Table 1 - Requirements:

Id	Functional/Non-Functional	Module	Description	Priority	Risk	Status	Implementation
1.	Functional	HR	The system must store the previous shifts.	MH	HIGH	ToDo	Not implemented because currently we store previous shifts in the RAM and all the information is erased when the power cause off. In the future we will use a database then we can implement it.
2.	Functional	HR	The system must show information about workers possible shifts inside shifts schedule.	MH	HIGH	Done	requirement can be applied
3.	Functional	HR	The system must support saving employee's personal information like name and id and employee's work information like bank account, salary, job terms, and start date.	MH	LOW	InProgress	requirement can be applied
4.	Functional	HR	The system must support managing lists of possible working days and hours per worker.	MH	LOW	InProgress	requirement can be applied
5.	Functional	HR	The system must support adding workers to shifts.	MH	LOW	InProgress	requirement can be applied
6.	Functional	HR	The system must prevent creating a shift without a shift manager.	MH	LOW	InProgress	requirement can be applied

7.	**Non-Functional**	**HR**	**The system must distinguish between regular employees and a manager.**	**MH**	**LOW**	**InProgress**	**requirement can be applied**
8.	**Functional**	**HR**	**The system must give regular employees the option to add new constrains.**	**MH**	**LOW**	**InProgress**	**requirement can be applied**
9.	**Functional**	**HR**	**The system must prevent placing an employee for a job which he is not qualified to do.**	**MH**	**LOW**	**InProgress**	**requirement can be applied**
10.	**Functional**	**HR**	**The system must send a message to the manager when he does not have enough employees for a shift.**	**MH**	**LOW**	**InProgress**	**requirement can be applied**
11.	**Functional**	**HR**	**The system must allow regular employees to see their shifts for the week.**	**MH**	**LOW**	**InProgress**	**requirement can be applied**
12.	**Functional**	**HR**	**The system must allow the manager to add a new constrain that the employees can use to declare a new constrain.**	**NTH**	**LOW**	**InProgress**	**requirement can be applied**
13.	**Functional**	**HR**	**The system should allow regular employees to see their personal and work information.**	**NTH**	**LOW**	**InProgress**	**requirement can be applied**
14.	**Functional**	**HR**	**The system should give the manager a preferable choice of an employee based on his constrains .**	**NTH**	**HIGH**	**InProgress**	**requirement can be applied**
15.	**Functional**	**HR**	**The system should send a message to the manager when an employee does not have enough shifts.**	**NTH**	**HIGH**	**InProgress**	**requirement can be applied**

Table 2 – open questions:

#	Topic	Open question
1.	Storing data	Which type of database we should use to store the data? Relational database or non-relational database
2.	Storing data	Which database we should use? (mongoDB, mysql, DynamoDB, etc)

Table 3 - questions:

#	Topic	Issue	Answer
1.	Employment	What do employment terms include?	Employment terms include job title and job type (full/part time)
2.	Assigning to shifts	how does the system need to show information about possible workers in certain day?	The manager chooses a day, shift(morning/evening) and a job title and the system will show all the available workers
3.	Job title	Does a worker can be assigned to two different positions?	No, every employee should focus on one position
4.	Constrains	Do the workers have constant constrains?	No, every employee can enter/delete/update constrains until Thursday, thus the manager create a new schedule to the next week in the weekend
5.	Urgent constrains	Do the workers have a permission to add a new constrain after the deadline is exceeded	No, they have to directly call the manager. The manager can alter the schedule any time
6.	Lack of workers	What should the system do when there is no enough workers in a certain shift?	The system should suggest an employee based on his constrain and working hours.