, CSS, MySQL and using the Play Framework.

The end result of this project is meant to be used in NETS for the future bookings of components in the Test Center.

Resumé

Test Centeret i NETS står for alle test systemer, terminaler, simulatorer samt linjer ud af huset.

I denne IT Diplom rapport bliver der designet og udviklet en ny web-baseret applikation til booking af alle komponenter i NETS Test Center. Dette skal gøre det lettere for brugerne samt administratorerne at få et overblik over anvendelsen af disse komponenter, samt at forenkle processen for at booke komponenterne.

Denne nye applikation vil blive baseret på indsamlet data fra interessenter i NETS, samt en analyse af brugernes behov og rutiner. Der vil undervejs blive lavet adskillige prototyper, der vil blive testet i samarbejde med brugerne, for at finde den mest optimale løsning. Applikationen vil blive udviklet i Java, HTML, CSS, MySQL samt ved hjælp af Play Framework.

Det endelige resultat udvikles med formålet at kunne anvendes i NETS til fremtidige bookinger hos Test Centeret.

Abbreviations

CSS:	Cascading Style Sheets
DTU:	Danmarks Tekniske Universitet
EER:	Enhanced Entity-Relationship
HTML:	Hypertext Markup Language
MVC:	Model View Controller
MySQL:	My Structured Query Language
WUI:	Web User Interface

Lists of tables and figures

Contents

Chapter 1

Analysis

1.1 Introduction

In NETS, the Test Center department provides different kinds of test solutions for both internal IT project and for external clients, who use NETS services; credit cards and money transfers. These test solutions will be referred to as components.

The purpose of this IT Diplom project is to develop a booking system for NETS' Test Center. Currently, the method for booking a component in the Test Center is handled manually through e-mails and Excel sheets. This method is both time consuming and cumbersome to maintain for the two groups within NETS, who have to do this task. The groups are the Kopi Team who administers the bookings, which they then report to the Test Center Team, who administer the components in the Test Center. The wish for an improved way of handling the bookings was motivated by these two groups. Replacing their former booking process with a web application will increase the synergy between the groups, as the web application will automatically provide the other group with necessary information, instead of continuously exchanging information through email.

Furthermore, the process of implementing this new web application involves the users who book the components in the Test Center. Adding these users to the web application will make it possible to simplify the booking process, as well as automatically share relevant information among the users and the test groups.

The new web application will also make it easier to maintain booking and component history, since such a feature will be part of the functionality of the web application. The final result of this project will be a beta version of the Test Center booking system. This report contains the details of the beta Test Center booking system and its origin and the development hereof.

1.1.1 NETS history

In 2010 The Danish company PBS merged with the Norwegian company BBS, which resulted in the name NETS for the whole company. PBS has focused on IT payment solutions for all of Denmark since 1968 and in 1983 PBS introduced Dankort into Denmark. BBS has worked with payment solutions since 1972 for all of Norway. Both companies have during this time expanded their markets to the surrounding countries and now in 2012 NETS handles the majority of the daily payments, remittances, and other types of transactions in northern Europe.

1.2 Success Criteria

For the project to be a success, the following success criteria have to be met:

- The project has to meet the priority 1 requirements (specified in the 1.4 Requirement chapter)
- The project has to be completed together with this report, and handed in before the project deadline, which is midnight the 10th of April 2012
- It is a requirement that a demonstration of the new application must be provided for:
    - Finn Gustafsson (DTU supervisor)
    - Kim Westergaard (NETS supervisor)
    - Tamim Ghaussy (Manager of NETS testcenter)
    - The department who has to administrate the application in the future

  Before the project deadline
- The future administrators of the application and Tamim Ghaussy have to accept the application as being able to perform its primary function, in the form of a usable booking system.

1.3 Stakeholders

The project stakeholders are anyone that are actively involved in the development of this new web application, or whose interests may be affected as a result of the new application. In table 1 the most important stakeholders are shown with an exposure value, which defines how involved the stakeholder will be when working with the new application, and an importance value showing how important their requirement requests are. More detailed descriptions of the values are contained in the following section, which describes the interaction applicable for each stakeholder category.

Table 1 the most important stakeholders

Stakeholder	Interaction with the new application	Exposure	Importance
Manager / *Owner*	Gather data for a statistic overviews of the actions made in the application	**1**	**4**
Kopi Team / *Admin*	Administer bookings of components made by users	**4**	**3**
Test Center / *Admin*	Administer all components and update the status for some of them, on a daily basis.	**4**	**3**
Tester / *User*	Books the components they require for their tests	**2**	**2**
Developer	Could be asked for further maintenance or updates of the application	**1**	**1**

1.3.1 Manager / *Owner*

It is within the manager's power to terminate the project at any time, or simply decide if the web application shall not be implemented in the company. This means the manager has a high importance; even with a minimal exposure to the application.

1.3.2 Kopi Team & Test Center / *Admin*

These two groups' work assignments are the foundation which this project is mainly based on. Accordingly, they have a high importance as the main source for high priority requirements. Therefore, their input can impact the project positively and/or negatively. The new application will have a positive impact on both groups' work processes, as the process will be more efficient. Therefore the employees of the Kopi Team and the Test Center should be co-operative.

1.3.3 Tester / *User*

When the application is implemented, the users will be forced to use it from then on. It is in the users' interest to be involved in the requirement gathering phase, to influence the outcome of the application. Their importance and exposure values have been estimated to a medium level, since any issues they might have with the application will require them to take the time to organize, to express their concern as a group to the manager, who will then decide the course of action so forth. Therefore, the testers' requirements must be taken into consideration.

1.3.4 Developer

The Developer is responsible for the planning and completion of the project. Practical experience and the ability to carry out the plan is required, which if lacking could be a risk to the project. The developer does however have a professional stake in the project in the form of career advancement.

1.4 Requirements

When introducing a new application into a company, it is advantageous that the stakeholders, who will have to work with the application, see it as an improvement compared to the previous solution. This is why they have been directly and iteratively involved in the requirement gathering process. This has been done by asking the stakeholders to describe the previous solution and writing down all their desired requirements for the new booking system, categorised by priority and criticality in a 1 to 10 scale. Priority is defined by how much the stakeholder wishes this feature, and the criticality being how essential the function is to have, for the application to perform its primary function. All requirements have been analysed and prioritised by the developer, by considering the requirements cohesion with each other as well as the highest critical and priority values. During the development progress additional prototype presentations were held for the stakeholders to verify that the specified requirements complied with their expectations. These presentations resulted in confirming or redefining the already stated requirements. New requirements were also added, after showing the stakeholders how the functions worked, displayed in the prototypes. This is based on agile development (Highsmith and Cockburn 2001). The end result from the requirement gathering is shown below in table 2, 3 and 4, listed by priority in these three tables where the priority one table contains the most important requirements.

Table 2 Requirements priority one, critical must have functionality

Req. ID	Summery	Description
P1.1	Web Server	The web application has to be installed on a web server within the company. It is essential to ensure as good an up time as possible for this web server, and ensure there is the possibility for backups.
P1.2	Programming language	It is necessary to choose a programming language which is installed on the web server, so the users can start the application from their browsers, not needing to install anything additional.
P1.3	User role	The user role is needed for the persons who need to book components in the test center, so they can execute their tests
P1.4	Information required for all users	The persons who needs to book components in the test center require a user in the application with the following information: • Mail • Name • Date of creation
P1.5	Information required for all projects	To book a component, it is required that the user works for a project, and the following information is required for all projects: • Project number • Comment (Note field) • Date of creation
P1.6	Information required for all components	Information required for all components: • Name • Edb centre nr • Status (up/down/unknown) • Location (where in the world) • Comment (Note field) (For transaction types and card types which the component works with)
P1.7	Information required for all	Information required for all bookings:

	bookings	• Booking receipt • Project number • Check in date • Check out date • Relevant component info • Relevant user info • Comment (Note field)
P1.8	Viewing and sorting data for users	The following data has to be available for the users: • All bookings owned by their projects • All bookings owned by their user • Status for all components
P1.9	User role actions	The user role needs to be able to: • Register them self as a user • Register a project • Book components • Cancel any of their own future bookings
P1.10	Admin role actions	The admin role is required to: • Update components status • Create new components • The possibility to delete any booking • Confirm and decline any bookings
P1.11	Admin views and data sorting	The admin role requires the following overviews available: • Status of all components • Which components require daily status updates • Calendar view of all booked components • Overview of all bookings
P1.12	History	It is necessary to store: • All bookings made • All component status changes This can be used to give an overview of the general status of the test center
P1.13	Report	The report has to be written in English to respect the fact that NETS is a company with many international employees
P1.14	Demonstrations before deadline	NETS have requested that a demonstration of the end product is provided for the manager and admins of the application, before the project deadline

Table 3 Requirements priority two, noncritical functionality

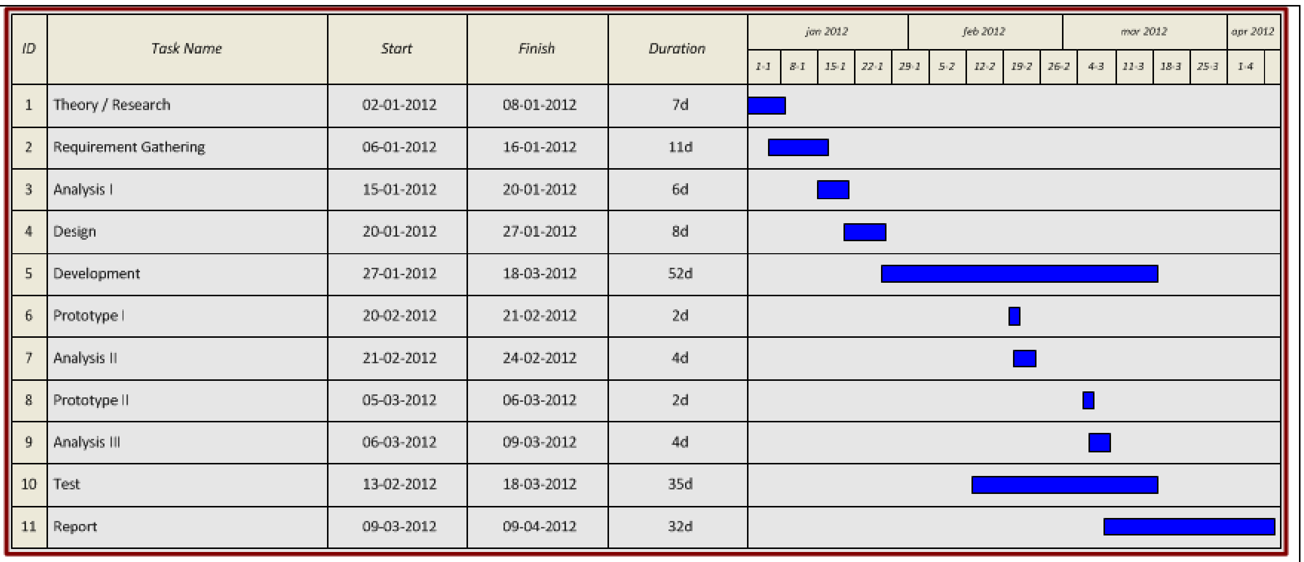
Req. ID	Summery	Description
P2.1	Display Interdependencies between the components	Some components require the same resource to work. Store these dependencies and display them in some way
P2.2	Automatic display of the interdependencies between the components	If a component using a resource is booked, it should be displayed on all components using this resource, that someone else have already booked the resource in this period
P2.3	User booking update	The users would like to have the possibility to update the booking information for their existing bookings
P2.4	Admin booking update	It would be nice for the admins to be able to update the booking information for existing bookings
P2.5	Update additional bookings at a time	To decrease the time spent on booking administration, it would be preferred to be able to update the booking date of multiple bookings at a time
P2.6	Date selected status history	In the future it would be interesting to see a overview of all the component updates that resulted in a bad status in a selected time period
P2.7	User Help	If there are any errors / problems / confusion it could be relevant to have referrals to show how to report these the correct way. Either by using the proper system within the company, or notifying which email to report the issue to.
P2.8	Add the project leader to a project	It would ease things if it was possible to see who the project leader was for the projects using the application
P2.9	Create additional bookings at a time	It could save time, if additional components could be selected to be booked at a time.

Table 4 Requirements priority three, nonessential extra functionality

Req. ID	Summery	Description
P3.1	Mail system	There are situations where it would save time for the administrators if the web application could send a mail.
P3.2	Export data	If the need to create statistics based on the data in the booking system arises, it could save time to have an export function.

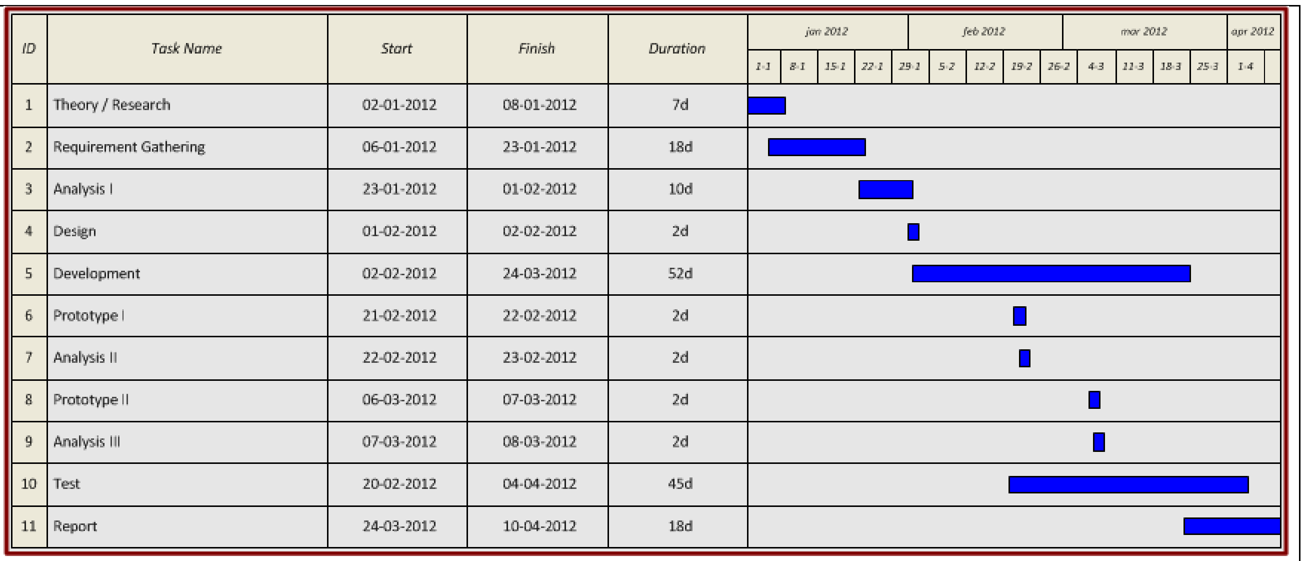
1.5 Estimation

The project plan has been roughly approximated in the start of the project. This was done to identify and place deadlines for all crucial tasks of the project. This is done to ensure that no tasks were forgotten or to minimise the risk that the work required for its completion were not underestimated. As the project progressed however it was necessary to revise the time plan, especially since the requirement gathering process was done gradually, making the projects size dynamic. Figure 1 is the first time plan created at the start of the project, where figure 2 is the final time plan of how the project actually progressed.



ID	Task Name	Start	Finish	Duration
1	Theory / Research	02-01-2012	08-01-2012	7d
2	Requirement Gathering	06-01-2012	16-01-2012	11d
3	Analysis I	15-01-2012	20-01-2012	6d
4	Design	20-01-2012	27-01-2012	8d
5	Development	27-01-2012	18-03-2012	52d
6	Prototype I	20-02-2012	21-02-2012	2d
7	Analysis II	21-02-2012	24-02-2012	4d
8	Prototype II	05-03-2012	06-03-2012	2d
9	Analysis III	06-03-2012	09-03-2012	4d
10	Test	13-02-2012	18-03-2012	35d
11	Report	09-03-2012	09-04-2012	32d

Figure 1 The first and not final time estimate for the project



ID	Task Name	Start	Finish	Duration
1	Theory / Research	02-01-2012	08-01-2012	7d
2	Requirement Gathering	06-01-2012	23-01-2012	18d
3	Analysis I	23-01-2012	01-02-2012	10d
4	Design	01-02-2012	02-02-2012	2d
5	Development	02-02-2012	24-03-2012	52d
6	Prototype I	21-02-2012	22-02-2012	2d
7	Analysis II	22-02-2012	23-02-2012	2d
8	Prototype II	06-03-2012	07-03-2012	2d
9	Analysis III	07-03-2012	08-03-2012	2d
10	Test	20-02-2012	04-04-2012	45d
11	Report	24-03-2012	10-04-2012	18d

Figure 2 The final time plan of how the project actually progressed

The final time plan roughly resembles the estimated time plan. The main difference between the plans was that the requirement gathering and testing ended up taking a lot longer than estimated. Some of the reasons being that these tasks where dependent on involving busy stakeholders, and no time could be taken from the development because of the need to fulfil the success criteria. This resulted in all the other tasks getting pushed forward in the time plan, cutting time from the last task which was the report.

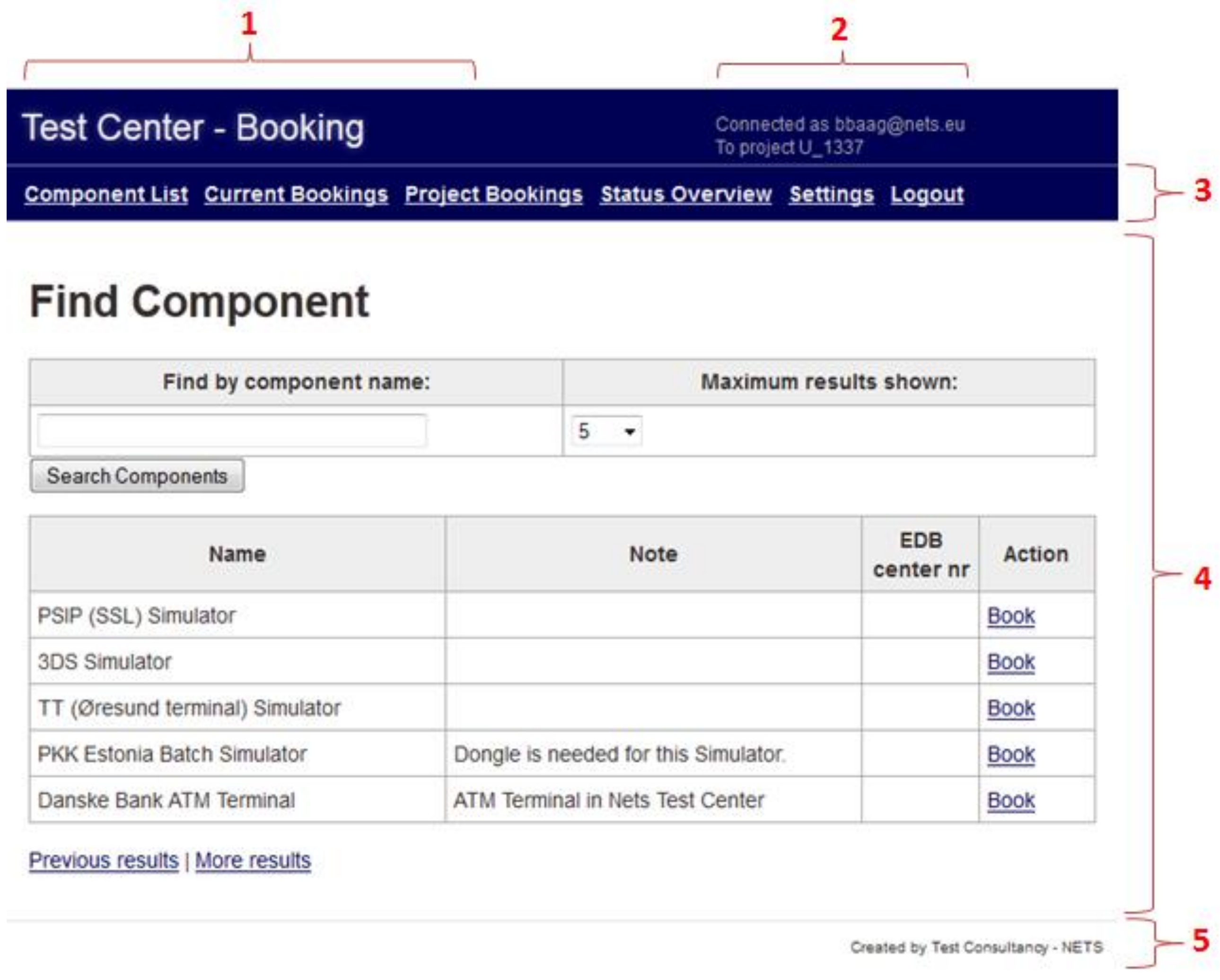
Chapter 2

Design

2.1 Design

The application design has been made in collaboration with the stakeholders, in the form of a mock-up drawing, which was created in the requirement gathering phase. Then later at the prototype presentations, the stakeholders could influence the design, which led to changes several times throughout the development phase. This was done to ensure that the end product would meet the stakeholders' expectations.

The final Web User Interface (WUI) was constructed with a login screen as start page, which was necessary to differentiate between the admins and users. The rest of the WUI consists mostly of screens containing informative tables providing the users and admins with their necessary views, as well as actions related to the views information. Figure 3 shows the final design for the bookable component list, which is accessible for the users of the application.


1
2
3
4
5
Test Center - Booking
Connected as bbaag@nets.eu
To project U_1337
Component List Current Bookings Project Bookings Status Overview Settings Logout
Find Component
Search Components
Previous results | More results
Created by Test Consultancy - NETS

Find by component name:	Maximum results shown:
	5

Name	Note	EDB center nr	Action
PSIP (SSL) Simulator			Book
3DS Simulator			Book
TT (Øresund terminal) Simulator			Book
PKK Estonia Batch Simulator	Dongle is needed for this Simulator.		Book
Danske Bank ATM Terminal	ATM Terminal in Nets Test Center		Book

Figure 3 the final design for the bookable component list, which is accessible for the users of the application.

The design shown in figure 3 is only one screen of many in the booking application. It does however display the outcome of several major design decisions. These decisions are the red number markings.

1. In the upper left corner the booking applications name is shown. It is shown on all screens throughout the whole application.
2. In the upper right corner the user information is displayed. If a user is logged in, the user mail and project number is shown. If an admin is logged in, only his admin mail will be shown. If no one is logged in, nothing is shown.
3. Just under the site header, the menu bar will be shown on all screens throughout the application, as long as a user or admin is logged in. The user and admins menu bars are different, since they have different tasks to fulfil.
4. This section contains the information and actions the users and admins require. The information is mostly contained in tables, and the actions usually encapsulated in a square. How section 4 is designed for the different screens variants, based on the function the screen must supply.
5. At the bottom right corner the department name, which the developer works in is displayed. This footer is shown on all screens throughout the whole application.

The rest of the booking applications screens can be found in appendix 6.4

2.2 Booking process

There are four types of users who will interact with the web application, as defined in the stakeholder chapter 1.3. This section will illustrate the booking process from start to end, by listing all the user actions the users will be using in the new booking application.

1. A project has finally reached the point in its development process where *testers* can test their new features on one of the company's test environments. Therefore the responsible *tester* books the components required for the tests through the booking application.
2. The *Kopi Team* notice the new booking and checks if anything should prevent this booking from being a success. When this is done, the booking can be accepted or denied.
3. On a daily basis a list of components is generated by the booking system. This list includes all components that are booked for the actual day, together with the components that are required always to work. It is the *Test Center Teams* job to ensure all components on this list are up and running all day. When a member of the *Test Center Team* have checked if a component is working, it should be updated in the booking application, which both the *tester* and *Kopi Team* will be able to see.
4. The *Manager* can at any time enter the booking system to see an overview of all booking and component status updates made.

2.3 Use case Diagram

The following two use case diagrams are meant to visualize all possible views and actions the user groups have in the application, as stated in the requirement P1.8, P.1.9, P.1.10 and P.1.11. There is only shown the primary functions of the application without any details.

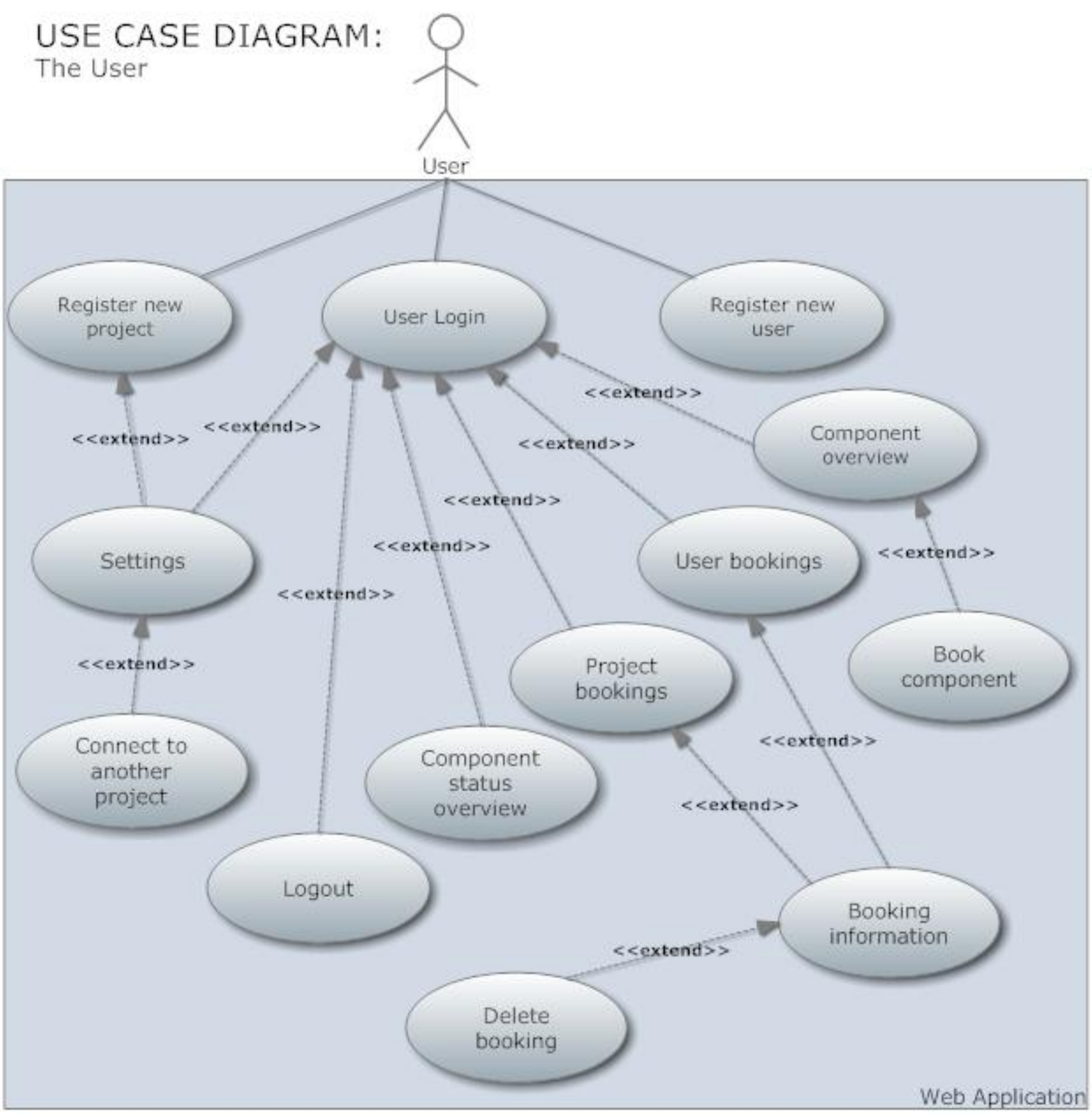


Figure 4 Use case diagram for the user, showing all their possible actions in the web application, as stated in the requirement P1.8 and P1.9

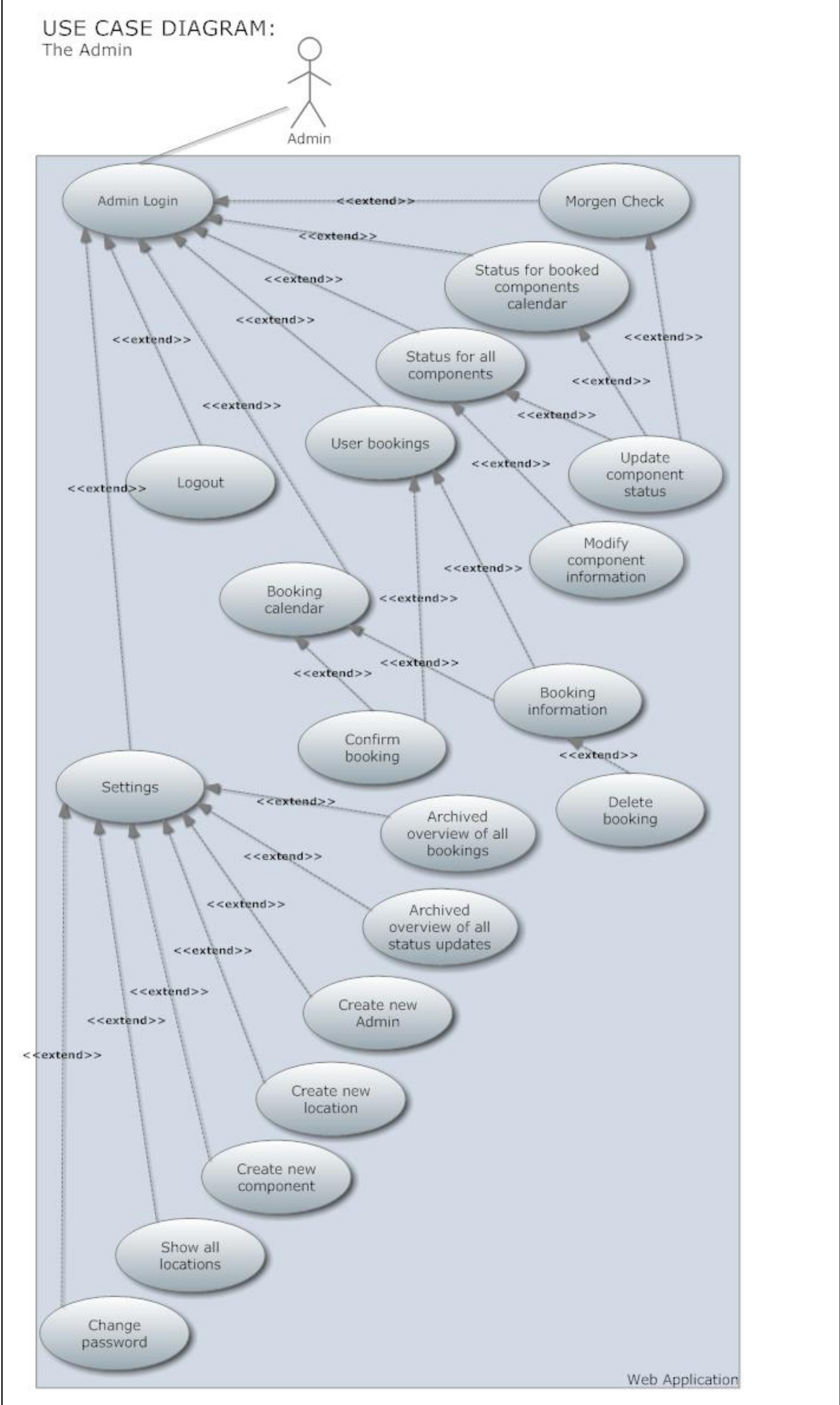


Figure 5 Use case diagram for the admin, showing all their possible actions in the web application, as stated in the requirement P1.10 and P.1.11

Chapter 3

Development

3.1 Theory

3.1.1 Choice of Development Framework

The available web server at NETS has Java and MySQL installed. This means that we have to code the web application with the use of these programming languages, for users to access the application directly through a browser, without having to install anything additionally. As stated in requirement P1.2.

Knowing that the application has to be developed in Java, a framework was searched for, focusing on the need for creating prototypes early in the development phase. The research showed that a framework called Play Framework seemed like a good candidate for not only making it possible to create prototypes at a early point in the development, but it also seemed qualified for creating a robust web application.

By choosing the Play Framework for the development of the web application, it follows that the following programming languages, Java, HTML, CSS and MySQL would be used.

3.1.2 Play Framework

Creating a web application with the Play Framework requires the model-view-controller (MVC) architectural pattern, which means these three layers are the cornerstones of the application. The model (MySQL) handles the usage of all raw data throughout the application. The view (CSS, HTML) represents the user interface, which the user will use for all interactions with the application. The Controller (Java) processes all events triggered in the application, which could be anything from a HTTP request to an action calling or changing some specific data in the database. Figure 6 shows how the MVC patterns are linked together for the play framework, and in this example an event is triggered that updates some data and returns a new view.

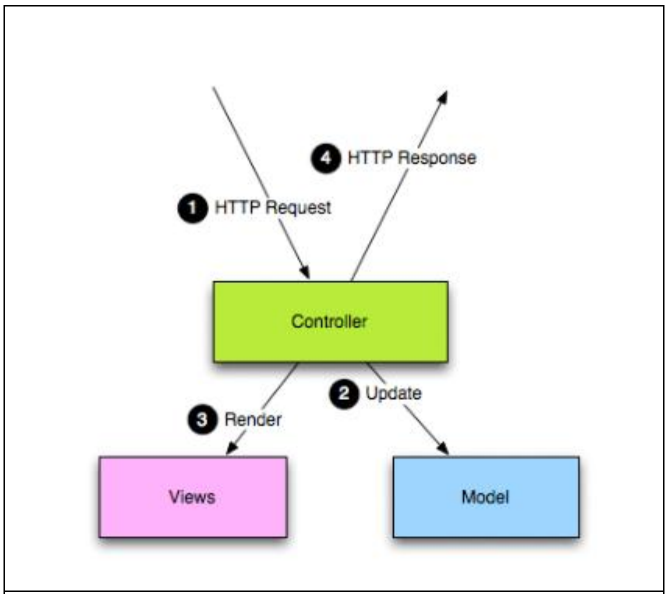


Figure 6 Play Framework MVC. From: http://www.playframework.org/documentation/1.0/main

3.2 Database (MySQL)

The Test Center booking application stores all data in a MySQL database. This data is the basis for the application, since it consists of all the necessary information required for the application to function, such as bookable components, bookings, users and so forth. Therefore it is essential to create a well structured database to give the application an optimal performance, which should be achieved by creating the database in the Third Normal Form. The data which is to be stored in the database was identified in collaboration with the stakeholders in the requirement gathering phase. This section of the report will describe the thoughts and efforts that have gone into designing and creating this database.

3.2.1 Database Design

When designing a database, several decisions have to be made. This section will reflect on some of these decisions.

All the database tables require at least one unique value per row in a selected column. This or these columns are known as the primary key of the table, which are used to uniquely identify specific rows of data. To relieve the users from constantly needing to remember all former unique values, for when they manually add new data into the database, it was decided to use auto increment. Auto increment means adding an extra column containing a number starting from one, which rises by one for each row of data added into the table. It was selected that the maximum value for the auto increment should be a bigint (2^{64}-1), which ensure a maximum unrealistic to reach. It was decided that to increase the database's performance it should be created in the third normal form, which is described in the next section; 3.2.2 Database Normalization.

3.2.2 Database Normalization

The whole purpose of normalization is to eliminate redundancy and anomalies in the database. By creating the database in the third normal form the most serious of anomalies and redundancy will be handled.

First normalization is achieved simply by each column in our database tables only containing a single value, and each row having at least one unique primary key column.

Second normalization is achieved by creating separate tables for each set of values that apply to multiple records, and relate the proper tables with foreign keys.

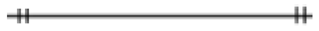
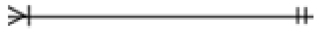
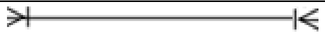
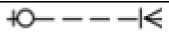
Third normalization is achieved by removing all columns not dependant on the key, and separating tables to avoid redundant data which can be altered to illogical contexts, like a component with two addresses.

3.2.3 Enhanced Entity-Relationship diagram (EER)

The database design has been illustrated with an Enhanced Entity-Relationship model (EER), which is a fusion of an entity-relationship diagram and a database model. This section will describe how to understand the EER diagram together with the actual diagram.

The graphical elements used in the EER diagram are described in table 5.

Table 5 EER diagram elements

A one to one relation	
A many to one relation	
A many to many relation	
A zero to many relation	
Primary key	🔑
Foreign key	◇
Attribute	◇
Unique attribute	◆

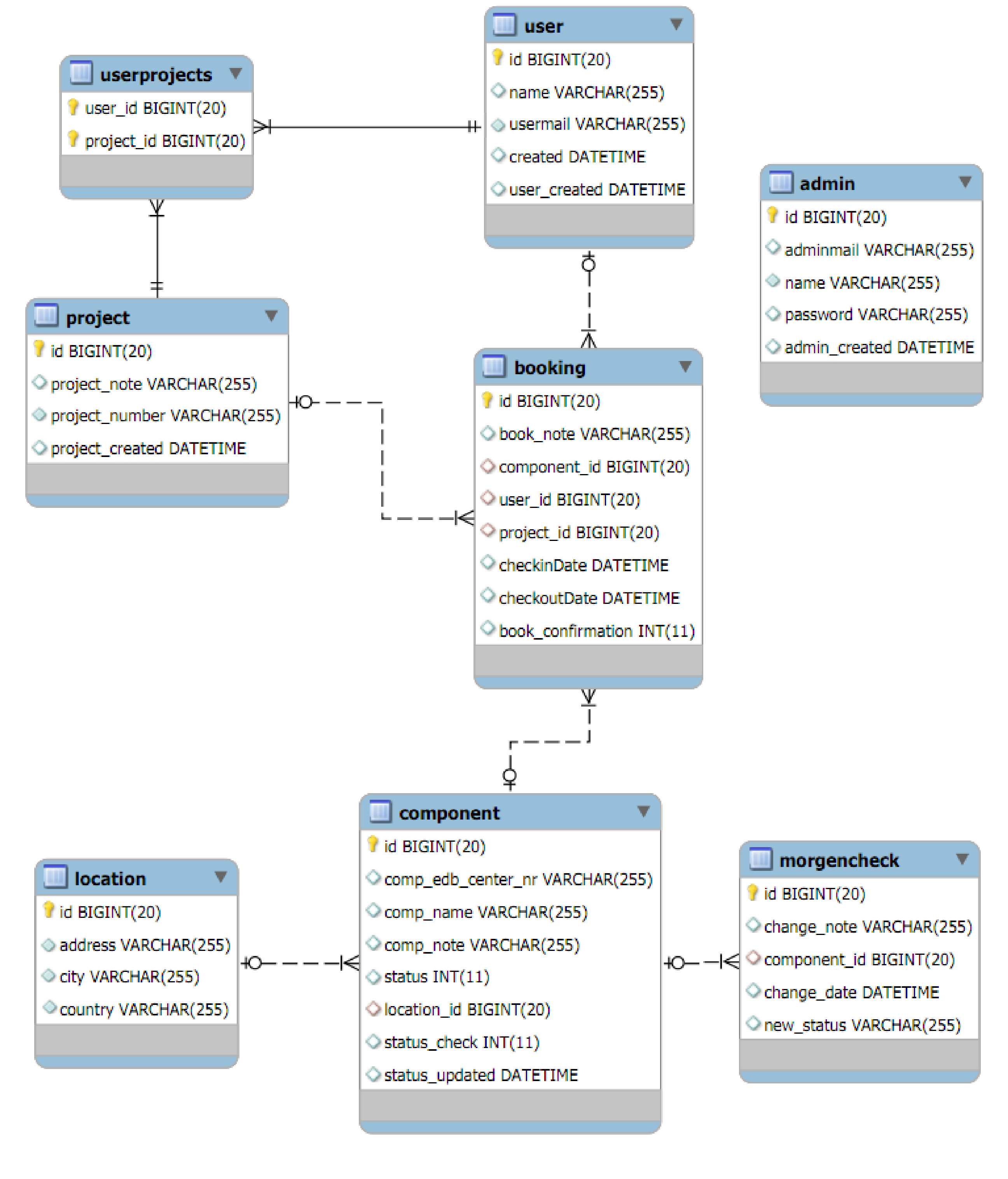


Figure 7 EER Diagram

3.2.4 Database Overview

This section will provide an overview of all the purposes of the database tables, meant to give a better understanding of figure 7. The overview is shown in table 6.

Table 6 Database table overview

Database table name	Purpose
Component	The Test Center at NETS has a multitude of components, which are all stored in this database table, with their individual notes and data.
Location	The database location table contains the world location of a Test Center, where no Test Center can be located in the same country.
morgencheck	The morgencheck is the name given by the Test Center team in NETS, for updating components status. This database table is used to log all status updates made for any component.
Booking	All bookings are stored in this database table.
User	All users data is stored in this database table
Project	All project data is stored in this database table
Userprojects	This database table is used to store the linkage between users and their respective projects.
Admin	All admin data is stored in this database table.

3.2.5 Database Reflections

There was successfully created a MySQL database corresponding to the desired specifications. Third normal form was achieved on all tables in our database, since all data in the tables are depended on the primary key, which means that there is basically no redundant data or anomalies. The only situation where the same data is being stored in two tables at once is whenever a component status update occurs, since the last update is stored in both the actual component in the component table, and in the morgencheck table, which is meant to store the history of all status updates. The thoughts behind this design, was to provide a tiny performance increase by avoiding the join between the component table and the morgencheck table every time a status overview is required, since the morgencheck table is going to grow rapidly each day.

3.3 Web Development

The purpose of this section is to describe the functions of the Test Center booking applications and how they were developed.

3.3.1 Usability

Since all interaction with the application is through the WUI, and a multitude of people will be using it when it is has been completed, it is very critical to achieve a high degree of user friendliness. This is because the applications primary function has to be a better solution than the previous, which will not be achieved if it is more complicated and takes longer to use. How the usability was achieved is described in chapter 2.1, 3.3.2 and 4.3

3.3.2 Heuristic Evaluation

- This section will evaluate if the applications final design is usable based on Jacob Nielsen's ten usability heuristic's. (Nielsen, J. (1994). 10 usability heuristics).
- **Visibility of system status**
  All screens throughout the application contain an informative title just under the menu bar, which shows what the actual page is all about.
- **Match between system and the real world**
  The language used in the application has been worked out with and by the stakeholders to ensure that it is understandable by the people who have to use the application. This is also the reason for the name 'Morgen Check' for the daily status update on the admin site, since that was the name requested. There have also been used describing image metaphors to illustrate the status for a component.
- **User control and freedom**
  There is never more than one screen layer to navigate to, from a top menu screen. From this additional layer screen there will always be a return button to the original screen, as well as the possibility to use the top menu to navigate back.
- **Consistency and standards**
  It has been decided to encapsulate the actions that require a confirmation in a smooth edges square to create consistency for the users. Additionally the same colours and font have been used throughout the application.
- **Error prevention**
  The stakeholders preferred, that there was no extra confirmation box after clicking confirm once anywhere in the application. The same is the case for the delete button, but to prevent accidental deletions it has been positioned to the far right of the screen, while all other buttons are located to the left.
- **Recognition rather than recall**
  The top menu has white and underlined links for navigation, while all other links are blue and underlined to distinct them from normal text. The most relevant information is displayed in the tables of data, however additional information can be found by clicking the right side of the table.

This is only the case for the tables that cannot contain all the data. Based on the stakeholders' feedback, it was deemed unnecessary with extra explanatory information on any screen.

- **Flexibility and efficiency of use**
  Since the application is primarily a booking system, the focus has been on making the process more effective compared to the former solution. This has been done by creating a search function for the component list and providing the users with as much information as possible, such as when the components they wish to book are available in the future, and the working status of all components.
- **Aesthetic and minimalist design**
  By involving the stakeholders in the design process, it was possible to create the application based on their wishes, with only the necessary information shown in the respective screens throughout the application.
- **Help users recognize, diagnose and recover from errors**
  The most common user caused error in the application is invalid typed data for creation or updates into the database. When these appear, an informative red error is either shown just under the menu bar, or at the actual field containing the problem, or both.
- **Help and documentation**
  There has not been implemented any user guide or help function for the application, but it has been considered, which has lead to requirement P2.7 which is a possible future feature.

3.3.3 General Functionality

This section will describe in detail the overall technical aspects of the web application.

- **CSS / Style sheets**
  Using CSS enabled an easy way to control the presentation of elements in the application. This was done by creating a master style sheet to specify the presentation of all elements, creating a consistent element design throughout all the screens in the application. By using a style sheet instead of designing the web screens individually, it has also become much easier to maintain and update the overall design in the future.
- **Login**
  When a user logs into the application the user's mail and project will be stored in a session, in the user's cookies, located on his computer. As long as this session exists the user is allowed to roam the user section of the application, and use its functionalities. A session is cleared by logging out or clearing the local cookies, which will result in the user being directed to the login screen when

trying to move around in the application. The same is also the case for the admin, where the session is based only on the admins mail. The security of a session is handled by an unencrypted security key, which grants a medium sense of security. However, since the key is not encrypted, it was decided not to store critical data in the sessions, such as the admin password. (http://scala.playframework.org/documentation/1.2/security)

- **Moving from screen to screen**

  When moving from one screen to another in the application, a series of events are triggered. First the session is checked for whether any user is logged on, and if no users are stored in the session the next screens content will be the login screen. Secondly the actual content of the screen is loaded, which consists of a Java function and its related database calls. Thirdly the master main.html is loaded, which contains the general design for all screens such as top menu and footer as shown in figure 3 Page 10. Finally the screen's content design is loaded, and ready to be displayed in the user's browser.

- **Screens**

  Each screen consists of a Java function, used to process the data used in the selected screen. To illustrate this, the function behind the Current Bookings screen has been selected.

```
public static void Current_Bookings() {
		Date actualDate = new Date();
		actualDate.~~setHours~~(00);
		actualDate.~~setMinutes~~(00);
		actualDate.~~setSeconds~~(00);

		List<Booking> bookings = Booking.find("SELECT b FROM Booking b WHERE 
b.checkoutDate >= ? AND b.user = ? ORDER BY b.checkinDate, b.checkoutDate", actualDate, 
userconnected()).fetch();

		render(bookings);
    }
```

Figure 8 Current bookings Java function

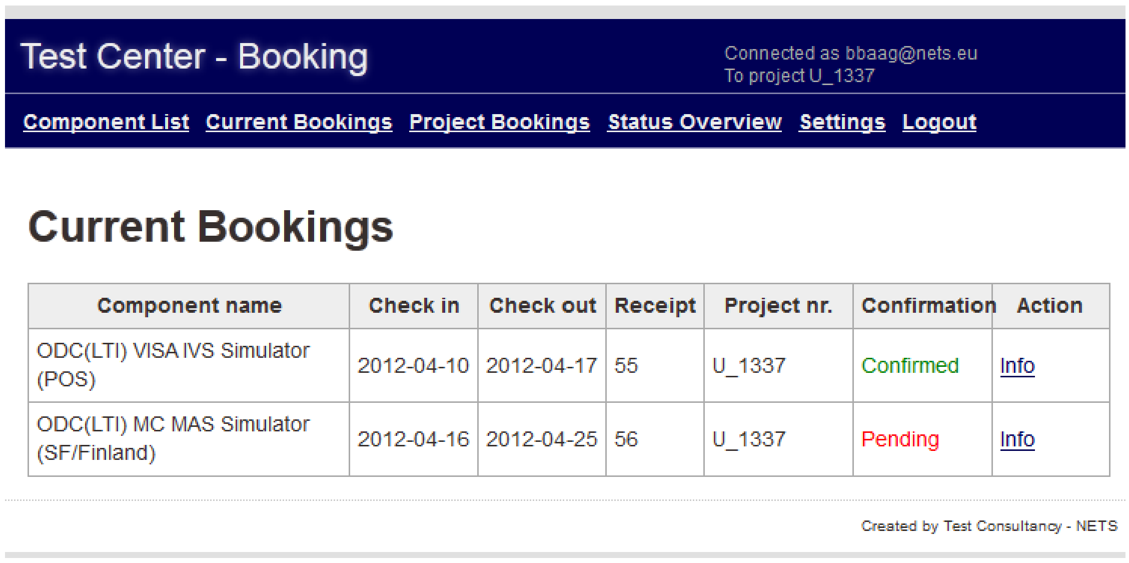
In the function shown in figure 8, the actual date is found and set to midnight. Thereafter a SQL query is sent to the database to find all current bookings for the connected user, which will be preserved as a list of objects. Next the HTML view is called together with the list of booked objects.

```
#{extends 'main.html' /}
#{set title:'Current_Bookings' /}

<h1>Current Bookings</h1>

#{ifnot bookings}
    <p>
        No Bookings Found
    </p>
#{/ifnot}
```

```
#{else}
   <table>
       <thead>
           <tr>
               <th width="30%">Component name</th>
               <th width="11%">Check in</th>
               <th width="11%">Check out</th>
               <th width="8%">Receipt</th>
               <th width="13%">Project nr.</th>
               <th width="12%">Confirmation</th>
               <th width="10%">Action</th>
           </tr>
       </thead>
       <tbody>
           #{list bookings, as:'booking'}
               <tr>
                   <td>${booking.component.comp_name}</td>
                   <td>${booking.checkinDate.format('yyyy-MM-dd')}</td>
                   <td>${booking.checkoutDate.format('yyyy-MM-dd')}</td>
                   <td>${booking.id}</td>
                   <td>${booking.project.project_number}</td>
                   <td>
                   #{if booking.book_confirmation == 0}
                    <FONT COLOR="#FF0000">Pending</FONT>
                   #{/if}
                   #{else}
                    <FONT COLOR="#008000">Confirmed</FONT>
                   #{/else}
                   </td>
                   <td>
                    <a href="@{userbookinginfo(booking.id)}">Info</a>
                   </td>
               </tr>
           #{/list}
       </tbody>
   </table>
#{/else}
```

Figure 9 Current booking HTML view

In figure 9 it is shown how the content of the screen Current Bookings has been designed by using HTML, Java script and the CSS style sheet. This code receives the list of booked objects which if empty writes “No Bookings Found”. If the list is not empty a table containing all the objects is created, which could look like figure 10.


Test Center - Booking

Connected as bbaag@nets.eu
To project U_1337

Component List Current Bookings Project Bookings Status Overview Settings Logout

Current Bookings

Component name	Check in	Check out	Receipt	Project nr.	Confirmation	Action
ODC(LTI) VISA IVS Simulator (POS)	2012-04-10	2012-04-17	55	U_1337	Confirmed	Info
ODC(LTI) MC MAS Simulator (SF/Finland)	2012-04-16	2012-04-25	56	U_1337	Pending	Info

Created by Test Consultancy - NETS

Figure 10 Current booking screen final presentation

The processes used as an example has been replicated with more and less advanced functions for the creation of the Test Center booking application.

3.4 Development Reflections

A challenging task for the booking application was the calendar function located in the admin section. After a long study of Java's way of working with dates and time, it was discovered how to specify the date needed into a calendar object, which then was used to create a date object. The date was required to be a date object to be used by the Play Framework database query, which then could find the information needed for the specific date.

All the code can be found on the CD delivered together with the report, where appendix 6.5 describes the location of the code files. Additionally appendix 6.2 provides an installation manual for creating a local instance of the application.

There has successfully been created a complete web application by using HTML, CSS, Java and MySQL. The communication between the module view controller layers were efficiently created and maintained by using the Play Framework, which enabled a fast upstart for the development work in the layers.

Chapter 4

Test Methods

4.1 Test strategy

Since the end product is to be used throughout the company, it is important to test it during the development phase. Tests will grant information about the applications quality, which can then be shown to the stakeholders to increase their interest and insight in the project. There exists a wide variety of software tests, which all grant different results. Therefore a selection of which tests were the most relevant ones for our application was done, which resulted in the following types of tests:

- Black box test
- Usability test
- Browser test

4.2 Black-box test

The black-box test has been completed by executing functionalities throughout the application, to ensure that they work as intended. The tests are aimed for the users and admins main functionalities, which have been carried out throughout the development phase, where table 7 and 8 shows the latest executed tests.

Table 7 Black box testing the user functionality

Nr.	Test Scenario	Expected result	Actual result	Ok?	**Appendix**
1.1	Create a new user and project followed by a login	The user and project is successfully created, as well as used to login immediately.	The user and project was successfully created, and could be used to login immediately.	Yes	**6.3.3 Page 41**
1.2	Use the component list search function to find a component with “London” in its name	Only the components with London in their name is shown in the table	Only the components with London in their name was shown in the table	Yes	**6.3.4 Page 42**
1.3	Select a component to book and enter a date in the past, and confirm the booking	An informative error message will appear.	An informative error message appeared.	Yes	**6.3.5 Page 42**
1.4	Use the expandable calendar in the date field to fill out a proper date and confirm the booking	The booking should be accepted and a message displaying the receipt number should be shown at the top of the screen	The booking was accepted and a message displaying the receipt number was shown	Yes	**6.3.6 Page 43**
1.5	Move to the *current bookings* screen by using the top menu, and click the *info* link at the new booking	The information for the booking should be displayed	The information for the booking was displayed	Yes	**6.3.7 Page 43**

	just created				
1.6	Delete the booking, and check that it has been removed from the *current bookings* list	The booking should be deleted and removed from the current bookings list	The booking was deleted and did not exist in the current booking list anymore	Yes	**6.3.8 Page 43**
1.7	Move to the *settings* screen by using the top menu and connect project to "U_1337", then move to the *project bookings* screen	Both the settings and *project bookings* screens should be shown successfully and the connection to the other project happened	Both the *settings* and *project bookings* screens were loaded successfully, and the connection to another project was successful	Yes	**6.3.9 Page 44**
1.8	Click the *status overview* link in top menu, followed by clicking logout	The *status overview* should be loaded, followed by a logout to the login screen	The *status overview* screen was loaded without issues, and the logout function worked as well	Yes	**-**

Table 8 Black box testing the admin functionality

Nr.	Test Scenario	Expected result	Actual result	Ok?	**Appendix**
2.1	Enter the admin login and login using user "Black@adm.eu" and password "Black123"	Access to the admin site should be granted and the *Morgen Check* screen shown	The admin site was successfully accessed and the *Morgen Check* screen shown	Yes	**6.3.10 Page 44**
2.2	Select the *update* link for the first component in the *checked daily* table, and complete the update with a new status and update note	The update link should show the screen required for updating the component, and the status should be updated based on the values typed.	The component was successfully updated	Yes	**6.3.11 Page 44**
2.3	Move to the *Status all* screen by using the top menu. From here click the *modify* link for the first component in the list	The *Status all* and *modify component* screens should be loaded successfully	The *Status all* and *modify component* screens were loaded successfully	Yes	**6.3.12 Page 45**
2.4	Modify the component by inserting "#" into the fields	The component should show an error in the location field, which can only contain a number	The application triggered a significant Runtime exception, which filled the whole screen. It was necessary to click back	No	**6.3.13 Page 45**

			to previous page on the browser to get back to the *modify component* screen		
2.5	Remove the "#" from the fields and change the note to "Black box tested"	The modification of the component should have happened successfully	The component was modified successfully and the *status all* screen was shown	Yes	**6.3.14 Page 45**
2.6	Move to the *status calendar* screen by using the top menu, and move through the weeks until a week without bookings have been found	The *status calendar* screen should be shown and the movement through the weeks should be working properly	The *status calendar* screen was shown and the movement through the weeks worked properly	Yes	**6.3.15 Page 46**
2.7	Move to the *booking calendar* screen by using the top menu, and move two month through the weeks or until a pending booking has been found	The *booking calendar* screen should be shown and the movement through the weeks should be working properly	The *booking calendar* screen was shown and the movement through the weeks worked properly	Yes	**6.3.16 Page 46**
2.8	Move to the *user bookings* screen by using the top menu, and click the first booking in the list's *info* link	The *user bookings* screen should be shown and the info link working properly	The *user bookings* screen was shown and the info link worked properly	Yes	**6.3.17 Page 47**
2.9	Click the confirmation to change it	The confirmation status should change, and a message should be displayed in the top of the screen	The confirmation changed, and the top message appeared	Yes	**6.3.18 Page 48**
2.10	Click the *settings* link in top menu, followed by clicking logout	The *settings* should be loaded, followed by a logout to the login screen	The *settings* screen was loaded without issues, and the logout function worked as well	Yes	**-**

4.2.1 Black box Test Reflections

The black box test shows a satisfying result, which is that the application works, except for one significant error in test nr. 2.4, that has to be handled in the future development, which will be discussed in the conclusion chapter 5.2

4.3 Usability Test

In order to best ensure that the application is intuitive for users, two usability tests were conducted on two types of stakeholders; a person from the Kopi Team (admin) and a Tester (user), which have all been described in chapter 1.3. Prototypes of the Test Center booking application were created for the completion of the usability tests described in this section. This was done to give the users a visual and interactive experience of the application. The usability test is based in the process described in (Molich, R. (2007)). These tests were executed late in the development phase, where almost all the main functionalities had already been implemented.

4.3.1 Usability goals

It was decided to focus on the following aspects for the application:

- **Useful:** The users must be able to use all the respective functionalities, without getting irritated.
- **Feature rich:** The application has to be seen as an improvement over the former solution, and provide the users with the feeling of getting additional useful features to work with.
- **Formal:** The overall theme of the applications design has been aimed to be formal, meaning that the users should find the applications design efficient and direct without getting confused where the information is located on the screen.

4.3.2 User Experience Goals

When a user is working with the application, the aspirations are that all the user's needs have been met, as well as the functions being usable. If a function is not intuitive, the user should be able to find helpful information on the actual page to get them through whatever obstacle they might encounter. The intention is that the users find the application a professional tool to handle their booking needs in the future. Generally users must be in favour of the content of the applications

4.3.3 Usability Test – Environment

The location for the usability tests was a meeting room, sized for four people, where the laptop in which the test performed, was plugged into a TV. The TV screen could be seen by the person testing and the person monitoring the test, who was sitting on the other side of the table taking notes, and asking if a list of actions are indeed possible to perform in the application. The person testing was asked to speak aloud any thoughts concerning the application.

4.3.4 Usability Test

Two usability tests were performed. The first was performed by a member of the Test Center Team (admin), while the other was by a tester (user). Both persons had not seen any of the prototypes for the application before helping with the usability tests. The persons are both considered to have a high technical understanding for computer systems and applications. The outcome of the usability sessions can be found in appendix 6.3.1 page 38-39.

4.3.5 Usability Test Reflections

None of the persons participating in the usability tests mentioned the general design, but it was easy to see that they knew where to focus when entering each screen. They both seemed content with the functionality provided by the application, but saw room for time saving improvements several places, which were noted as nice to have requirements.

They both had the same characteristics in the sense of a high understanding for computer systems and applications, which resulted in them taking the usability test as a challenge that needed completion fast, to show their worth. Additionally it meant that they were armed with the knowing of what is possible to create in an application, which they shared freely after analysing the screens throughout the application. In general it would have been preferred to do the usability tests on at least two to three more users with more varieties in their characteristics. However this was not possible since all the stakeholders in NETS have a high understanding for computers.

Overall the usability tests have been a success, since the user experience goals where fulfilled and usable feedback was gathered, which can be used to improve the application.

4.4 Browser Test

The standard browser in NETS is Microsoft internet explorer, but it is permitted for the employees to install the following four browsers on their computers. Therefore the application was tried out on Microsoft internet explorer, Firefox, Google Chrome and Safari.

There were no graphical differences, except for different lighting in the selected text fields. However there was a functionality difference. Firefox was the only browser that could show a dropdown list of objects from a text field, in all the other browsers the text field was just an ordinary text field. This can be seen in appendix 6.3.2 page 39 – 41.

4.4.1 Validation

The Firefox add-in “Developer” was installed and used to validate the Test Center booking applications HTML5 and CSS. The outcome resulted in few accepted HTML5 screens and several screens with minor errors in the form of obsolete code that the validation wanted created completely with CSS instead of in HTML. The CSS validation resulted in minor errors as well, in the form of colour selection for the design. Since these tests were carried out late in the development phase, they were noted to be debugged in the future development.

Chapter 5

Conclusion

5.1 Future Development

The managers have come to the agreement, that I will continue working for the company, and finish the Test Center booking application completely. Additionally after the last prototype presentation provided for the admins and managers, it was decided that a beta of the application should be installed on NETS web server. The purpose of this being that all stakeholders can try the application from their workstations and send feedback, which will be evaluated and presented to the managers, who will decide on the last changes to the application, before the final release date of the application. Of course only if the application receives positive feedback from both admins and users in its beta phase.

New possibilities will be available for the future development, since a company computer will be provided granting access to all internal development programs. This will allow the use of Extreme programming (Graham, D. et al. 2007) test method, which states that all possible test cases should be thought out and automated. After a new feature has been added to the application, the automated regression test should be executed to ensure that no former functionality has been lost due to the new feature. The program provided by NETS to create this automated regression test is called Quicktest Professional.

5.2 Conclusion

The purpose of the project was to develop a booking system for NETS' Test Center in order to replace the existing manually driven e-mail and excel based process with a web based solution.

After gathering the first requirements and analysing them, priorities were listed from 1 to 3. One being the most important which were the most needed functionalities required to create the application successfully. Two were non-critical functionalities and three were non-essential extra functionalities.

Throughout the development process additional prototypes of the application were created and shown to the stakeholders. This resulted in new requirements as well as the confirmation or redefining of already stated requirements, to ensure that the stakeholders needs for the application where fulfilled.

The Play Framework was a good help for the application development, providing efficient tools as well as connections between the model view controller layers. Additionally the frameworks ability to display code changes directly in a local instance of the web application was a great help for prototype presentations.

As the development progressed, it was clear that the iterative process helped specify the requirements as well as involve the stakeholders, which has lead to the creation of a web based booking system that the stakeholders acknowledge being able to perform its primary functions, and would like to try. The end product of this project has therefore become a beta version of the Test Center booking system, which is expected to replace the former manual booking procedure.

The application has been tested by a Black-box test which showed that all the functionalities worked, except when modifying an already existing components location with an invalid value, which results in an error message without meaning, with no impact on the program. This should be solved before a final implementation of the application.

The application has also been subjected to usability testing, which showed that the general design and functionality were usable, but with room for improvement, which are all included in the priority two requirements.

All priority one requirements have successfully been implemented in the application, which means that the main functionalities are working. For the successful deployment of the application into NETS, it would be preferred to have most of priority two requirements also implemented, to fulfil the stakeholder’s expectations.

The project can be deemed successful based on the success criteria, which has all been met.

Chapter 6

Appendix

6.1 References

The following books and web pages have been used for the completion of this project.

- Rolf Molich (2007). Usable Web Design. Chapter 11. Think Aloud testing. Nyt Teknisk Forlag 2007.

- Dorothy Graham, Erik Van Veenendaal, Isabel Evans, Rex Black (2008). Foundations of Software Testing. Chapter 2, page 40. Course Technology.

- Jim Highsmith and Alistair Cockburn. (2001) Agile Software Development: The Business of Innovation. *Computer* page 120 to 122.
  http://ieeexplore.ieee.org.globalproxy.cvt.dk/stamp/stamp.jsp?tp=&arnumber=947100&tag=1
  Found: 01-04-2012

- Jacob Nielsen (1994). 10 usability heuristics
  http://www.useit.com/papers/heuristic/heuristic_list.html
  Found: 01-04-2012

- The following link is for the Play Frameworks home page

  http://www.playframework.org/

- Figure 6 was found on page http://www.playframework.org/documentation/1.0/main

- The following link describes the security in the Play Framework Sessions

  http://scala.playframework.org/documentation/1.2/security

- The following link contains all the programs used for the creation of the database

  http://dev.mysql.com/

- The following link contains the “developer” add-in for Firefox, which was used for the browser test

  https://addons.mozilla.org/da/firefox/addon/web-developer/

- The following link is a Java library, used for the development of the application

  http://docs.oracle.com/javase/1.4.2/docs/api/

6.2 Installation manual

The Test Center booking application has been designed for installation on a web server with its associated database. It is however possible to create a local instance of the application as well as database, which this installation manual will focus upon.

1. Make sure that there is a working Java 5 or later installation on the computer.
2. Now install the database which is done by installing MySQL server version 5.5.20 or better. On the CD delivered with the report there is a file called "mysql Server-5.5.20-win32.msi". Executing this file will install the database server.
   Be sure to create the server on localhost:3306 with password "Fisk"
3. The database schema has to be created manually before the Play Framework can work with it, therefore it is necessary to install MySQL workbench version 5.2.37 or better. On the CD delivered with the report there is a installation file for this called "mysql-workbench-glp-5.2.37-win32.msi"
4. When the MySQL workbench has been installed, it should be possible to connect to the local database server on port 3306 with password Fisk. Thereafter a list of schemas will be presented to the left of the workbench. Here a schema called "test_center_booking" should be created.
5. Now Play can be installed, which is done by finding the folder Play on the CD delivered with the report, and moving it to the local C drive.
6. When this is done, open a command prompt and write:

   ```
   Cd\

   Cd play

   play run TCB
   ```

7. By opening a browser (Firefox, Microsoft Internet Explorer, Safari or Google Chrome) this application can be accessed by writing the URL: http://localhost:9000
8. When the application has been started once the database will have been created, which by refreshing the MySQL workbench should appear.
   To gain access to the admin site of the application, an admin has first to exist. This can be done by writing the following into the database from the MySQL workbench:

   ```
   INSERT INTO `test_center_booking`.`admin` (`id`, `adminmail`, `name`, `password`,
   `admin_created`) VALUES (1, 'Adm@adm.eu', 'Admin', 'Admin', '2012-04-10 12:00:00');
   ```

   This will make the admin email: Adm@adm.eu and password: Admin

9. It is now possible to add components as an admin in the application, or directly into the database. Which will provide data required to work with the main functionality of the application
10. Now the Test Center booking application has been installed and all the possible functions unlocked.

6.3 Test Results

6.3.1 Usability Test

Test Center Team

At first the admin was asked which expectations were held for the application, where the response was that it was a not fully completed application, in which areas could be lacking data, since this test would provide information required to improve the application.

The test started from the *user login* screen, where the admin spent around 30 seconds to study the applications design before clicking the *admin login* link. Have been given the admin mail and password in advance, the *admin login* screen was bypassed within 10 seconds. Now met with the *Morgen Check* screen the admin tried to click the status metaphors to change the component status. After around 30 seconds the table was fully understood and a component update had been done successfully. It was here mentioned by the admin that it would be preferred to have the status changes directly in the tables instead of selecting them individually. Next the admin was asked to find if a status update was necessary on specific dates, which lead to the admin clicking the status calendar. Here the first thing the admin mentioned was a compliment, that there actually existed a calendar, but also a concern that the actual screen could be very bothersome and overwhelming to use if there where to many components that require a status update a single week. Next the admin was asked if it was possible to update a component, which first got the admin to move to the *settings* where the screen was studied from top to bottom in around 25 seconds, before selecting the *status all* link in the top menu. Here modify was quickly selected and a component updated.

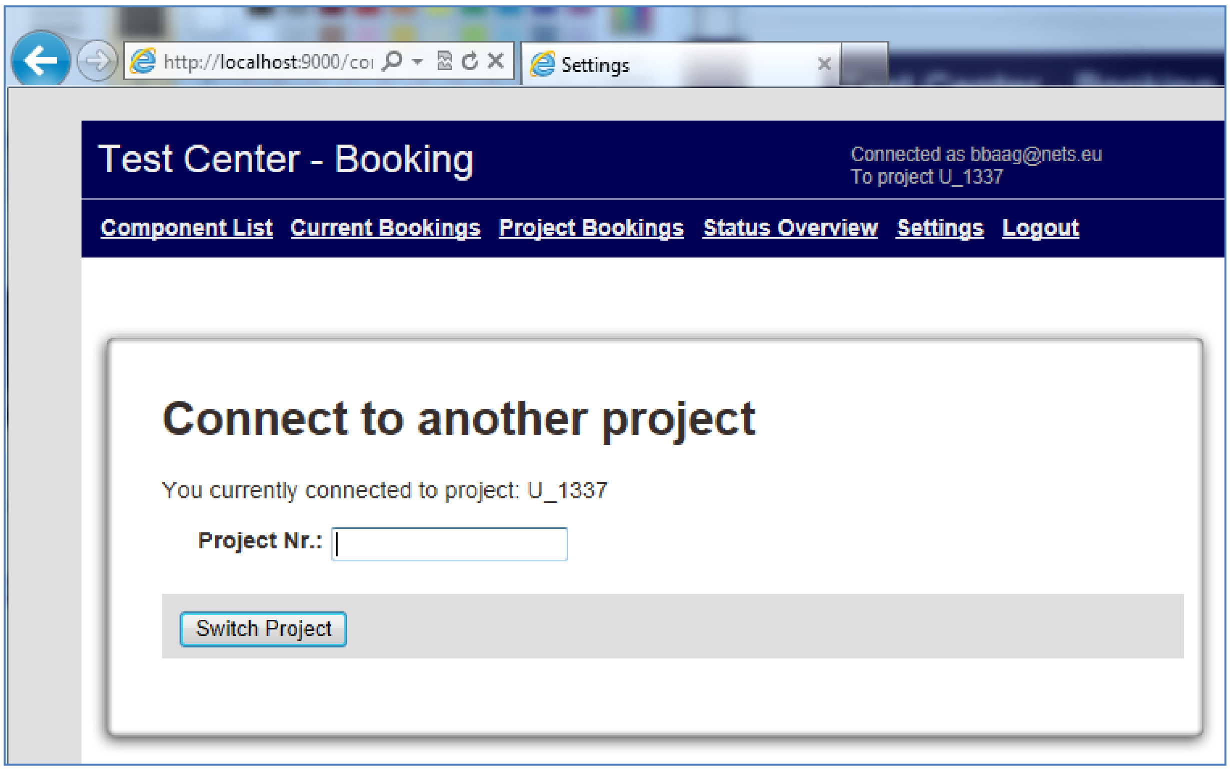
Usability Test (Tester)

At first the user was asked which expectations where held for the application, where the response was that it would be an improved way of handling bookings in the future.

The test started from the *user login* screen, where the user had to create a new user and project, which took a little under one minute. After entering the application, several seconds were used to click through the component list. Here the task was to book a specific component, which the user found by expanding the component list fully and using the browsers search function to locate it. On the *booking* screen the user noticed the bottom table showing all other bookings of this specific component, and tried deliberately to create an overlapping booking, which was accepted by the system, which made the user comment that some kind of warning was expected, additionally it was commented that it sometimes could be beneficial if the booking system had a shopping cart function, to book additional components at once. The whole booking process for the first booking took around 30 seconds. Next the user moved to the *current bookings* screen where the user quickly found the new booking that was just created, and clicked the *info* link for it. This resulted in the comment that there had to be some kind of update functionality, again as a shopping cart, because additional booked tests could require to be postponed due to a multitude of situations in their development project. The task was to find out if it was possible to change the booking date, which the user answered no. The last task was to see if it was possible to change project to one provided for the test. This resulted in the user logging out and entering again using the user mail created in the start of the test and the new project number. It was only after the tasks were completed and the user was allowed to freely roam the application, that the functionality to change projects while still logged on, was discovered in the settings, which made the user comment that it was not bothersome to log out and in, and that having the functionality in the settings is a bonus.

6.3.2 Browser Test

Microsoft Internet Explorer



Test Center - Booking

Connected as bbaag@nets.eu
To project U_1337

Component List Current Bookings Project Bookings Status Overview Settings Logout

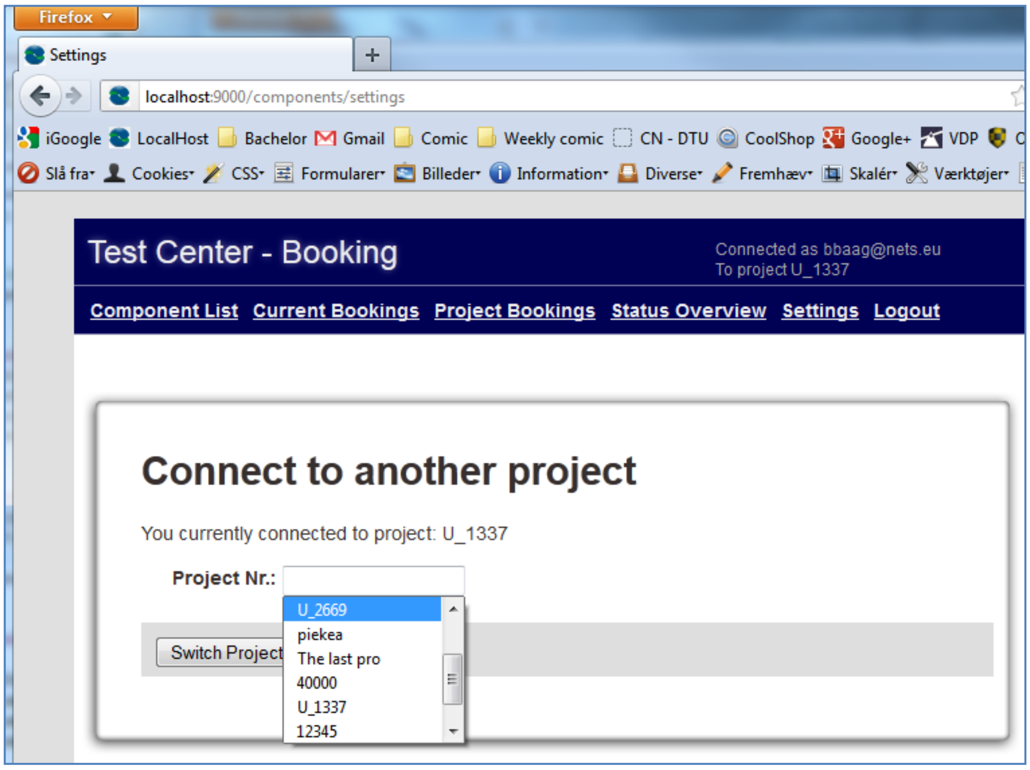
Connect to another project

You currently connected to project: U_1337

Project Nr.:

Switch Project

Firefox


Firefox
Settings
localhost:9000/components/settings
iGoogle LocalHost Bachelor Gmail Comic Weekly comic CN - DTU CoolShop Google+ VDP
Slå fra Cookies CSS Formularer Billeder Information Diverse Fremhæv Skalér Værktøjer

Test Center - Booking

Connected as bbaag@nets.eu
To project U_1337

Component List Current Bookings Project Bookings Status Overview Settings Logout

Connect to another project

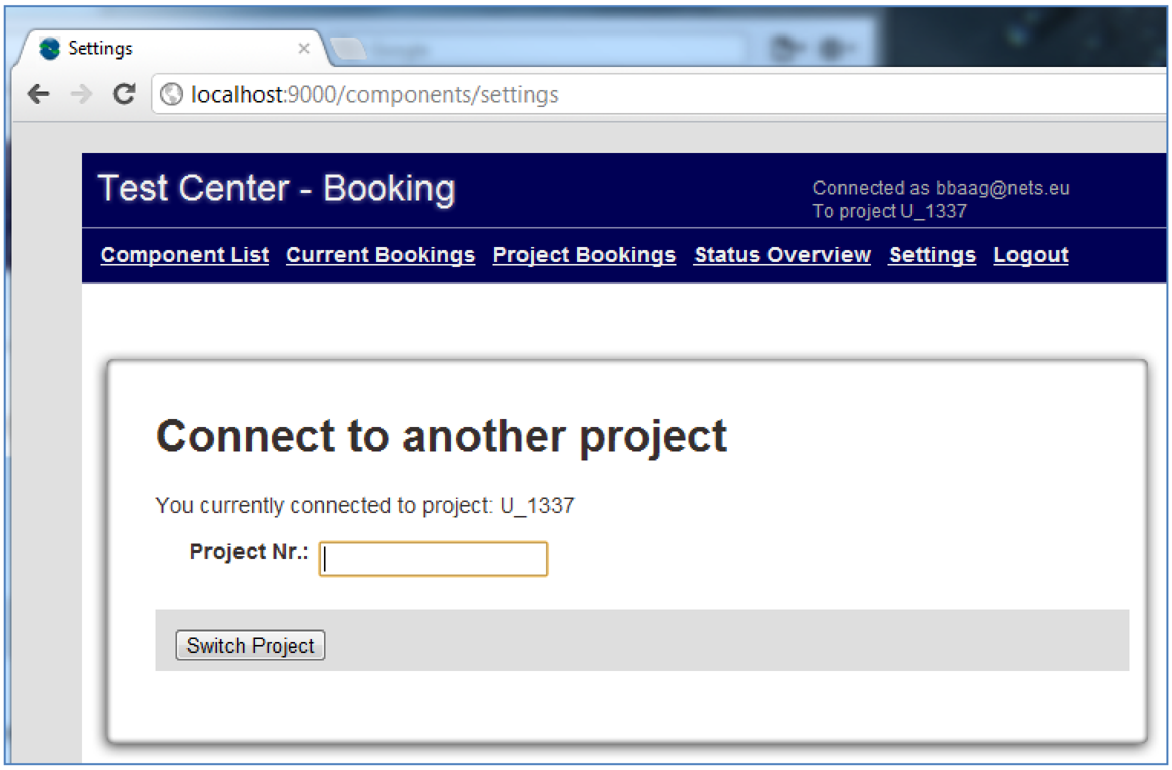
You currently connected to project: U_1337

Project Nr.:

U_2669
piekea
The last pro
40000
U_1337
12345

Switch Project

Google Chrome



Test Center - Booking

Connected as bbaag@nets.eu
To project U_1337

Component List Current Bookings Project Bookings Status Overview Settings Logout

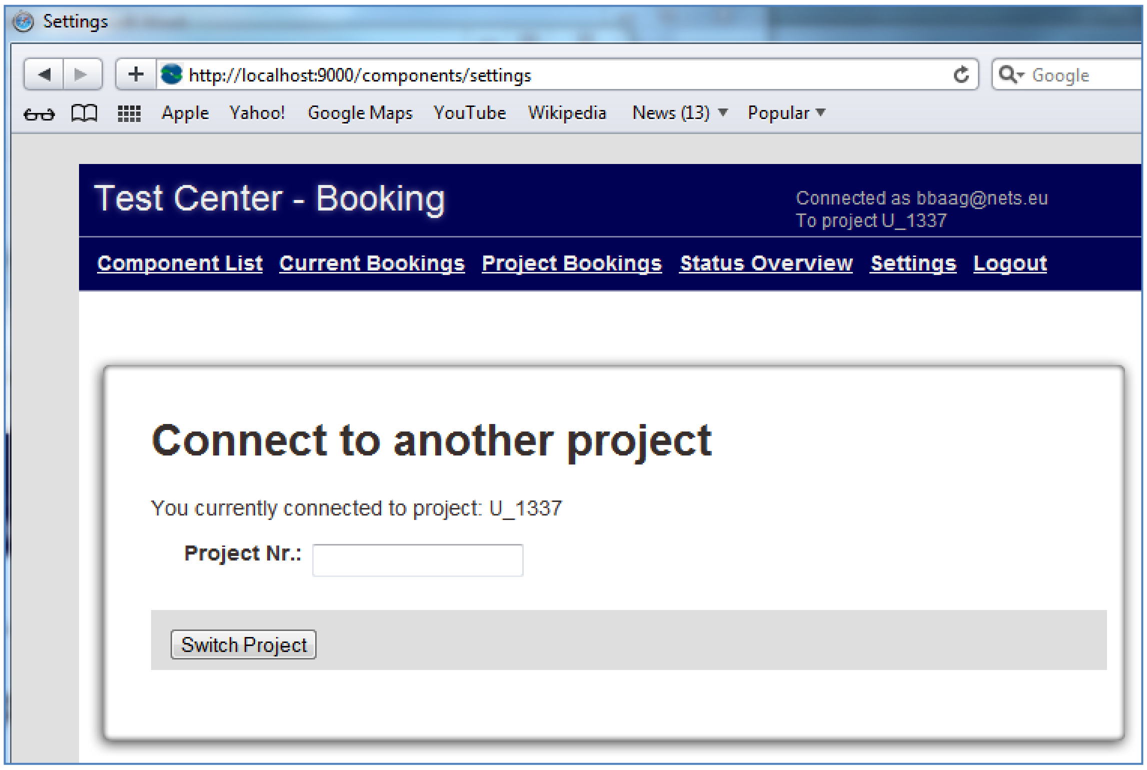
Connect to another project

You currently connected to project: U_1337

Project Nr.:

Switch Project

Safari



Settings

http://localhost:9000/components/settings

Apple Yahoo! Google Maps YouTube Wikipedia News (13) Popular

Test Center - Booking

Connected as bbaag@nets.eu
To project U_1337

Component List Current Bookings Project Bookings Status Overview Settings Logout

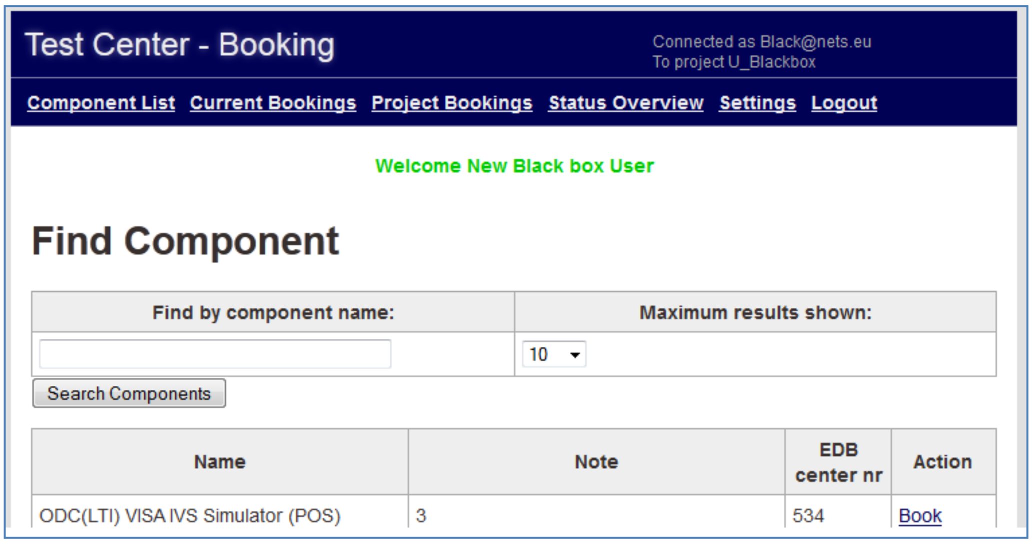
Connect to another project

You currently connected to project: U_1337

Project Nr.:

Switch Project

6.3.3 Black box result 1.1



Test Center - Booking

Connected as Black@nets.eu
To project U_Blackbox

Component List Current Bookings Project Bookings Status Overview Settings Logout

Welcome New Black box User

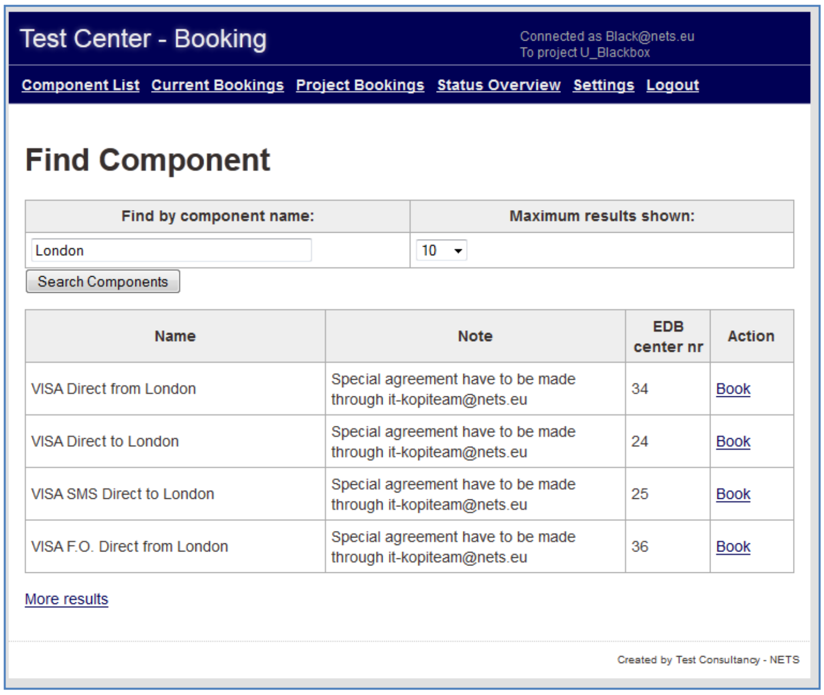
Find Component

Find by component name:	Maximum results shown:
	10

Search Components

Name	Note	EDB center nr	Action
ODC(LTI) VISA IVS Simulator (POS)	3	534	Book

6.3.4 Black box result 1.2



Test Center - Booking

Connected as Black@nets.eu
To project U_Blackbox

Component List Current Bookings Project Bookings Status Overview Settings Logout

Find Component

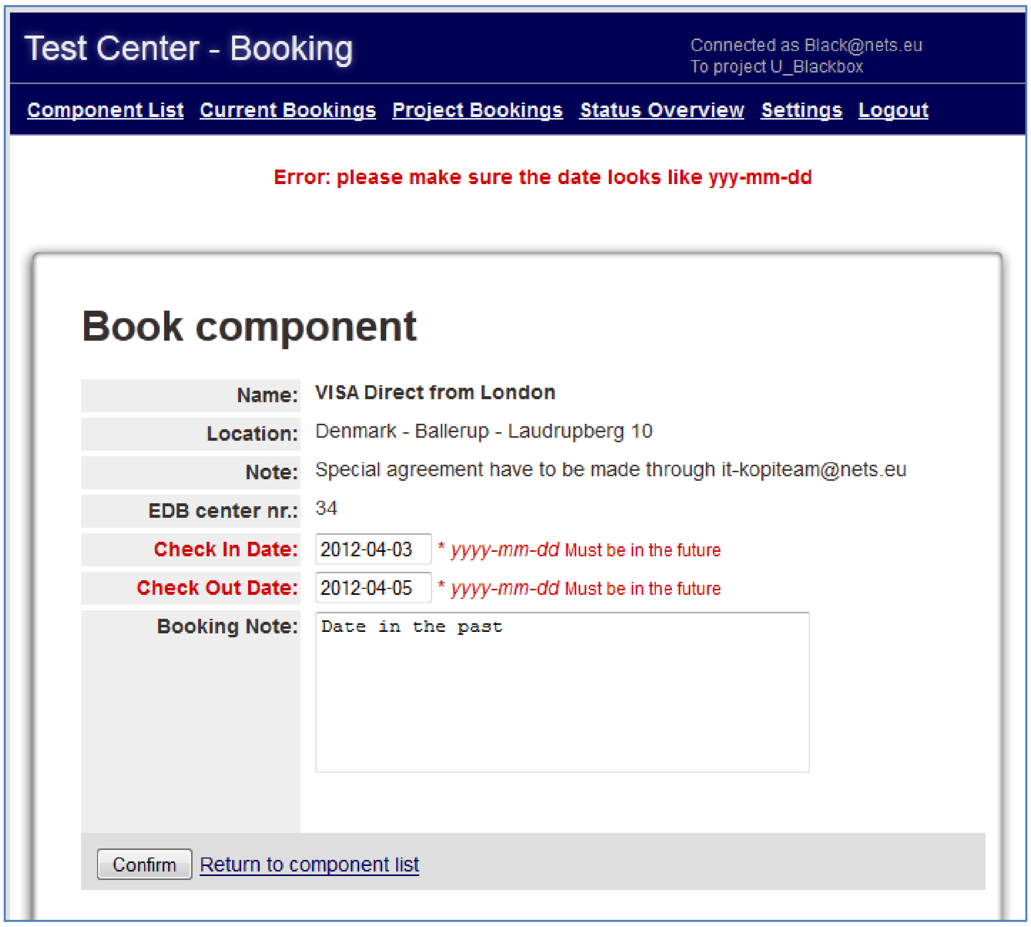
Find by component name:	Maximum results shown:
London	10

Name	Note	EDB center nr	Action
VISA Direct from London	Special agreement have to be made through it-kopiteam@nets.eu	34	Book
VISA Direct to London	Special agreement have to be made through it-kopiteam@nets.eu	24	Book
VISA SMS Direct to London	Special agreement have to be made through it-kopiteam@nets.eu	25	Book
VISA F.O. Direct from London	Special agreement have to be made through it-kopiteam@nets.eu	36	Book

More results

Created by Test Consultancy - NETS

6.3.5 Black box result 1.3



Test Center - Booking

Connected as Black@nets.eu
To project U_Blackbox

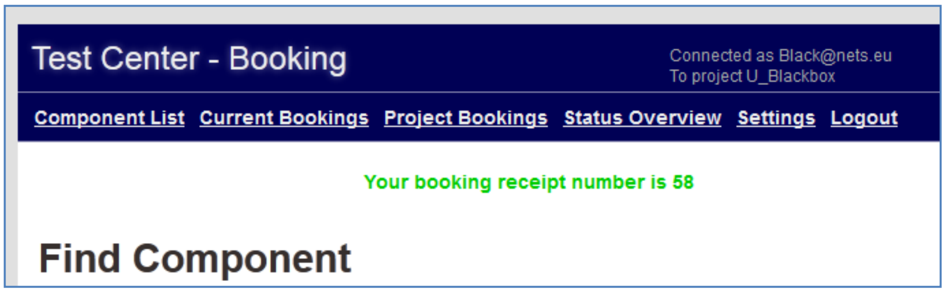
Component List Current Bookings Project Bookings Status Overview Settings Logout

Error: please make sure the date looks like yyy-mm-dd

Book component

Name:	VISA Direct from London
Location:	Denmark - Ballerup - Laudrupberg 10
Note:	Special agreement have to be made through it-kopiteam@nets.eu
EDB center nr.:	34
Check In Date:	2012-04-03 * yyyy-mm-dd Must be in the future
Check Out Date:	2012-04-05 * yyyy-mm-dd Must be in the future
Booking Note:	Date in the past

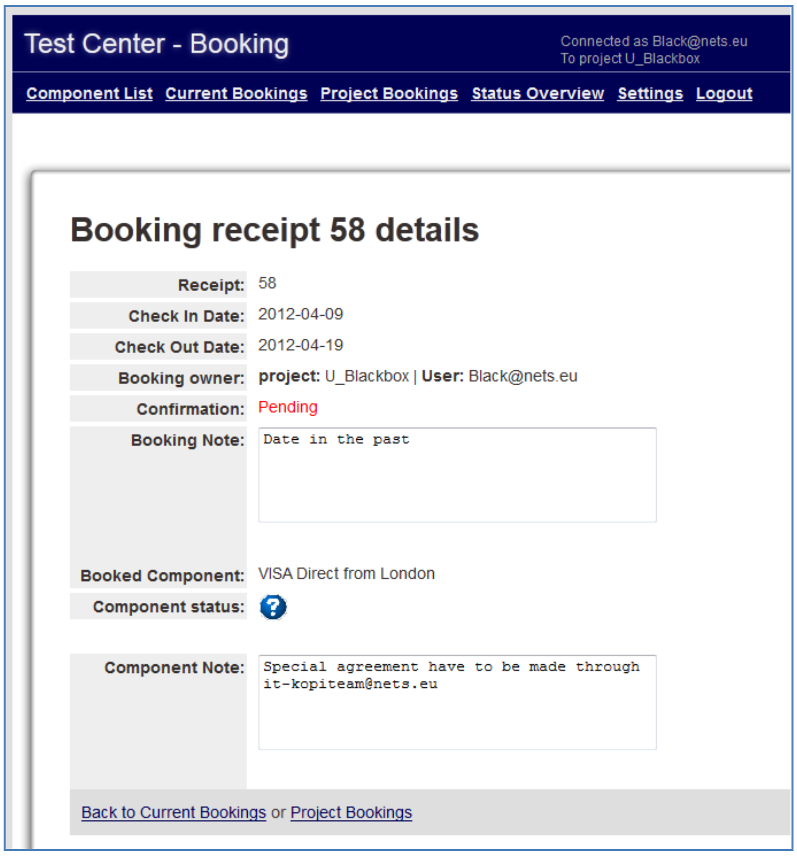
6.3.6 Black box result 1.4



Your booking receipt number is 58

Find Component

6.3.7 Black box result 1.5

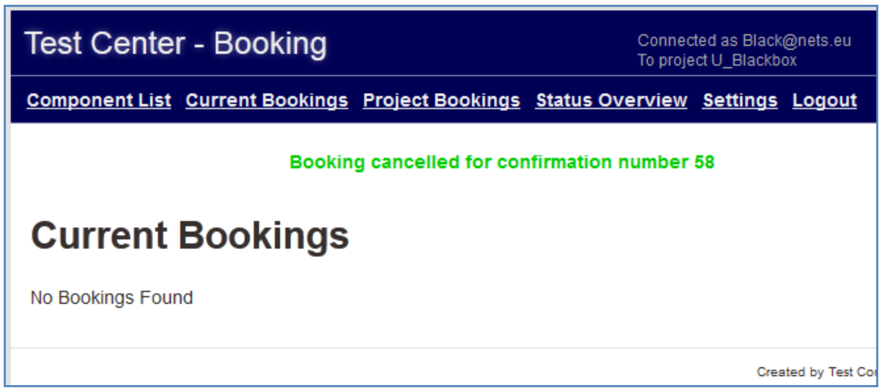


Booking receipt 58 details

Receipt:	58
Check In Date:	2012-04-09
Check Out Date:	2012-04-19
Booking owner:	**project:** U_Blackbox | **User:** Black@nets.eu
Confirmation:	Pending
Booking Note:	Date in the past
Booked Component:	VISA Direct from London
Component status:	?
Component Note:	Special agreement have to be made through it-kopiteam@nets.eu

Back to Current Bookings or Project Bookings

6.3.8 Black box result 1.6



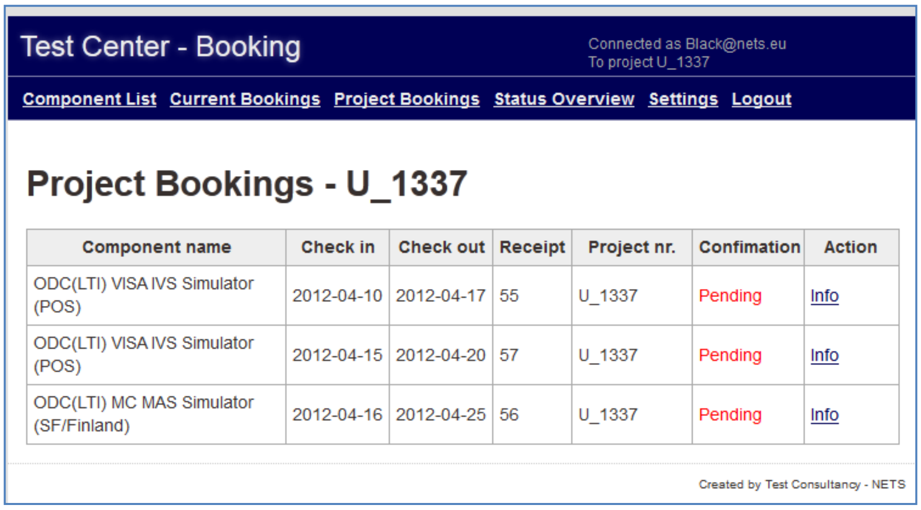
Booking cancelled for confirmation number 58

Current Bookings

No Bookings Found

Created by Test Co

6.3.9 Black box result 1.7



Test Center - Booking

Connected as Black@nets.eu
To project U_1337

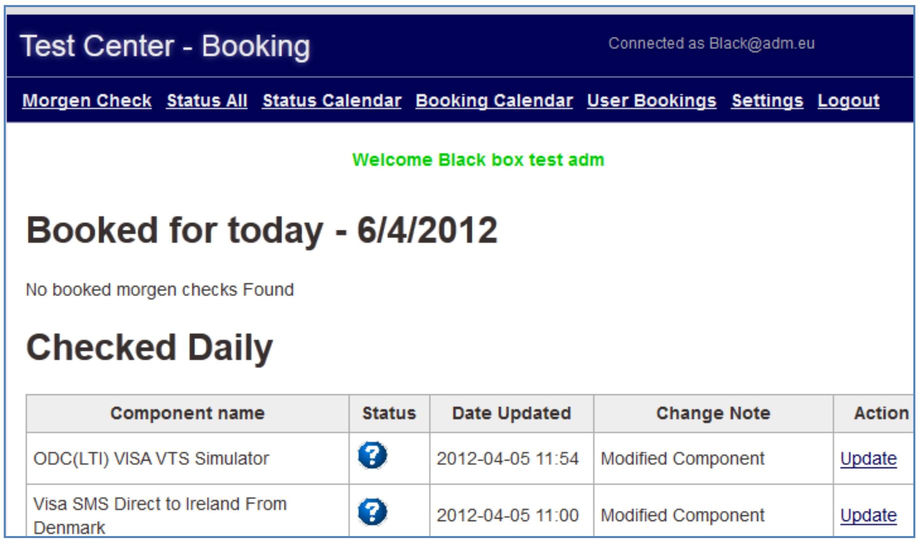
Component List Current Bookings Project Bookings Status Overview Settings Logout

Project Bookings - U_1337

Component name	Check in	Check out	Receipt	Project nr.	Confimation	Action
ODC(LTI) VISA IVS Simulator (POS)	2012-04-10	2012-04-17	55	U_1337	Pending	Info
ODC(LTI) VISA IVS Simulator (POS)	2012-04-15	2012-04-20	57	U_1337	Pending	Info
ODC(LTI) MC MAS Simulator (SF/Finland)	2012-04-16	2012-04-25	56	U_1337	Pending	Info

Created by Test Consultancy - NETS

6.3.10 Black box result 2.1



Test Center - Booking

Connected as Black@adm.eu

Morgen Check Status All Status Calendar Booking Calendar User Bookings Settings Logout

Welcome Black box test adm

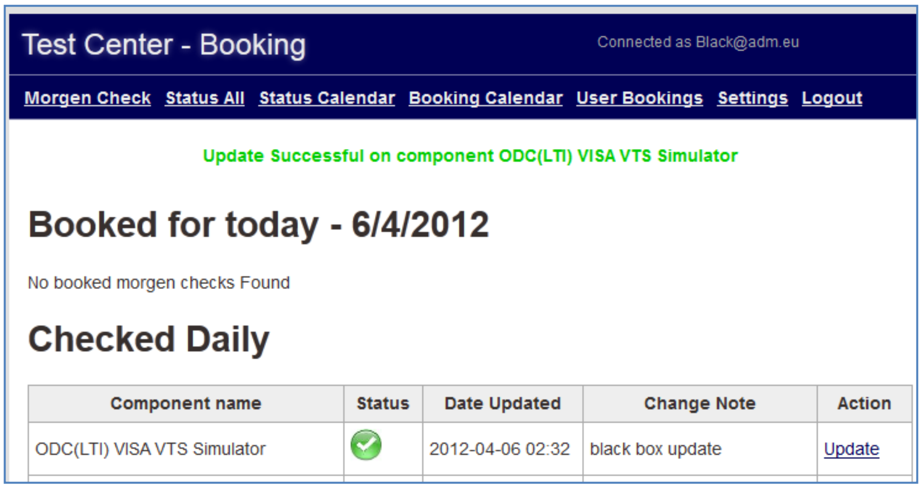
Booked for today - 6/4/2012

No booked morgen checks Found

Checked Daily

Component name	Status	Date Updated	Change Note	Action
ODC(LTI) VISA VTS Simulator	?	2012-04-05 11:54	Modified Component	Update
Visa SMS Direct to Ireland From Denmark	?	2012-04-05 11:00	Modified Component	Update

6.3.11 Black box result 2.2



Test Center - Booking

Connected as Black@adm.eu

Morgen Check Status All Status Calendar Booking Calendar User Bookings Settings Logout

Update Successful on component ODC(LTI) VISA VTS Simulator

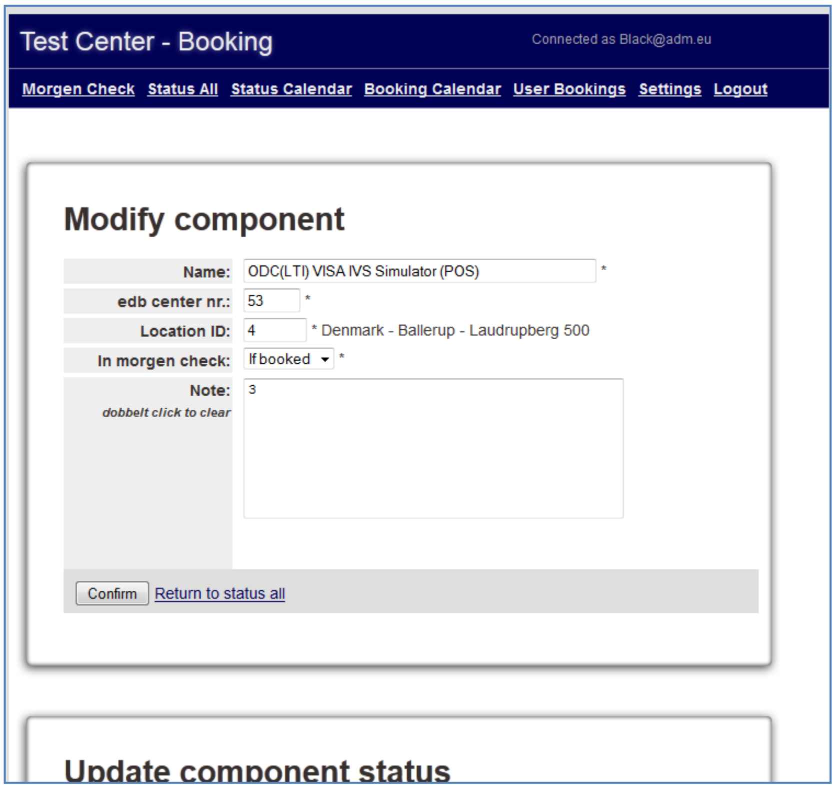
Booked for today - 6/4/2012

No booked morgen checks Found

Checked Daily

Component name	Status	Date Updated	Change Note	Action
ODC(LTI) VISA VTS Simulator	✓	2012-04-06 02:32	black box update	Update

6.3.12 Black box result 2.3



Test Center - Booking

Connected as Black@adm.eu

Morgen Check Status All Status Calendar Booking Calendar User Bookings Settings Logout

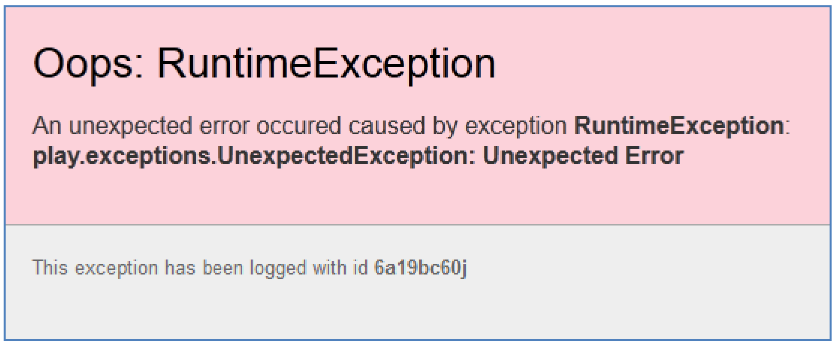
Modify component

Name:	ODC(LTI) VISA IVS Simulator (POS) *
edb center nr.:	53 *
Location ID:	4 * Denmark - Ballerup - Laudrupberg 500
In morgen check:	If booked *
Note: *dobbelt click to clear*	3

Confirm Return to status all

Update component status

6.3.13 Black box result 2.4

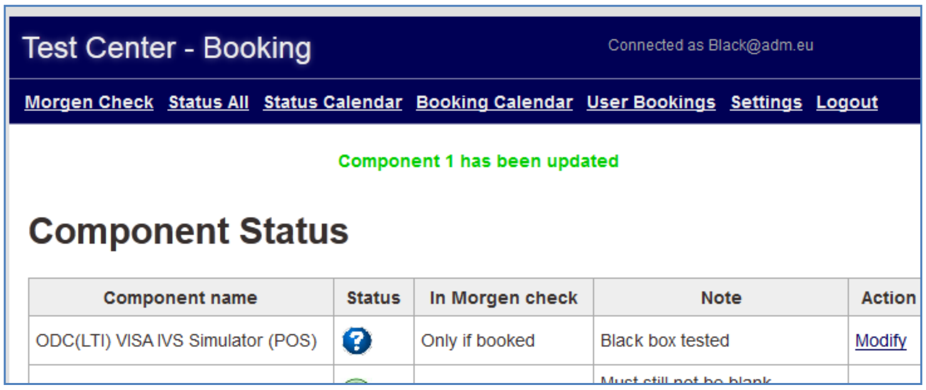


Oops: RuntimeException

An unexpected error occured caused by exception **RuntimeException**: **play.exceptions.UnexpectedException: Unexpected Error**

This exception has been logged with id **6a19bc60j**

6.3.14 Black box result 2.5



Test Center - Booking

Connected as Black@adm.eu

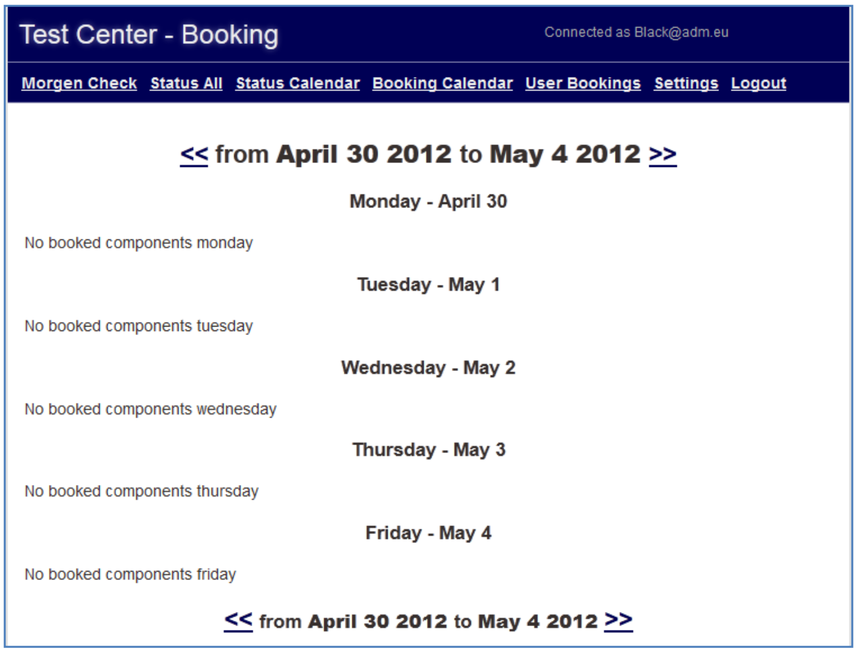
Morgen Check Status All Status Calendar Booking Calendar User Bookings Settings Logout

Component 1 has been updated

Component Status

Component name	Status	In Morgen check	Note	Action
ODC(LTI) VISA IVS Simulator (POS)	?	Only if booked	Black box tested	Modify

6.3.15 Black box result 2.6



Test Center - Booking

Connected as Black@adm.eu

Morgen Check Status All Status Calendar Booking Calendar User Bookings Settings Logout

<< from **April 30 2012** to **May 4 2012** >>

Monday - April 30

No booked components monday

Tuesday - May 1

No booked components tuesday

Wednesday - May 2

No booked components wednesday

Thursday - May 3

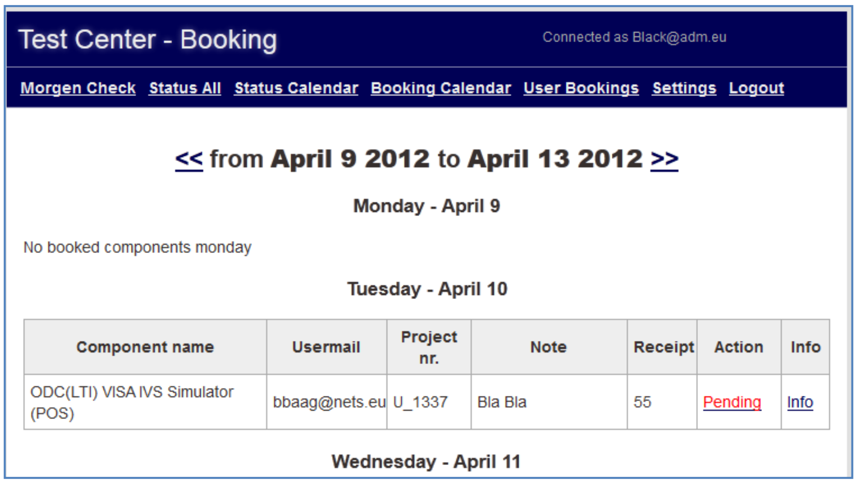
No booked components thursday

Friday - May 4

No booked components friday

<< from **April 30 2012** to **May 4 2012** >>

6.3.16 Black box result 2.7



Test Center - Booking

Connected as Black@adm.eu

Morgen Check Status All Status Calendar Booking Calendar User Bookings Settings Logout

<< from **April 9 2012** to **April 13 2012** >>

Monday - April 9

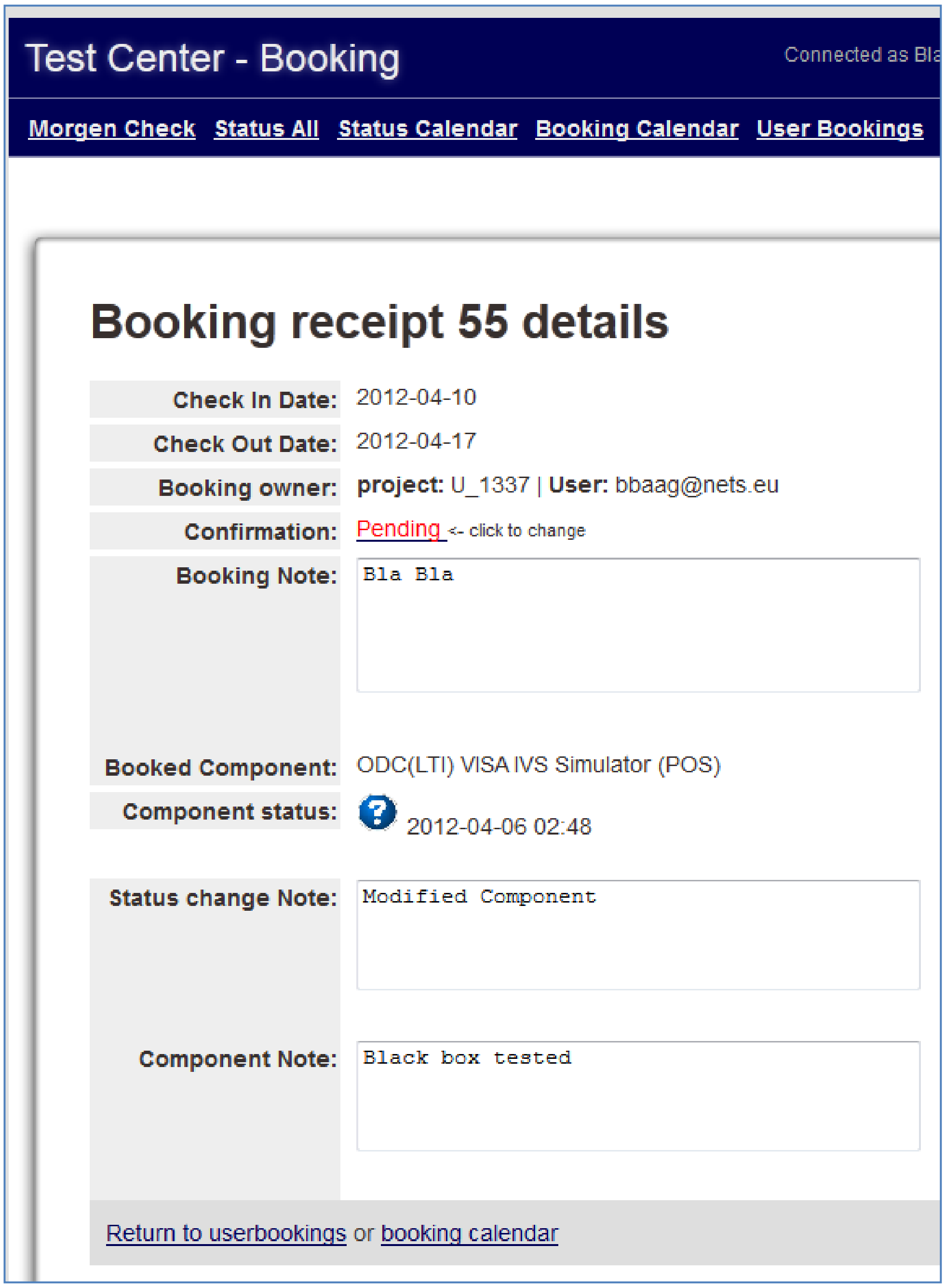
No booked components monday

Tuesday - April 10

Component name	Usermail	Project nr.	Note	Receipt	Action	Info
ODC(LTI) VISA IVS Simulator (POS)	bbaag@nets.eu	U_1337	Bla Bla	55	Pending	Info

Wednesday - April 11

6.3.17 Black box result 2.8

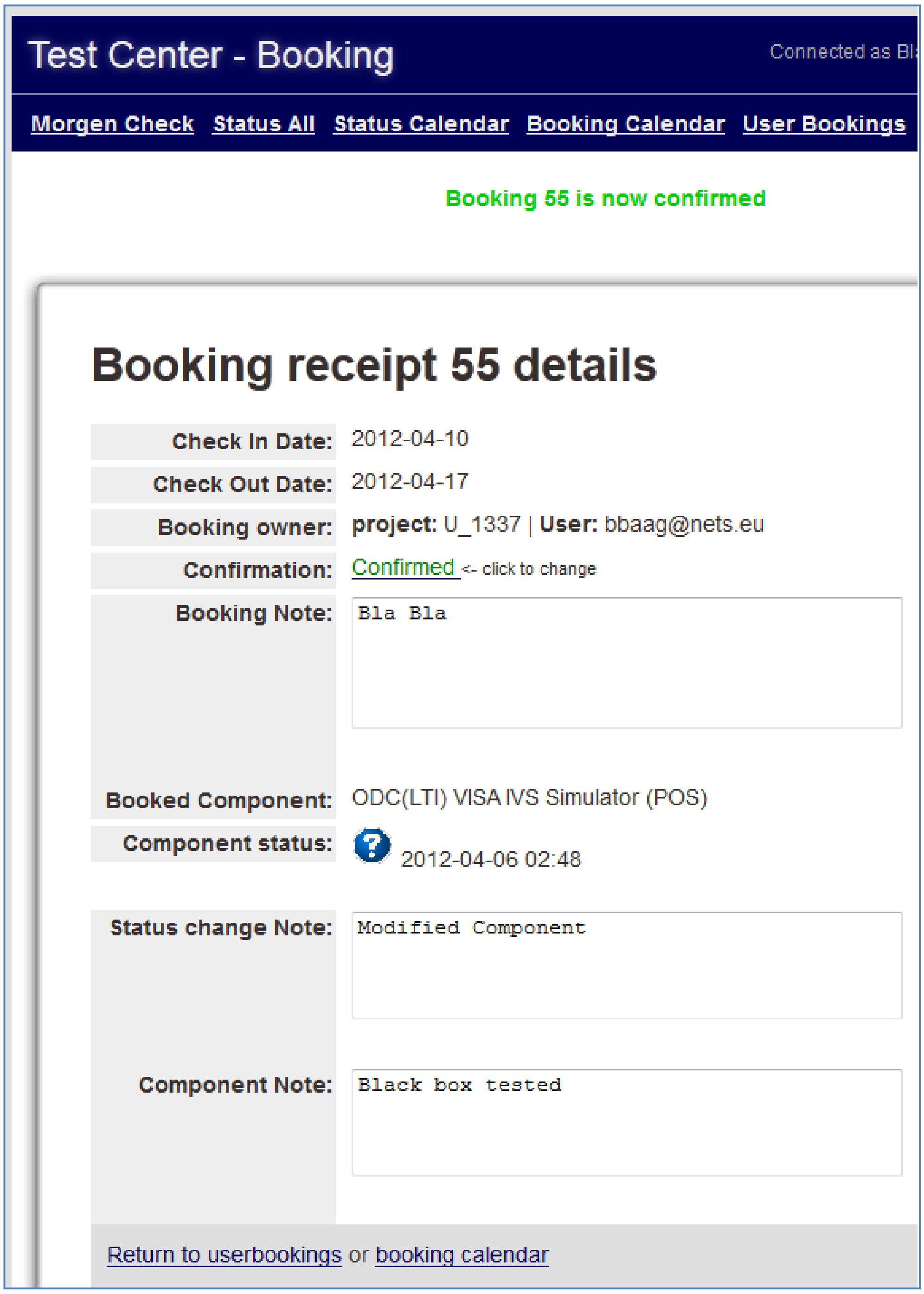


Booking receipt 55 details

Check In Date:	2012-04-10
Check Out Date:	2012-04-17
Booking owner:	**project:** U_1337 | **User:** bbaag@nets.eu
Confirmation:	Pending <- click to change
Booking Note:	Bla Bla
Booked Component:	ODC(LTI) VISA IVS Simulator (POS)
Component status:	? 2012-04-06 02:48
Status change Note:	Modified Component
Component Note:	Black box tested

Return to userbookings or booking calendar

6.3.18 Black box result 2.9



Test Center - Booking

Connected as Bl

Booking 55 is now confirmed

Booking receipt 55 details

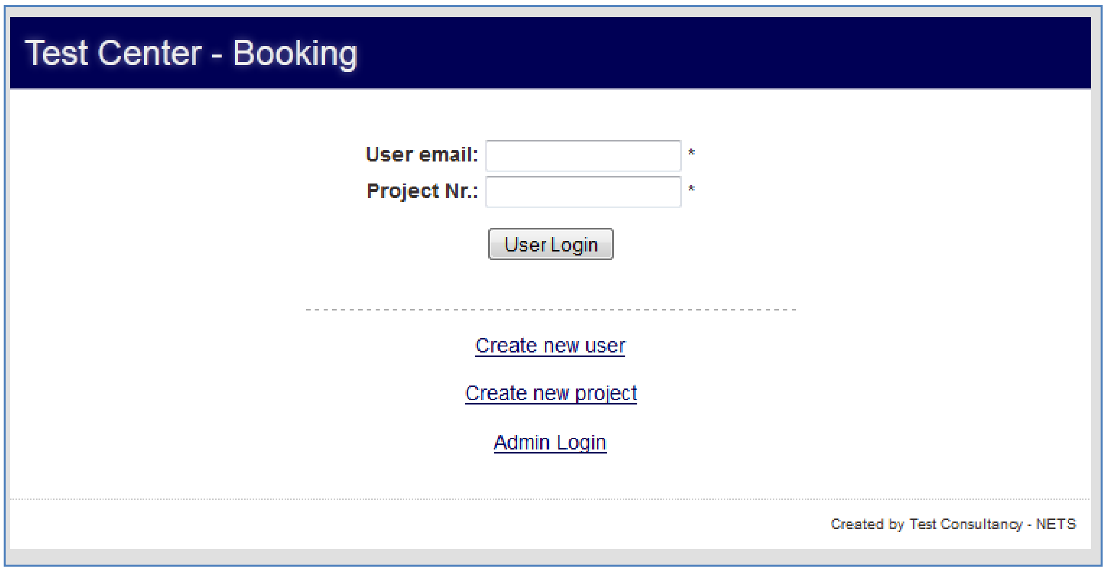
Check In Date:	2012-04-10
Check Out Date:	2012-04-17
Booking owner:	**project:** U_1337 | **User:** bbaag@nets.eu
Confirmation:	Confirmed <- click to change
Booking Note:	Bla Bla
Booked Component:	ODC(LTI) VISA IVS Simulator (POS)
Component status:	? 2012-04-06 02:48
Status change Note:	Modified Component
Component Note:	Black box tested

Return to userbookings or booking calendar

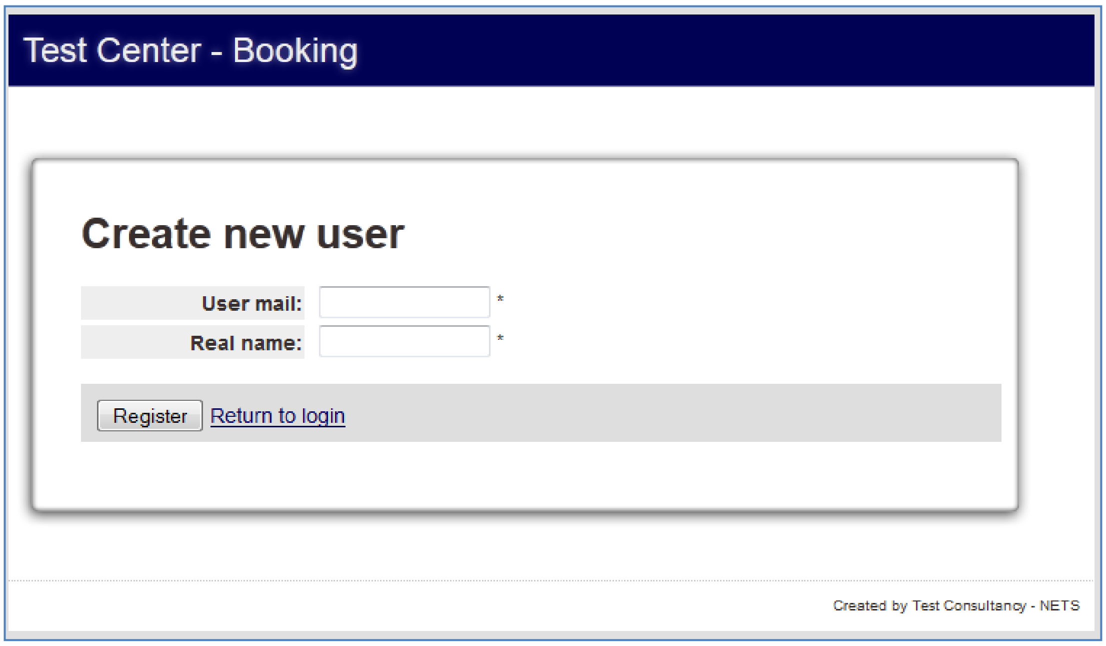
6.4 Final Design

The following pictures show the finalised design of the Test Center booking application.

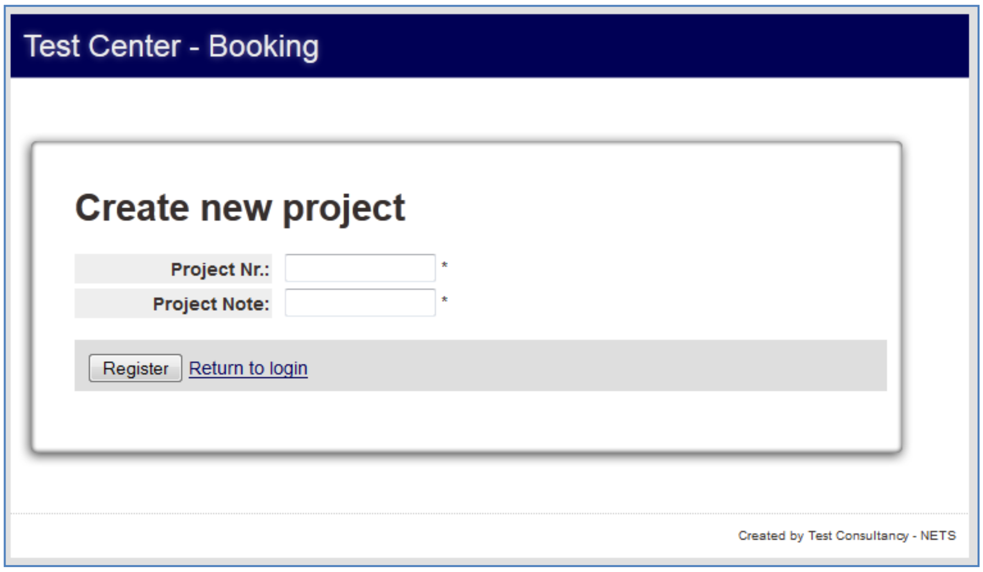
6.4.1 User Login



6.4.2 Create New User



6.4.3 Create New Project



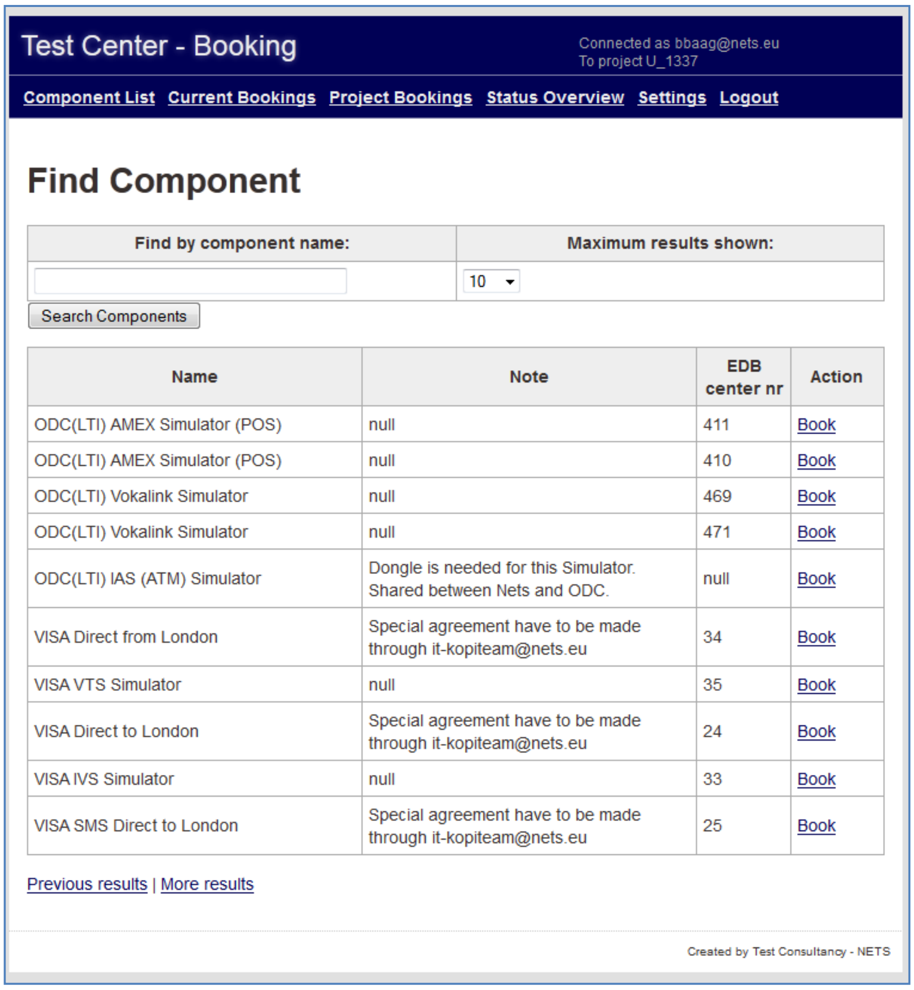
Test Center - Booking

Create new project

Project Nr.:	*
Project Note:	*

Created by Test Consultancy - NETS

6.4.4 Component List (User site)



Test Center - Booking

Connected as bbaag@nets.eu
To project U_1337

Component List Current Bookings Project Bookings Status Overview Settings Logout

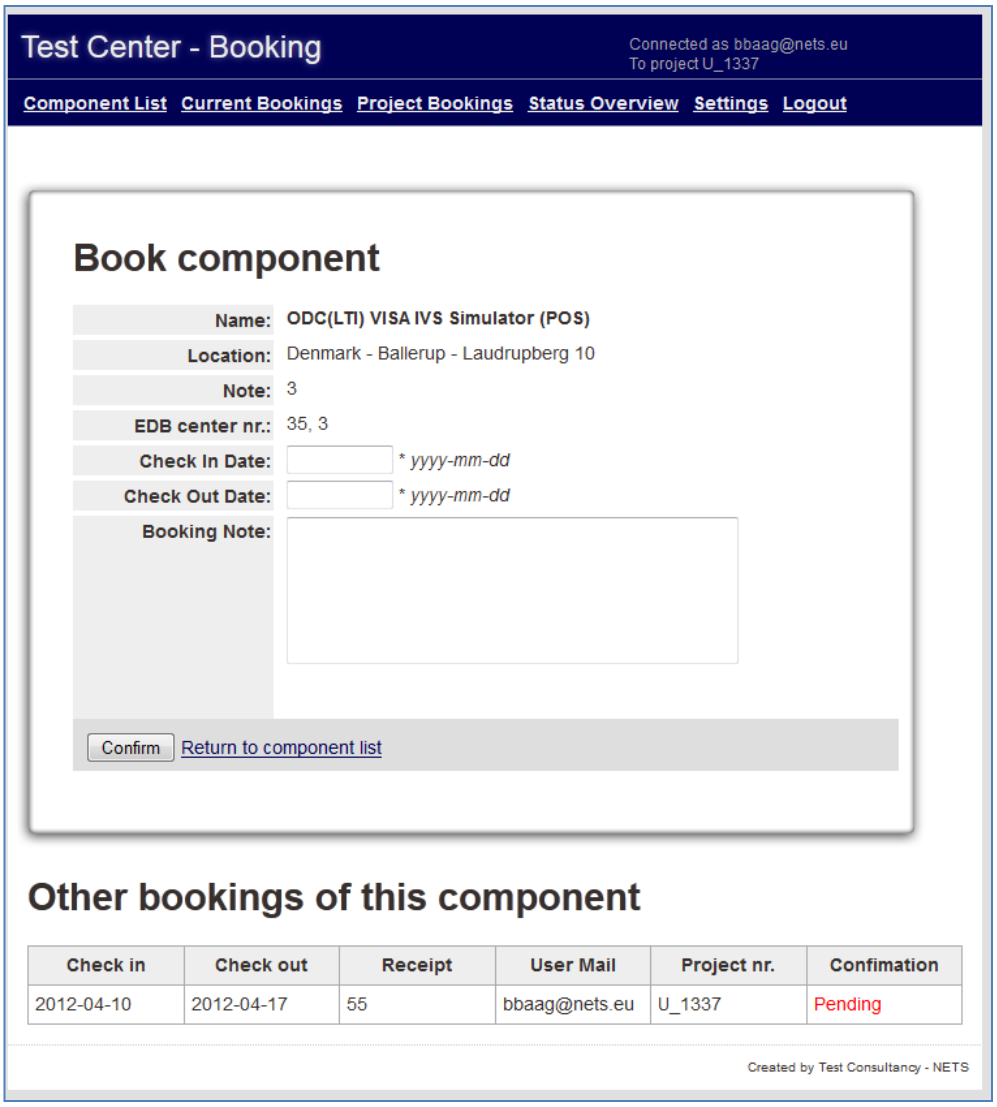
Find Component

Find by component name:	Maximum results shown:
	10

Name	Note	EDB center nr	Action
ODC(LTI) AMEX Simulator (POS)	null	411	Book
ODC(LTI) AMEX Simulator (POS)	null	410	Book
ODC(LTI) Vokalink Simulator	null	469	Book
ODC(LTI) Vokalink Simulator	null	471	Book
ODC(LTI) IAS (ATM) Simulator	Dongle is needed for this Simulator. Shared between Nets and ODC.	null	Book
VISA Direct from London	Special agreement have to be made through it-kopiteam@nets.eu	34	Book
VISA VTS Simulator	null	35	Book
VISA Direct to London	Special agreement have to be made through it-kopiteam@nets.eu	24	Book
VISA IVS Simulator	null	33	Book
VISA SMS Direct to London	Special agreement have to be made through it-kopiteam@nets.eu	25	Book

Created by Test Consultancy - NETS

6.4.5 Book Component (User site)



Test Center - Booking

Connected as bbaag@nets.eu
To project U_1337

Component List Current Bookings Project Bookings Status Overview Settings Logout

Book component

Name:	ODC(LTI) VISA IVS Simulator (POS)
Location:	Denmark - Ballerup - Laudrupberg 10
Note:	3
EDB center nr.:	35, 3
Check In Date:	* yyyy-mm-dd
Check Out Date:	* yyyy-mm-dd
Booking Note:	

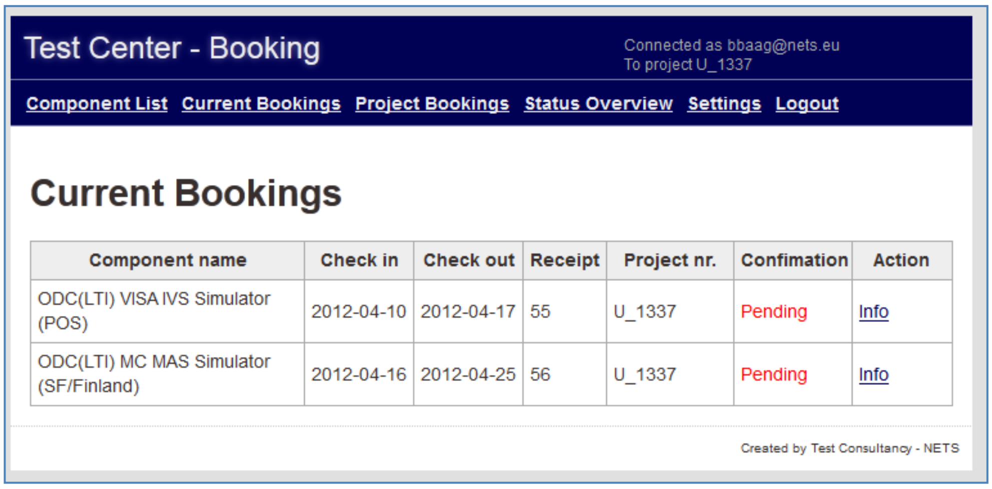
Confirm Return to component list

Other bookings of this component

Check in	Check out	Receipt	User Mail	Project nr.	Confimation
2012-04-10	2012-04-17	55	bbaag@nets.eu	U_1337	Pending

Created by Test Consultancy - NETS

6.4.6 Current Bookings (User site)



Test Center - Booking

Connected as bbaag@nets.eu
To project U_1337

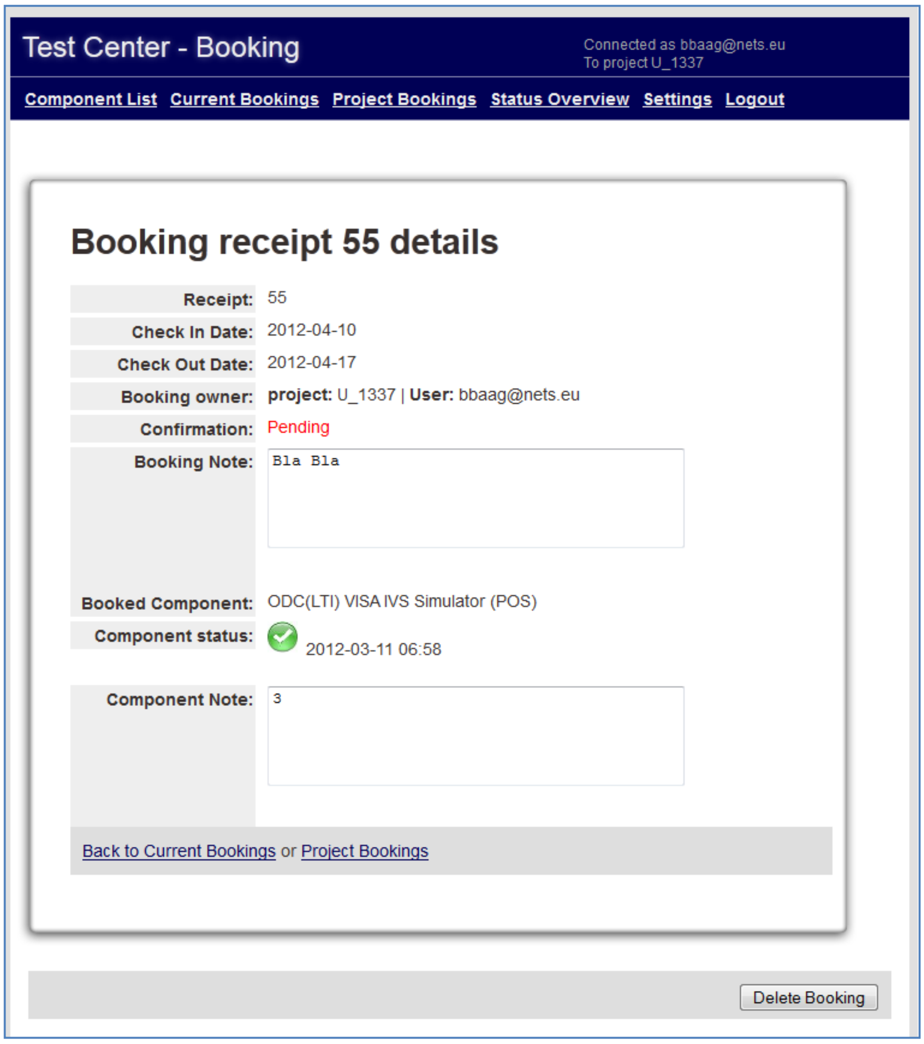
Component List Current Bookings Project Bookings Status Overview Settings Logout

Current Bookings

Component name	Check in	Check out	Receipt	Project nr.	Confimation	Action
ODC(LTI) VISA IVS Simulator (POS)	2012-04-10	2012-04-17	55	U_1337	Pending	Info
ODC(LTI) MC MAS Simulator (SF/Finland)	2012-04-16	2012-04-25	56	U_1337	Pending	Info

Created by Test Consultancy - NETS

6.4.7 Booking Information (User site)



Test Center - Booking

Connected as bbaag@nets.eu
To project U_1337

Component List Current Bookings Project Bookings Status Overview Settings Logout

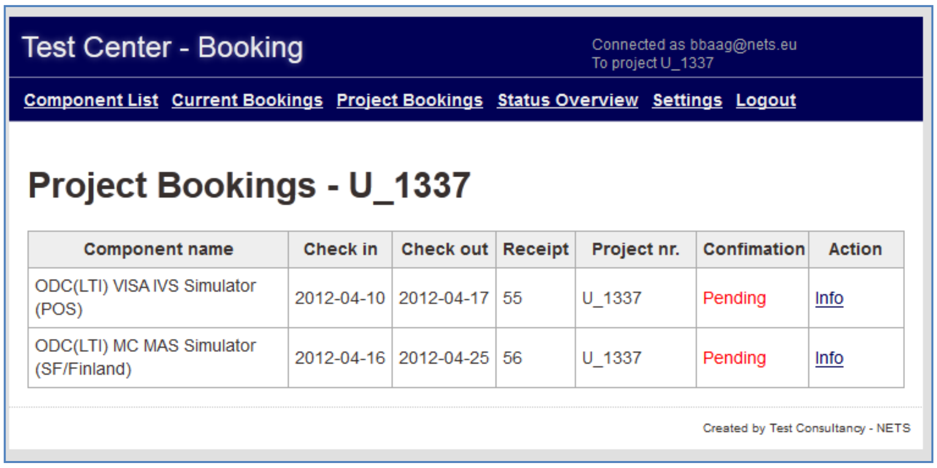
Booking receipt 55 details

Receipt:	55
Check In Date:	2012-04-10
Check Out Date:	2012-04-17
Booking owner:	**project:** U_1337 | **User:** bbaag@nets.eu
Confirmation:	Pending
Booking Note:	Bla Bla
Booked Component:	ODC(LTI) VISA IVS Simulator (POS)
Component status:	✓ 2012-03-11 06:58
Component Note:	3

Back to Current Bookings or Project Bookings

Delete Booking

6.4.8 Project Bookings (User site)



Test Center - Booking

Connected as bbaag@nets.eu
To project U_1337

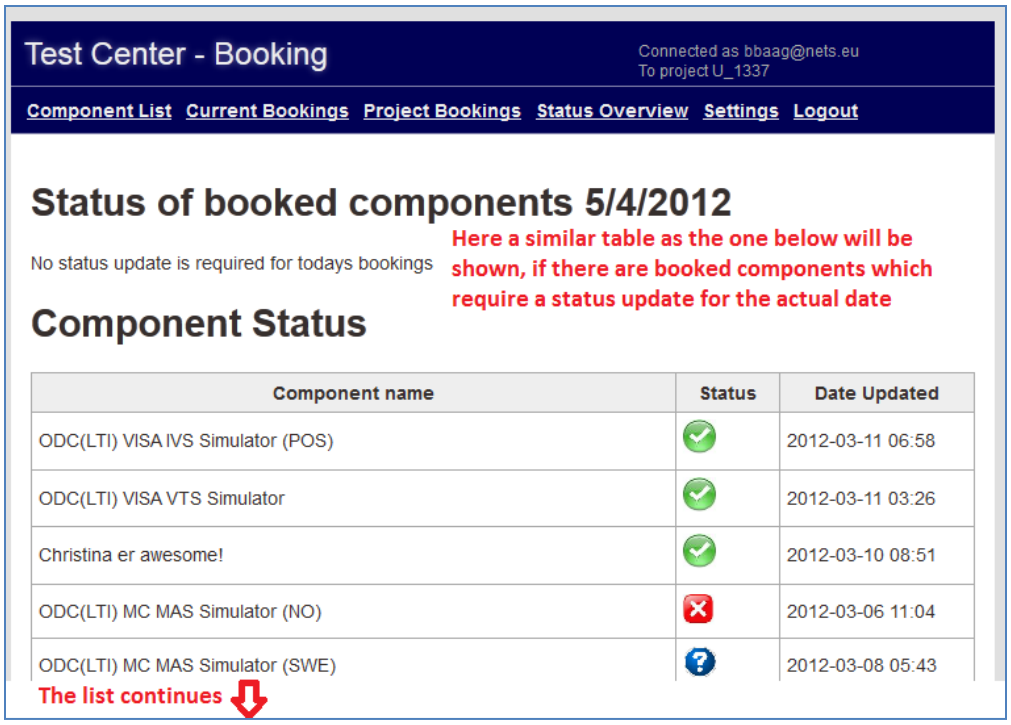
Component List Current Bookings Project Bookings Status Overview Settings Logout

Project Bookings - U_1337

Component name	Check in	Check out	Receipt	Project nr.	Confimation	Action
ODC(LTI) VISA IVS Simulator (POS)	2012-04-10	2012-04-17	55	U_1337	Pending	Info
ODC(LTI) MC MAS Simulator (SF/Finland)	2012-04-16	2012-04-25	56	U_1337	Pending	Info

Created by Test Consultancy - NETS

6.4.9 Status Overview (User site)



Test Center - Booking

Connected as bbaag@nets.eu
To project U_1337

Status of booked components 5/4/2012

No status update is required for todays bookings

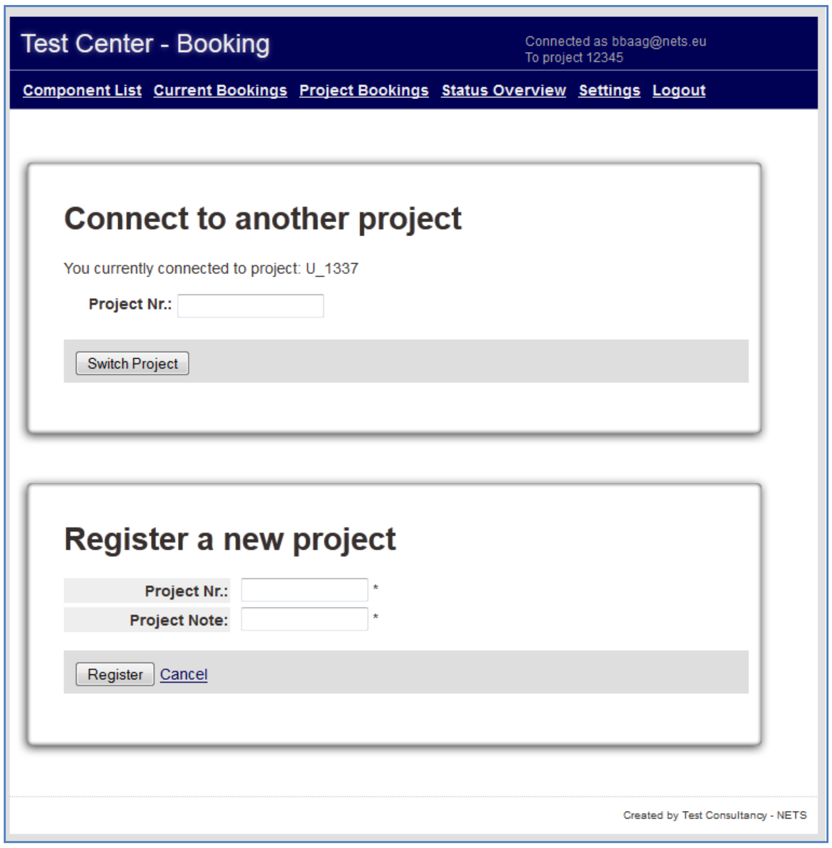
Here a similar table as the one below will be shown, if there are booked components which require a status update for the actual date

Component Status

Component name	Status	Date Updated
ODC(LTI) VISA IVS Simulator (POS)	✓	2012-03-11 06:58
ODC(LTI) VISA VTS Simulator	✓	2012-03-11 03:26
Christina er awesome!	✓	2012-03-10 08:51
ODC(LTI) MC MAS Simulator (NO)	✗	2012-03-06 11:04
ODC(LTI) MC MAS Simulator (SWE)	?	2012-03-08 05:43

The list continues ⇩

6.4.10 Settings (User site)



Test Center - Booking

Connected as bbaag@nets.eu
To project 12345

Connect to another project

You currently connected to project: U_1337

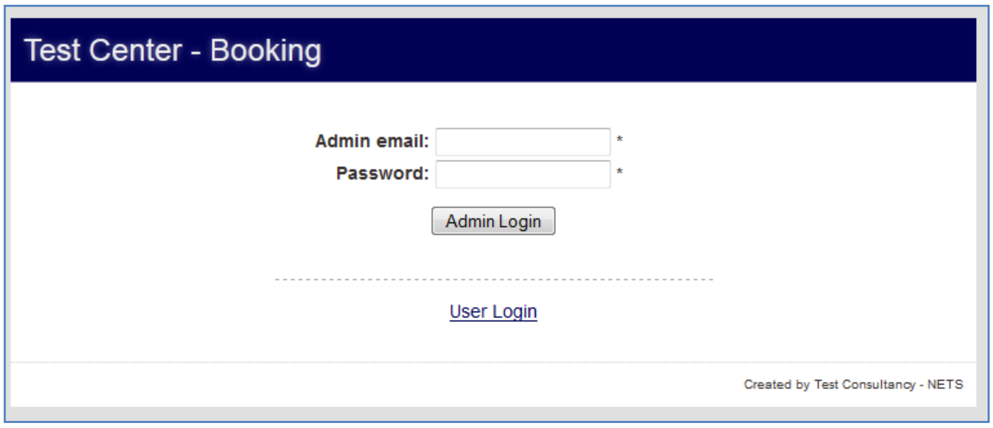
Project Nr.:

Register a new project

Project Nr.: *
Project Note: *

Created by Test Consultancy - NETS

6.4.11 Admin Login



Test Center - Booking

Admin email: *

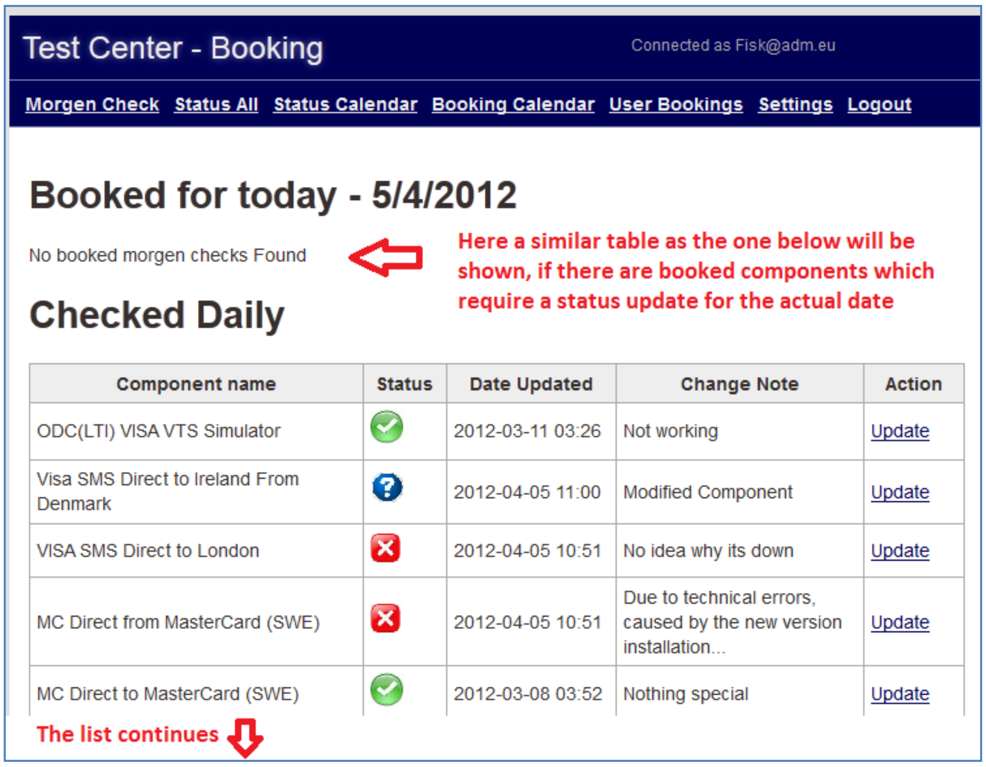
Password: *

Admin Login

User Login

Created by Test Consultancy - NETS

6.4.12 Morgen Check (Admin site)



Test Center - Booking

Connected as Fisk@adm.eu

Morgen Check Status All Status Calendar Booking Calendar User Bookings Settings Logout

Booked for today - 5/4/2012

No booked morgen checks Found

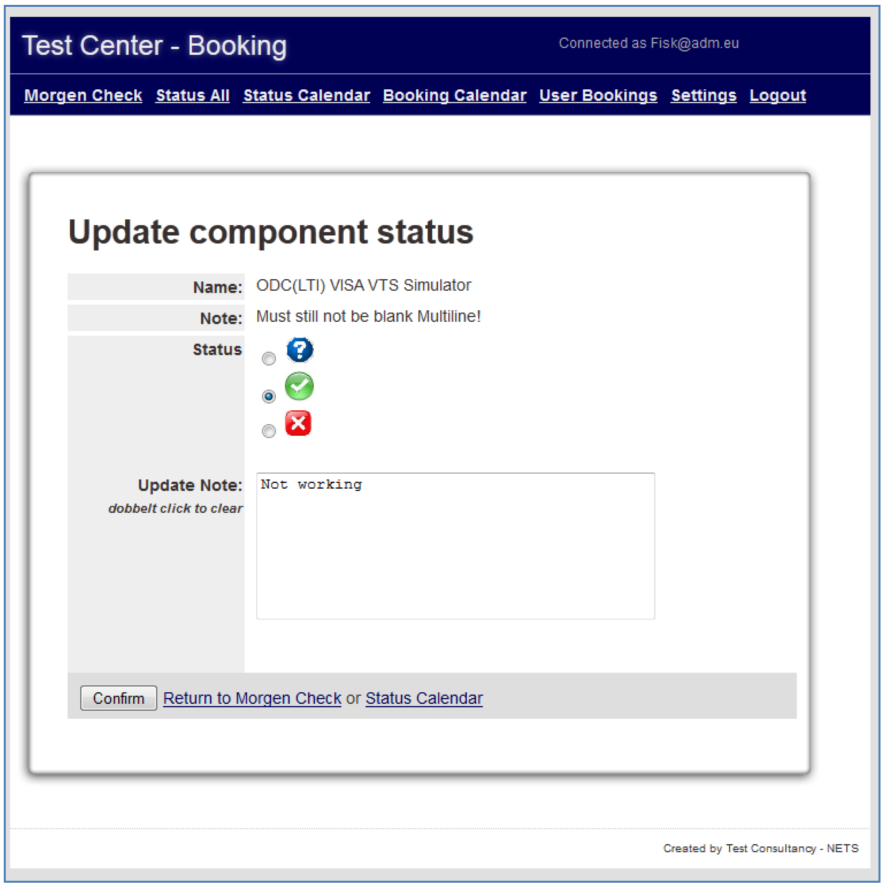
Here a similar table as the one below will be shown, if there are booked components which require a status update for the actual date

Checked Daily

Component name	Status	Date Updated	Change Note	Action
ODC(LTI) VISA VTS Simulator	✓	2012-03-11 03:26	Not working	Update
Visa SMS Direct to Ireland From Denmark	?	2012-04-05 11:00	Modified Component	Update
VISA SMS Direct to London	✗	2012-04-05 10:51	No idea why its down	Update
MC Direct from MasterCard (SWE)	✗	2012-04-05 10:51	Due to technical errors, caused by the new version installation...	Update
MC Direct to MasterCard (SWE)	✓	2012-03-08 03:52	Nothing special	Update

The list continues

6.4.13 Status Update (Admin site)



Test Center - Booking

Connected as Fisk@adm.eu

Morgen Check Status All Status Calendar Booking Calendar User Bookings Settings Logout

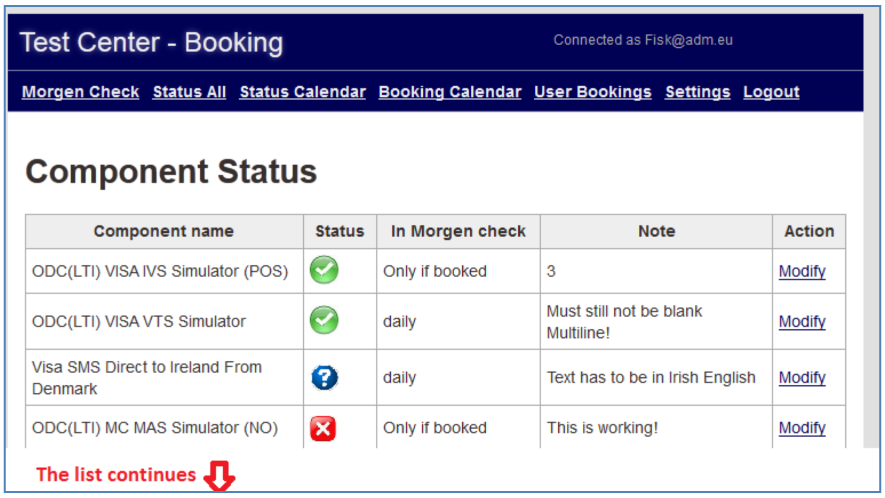
Update component status

Name:	ODC(LTI) VISA VTS Simulator
Note:	Must still not be blank Multiline!
Status	○ ? [✓ ● ✓] ○ ✗
Update Note: *dobbelt click to clear*	Not working

Confirm Return to Morgen Check or Status Calendar

Created by Test Consultancy - NETS

6.4.14 Status All (Admin site)



Test Center - Booking

Connected as Fisk@adm.eu

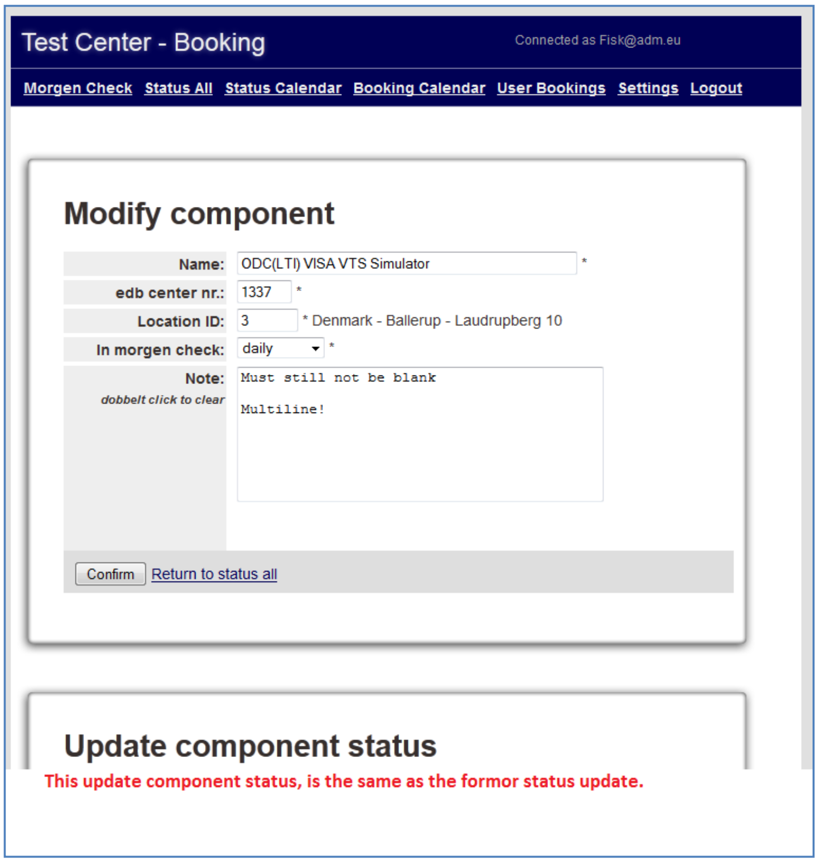
Morgen Check Status All Status Calendar Booking Calendar User Bookings Settings Logout

Component Status

Component name	Status	In Morgen check	Note	Action
ODC(LTI) VISA IVS Simulator (POS)	✓	Only if booked	3	Modify
ODC(LTI) VISA VTS Simulator	✓	daily	Must still not be blank Multiline!	Modify
Visa SMS Direct to Ireland From Denmark	?	daily	Text has to be in Irish English	Modify
ODC(LTI) MC MAS Simulator (NO)	✗	Only if booked	This is working!	Modify

The list continues ⇩

6.4.15 Modify Component (Admin site)


Test Center - Booking
Connected as Fisk@adm.eu
Morgen Check Status All Status Calendar Booking Calendar User Bookings Settings Logout

Modify component

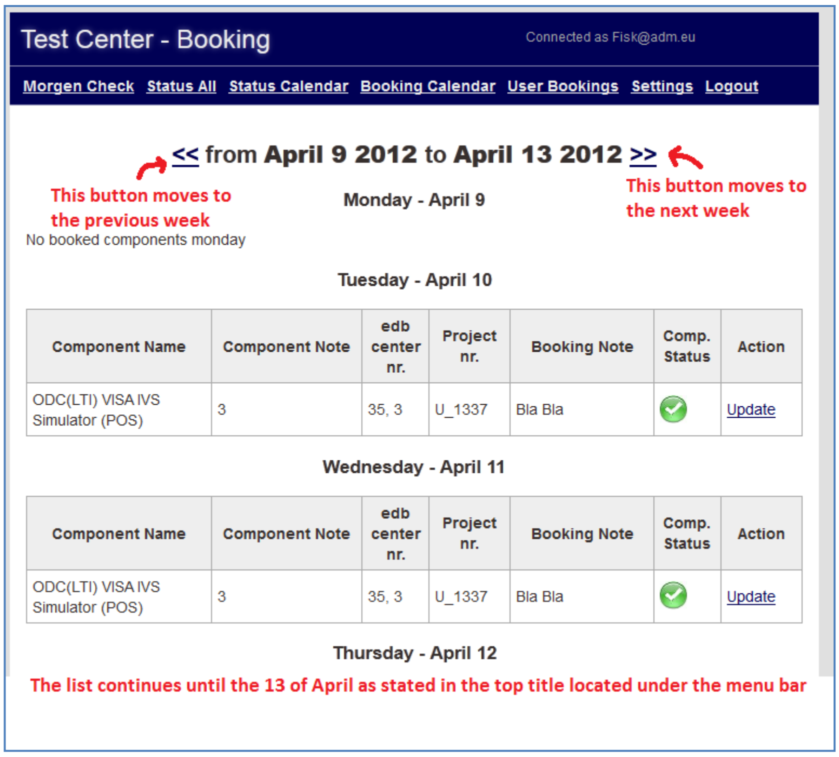
Name:	ODC(LTI) VISA VTS Simulator *
edb center nr.:	1337 *
Location ID:	3 * Denmark - Ballerup - Laudrupberg 10
In morgen check:	daily *
Note: *dobbelt click to clear*	Must still not be blank Multiline!

Confirm
Return to status all

Update component status

This update component status, is the same as the formor status update.

6.4.16 Status Calendar (Admin site)


Test Center - Booking
Connected as Fisk@adm.eu
Morgen Check Status All Status Calendar Booking Calendar User Bookings Settings Logout

<< from April 9 2012 to April 13 2012 >>

This button moves to the previous week

This button moves to the next week

Monday - April 9

No booked components monday

Tuesday - April 10

Component Name	Component Note	edb center nr.	Project nr.	Booking Note	Comp. Status	Action
ODC(LTI) VISA IVS Simulator (POS)	3	35, 3	U_1337	Bla Bla	✓	Update

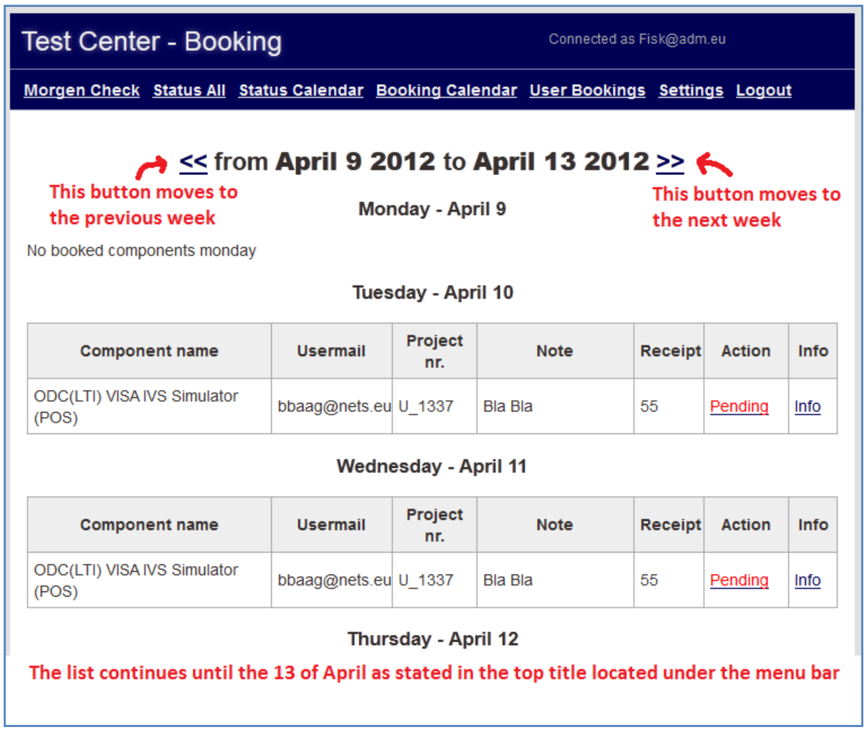
Wednesday - April 11

Component Name	Component Note	edb center nr.	Project nr.	Booking Note	Comp. Status	Action
ODC(LTI) VISA IVS Simulator (POS)	3	35, 3	U_1337	Bla Bla	✓	Update

Thursday - April 12

The list continues until the 13 of April as stated in the top title located under the menu bar

6.4.17 Booking Calendar (Admin site)



<< from April 9 2012 to April 13 2012 >>

Monday - April 9

No booked components monday

Tuesday - April 10

Component name	Usermail	Project nr.	Note	Receipt	Action	Info
ODC(LTI) VISA IVS Simulator (POS)	bbaag@nets.eu	U_1337	Bla Bla	55	Pending	Info

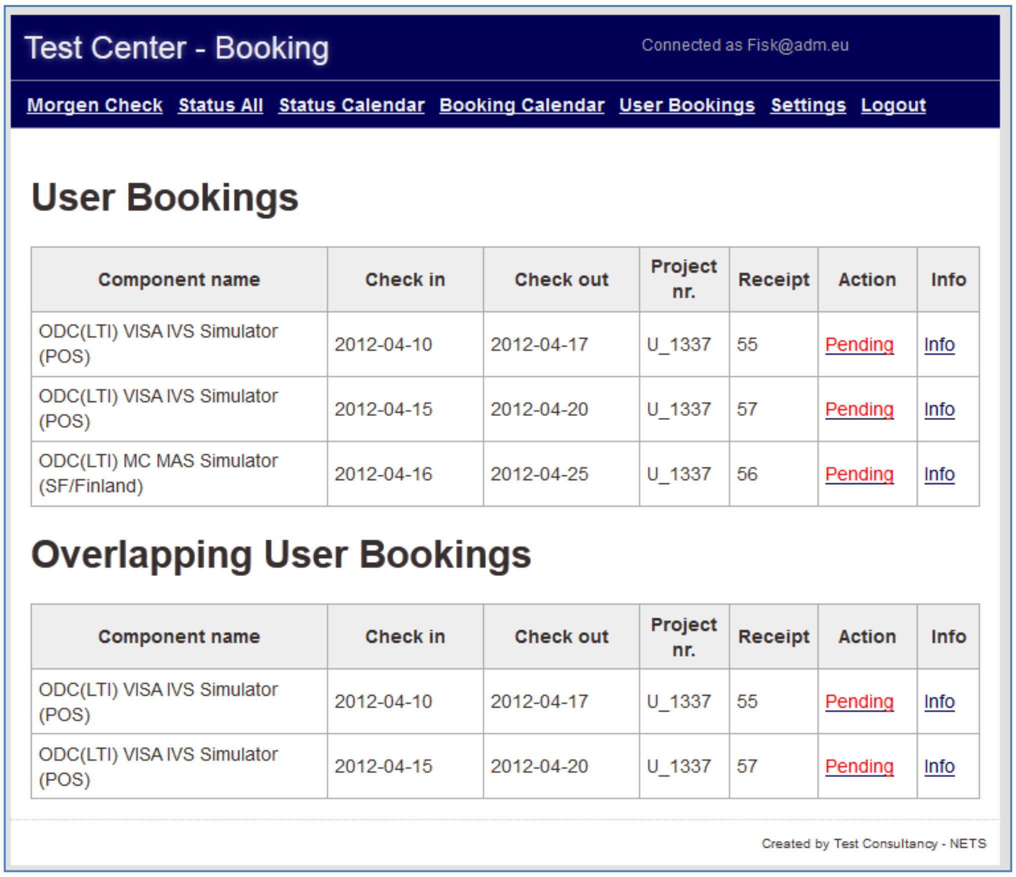
Wednesday - April 11

Component name	Usermail	Project nr.	Note	Receipt	Action	Info
ODC(LTI) VISA IVS Simulator (POS)	bbaag@nets.eu	U_1337	Bla Bla	55	Pending	Info

Thursday - April 12

The list continues until the 13 of April as stated in the top title located under the menu bar

6.4.18 User Bookings (Admin site)



User Bookings

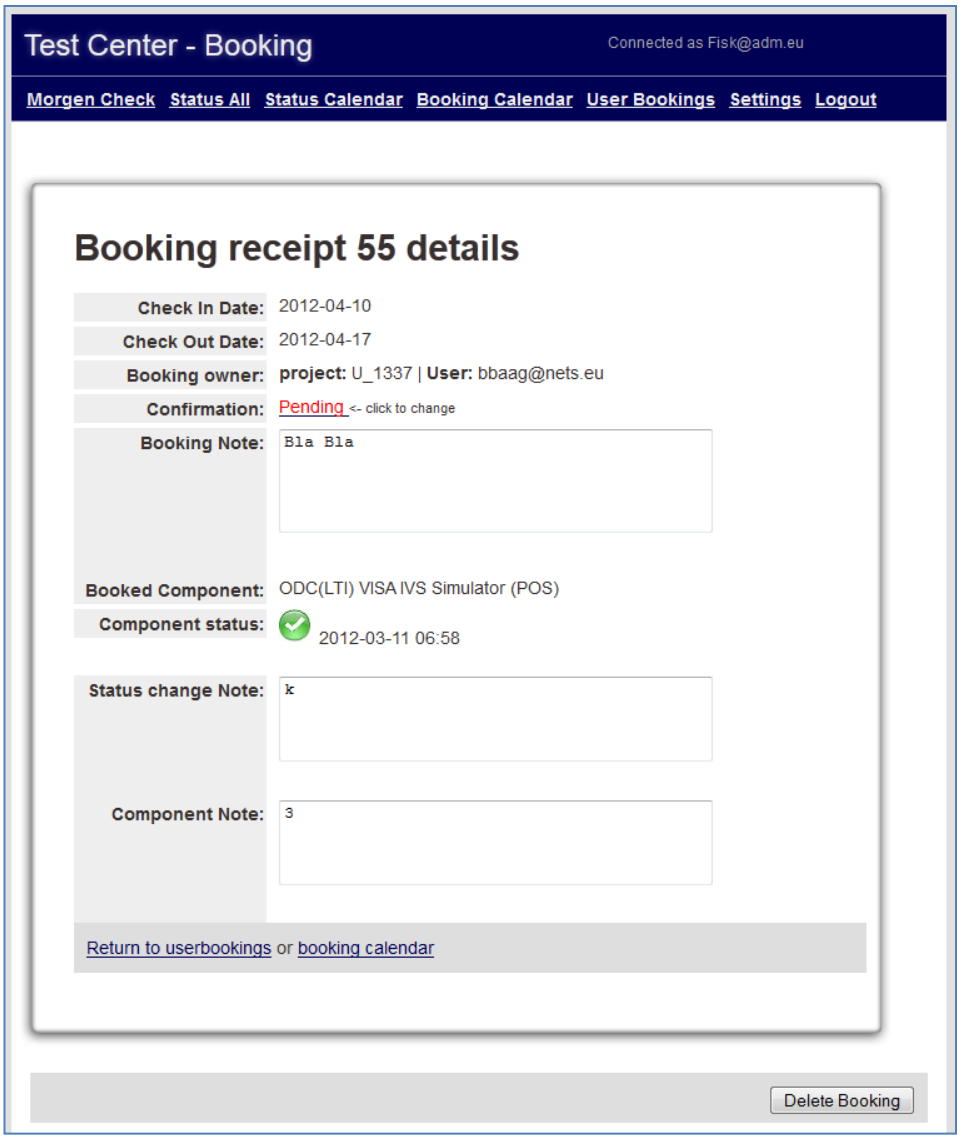
Component name	Check in	Check out	Project nr.	Receipt	Action	Info
ODC(LTI) VISA IVS Simulator (POS)	2012-04-10	2012-04-17	U_1337	55	Pending	Info
ODC(LTI) VISA IVS Simulator (POS)	2012-04-15	2012-04-20	U_1337	57	Pending	Info
ODC(LTI) MC MAS Simulator (SF/Finland)	2012-04-16	2012-04-25	U_1337	56	Pending	Info

Overlapping User Bookings

Component name	Check in	Check out	Project nr.	Receipt	Action	Info
ODC(LTI) VISA IVS Simulator (POS)	2012-04-10	2012-04-17	U_1337	55	Pending	Info
ODC(LTI) VISA IVS Simulator (POS)	2012-04-15	2012-04-20	U_1337	57	Pending	Info

Created by Test Consultancy - NETS

6.4.19 Booking Information (Admin site)

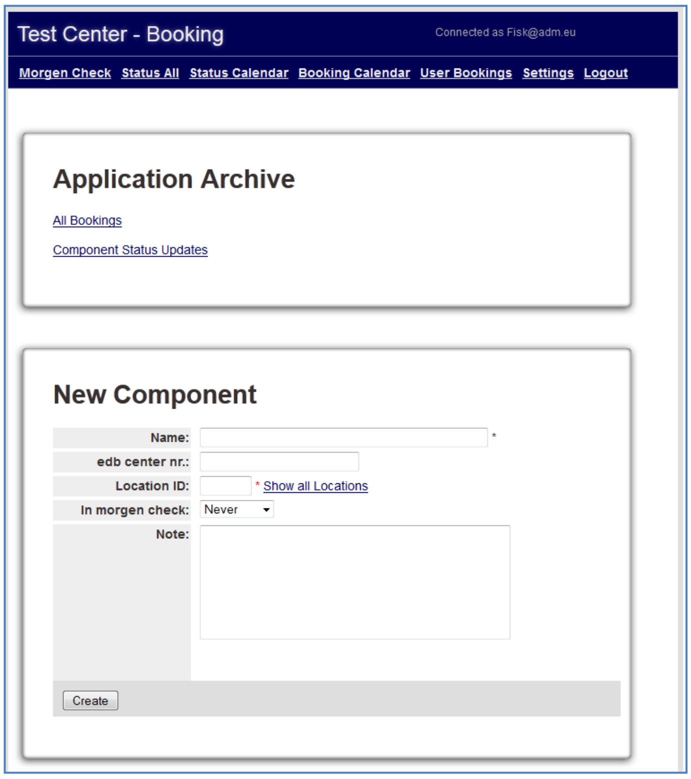


Booking receipt 55 details

Check In Date:	2012-04-10
Check Out Date:	2012-04-17
Booking owner:	**project:** U_1337 | **User:** bbaag@nets.eu
Confirmation:	Pending <- click to change
Booking Note:	Bla Bla
Booked Component:	ODC(LTI) VISA IVS Simulator (POS)
Component status:	✓ 2012-03-11 06:58
Status change Note:	k
Component Note:	3

Return to userbookings or booking calendar

6.4.20 Settings part 1 (Admin site)



Test Center - Booking

Connected as Fisk@adm.eu

Morgen Check Status All Status Calendar Booking Calendar User Bookings Settings Logout

Application Archive

All Bookings

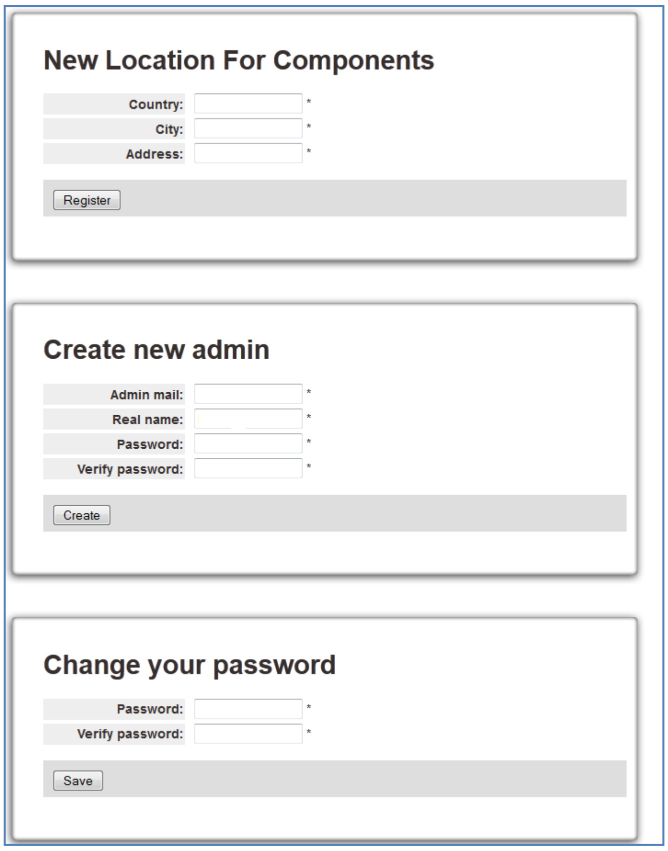
Component Status Updates

New Component

Name:	*
edb center nr.:	
Location ID:	* Show all Locations
In morgen check:	Never
Note:	

Create

6.4.21 Settings part 2 (Admin site)



New Location For Components

Country:	*
City:	*
Address:	*

Register

Create new admin

Admin mail:	*
Real name:	*
Password:	*
Verify password:	*

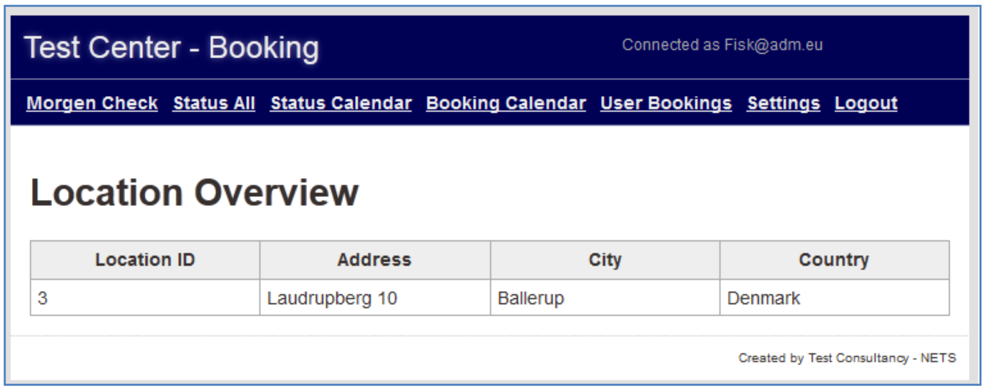
Create

Change your password

Password:	*
Verify password:	*

Save

6.4.22 Component Location List (Admin site)



Test Center - Booking

Connected as Fisk@adm.eu

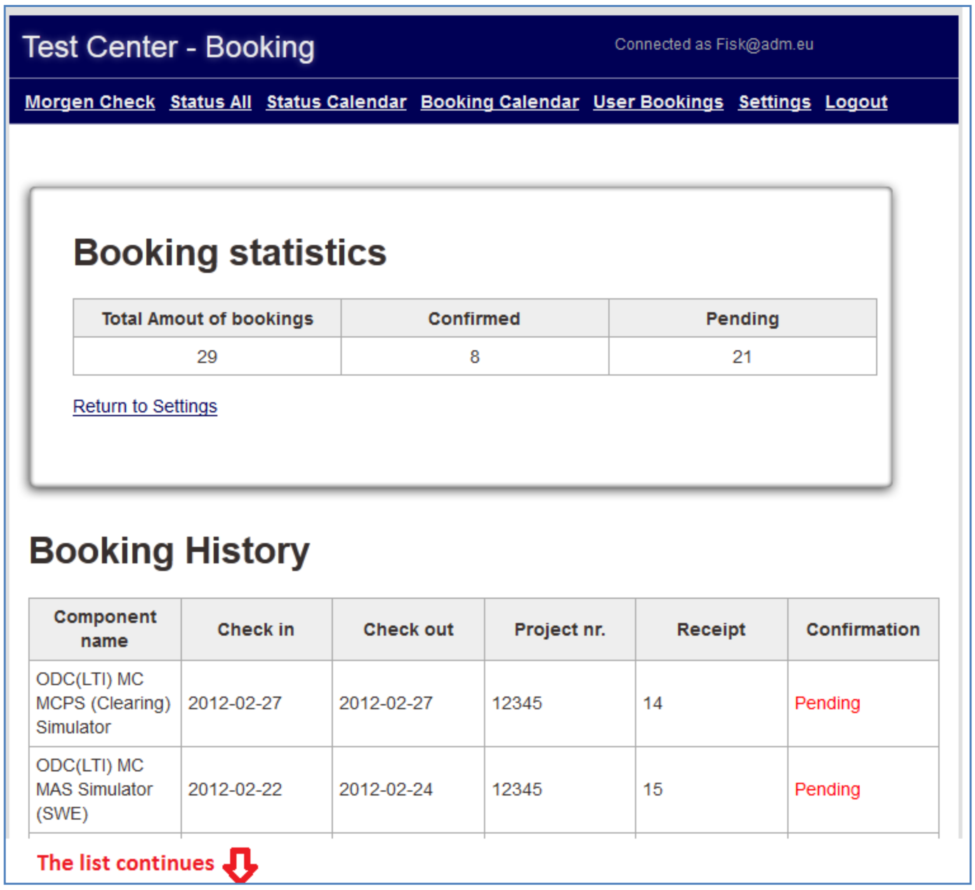
Morgen Check Status All Status Calendar Booking Calendar User Bookings Settings Logout

Location Overview

Location ID	Address	City	Country
3	Laudrupberg 10	Ballerup	Denmark

Created by Test Consultancy - NETS

6.4.23 Booking Statistics (Admin site)



Test Center - Booking

Connected as Fisk@adm.eu

Morgen Check Status All Status Calendar Booking Calendar User Bookings Settings Logout

Booking statistics

Total Amout of bookings	Confirmed	Pending
29	8	21

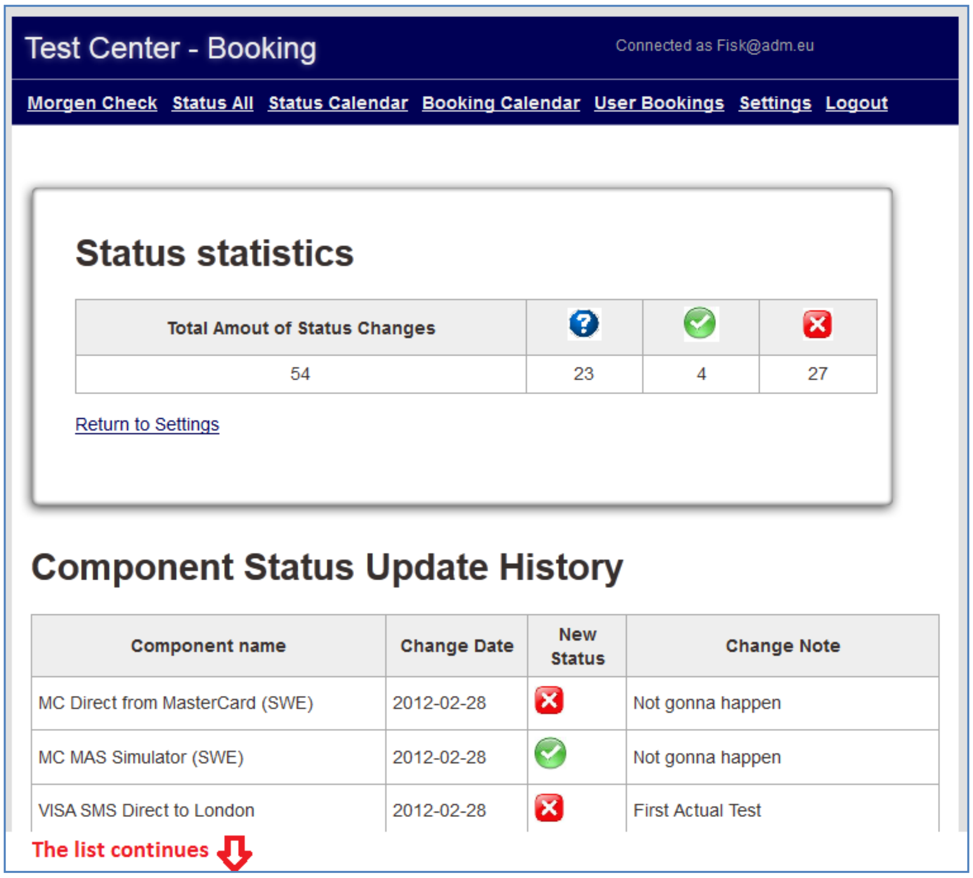
Return to Settings

Booking History

Component name	Check in	Check out	Project nr.	Receipt	Confirmation
ODC(LTI) MC MCPS (Clearing) Simulator	2012-02-27	2012-02-27	12345	14	Pending
ODC(LTI) MC MAS Simulator (SWE)	2012-02-22	2012-02-24	12345	15	Pending

The list continues ⇩

6.4.24 Status Statistics (Admin site)



Test Center - Booking
Connected as Fisk@adm.eu
Morgen Check Status All Status Calendar Booking Calendar User Bookings Settings Logout

Status statistics

Total Amout of Status Changes	?	✓	✗
54	23	4	27

Return to Settings

Component Status Update History

Component name	Change Date	New Status	Change Note
MC Direct from MasterCard (SWE)	2012-02-28	✗	Not gonna happen
MC MAS Simulator (SWE)	2012-02-28	✓	Not gonna happen
VISA SMS Direct to London	2012-02-28	✗	First Actual Test

The list continues ⇩

6.5 Location of Application Code

On the CD delivered with the report there is a folder called Play which contains the Play framework and the applications code. The code can be found in the folder TCB.

- The CSS Style sheet can be found in the following folder: TCB -> public -> stylesheets -> *main.css*

- The HTML views can be found in the following folder: TCB -> app -> views
  Here the *main.html* contains the general design of the screens, described in chapter 2.1
  The folders located at this path contain:
  Application contains the design of the login screens
  Adminsite contains the design of the admin site screens
  Components contains the design of the user site screens

- The Java functions can be found in the following folder: TCB -> app -> controllers
  Here the functions are divided among the three java files:
  Application contain the functions used on the long in screens of the application
  Adminsite contain the functions used on the admin site of the application
  Components contain the functions used on the user site of the application

- The setup of database tables used by the Play Framework can be found in the following folder:
  TCB -> app -> models