



# **Let's Help Bo Design Description**

**Version 1.4**



Let's Help Bo	Version: 1.4
Design Description	Date: 2013-01-18

### Revision History

<b>Date</b>	<b>Version</b>	<b>Description</b>	<b>Author</b>
2012-11-07	0.1	Initial Draft	Niklas Gillström
2012-11-08	1.0	First version	All group members
2012-11-24	1.1	Updated class diagrams, database schema and figure 4-9. Corrected spelling errors.	Nives Bučić, Danijel Jambrečina
2012-12-27	1.2	Updated class diagrams, database schema and GUI screenshots.	Nives Bučić, Danijel Jambrečina
2013-01-14	1.3	Updated class diagrams	Nives Bučić, Danijel Jambrečina
2013-01-18	1.4	Final review	Antonio Gallucci, Rasul Niyazimbetov

Let's Help Bo	Version: 1.4
Design Description	Date: 2013-01-18

## Table of Contents

1.	Introduction	5
1.1	Purpose of this document	5
1.2	Intended Audience	5
1.3	Scope	5
1.4	Acronyms and abbreviations	5
1.5	References	6
2.	Software architecture	6
2.1	Conceptual design	6
2.1.1	User management component	7
2.1.2	Inventory component	7
2.1.3	Map component	7
2.1.4	Work order component	7
2.2	Architecture	8
2.3	System specification	8
2.4	External Components	9
3.	External interfaces	9
3.1	Hardware Interfaces	9
3.2	Software Interfaces	9
3.3	Communication Interfaces	10
3.4	User Interfaces	10
4.	Detailed software design	11
4.1	Implementation modules / components	11
4.1.1	Inventory component class diagram	12
4.1.2	Map component class diagram	13
4.1.3	Work order component class diagram	14
4.1.4	User component class diagram	15
4.2	Data Types / Formats	15
4.3	Database Model	16
4.4	Web site organization	17
4.5	Interfaces to External Systems	19
4.6	Algorithms	19

Let's Help Bo	Version: 1.4
Design Description	Date: 2013-01-18

## 1. Introduction

### 1.1 Purpose of this document

The purpose of this document is to provide information about Let's Help Bo project design. The goal of this project is to create software that supports the machine operators in their working activities. In addition, it provides other functionalities that aim to help in the daily activities of other people that work in the mine during their daily activities. The main actor of the system is the machine operator, but there are also other actors that will have a part in the system. LHB has been developed taking into account the environment in which it will be used and made it as usable as possible for the final users. The system is available for different platforms: smartphones, laptops, desktops and tablets.

### 1.2 Intended Audience

The intended audience for this document is:

- Team members
- Supervisor
- Customers

### 1.3 Scope

The architecture described is a server client architecture, where all components are on the server side and the output from the components is displayed to users that can interact with the system from the client side. This document covers the following design parts used in the project:

- Software architecture
- Identified components
- External interfaces
- Software interfaces
- Class diagrams
- Database model
- Applications organization
- Algorithms

### 1.4 Acronyms and abbreviations

Acronym or abbreviation	Definitions
LHB	Let's Help Bo
DSD	Distributed Software Development
UM	User management component
WC	Work order component
MC	Map component
IC	Inventory component
SDK	Software Development Kit
RDBMS	Relational Database Management System
GUI	Graphic user interface
MVC	Model View Controller

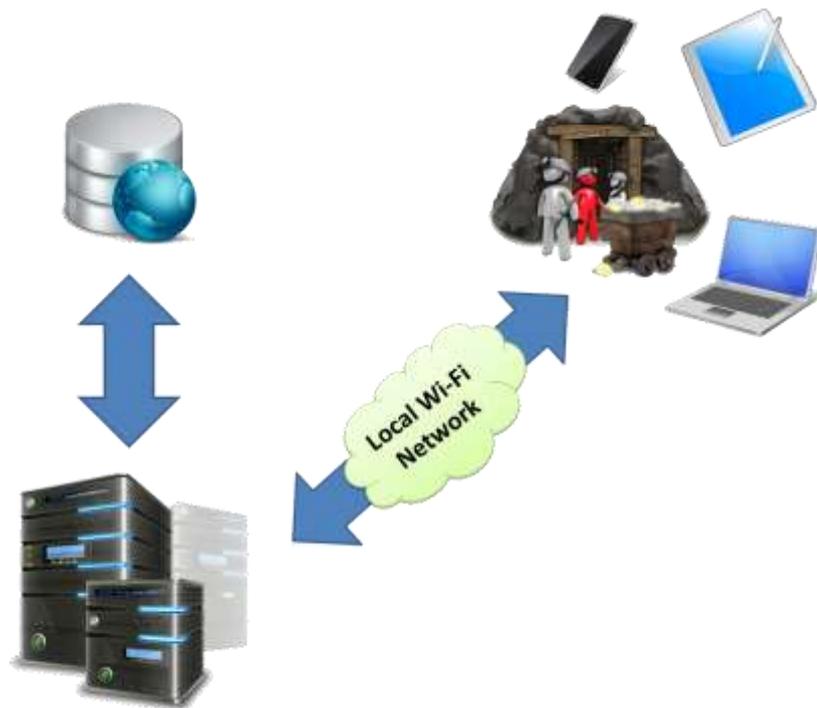
Let's Help Bo	Version: 1.4
Design Description	Date: 2013-01-18

## 1.5 References

- Project DSD home page  
[http://www.fer.unizg.hr/rasip/dsd/projects/lets\\_help\\_bo](http://www.fer.unizg.hr/rasip/dsd/projects/lets_help_bo)

## 2. Software architecture

### 2.1 Conceptual design



**Figure 2-1 Conceptual design**

The overall system is based on a central server running the main software, local database and client applications connected to the central server by an internal wireless network. Client applications are represented with mobile and web interfaces.

Let's Help Bo	Version: 1.4
Design Description	Date: 2013-01-18

The system is built out of several components which interact among each other.

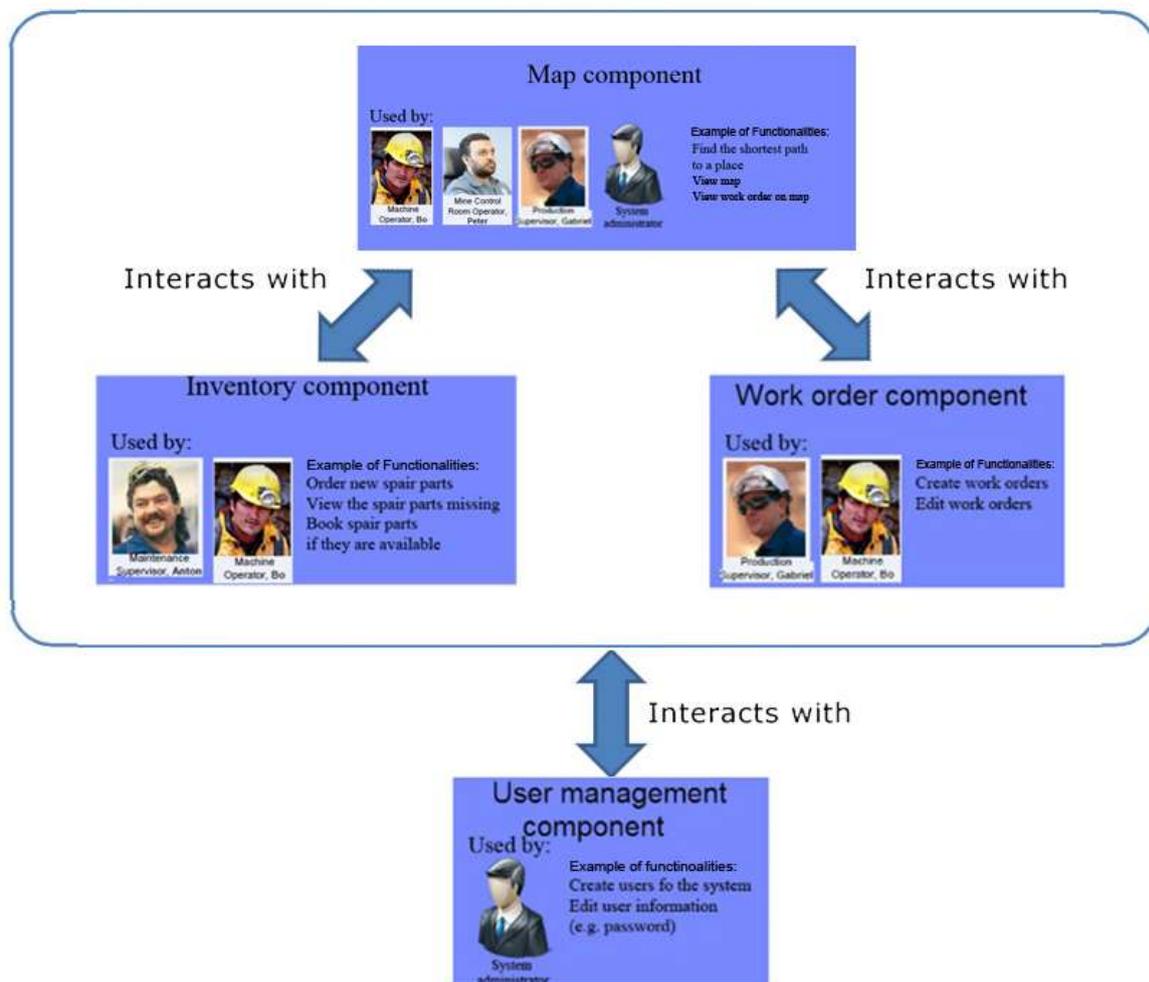


Figure 2-2 System components

### 2.1.1 User management component

This component deals with user administration which includes creating user accounts, editing their information or deleting them. These operations are managed by the system administrator.

### 2.1.2 Inventory component

This is the main component which involves booking of the spare parts by the machine operators. It offers order status and availability information. It is responsible for sending notifications regarding order completion and inventory status. Also, administration of warehouse composed of spare parts ordering and checking the inventory for missing parts. This is managed by the maintenance supervisor.

### 2.1.3 Map component

Hold the map of the mine with warehouse, users, vehicles, rooms and work order locations, tunnel paths and tunnel statuses. It provides the shortest path functionality that is used to show the shortest path to the user from his position to the warehouse or work order place.

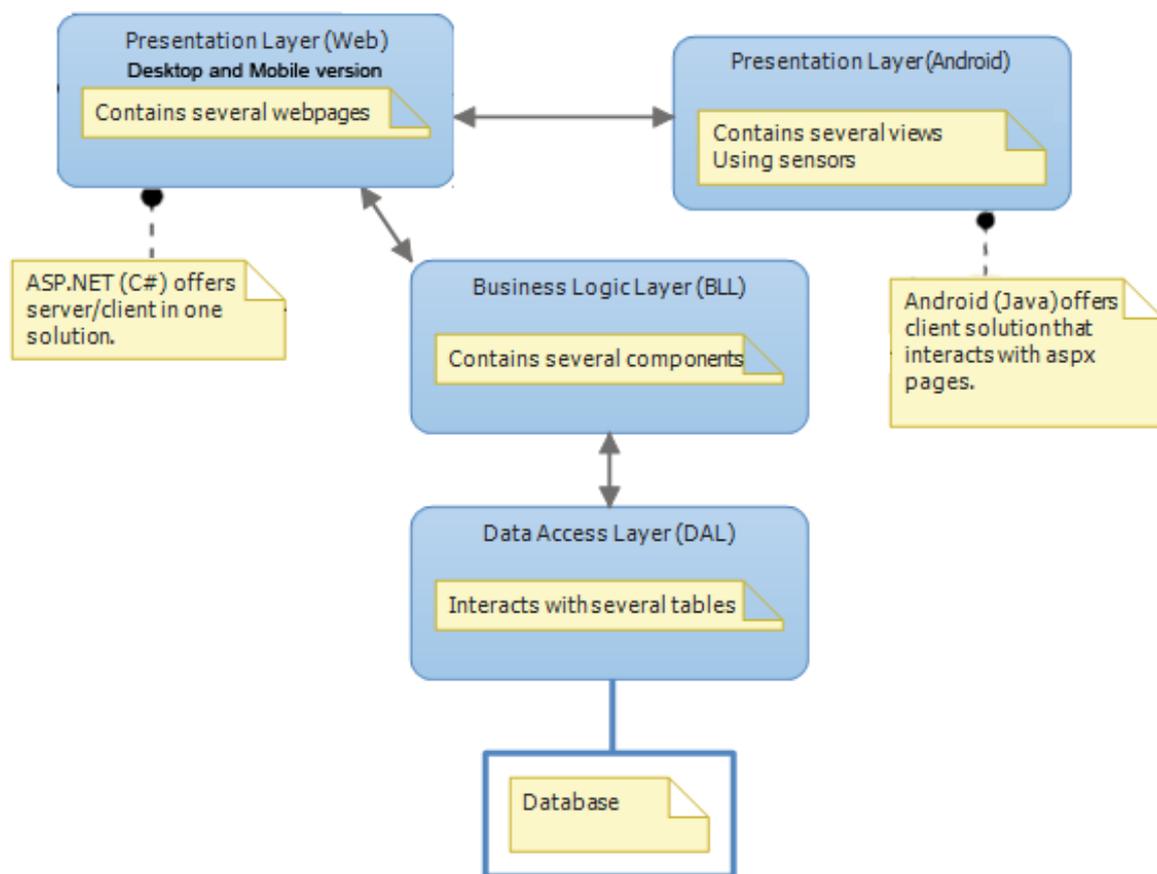
### 2.1.4 Work order component

Responsible for create and edit work orders. A work order is an action that needs to be performed and it is assigned to a machine operator.

Let's Help Bo	Version: 1.4
Design Description	Date: 2013-01-18

## 2.2 Architecture

The system architecture is based on three tiers along with a database.



**Figure 2-3 System architecture**

The data access layer offers an interface between the database and the business logic. The business logic layer contains the functionalities of the central system. The presentation layer contains the client applications which provide the user interface for the system.

## 2.3 System specification

The system follows client-server architecture with a mobile device (tablet, smartphone) a small laptop or a desktop computer playing the role of the client, and a central server that handles all the incoming requests from the devices connected to the system. During the development phase, the central server role will be fulfilled by a virtual machine running on Microsoft Windows Server 2008 R2-Service Pack 1. This would be also the target configuration for running on real servers in production mode, after shipping the product.

### Backend

The Microsoft SQL Server 2008 Express database management system will be used for the database, and the Logic Layer that interacts with the database will be implemented in C# using the .NET 4.0 Framework.

Let's Help Bo	Version: 1.4
Design Description	Date: 2013-01-18

## Frontend

As specified there are two presentation layers, one for the Web interface and another for mobile devices. The Web interface will be developed using ASP.NET, HTML5/CSS3 and JavaScript. For the mobile devices, there isn't a specific device needed. The project will experiment with Android devices mainly, but the presentation layer will be developed using HTML5, CSS3 and JavaScript. Therefore, it is expected to work across all types of smartphones that have browser applications installed.

## 2.4 External Components

Microsoft® SQL Server® 2008 Express is a RDBMS developed by Microsoft, and is a free edition of their family of commercial solutions SQL Server®. The Express version is a good and practical solution for building desktop and small server applications, and will fit very well in the project since the other tiers of the system architecture will be also using Microsoft® technologies. Being a free product, it'll help keeping the costs of this project fairly low.

Regarding the previous assertions, the .NET Framework is also a major block in this project. It encloses a large number of libraries ranging from low-level to high-level functionality, and the vast amount of code functionality already implemented in this software framework will spare the developers from unnecessarily reinventing the wheel and be a good stepping stone for this project.

## 3. External interfaces

There exists one external system that the system will use:

- Global Positioning System (GPS)

The system will use GPS to get one or more reference points when Bo is outside the mine. The reference points are used to get GPS coordinates in the mine where the GPS normally doesn't work. The route between locations can then be displayed to the user.

### 3.1 Hardware Interfaces



Figure 3-1

Our application will use the built-in GPS to get a reference point so the map component can work when Bo is inside the mine (GPS does not normally work inside the mine). The GPS will be used once outside the mine and after that, everything else will be calculated.

### 3.2 Software Interfaces

In order to provide support for most of the existing smart phones and tablets in the current market, the application for Mobile platform also will be developed as a web application. It will have an external interface of a native application. Nevertheless the system still will be possible to access from web browser of smartphones.

Let's Help Bo	Version: 1.4
Design Description	Date: 2013-01-18

We will design the web pages with the proper characteristics of native applications for smartphones. The client-side application embeds some web pages using Web View in application layout.

### 3.3 Communication Interfaces

Our web application will communicate with the database through ASP.NET middleware. Using ASP.NET we can dynamically create the web pages, fill them with the information from the database (using scripts) and present them to the user in the graphical user interface.

Since all of our application logic and data layers are on the server side it's integral to have access to either the internal Wi-Fi network or to the 3G network (outside the mine).

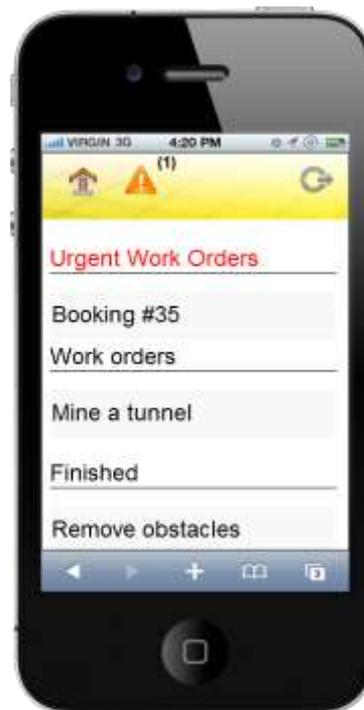
### 3.4 User Interfaces

The interfaces of applications are provided for each platform (Mobile or desktop) individually due to increase of usability. The system and all their components representations are device orientated. The usage of the system is orientated into several user types. All users are separated with authentication access and have different, additional or limited views in GUI.

- Client - has high restricted access and can only edit very limited data. (for instance, on work order component can make check in if work is finished)
- Manager - has high level access (Can add, edit, delete components data).



**Figure 3-2 Main page**



**Figure 3-3 Work order component**

**Figure 3-2** illustrates the main page of Mobile application for clients. Here clients can access all components by clicking the relative buttons.

**Figure 3-3** Illustrates mobile application with work orders component realization for Client users. The navigation between work orders dates can be held by pressing buttons left right on top bar or swiping the screen to left or right. Each Task can be marked as done directly in this page.

Application for Desktop platform also follows the same structure of authentication and organization with components as in Mobile platform.

Let's Help Bo	Version: 1.4
Design Description	Date: 2013-01-18



**Figure 3-4 Booking process**

**Figure 3-4** illustrates the booking process with machine operator access in our web application for Desktop platform. Since, on desktop platforms there is a large space; it gives us possibility to insert more information on one screen compare to mobile platform.

At the header of web application there is a navigation of components. On the left side, there is a menu bar that change for each user role. The menu items list is differentiated by access type in web application (e.g. Administrator has different menu items compare to Machine operator). Log out form and navigation for settings are the same for all type of users and it is located on top right sight of the application.

## 4. Detailed software design

### 4.1 Implementation modules / components

In this chapter the class diagrams for the following components will be presented:

- Inventory component
- Map component
- Work order component
- User component

Let's Help Bo	Version: 1.4
Design Description	Date: 2013-01-18

### 4.1.1 Inventory component class diagram

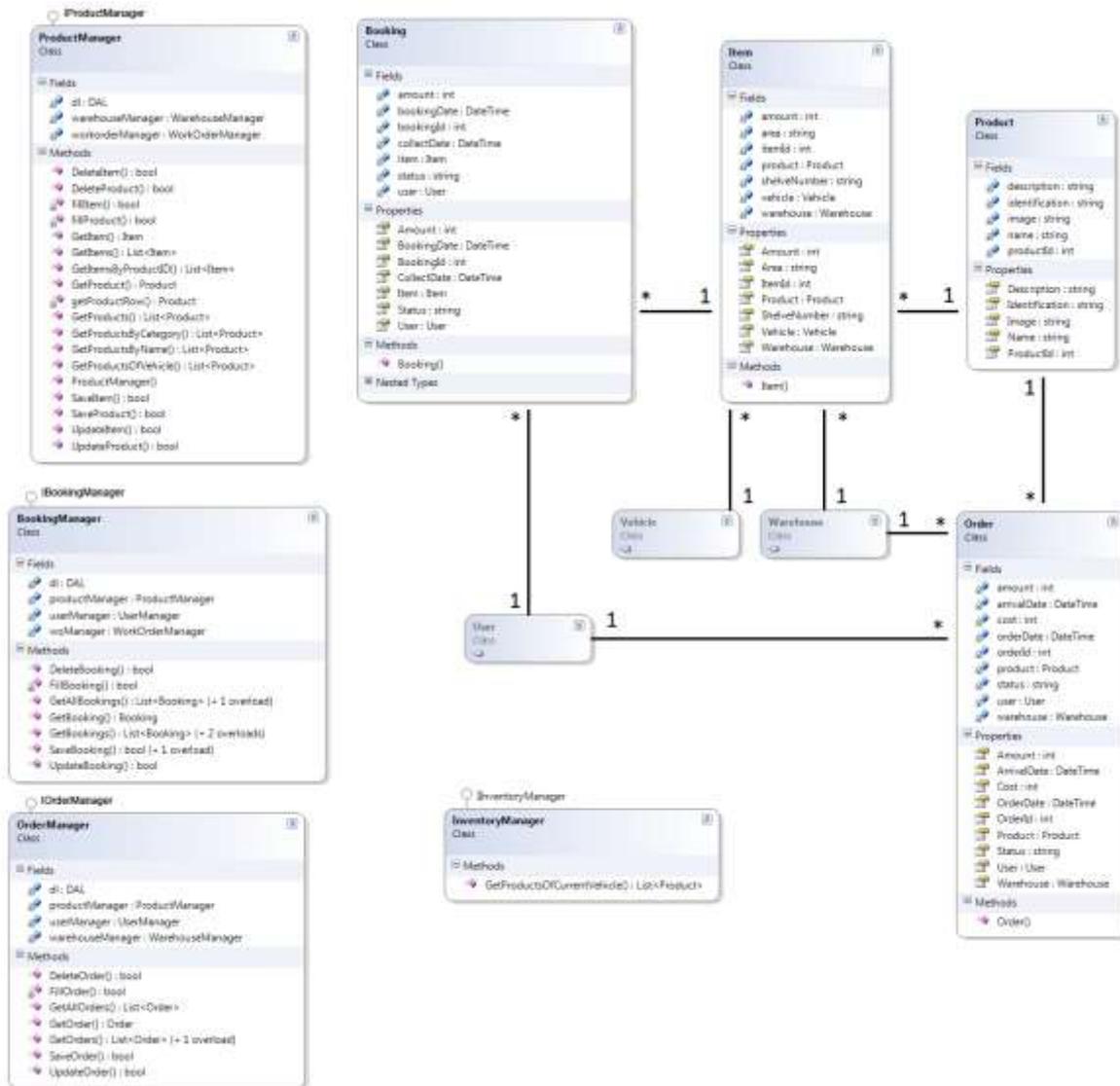


Figure 4-1 Inventory component

In this component there are all the classes to manage the functionalities related to the inventory system. The machine operator can book only one spare part per time, but several items for it.

Let's Help Bo	Version: 1.4
Design Description	Date: 2013-01-18

### 4.1.2 Map component class diagram

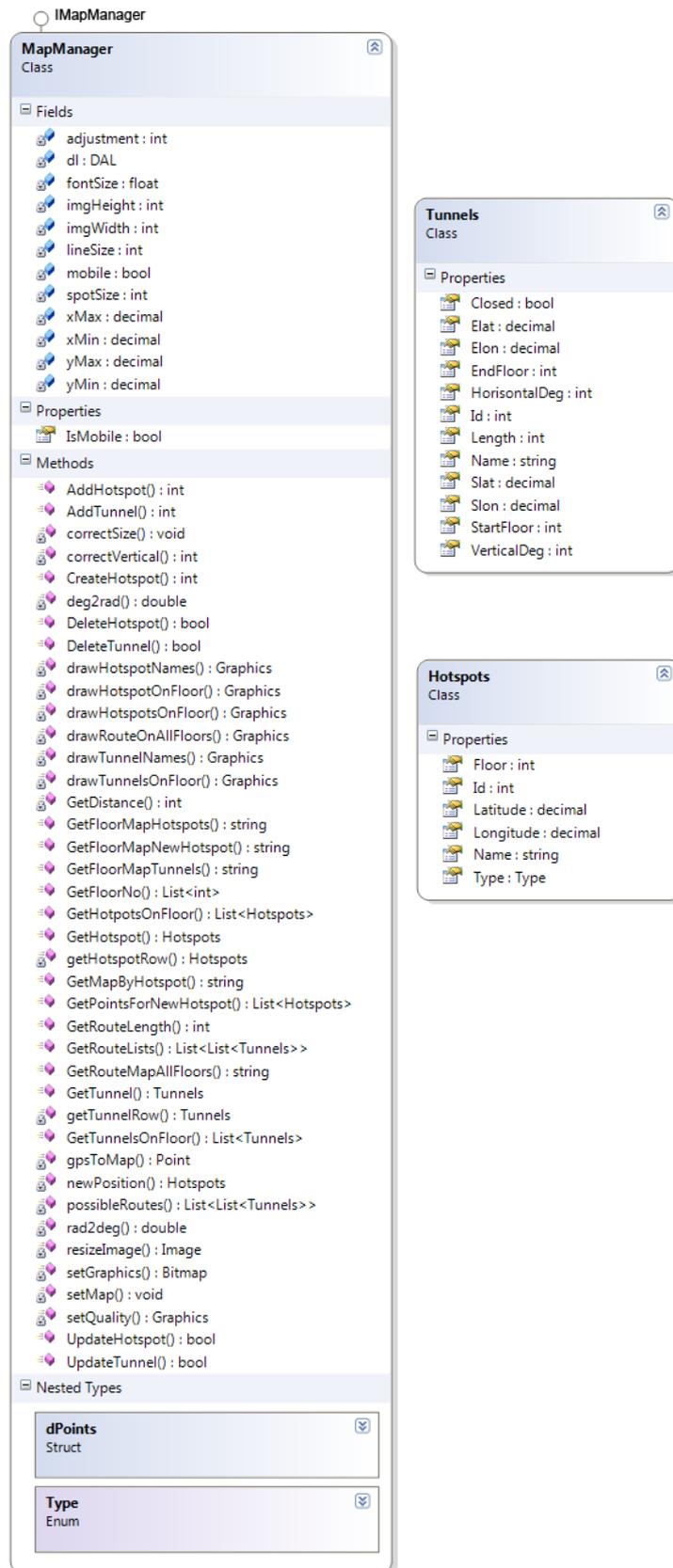


Figure 4-2 Map component

Let's Help Bo	Version: 1.4
Design Description	Date: 2013-01-18

This diagram contains all the classes related to the map component. A map can be seen as a set of tunnels and hotspots. Each hotspot has its coordinate on the map, and it can represent a user, room, vehicle, warehouse, work order or production front.

#### 4.1.3 Work order component class diagram

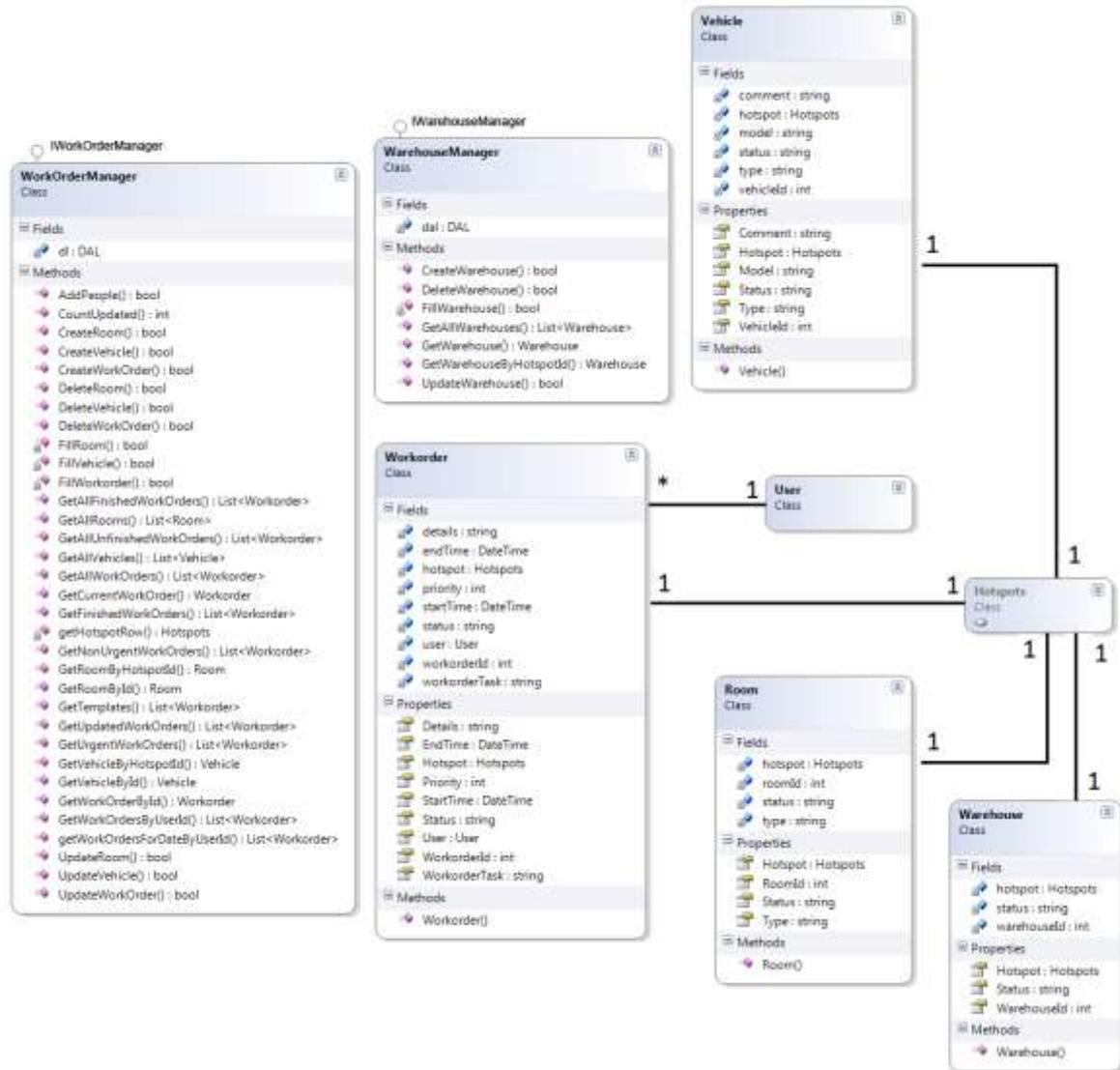
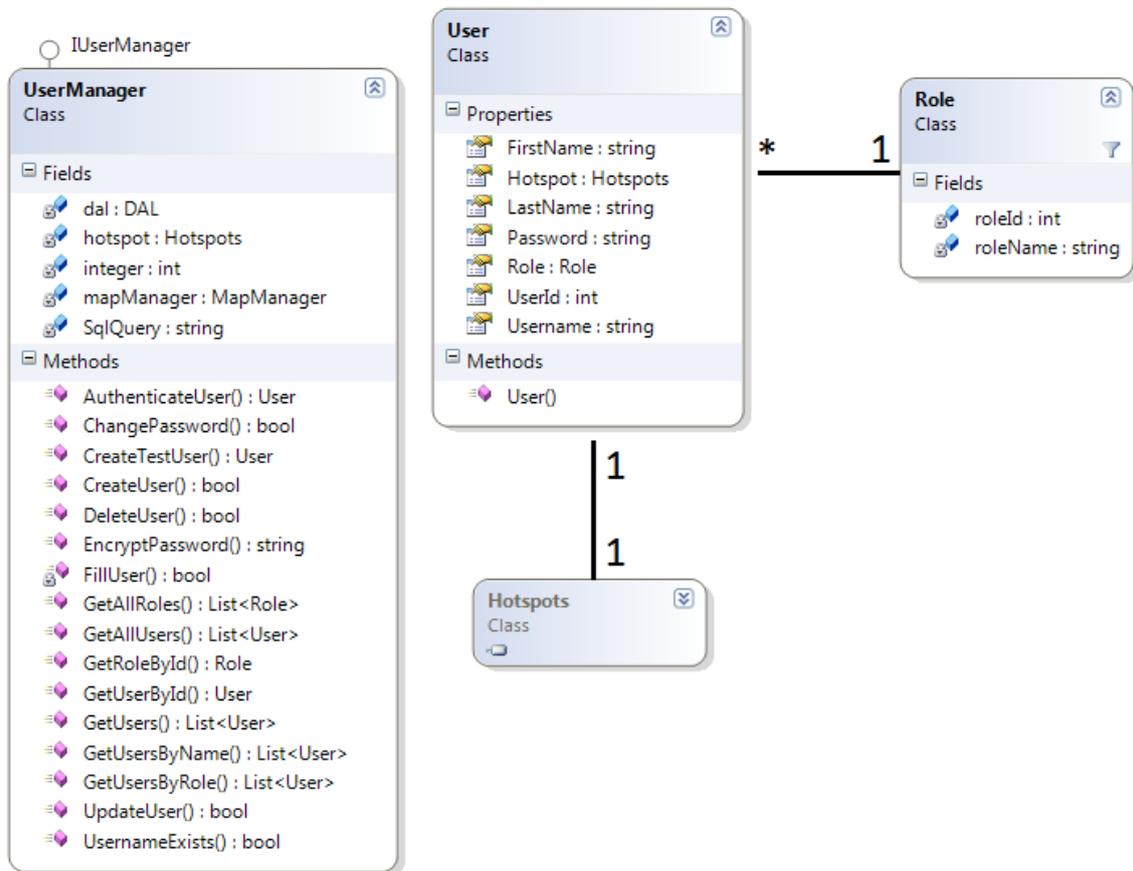


Figure 4-3 Work order component

In the work order component, we have the work order class in which all the data about the work orders is kept. There are also entities that represent the vehicles, rooms and warehouses in the mine. Machine operator can have several work orders, Depending on work order type; work order start and end times can be at the same time or cannot match.

Let's Help Bo	Version: 1.4
Design Description	Date: 2013-01-18

#### 4.1.4 User component class diagram



**Figure 4-4 User component**

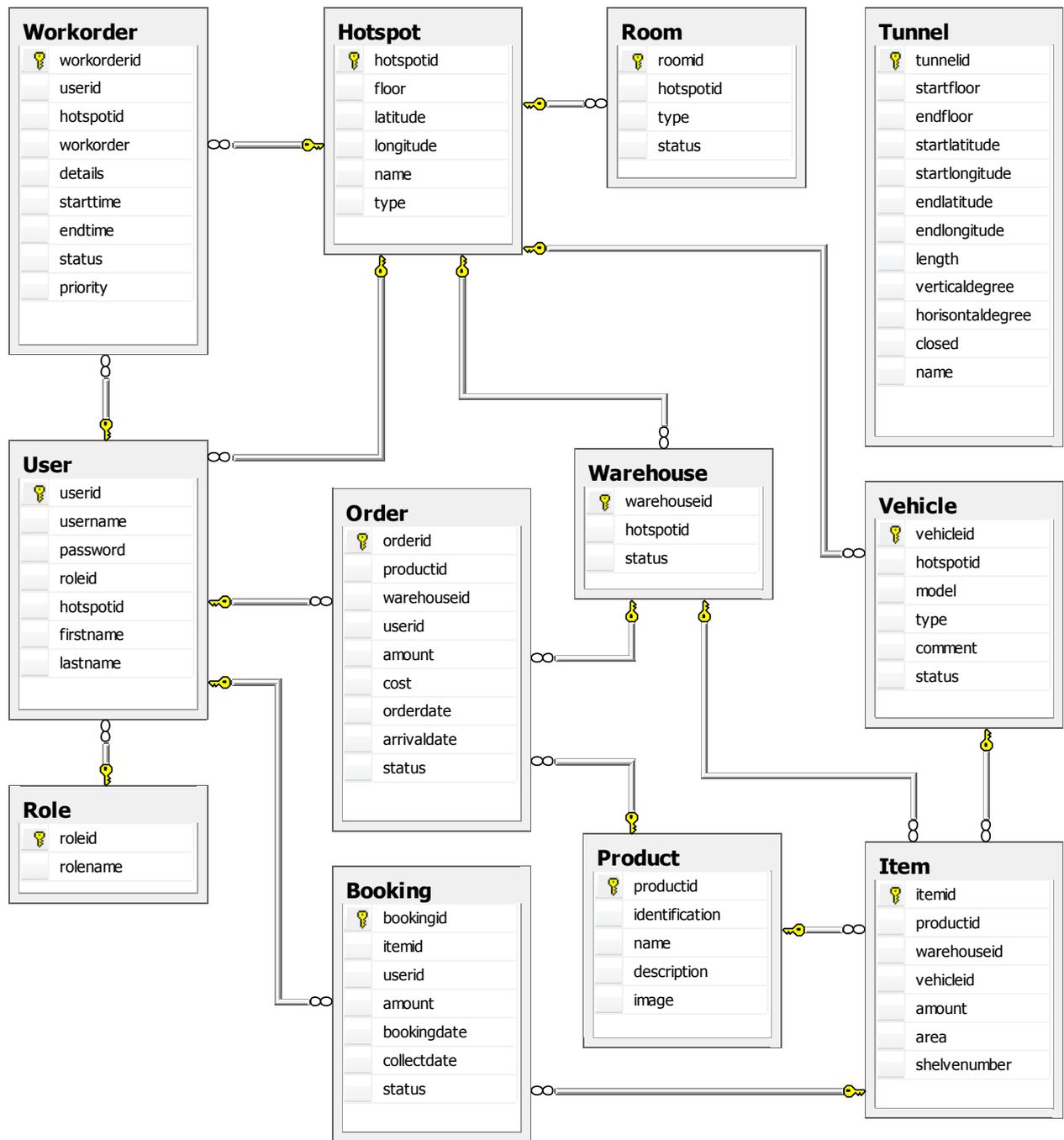
In the user component there is all the information about the users that exist in the system. The users can log in with username and password and each user has a specific role and hotspotid.

#### 4.2 Data Types / Formats

The product contains the following file formats:

1. Document file formats
  - .doc
  - .docx
  - .pdf
  - .ppt
  - .pptx
  
2. Source code/database formats
  - .mdf
  - .cs
  - .aspx
  - .css
  - .js
  - .dll
  - .CONFIG

### 4.3 Database Model



**Figure 4-5 Database scheme**

On **Figure 4-5** is shown the database model that will be used. It is composed of 12 entities which will be transformed into 12 tables. Table Booking stores information about bookings of spare parts - booking id, amount of booked spare parts, booking date and actual collect date, status of booking, user who made booking and item that was booked. Item stores item id, amount of that item that is stored in warehouse which is determined with warehouse id, area and shelf number of warehouse where the items are stored, product id which determines what type is the item and vehicle id which determines to which vehicle item belongs. Table Product stores general information about products such as product id, identification, name, description and image. Table Order has similar attributes like table Booking. The difference is that it also has warehouse id of the warehouse where the products will be stored, product id that shows which product was ordered and the cost of the order. Table Work Order is composed of work order id, information about work order such as work order details, start

Let's Help Bo	Version: 1.4
Design Description	Date: 2013-01-18

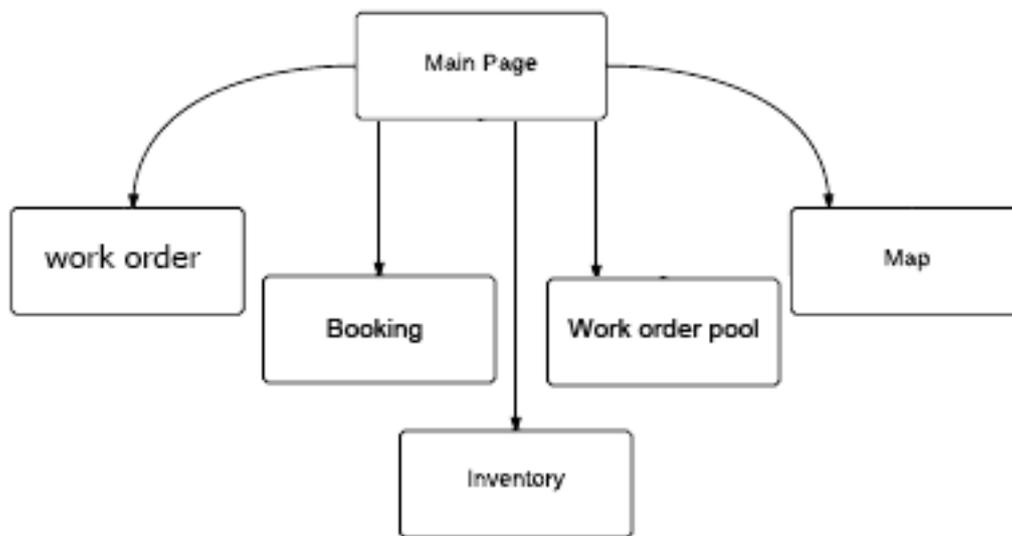
time, end time, status and priority. Besides that, it stores user id of the user who is working on that work order and hotspot id of hotspot on which the user is working. In table User there are user id, user name, password, id of user's role, id of user's hotspot, first name and last name. Table Tunnel stores information about the position, size and status of the tunnel. Hotspot has information about position, name, type and id. Room, Warehouse and Vehicle are variations of hotspots so they have detailed information about it, for example a vehicle has status, type and model.

#### 4.4 Web site organization

Since the usage of the product is divided into two platforms, desktop and mobile, as it was already mentioned, it will be developed as web application with consideration for different platform orientations.

In order to provide security of product there will be authentication check where it separates machine operator and supervisors and gives access for different menu items, tools, etc.

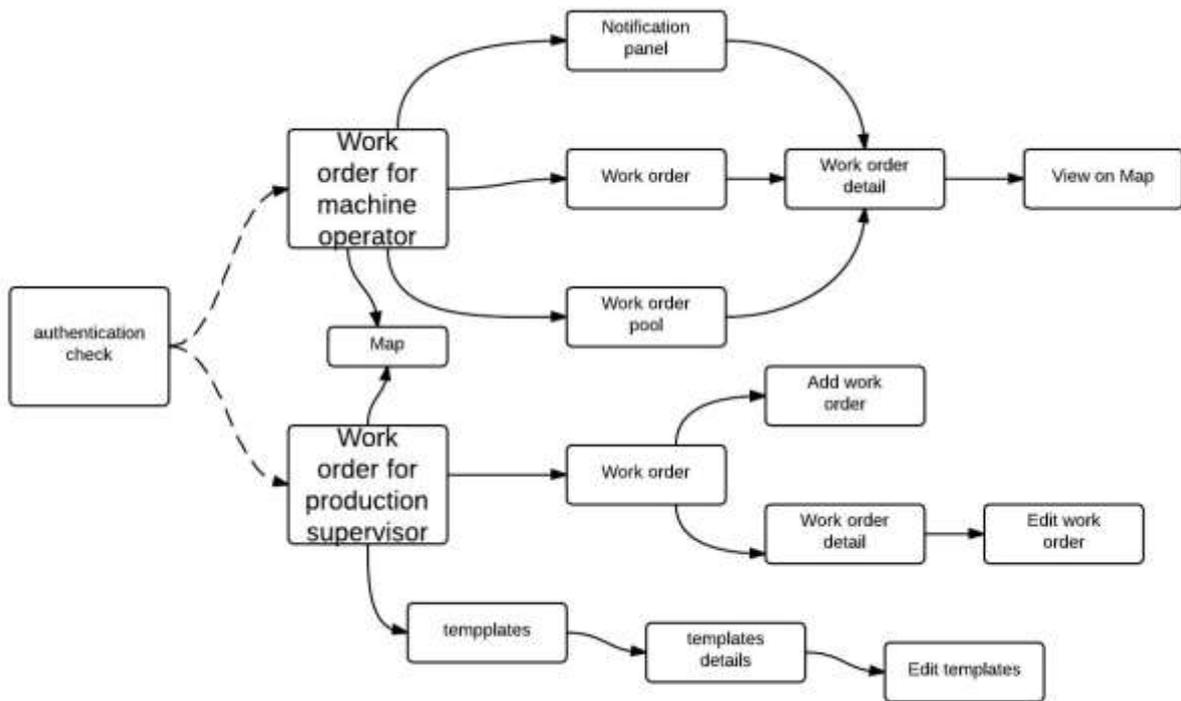
The diagrams below shows what user will see and navigation directions on applications. The interval arrows show authentication script.



**Figure 4-6 Main page structure for machine operator**

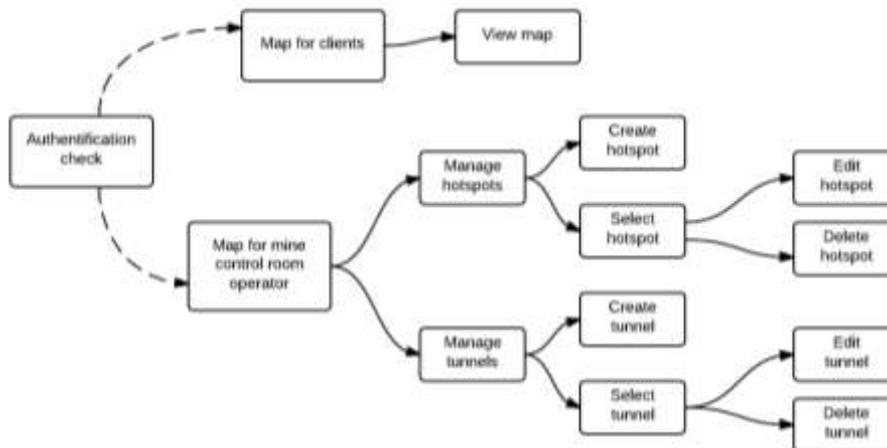
**Figure 4-6** shows simple structure of main page for mobile and desktop platform interface available for the machine operator. It will consist of links to Work orders, Map, Inventory, Booking and Work order pool functionalities. The access for the Main Page is held by authentication for each user. This procedure is held once, until the cookies will exist on platform. After the authentication step each component service can be accessed with providing menu and items for Machine operator and supervisors separately. At the main page user can navigate to one of each component. As an example for separated access we can see below on **Figure 4-7**.

Let's Help Bo	Version: 1.4
Design Description	Date: 2013-01-18



**Figure 4-7 Pages structure for machine operator and production supervisor**

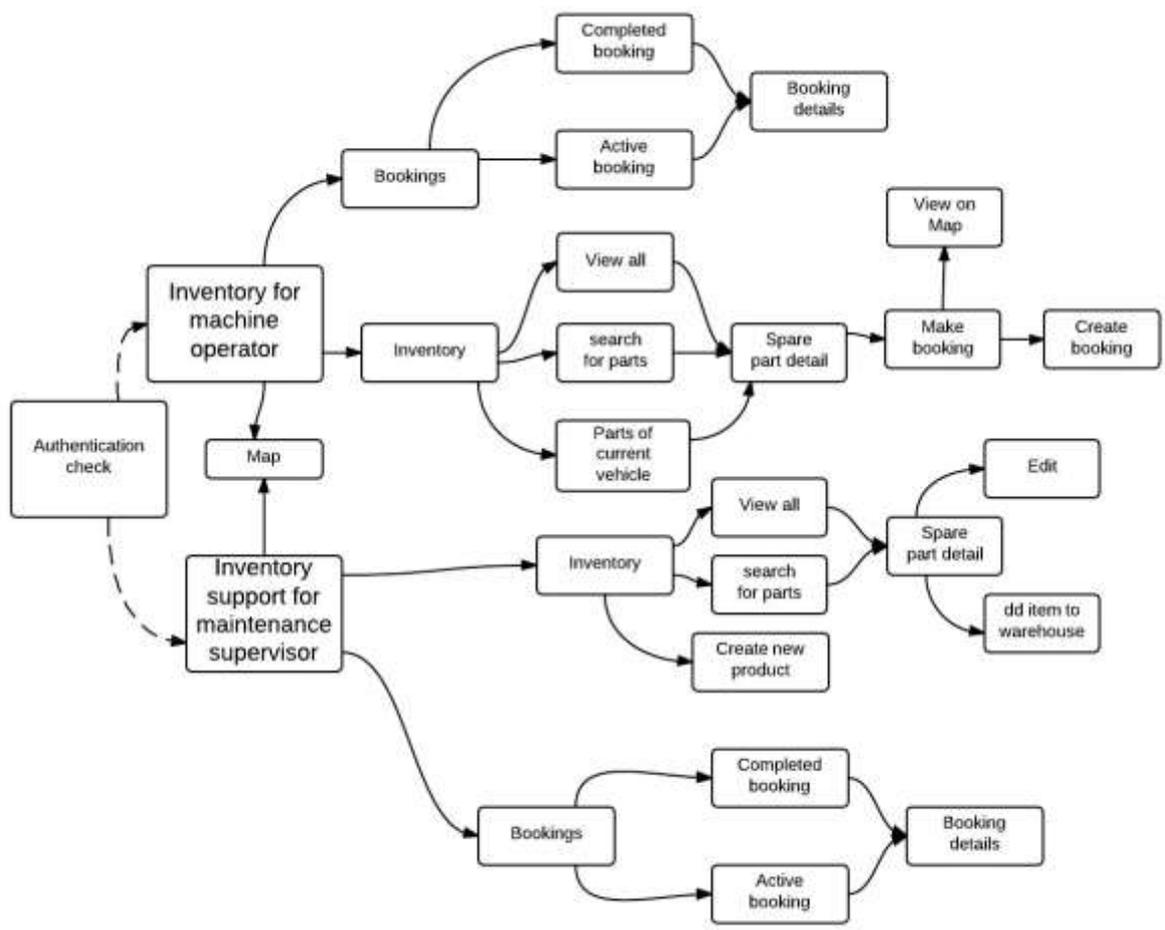
On **Figure 4-7** we can see after authentication checking was held each type of user has different access to the system. Machine operator can only see their work orders. Production supervisor has access for edit, and add work orders pages. As it can be seen the navigation of Machine operator and Production supervisor are different. Both, machine operator and production supervisor has the map component on their pages navigation. They have same access to all provided functionalities; hence this page is used for all users except users with mine control room operator.



**Figure 4-8 Map**

**Figure 4-8** illustrates Map component. All the users except for the mine control room operator can access to the view map functionality. Instead, the mine control room operator can manage the map accessing to the relative functionality.

Let's Help Bo	Version: 1.4
Design Description	Date: 2013-01-18



**Figure 4-9 Inventory support**

**Figure 4-9** illustrates navigation of Inventory support component. Some of the pages are linked to another component (map).

#### 4.5 Interfaces to External Systems

The only interface exposed by this product to external systems is the user interface. The functionality of the interface is the functionality obtained when browsing a web page, but within the user interface all functionality from the components can be retrieved with the proper credentials.

#### 4.6 Algorithms

The algorithm to get all possible routes uses the same technique as depth first algorithm. This algorithm is customized to function in 3D space with a limited amount of movements in vertical space to optimize performance. The route generation algorithm is based on connection coordinates which is from the GPS system, latitude and longitude. The start point is always the position the user is at, the endpoint can be several places. For example if the user wants to see possible routes to different warehouses for his own position. If the connection coordinates matches some location that is added the search on that branch continue, but if there are no possible connections then that route is excluded because it didn't lead to the destination point. In the case that in the tunnel end or beginning occurs a split into multiple tunnels all tunnels but one is added to a queue and before they are added the route to that point are copied and the next tunnel are added to avoid infinite loops. If the tunnel goes to another floor a check is made so it is a floor in the right direction and if it is then it is processed as a new tunnel with sub-tunnels. After each tunnel either reaches an end-tunnel or have a successful path to the destination the last tunnel added to the queue are processed next then this continue until the queue is empty.