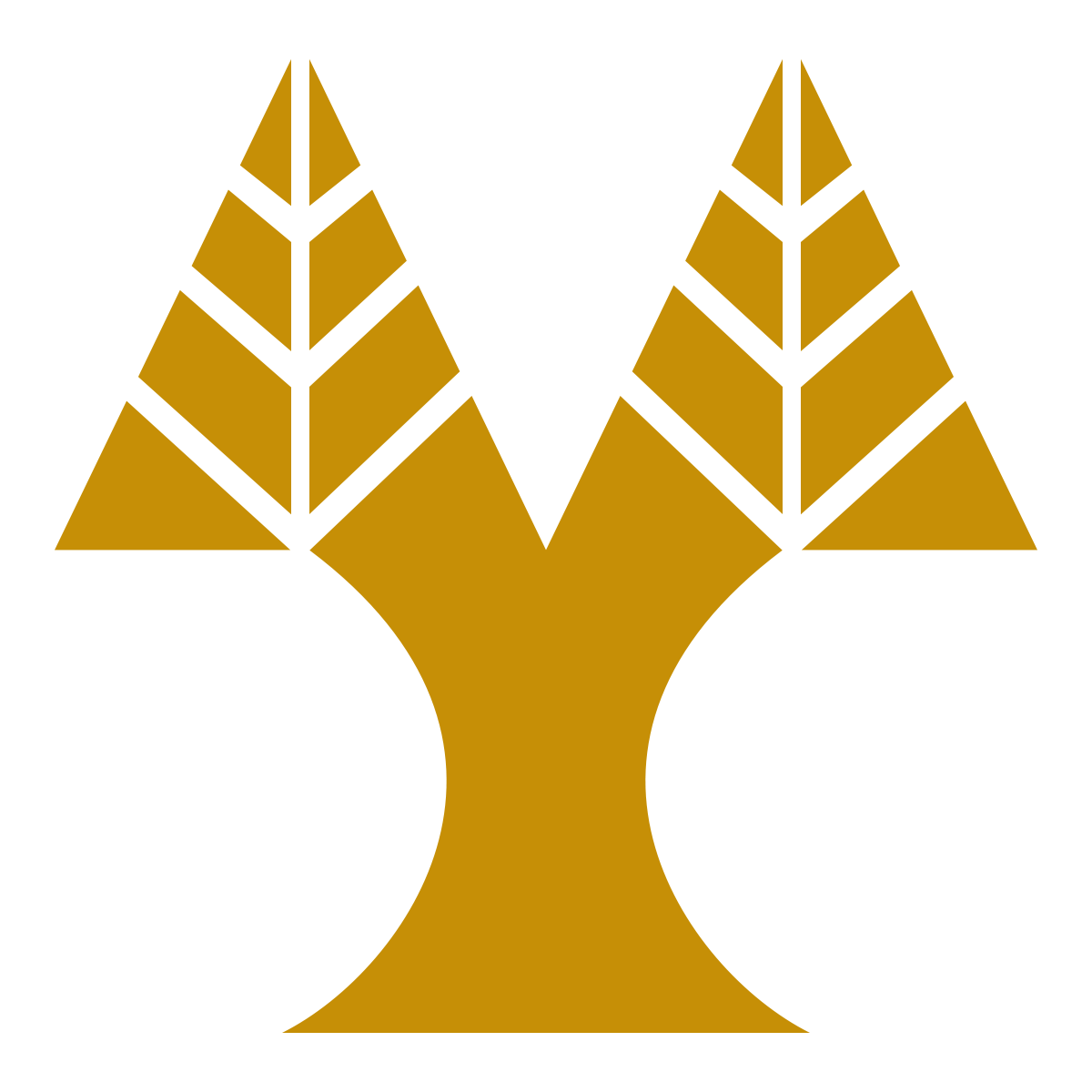


SPECIFICATIONS DOCUMENT



**Katsi Eleni**

**Mina Mariana**

**Agroti Louiza**

**Kouppi Maria**

**Contents**

**1.**               **Introduction**

1.1             Purpose

1.2             Scope

1.3             Definitions, Acronyms, and Abbreviations

1.4             References

**2.**               **Overall Description**

2.1             Product Perspective

*2.1.1            System Interfaces*

*2.1.2            User Interfaces*

*2.1.3            Hardware Interfaces*

*2.1.4            Software Interfaces*

*2.1.5            Communications Interfaces*

*2.1.6            Memory Constraints*

*2.1.7            Site Adaptation Requirements*

2.2             User Characteristics

2.3             Constraints

*2.3.1*  *Server*

*2.3.2 Response time*

*2.3.3 Storage*

*2.3.4 Software*

*2.3.5 Administrator*

2.4             Assumptions and Dependencies

**3.**               **Specific Requirements**

3.1             Software Product Features

*3.1.1*       *Employee’s Features*

*3.1.1.1         Associated Functional Requirements*

*3.1.1.1.1       Login*

*3.1.1.1.2       Clock out*

*3.1.1.1.3       Working Hour’s Clock*

*3.1.1.1.4       Leave Requests*

*3.1.1.1.5       View The Remaining Leave Days*

*3.1.1.1.6       Edit Personal Information*

*3.1.1.1.7       Contact Us*

*3.1.2        Manager’s Features*

*3.1.2.1         Associated Functional Requirements*

*3.1.2.1.1       Add User*

*3.1.2.1.2       Modify User*

*3.1.2.1.3       View Leave Requests*

*3.1.2.1.4      View Employee’s State*

*3.1.2.1.5      Delete Employee*

3.2             Performance Requirements

3.3             Software System Attributes

*3.3.1*       *Reliability*

*3.3.2*       *Availability*

*3.3.3*       *Security*

*3.3.4*       *Maintainability*

*3.3.5*       *Portability*

3.4             Logical Database Requirements

3.5              Other Requirements

**4.**               **Appendices**

1. **Introduction**

Statare Brands Ltd is based in Cyprus and owns the rights for various selling products and IT services.  The Statare Group is privately owned and has its main businesses in Norway, Sweden and Poland, while being under continuous growth.  The core business is the production and sales of storage solutions & wardrobes (mainly wardrobes made on the customer's own measures). The Group headquarters are located in Paphos, and employs around 200-500 employees worldwide [1].

**Webpages:**

[***http://statare.com***](http://statare.com/)

[***https://www.linkedin.com/company/statare-brands-ltd***](https://www.linkedin.com/company/statare-brands-ltd)

**1.1       Purpose**

This document describes in a specific way the requirements of this project.  This project will be a web app that works as a time attendance system for Statare LTD.  It refers to both the development team as well as the customer (Statare LTD) and can act as a contract between the development team and the customer.  Specifically, the scope of this system is the improvement of processes that concern the work force of this company. The purpose of this document is the detailed description of the functional and non-functional requirements of the system.

**1.2       Scope**

The project that we have undertaken is a web app “Universal Time and Attendance System” that helps with the management of the workforce of a company.  Its main activity will be the strict recording of working hours of all employees.  Additionally the system will allow the user to apply for an annual leave, to look at the hours they’ve worked during the month, to be able to amend their personal record and some other functions that will be mentioned later in this section.

Every company’s employee will have the chance to clock in – login from the moment they start working as well as the ability to clock out – logout when they finish working.  In this way, the hours they’ve worked will be recorded and the employee will not only have the chance to see the hours they’ve worked that month, but to see the current day’s hours they’ve worked as well.  Additionally, they can apply for leave.  They can do that by stating the reason for the leave request as well as the days requested, they can be informed of their remaining leave days and see the public holidays depending on the place of employment.  As a final point, they will also be able to change some of their personal details that are stored in the system.

Moreover, company’s managers, apart from the above functions, will be allowed to take further actions that will facilitate the running of their team.  For instance, they will be able to approve or turn down their subordinates leave requests.  They will be able to see their personal details as well as edit them.  They will be able to see which employees are working at a given point of time and who are absent from their offices, to verify whether the working hours based on clock in – out are true based on their corded place, and they can characterize them as valid or invalid, accordingly.  Finally they will be able to add, update or delete a user’s record from the system.

The Web App will be adjustable by which country the employee is working.  It will adjust the language, the number of days an employee is allowed as an annual holiday and the public holidays.  The system will be available in the following languages: Greek, English, Polish, Norwegian, German and Swedish.

**1.3       Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Time and Attendance system** | Systems that are used to monitor and track when employees start and stop work. [2] |
| **SSN** | Social Security Number. |
| **UCY** | University Of Cyprus. |
| **TCP** | Transmission Control Protocol. |
| **IP** | Internet Protocol. |
| **API** | In [computer programming](https://en.wikipedia.org/wiki/Computer_programming), an Application Programming Interface is a set of [subroutine](https://en.wikipedia.org/wiki/Subroutine) definitions, [protocols](https://en.wikipedia.org/wiki/Protocol), and tools for building application software. [3] |
| **XML** | In [computing](https://en.wikipedia.org/wiki/Computing), Extensible Markup Language (XML) is a [markup language](https://en.wikipedia.org/wiki/Markup_language) that defines a set of rules for encoding [documents](https://en.wikipedia.org/wiki/Electronic_document) in a [format](https://en.wikipedia.org/wiki/File_format) that is both [human-readable](https://en.wikipedia.org/wiki/Human-readable_medium) and [machine-readable](https://en.wikipedia.org/wiki/Machine-readable_data). [4] |
| **IT** | Information Technology |
| **DoS attacks** | In [computing](https://en.wikipedia.org/wiki/Computing), a denial-of-service attack (DoS attack) is a [cyber-attack](https://en.wikipedia.org/wiki/Cyber-attack) where the perpetrator seeks to make a machine or network resource unavailable to its intended [users](https://en.wikipedia.org/wiki/User_(computing)) by temporarily or indefinitely disrupting [services](https://en.wikipedia.org/wiki/Network_service) of a [host](https://en.wikipedia.org/wiki/Host_(network)) connected to the [Internet](https://en.wikipedia.org/wiki/Internet). [5] |
| **SQL injection** | A [code injection](https://en.wikipedia.org/wiki/Code_injection) technique, used to [attack](https://en.wikipedia.org/wiki/Attack_(computing)) data-driven applications, in which nefarious [SQL](https://en.wikipedia.org/wiki/SQL) statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker). [6] |

**1.4       References**

[1]  STATARE BRANDS LTD: [*https://www.linkedin.com/company/statare-brands-ltd*](https://www.linkedin.com/company/statare-brands-ltd)

[2]  Time attendance system: [*https://en.wikipedia.org/wiki/Time\_and\_attendance*](https://en.wikipedia.org/wiki/Time_and_attendance)

[3]  API: [*https://en.wikipedia.org/wiki/Application\_programming\_interface*](https://en.wikipedia.org/wiki/Application_programming_interface)

[4]  XML: [*https://en.wikipedia.org/wiki/XML*](https://en.wikipedia.org/wiki/XML)

[5]  DoS attacks: [*https://en.wikipedia.org/wiki/Denial-of-service\_attack*](https://en.wikipedia.org/wiki/Denial-of-service_attack)

[6] SQL injection: [*https://en.wikipedia.org/wiki/SQL\_injection*](https://en.wikipedia.org/wiki/SQL_injection)

**2.          Overall Description**

**2.1       Product Perspective**

The Universal Time and Attendance System is not independent. That is, it will communicate with two other existing company systems, Payroll system. The web application extracts a report with the employee's hours of work for the Payroll system.

**2.1.1**     **System Interfaces**

Using the Active Directory, the system’s users will be authenticated and authorised.  This is used to specify if a user is an employee or a manager.  When the user enters the system using its credentials, active directory knows the privileges the user has and informs the system about the user’s position.

Furthermore, the system will communicate through API with the company’s existing payroll system. In will export an XML file and a JSON file, which will contain information about the total hours that each employee worked, as well the overtimes, in order to calculate the final salary in the existing payroll system.

**2.1.2     User Interfaces**

When the user will login into the system, a screen will appear. That screen will ask him to insert his username and password. If the user inserts wrong password, then the “Wrong Password” message will be shown. If the user inserts username and password correctly, then a screen will appear that will be the same for both employees and managers. From there they can make clock in, clock out, break, return from break and log on to dashboard:

* Clock in (when this button is pressed, the system starts counting the working time)
* Clock out (when this button is pressed, the system stops counting the working time after entering the password)
* Break (when this button is pressed the system starts counting the time of break)
* Return from break (when this button is pressed the system stops counting the time of the break and continue to counting the working time as clock in)
* Log on to the dashboard (when the user clicks on this button the system will enter it into his account)

On the manager screen, the Log on to dashboard button will include the following various buttons each with its own function:

* By clicking on the View Hours button, you can see his/her hours of work (when he/she clock in, clock out, hours worked on a particular day, and the total average of hours he worked today).
* By clicking on the Add Employee button, the manager can add an employee to his/her group and record his/her personal details, such as his/her role in the company and annual leaves.
* By clicking on the Edit Employee button, the manager can edit employee’s personal details who is in his/her group. For example, change the salary of employee.
* By clicking on the View Requests button you will be able to see his/her employees' leave requests in a list. The manager will have the option to accept or reject the leave request by clicking on the A(accept) or R(reject) button. Once the manager clicks on one of the two buttons, the state of leave request will be updated accordingly in database.
* By clicking on the Average Report button, a manager should be give a range of dates, the format which in export the report(JSON or XML) and click the export button for export a report.
* By clicking on the Employee Status button you can see which employees are clock in, who are clock out and who are in break.
* By clicking on the Payroll Report button will generate a report that will record details of how many hours each employee has worked on. This report will be exported to XML or JSON format.
* By clicking on the Profile button you can modify his/her personal details.
* Pressing the Leave Request button will send a leave request to the manager. He/She will choose the days he/she wants the leave and the reason he/she want leave.
* By pressing the Delete Employee button, the manager will be able to delete an employee from his group. But the personal details of the deleted employees will continue to save in database, on another array.

On the employee screen, the Log on to dashboard button will include the following various buttons each with its own function:

* By clicking on the Profile button you can modify his/her personal details.
* Pressing the Leave Request button will send a leave request to the manager. He/She will choose the days he/she wants the leave and the reason he/she want leave.
* By clicking on the View Requests button you will be able to view his/her leave requests in a list. There he/she will be able to see whether the manager has accepted the request or not.
* By clicking on the View Hours button, you can see his/her hours of work (when he/she clock in, clock out, hours worked on a particular day, and the total average of hours he worked today).

**2.1.3     Hardware Interfaces**

The UCY’s server will be used for the database.

**2.1.4     Software Interfaces**

The web app will be written on HTML, CSS and Javascript.  It will also communicate with the database to storage all of the system’s data.

We will use the ucy’s server along with PHP code and a mySQL database to implement the system.

For PHP:

* Name: PHP
* Version Number: 5.6.31
* Source: PHP agency

For mySQL:

* Name: mySQL
* Mnemonic: unlimited
* Version Number: 5.7.19
* Source: mySQL

**2.1.5     Communications Interfaces**

The web app communicates with the Internet, as a result, TCP and IP protocols will be used.

**2.1.6     Memory Constraints**

The only part of the system that will need to be installed on the server of the company, is the database. The customer requests that the database can access unlimited records and clarifies that he doesn’t want to limit the database’s size.  Also, he doesn’t have any problem in terms of memory usage.

**2.1.7     Site Adaptation Requirements**

The web app must work on desktops with windows operating system and tablets with android operating system.

**2.2       User Characteristics**

The system’s users are divided into employees and managers.  Employees are divided to simple employees and sellers. There are manager of each department (leader managers, IT, shop, factory, truck etc) that the company has, who manage, supervise and control the employees of their department.

More details for each user are described in section 3.2. (Performance Requirements)

**2.3       Constraints**

(a)    We are unable to use other equipment except that of the company.  This limits us in terms of the choices and the methods that we will use.

(b)    Any coding of the system should be written in specific programming languages, like HTML, CSS and Javascript, and database management language like T-SQL for better maintainability.  This is because these are the programming languages that the company uses.

(c)    The system cannot be used by any other users except the company’s employees and managers.

(d)    The user cannot login from other locations except from the company’s address and the customers’ addresses.

(e)    There will be a list with the customers’ addresses in order to reassure managers that the employees are indeed in a customer’s house and they accept the request.

(f)    If the user forgets to logout, the system will check if they are within a small range radius near the location they were working and it will automatically log them out.

(g)    Safety and security considerations are quite important for a web app used by

companies. There are various security threats such as DOS attacks, SQL injections

and not installed antivirus.

### Server

Possible threat to the server, the huge amount of data as it will contain data for individuals inside and outside of Statare. Continuous personnel growth requires that we have enough space and the necessary memory to support the system. First of all, we think the system will support:

At least 1TB of memory

16 - 32 GB of RAM

MySQL database support

### Response time

The system response must be made without delays. In order for our system to be efficient, simple and flexible, the response time must be within a few seconds. For example, Clock in to computer responds to a minimum of less than 0.5 sec while the worst case response reaches 1 sec.

### Storage

The system saved the personal details of employees in a database. The database is on the server of company.

### Software

Windows system support.

### Administrator

Administrators are managers who will have full access to the system. Also, the managers of each department give passwords to their employees and have access only to their employee data.

**2.4       Assumptions and Dependencies**

**Assumptions:**

We assume that the network will always be active.

Additionally, we assume that the system does not depend on any other system of the company. Nonetheless, an XML file will be sent every month through API. The API will be used for the interaction between the UCY’s server and the company’s server. Though, in the end of this project, the company will install on their servers all of the system’s information.

**Dependencies:**

A fast internet connection is needed and it is one of the most important factors for the proper and right operation of the system.

Finally, devices must be of high performance in order to respond as fast as possible to the processes of the system to prevent delays or crashes.

**3.          Specific Requirements**

**3.1       Software Product Features**.

By using case cases we find the functionalities of our system. We need to check in specific cases the response of our system by making scenarios and cases to check the correctness of our system (that is, how we will manage these events).

**3.1.1**     **Employee’s Features**

**3.1.1.1**  **Associated Functional Requirements**

**3.1.1.1.1       Login**

Introduction: The “Clock in - Login” function is responsible for letting the system’s user to have access to the system.  In the text below we specify how it works.

Inputs:  Employee’s username and password.

Processing:Check whether the credentials of the user are valid.  Compare them with the credentials of each user account in the database. If the credentials are valid then the employee is login in the system.

Outputs: If the input is valid, the homepage of the web app is being shown to the user, otherwise an error message is being shown.

**3.1.1.1.2       Clock out**

Introduction:  The “clock out” function is responsible for logging the users out of the system.

Inputs: Press the clock out button.

Processing:  Get the exact number of the working hours from the system’s clock (counter) and save it into the database.

Outputs: Message that you clocked out.  If the user clocked out from the system, the clock stops.

**3.1.1.1.3       Working Hours’ Clock**

Introduction: The “Working Hours’ Clock” function is responsible for informing the users about how many hours they have been working on that day.

Inputs: None

Processing: Shows to the employee their counter-clock value.  Specifically, it shows to the user for how many hours have been working for that specific day. This value is available as long as the user has logged into the system and it is placed at their home page.

Outputs: Counter - Clock value.

**3.1.1.1.4       Leave Requests**

Introduction: The “Leave Request” function defines how the user will be able through the system to make a leave request to their manager.

Inputs: Text describing the reason for being absent along with the days you want to be absent.

Processing: If you press the submit button then the content of the text is saved in the database.  Also, a notification is sent to the manager.  If you press the cancel button then the text is deleted and nothing from the actions above takes place.

Outputs: Message that your request was sent successfully.

**3.1.1.1.5       View the Remaining Leave Days**

Introduction: The “View the Remaining Leave Days” function defines the way that the employees will have access to their remaining leave days.

Inputs: Push the requesting leave button.

Processing: By default, the web app shows the employee’s remaining leave days while they try to form a request.

Outputs: Text representing the remaining leave days of the employee.

**3.1.1.1.6       Edit Personal Information**

Introduction:The “Edit Personal Information” function specifies how the users can edit their personal data.

Inputs:Select the button “view profile” and then place string in any of the following fields:

Name, Surname, Username, Password, Address, E-mail, ID, SSN, Gender, Phone Number, Emergency Phone Number and Date of Birth.

Processing:Every user has the ability to change their personal information by selecting the personal information section.  Then, they can choose the proper field they want to modify.  If they want to apply the changes they press ‘save’ and the database will be updated or they press ‘cancel’ and the changes don’t take place.

Outputs:The page consists of their personal information is presented with the informed information.

**3.1.1.1.7       Contact Us**

Introduction:  The “Contact Us” function’s aim is to help users quickly overcome any difficulty concerning the system’s use.

Inputs:  Press the ‘Contact Us’ button on the initial page of the web app and write you message.

Processing:  By pressing the button a text box appears where you can write your problem. Then you have two options; to send the message or to cancel it.  If you send it, the message is sent to manager’s email.  Then, the manager will answer to your problem using the email your manager entered in your personal details when he registered you to the time attendance system.

Outputs: Message that employee’s message has been sent successfully.

**3.1.2**     **Manager’s Features**

**3.1.2.1** **Associated Functional Requirements**

**3.1.2.1.1       Add User**

Introduction: The “Add User” function gives the chance to the manager only, to add a new user to the system.

Input:  Selection of the button “Add user” and add personal details of a user. (name, surname, gender, date of birth, address, phone, emergency phone, e-mail, address, ID, SSN, type, salary, country of work and no. manager that will be responsible for them)

Processing:  If all necessary fields are completed, then, an extra record is inserted on the employee’s table of the specific team of the database, otherwise, a message is shown saying that some information is missing in order to enter the user.

Output: Record entered successfully and username and password of the new user are produced or an error message saying ‘there are missing fields’ appears.

**3.1.2.1.2       Modify User**

Introduction:  The “Edit Employee” function’s aim is to give the chance to the managers to edit the personal details of their subordinates.

Input: Selection of the button “View Employees” and then select the employee whose information needs to change, and add the changes to the following fields:

name, surname, address, phone, emergency phone, e-mail, ID, position, SSN, salary, type of employee, country of work,  monthly working hours-location of them and working hours.

Processing: The program proceeds to show the list of the subordinate employees when the manager makes the changes they can choose to save or cancel the changes.  If they choose to save them, then the proper fields change in the database too.

Outputs:  the program shows the page with the modified information about the employee.

**3.1.2.1.3       View Leave Requests**

Introduction:  The “View Leave Requests” function’s aim is to make the procedure of accepting or rejecting a request for the managers a lot easier.

Input: The manager can select the button with the ‘Leave Request’.

Processing: The program shows the page with the list of all the requests from the employees, the manager can accept or decline using the buttons that are next to every request.  The employee will get a notification about the review of their request.

Outputs: The system based on the selection of the manager will colour the request with red, if the request was declined, or green, if the request was accepted.

**3.1.2.1.5        View Employee’s State**

Introduction:  The “View Employee’s State” function’s aim is to let the manager know the state of his employees.  For example, if they are currently clock in, on vacation, etc.

Input: Press the button ‘View Employees’.

Processing: The program shows the page with the list of all the employees; the employees that are currently absent with leave requests are grey and they seem offline.  For the other employees that are working regularly there is a state that shows if they are currently clocked in or clocked out, this information comes from the database based on the actions of every employee.

Output: The list with all the employees and their current state.

**3.1.2.1.6        Delete Employee**

Introduction: The “Delete Employee” function’s aim is to give the chance to the manager to remove an employee from the system if they stop working for their team.

Input: Press the button “Delete Employee”.

Processing:  A list of all employees is shown.  The manager can select the record of the employee they want to remove and then press the button delete.  Then the password and the username of the employee are removed from the valid credentials of the database.

Output: Message that the user record was removed successfully.

**3.2       Performance Requirements**

Performance requirements are divided into static and dynamic requirements, which are the requirements that are placed on the software or on human interaction with the software.

**-  Static:**

One of the static requirements is that the application’s transactions should be as fast as possible for every action each user makes.  In best case scenario, every transaction should take at least 1 second, and in worst case scenario it would take about 3 seconds.  Another requirement is that the application shall support simultaneous users, which are going to be around 200 - 500 users at first.  Those simultaneous users would be the leader managers, the managers, the sellers and the rest of the employees.  The company’s managers are divided into the following areas: IT, shop, factory, truck, team and leader. Moreover, the application will handle personal information of each employee, for example, their name, surname, username, password, ID, gender, date of birth, phone number, emergency phone number, email, address, country and SSN.  Also, it will handle each employee’s salary, position, annual leave, working hours, location, their type of employment -if they are part time fixed salary, or fixed salary with overtime) and public holidays. Each user’s profile will take around 150–200 MB.

**-  Dynamic:**

One of the dynamic requirements is that the web app’s transactions and tasks are to calculate the hours that the employees are working. The duration of their break depends on what position the employee has and on the policies that each country applies. Moreover, it will send a request to the manager when an employee wants to have an annual leave. Furthermore, the data that will be processed within a certain time period for both normal and peak workload conditions are the working hours of each employee, the annual leave each of them is entitled to and their location. Those data are going to take around 50 GB of ram and 2TB of hard disk, though it depends on the exact number of the employees that the company has.

**3.3       Software System Attributes**

Software system attributes are divided to reliability, availability, security, maintainability and portability.  It can provide a partial list of system attributes that can serve as requirements that should be objectively verified.

**3.3.1     Reliability**

One of the factors that affect the reliability of our software system is the database.  The database is usually a critical piece of infrastructure in an organization; when the database is down, many things stop working, so database reliability is often a serious concern.  One way to solve that problem is to backup the data of the software system on another server that the company has. Another way to solve that problem is to be developed to hold at least 3 months of data and backup can be done daily, perhaps twice a day.

**3.3.2     Availability**

The application should be available online on a web page. It would be available for every person that is a part of the company. The web app will be set as a start-up app and full screen on the tablets that the company´s employees use.

**3.3.3     Security**

To protect the software from accidental or malicious access, misuse, or modification each user shall have their own unique username and password to enter the system. In this way each user´s personal details will be safe and uncorrupted.

Each user will have access to different data. For instance, the manager will have access to all of the information of each employee that works on the department that he is responsible for. He will have access to each employee’s personal information, annual leaves, working hours, salary, position and location. On the other hand, the rest of the employees will only have access to their own personal information, working hours and annual leaves.

**3.3.4     Maintainability**

One of the attributes of the software that relate to maintenance is fixing a bug that will may occur, without introducing a new one, and at the same time fixing it in a way that it will prevent another bug to occur in the further future.  Another one is to add new features that the company would like to have, without though occurring new bugs.

Furthermore, there may be a need to improve the usability, for instance, to make the user’s interface even simpler than before, to make it as simple as it can get.  There also may be a need to make changes to the web app so that it would be able to support new operating systems, environments or tools.

.

**3.3.5     Portability**

Our web app will have the ease of porting the software to other host machines and / or operating systems because it’s a web app.  As a result, it would be available for every host machine and operating system.  Though, the web app for this project will be working only on Google Chrome and any other browser that uses HTTP 1.1.

In addition to this, the web app will transfer data to the database through API.  For instance, to transfer the working hour of each employee to the company’s existing payroll system by exporting an XML file and send it to it.  Another restriction that the database shall have is that the employee’s that are working in the company as part-time employees they have specific working hours and they do not have overtime hours.  As a result, it’s impossible for them to overcome their working hours.  Also, another restriction is the limit that each employee has for their annual leaves.

**3.4       Logical Database Requirements**

In this section it will be clarified what kind of information will be placed into the database.

First of all, the types of information that will be used for various functions are:

❖    Varchar(35), for the employee’s Name, Surname, Username, Password, Address, E-mail, Position and Location.

❖    Varchar(15), for the employee’s ID, SSN.

❖    Char(1), for the employee’s Gender.

❖    Int, for the employee’s Phone Number, Emergency Phone Number, No. Country and No. Manager.

❖    Date, for the Public Holidays and for the employee’s Date of Birth.

❖    Float, for the employee’s Salary, Working Hours, Overtime Hours and Annual Leaves.

❖    Char(3), for the employee’s Type, if they are Part-time, Fixed Salary or Fixed Salary with Overtime.

❖    Char(2), for the employee’s Status if they are offline, online or on annual leave.  
  
The information above apply for the manager too. Moreover, database’s frequency of use would be daily. This is because every time that the user is logging into the system it would upload information from the database to the application and then it would update every modification that has been made every time that the user is logging out.

The database’s entities and their relationships are the following:

* The manager, which he should have a relationship with every employee that he is responsible for and belongs in the same department as he does.  Also, if there is a leader manager that is responsible for him then he should have a relationship with him too.  He shall be able to view every employee that he has relationship with, to view every of their personal information, their annual leave request, their annual leaves they are left to them, and their current situation.  Their current situation it would be specified by the fact that if they are on a break, if they are working or if they have annual leave on that moment.
* The employee, which he should have a relationship with his manager.

The database’s integrity constraints are that the manager has access only to his own account and he won’t be able to have access to employees that do not belong at the same department as he does.  He, also, won’t be able to modify some of the employee’s fields, for instance, he won’t be able to change the annual leaves they are left to them, neither to change their working hours.  At last, the employees would have access only to their own accounts and they would be able to edit only specific fields, for example, they won’t be able to change their salary, neither their working hours, or their annual leaves they are left to them.

The data retention requirements are to ensure that all the necessary data is stored properly.

**3.5       Other Requirements**

There are no other requirements.

**4.          Appendices**

1.         What are the functions of the system?

2.         Are the functions of the system feasible now? If yes, how?

3.         Are there any forms that connect with the functions of the system? Are these forms incorporate to the system? Can we get access to them?

4.         The system is addressed both to the employees and the user. Which functionalities do you want the system to support (what will the employee be able to see and do)?

5.         What work departments does the company have?

6.         Do you only have full-time or part-time employees as well?

7.         How many hours do the employees work every week?

8.         Would you prefer the system to count the hours only or the salary of the employees as well?

9.         Do you want the system to allow the employees to enter even if they exceed the weekly working hours? If yes, does it count as over-time or are they deducted from the working hours of following week?

10.       Will the system be accessible from everywhere, or just company premises?

11.       Will Geo-blocking working for specific employees? Are there any employees working outside the company? If yes, is the access to the system necessary?

12.       When the application is running will Geo-blocking keep working or will check the location of the employee when he/she tries to log in?

13.       Would you prefer to give the ability to the employee to notify you if they cannot come to work the specific day?

14.       Would you prefer the system to show the employee how many days off he has left?

15.       Which platform will support the system?

16.       Which operating system will the application support?

17.       Which is the worst time response of the system to the user?

18.       Which language will the template be written?

19.      Would you rather have the GitHub repository in public or in private?