

# Functional Requirements Document

## Time Manager

Version	Description of Change	Author	Date
1.0.0	Added project context and purpose (1)	Lycia/Jordan	17/09/2019
1.0.1	Added Context Diagram (2.1)	Jordan	18/09/2019
1.0.2	Added interface requirement (3.1)	Jordan	22/09/2019
1.0.3	Added functional and hardware/software requirements (2.3 - 3.2)	Carolina	24/09/2019
1.0.4	Data Flow Diagram (2.2)	Lycia	25/09/2019
1.0.5	Other requirements (3)	Carolina, Jordan, Lycia	30/09/2019

## **1 INTRODUCTION**

A time manager is used in every company to help the time management of employee for every task they are assigned to. It helps the manager to make sure that things are in order like employee are actively working and doing their tasks that the schedule is respected. For example, it is used in company like ESN to keep track of the time spend to make a project because it's generally paid by the time used.

### **1.1 Purpose**

The time manager will help employee to check in their presence in the company and to check out when they finish their day/work. A work schedule will be sent weekly to every employee, according to their tasks. It will help him get a better management of their working time but also limit the number of night shift he will do.

For the managers the usefulness is in the control of the time, the division of labour and the management of night shift, to look after their team.

### **1.2 Scope**

In the morning employee have to connect to the time manager and check in to start registering the starting time of their work, until when it's time to check out.

The manager get access to the list of employees they manage, they can also see their work time, in case of absence or error in the schedule they are notified.

The admin is here to take care of everything about the application, access and error gestion.

### **1.3 Background**

The Gotham Dev society is in charge of the development of this project, the "recette" and the delivery for the production server of the client, according to the deadlines negotiated. This document is made by this company to help the client to get a better understanding of the scope of this project.

### **1.4 References**

- First meeting to establish contact between members of the team, speak about their speciality and stuff
- Second meeting with the team manager to show the progress
- Third meeting to establish a norm for the interaction between front and back end

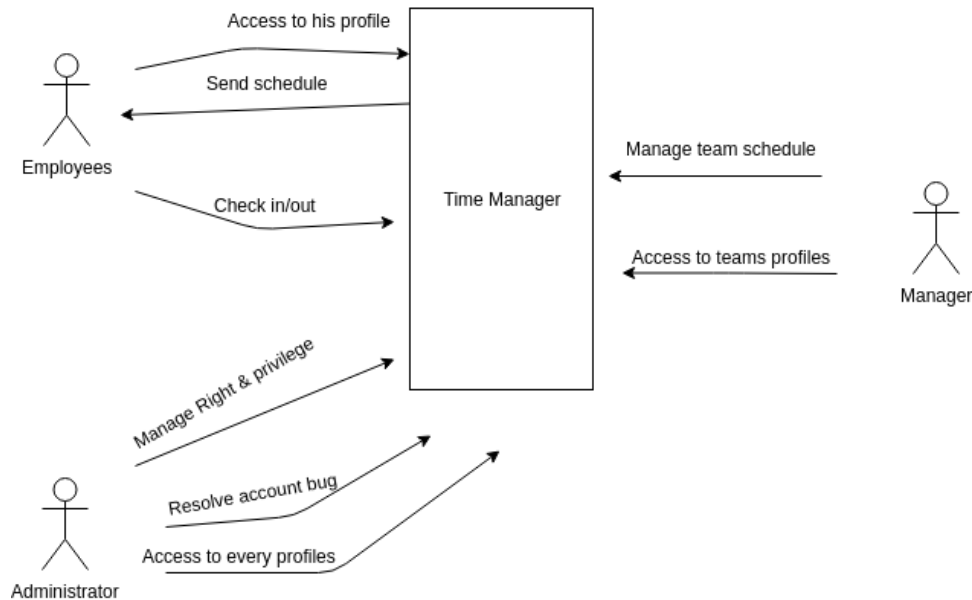
### **1.5 Document Overview**

This document is divided into two parts. First of all, we will see the customer's expectations on a technical level thanks to a context diagram, a data flows diagrams and functionality table. Then in a second step we will look at the customer's requests in a general way, i. e. on safety of user data, availability, accessibility, constraints and limitations.

## 2 FUNCTIONAL REQUIREMENTS

### 2.1 Context

**Diagram 1 – Features for the time manager users**



### 2.2 User Requirements

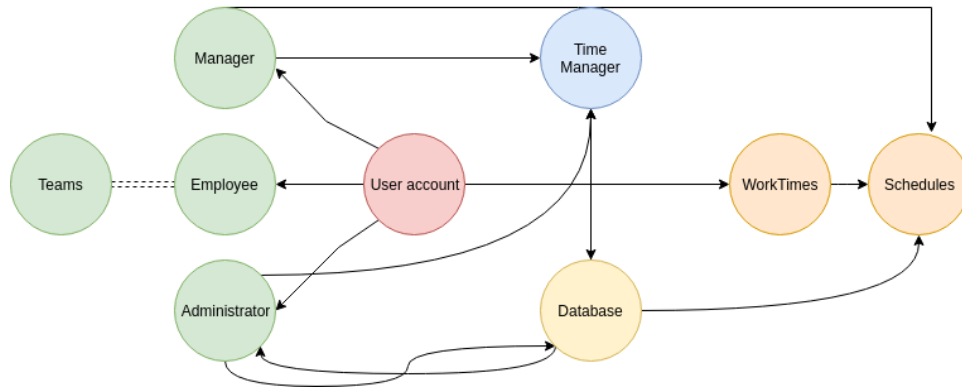
Training is required for any use of the application. For this, the administrator receives a complete training in order to train in turn all the employees and the manager access to the platform requires for any user an authentication whether the manager, the administrator or the employee. When a major new version of our time manager would be released, all persons will have the choice to assist to a training.

The manager has a view of all employees to manage the arrivals and departures of each with a detail of all hours pointed. In order to regularize the wages, it has a system of hours counter of day and night. He can also provide all schedules to employees and edit them.

The employee has access to his own detailed schedule for the week. He must follow the imposed schedules and point on time for each arrival / break / departure.

The administrator has access to the rights of each user, he manages the problems related to the scores. He also assigns user accounts to new employees with an email and a password.

## 2.2 Data Flow Diagrams



## 2.3 Functional Requirements

The functional requirements of the system can be divided in 3 parts: User, Teams and Working times. The main requested features are:

- Employee should create an account and login from the system for clock in/out their working periods.
- The system should allow employee to record/edit their working time manually even for finished days.
- Employee get notified when they receive their work schedule and have access to it.
- Manager should be able to manage their employee, modify their working time, work schedule and regulate their nightshift.
- Manager should create a team, organize employee by team and manage/follow employees working times.
- Manager should choose members of his team and active/put off the clock for all these members (this feature is very useful for employees that have same working schedules in a team)
- Teams must have settings that the manager must adjust like: number of days between consecutive nightshift allowed, maximum times of work per day for an employee, allowed additional hours, ... Manager should have the choice to block when limit is mad or not block but have an alert message.
- An alert system for managers when rules of teams just below are not respected or overflowed.
- Only administrator have access to all account: creation, edition, suppression of them and manage user's permissions. He can promote an employee to manager and the opposite.

### 2.3.1 Functional Requirements User

#### Exhibit 1 - Requirements User

Section/ Requirement ID	Requirement Definition
FR1.0	Authentication of a user
FR1.1	Register a new user
FR1.1.1	Create a new user account with first name, last name, email, password and with employee role by default
FR1.1.2	After the user is created, he should access to the Time Manager app
FR1.2	Login of a user
FR1.2.1	User can sign in with email and password then there should be a verification of credentials
FR1.2.2	After the login, user should access to the Time Manager app
FR1.3	Logout of a user
FR1.3.1	User should logout and after the access to the app should be denied
FR2.0	Manage user's account
FR2.1	Get user account information
FR2.1.1	Get the user first name, last name, email and user role from user id
FR2.2	Update user account information
FR2.2.1	User can change his/her last name, first name and email
FR2.3	Delete user account
FR2.3.1	All user information's should be deleted: user data, user working times, clock data of user and remove user from all enrolled teams
FR2.4 - [ADMIN]	Creation of a new user
FR2.4.1	Create a new user with first name, last name, email, password and type value in [Admin, Manager, Employee]
FR2.5 - [ADMIN]	Change user type (it's the role of the user in the company)
FR2.5.1	Promote user role to manager or admin or downgrade it to an employee

### 2.3.2 Functional Requirements Teams

#### Exhibit 2 - Requirements Teams

Section/ Requirement ID	Requirement Definition
FR3.0 [ADMIN – MANAGER]	Manage teams
FR3.1	Creation of a new team
FR3.1.1	Create a new team with team name

FR3.2	Modify team information
FR3.2.1	Change team name
FR3.3	Get team information
FR3.3.1	Get the team name, creation date, manager ID and team ID
FR3.4	Get all team of a manager
FR3.4.1	Show a list of teams (team name, creation date, manager ID and team ID) that belongs to a manager
FR3.5	Delete an existing team
FR3.5.1	Delete a team by team ID, this should remove team members before
FR3.6	Add a member to team
FR3.6.1	Add user to an existing team
FR3.7	Remove a member from a team
FR3.7.1	Delete user from an existing team
FR3.8	Get all members of a team
FR3.8.1	Show a list of all members (employee) which are in the given team
FR3.9	Adjust team setting for limit employee working times, this feature is a benefit for user health conditions

### 2.3.3 Functional Requirements Working times & Clock

#### Exhibit 3 - Requirements Working times & Clock

Section/ Requirement ID	Requirement Definition
FR4.0	Manage working times
FR4.1	Add a working time with start date, end date and user ID
FR4.2	Edit a working time information's by working time ID
FR4.3	Delete a working time by its ID
FR4.4	Get a working time information's
FR4.5	Get all working time of a user
FR4.6	Get the average of daily/weekly hours of working time of a team over a given period
FR4.7	Get the daily/weekly hours of working time of an employee over a given period
FR5.0	Manage clocks
FR5.1	Clock in, it must create a new working time with the current time for start date
FR5.2	Clock out, it must update working time with the current time for end date
FR5.3	Get the ongoing Clock information's

### **3 OTHER REQUIREMENTS**

#### **3.1 Interface Requirements**

For this requirement, we have to take in account 3 type of interface for each users: Employee, Manager and Admin.

The manager will get some statistic about his team working time, night hours in his dashboard, he gets access to his team and can modify working time of people inside, he is able to check-in/out for himself like everyone else and can send employee their schedule.

The Admin get access to everything the manager has, but with more liberty like access to every team and modify anything.

The employee get access to his own dashboard with statistic about himself and can check-in/out, he is able to check his schedule too

The interface must be user-friendly, the text should be clear and in a correct size for reading. Features should be easy to use and adapt depending the user's type.

##### **3.1.1 Hardware Interfaces**

The hardware interfaces supported by the system are: Computer, tablets and smartphones.

##### **3.1.2 Software Interfaces**

The Time Manager app is implemented using AngularJS technology. The use of progress web app (PWA) let us to install the app in any device (laptop, mobile or tablets, ...).

The app call a Rest API implemented in Java JEE with the framework Spring boot. The calls are made via https protocol. We use Docker for deploy the app and the API.

#### **3.2 Hardware/Software Requirements**

We strongly recommend a device fewer than 5 years old.

For hardware requirements:

- Processor: Minimum 1 GHz; Recommended 2GHz or more
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Memory (RAM): Minimum 1 GB; Recommended 4 GB or above

For Software requirements:

- Web app is available over all web navigators and in all OS (Windows, Linux and Mac)
- Mobile app is compatible with Android IOS and Windows phone

- Watch app is available in the Android and the IOS system

### **3.3 Security and Privacy**

#### **A. Consequences of the breaches of security in the application:**

1. In the case of data loss, users will no longer have access to schedules. Arrival / break / departure times will be lost. A wage problem will therefore be generated.
2. The accounts are locked if there is a disclosure of secrets or sensitive information.
3. If there is a corruption of the software or viruses, the application is blocked the time of the treatment of the devices.

#### **B. State the type(s) of security required. Include the need for the following as appropriate:**

1. Physical security.
2. Access by user role or types.
3. State access control requirements by data attribute. For example, one group of users has permission to view an attribute but not update it while another group of users has permissions to update or view it.
4. State access requirements based on system function. For example, if there is a need to grant access to certain system functions to one group of users, but not to another. For example, "The system shall make Function X available to the System Administrator only".
5. State if there is a need for certification and accreditation of the security measures adopted for this application
6. The data of one person cannot be accessed by another person due to the efficient authentication system (JWT Auth). All data access is based in this auth token which is unique for every user. Only you, your manager's and Admin users can access to your data.
7. We prevent also database attack by SQL or NOSQL injections. Input values are checked with security.
8. Passwords are stored encoded in the database and when the app sends login data to the API, data are encrypted by the https protocol.

### **3.4 System Availability**

The system is available 24/7 beside on Monday between 4:00 am and 5:00am for the daily maintenance



### **3.5 Data Retention**

The system will retain various information such as first name, last name and email about the employee, manager, administrator and also their past schedule, working times, position in the company etc... as long as they are still working for the company and for 5 more year when they retire/leave.

### **3.6 Error Handling**

The system has robust error management that handles errors related to data validation, data access permissions depending on user roles, configuration problems, data loss, or server failure ... In web app and mobile app user would be warned by occuring error from server side by an error message in popup.

### **3.6 Validation Rules**

All inputs are checked, input can't be empty and can't contain not allowed characters. All call to the server is valid if the send JWT token match with the user and role. All request params and request bodies are checked against injections (SQL and NoSQL).