2023

Fonteyn Vakantieparken

DESIGN DOCUMENT

MARC R. KOCK, MD FARHAN TAHMID, ROWEN DE VRIES, NIKOLA HRISTOV, MURTHID AL-HABSI.

FONTYS | Eindhoven

Table of Contents

AGREEMENTS

The main agreement among the group to create the project is “Finish the must-have elements first, then add the extra features.” We as a team also agree to complete the parts assigned to individuals themselves, in-time. At all the time of the development process, all the team members will keep the deadline and other risk factors in mind. The requirement of the client gets the top priority in the process.

The completion is estimated to be within 7 weeks of the start of the project. All the further changes in the plan will be agreed between the 4. The tutor will have an overview of the progress every day.

We all must hold each other responsible and motivate each other to complete the task. Rather than having one leader we choose to make decisions as a group and the role of leader is more meant to keep each other in check so we don’t procrastinate or hand in low quality work.

Communication is key to completing the project. That’s why we have made some rules we must abide by to ensure the quality of our teamwork as well as our final product.

- Let your team members know when you are: late with an assignment, not coming to school or you will be late.
- Communicate regularly about what and when you will be doing a task and give each member a specific time to show what they have done.
- Ask for feedback every day.
- In case of a disagreement, we vote.

These are the important rules to keep the team functioning.

REQUIREMENTS

These will be all the necessary components for our system to function in an acceptable manner. In accordance with the project requirements and our own requirements as to what we would like to add to our project.

Requirement ID	*Requirement Statements*	*MoSCoW*
FR001	Cost Management	Must
FR002	Cloud Migration	
FR003	Management Dashboard	
FR004	Fault Management	
FR005	Security	
FR006	Service Management	
FR007	Website	
FR008	RFID Entry System	Should
FR009	Backup Servers	
FR010	Failover Measure	
NFR01	SoD (Segregation of Duty) Analysis	Could
NFR02	Environment sensors	
NFR03	IPv6	Won't

Table 1: Functional requirements

SYSTEM SETUP

The whole system is built on a specific set of tasks. These are as follows:

1. **Creating a Network Infrastructure Diagram** and defining its communications briefly. Therefore, a brief understanding of the network and its components is defined.
2. **Creating a LAN with segmentation**. It prepares the base to be built on.
3. Installing the Hypervisor. The Hypervisor in this case is Hyper-V type 1. It is the base the system will be built on. It will host the Windows Server OS in our on premises hosted server.
4. **Creating a firewall** in the network. In this project, it is pfSense v2.1.5. This firewall acts as a gateway for the network and connects the devices in the network. Proper internet access needs to be configured supported by another DHCP server with rules. Also configuring the internal hosts to connect to the internet is necessary.
5. Creating necessary services. Said services are:
    a. **Fault management.** In the event of system failure, there will be a system put in place that will immediately alert the staff to the occurrence of such failure and proper mitigation measures will be taken.
    b. **Change management**. Should hotel administrators wish to implement grandiose changes into the environment there should be a framework in place that will help facilitate a seamless shift to the new flow chosen by the company.
    c. **Webserver**. The webserver will contain the website and host it. All the HTTP requests and responses will be protocoled by this server.
        i. Configuring webserver.
        ii. Creating and testing a test website.
        iii. Creating certificates for the website.
    d. **Storage Server**. All the files and data will be stored on this server. It will also enable us to secure and manage that data easily. It also contains the necessary databases.
        i. Configuring storage server.
        ii. Installing DB
6. **Installing necessary workspace management tools.** These tools will help manage the system and services.
7. **Setting up VPN for remote employees.**
8. **Setting up load balancing in whole system.**
9. **Setting up backups for whole system.**
10. **Azure Cloud.** The cloud has a VM set up with the webserver, database and cost management systems. The KPI monitoring are also built into azure, so it doesn't need to be monitored manually.

The system then will be able to serve the organization properly with functional servers, website, and services with real time monitoring. The whole network will also be secured. Staffs and visitors will be able to use the new system properly.

NETWORK CONFIGURATION

Our network will have a tree topology, that is separated into several different independent networks.

The network segmentation was done so that different company sectors are isolated from each other. There is a different network for the following company branches:

- Hotel administration
- IT management and maintenance (the virtualized servers will be run on the same network)
- Other hotel staff – security, cleaners, kitchen staff, etc.
- Guest network

That solution was implemented with security in mind – if a node in a network gets compromised by a malicious party, the bad actors will need to spend more time on exploring the network in search of what they need. This way there is a better chance of mediation taking place on time.

NETWORK DRAWING & DESCRIPTION

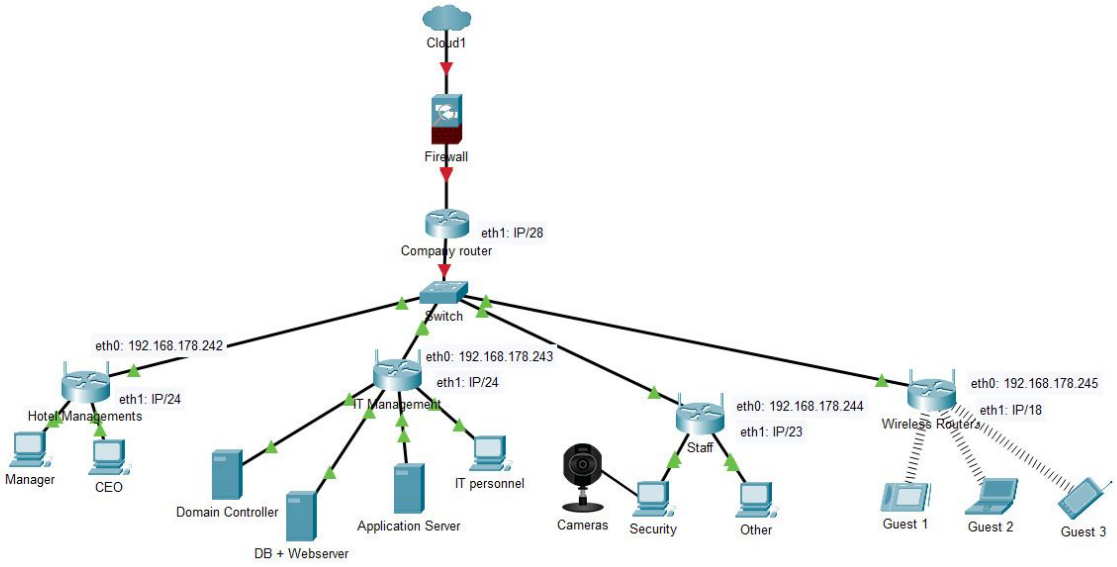

Figure 1: Network diagram V.1

The network has 4 different networks connected to a switch, which is also connected to the central router. The reason for the splitting of the network is due to security reasons. If one network gets compromised, the other networks and users are usually safe. It also makes sure that only the people who need to access the network are in the network and no one else. The central router has a firewall built before it, so it filters out necessary internet ping and access.

The 4 networks are Hotel Management, IT Management, Staff and Guest Network. Among these, the security cameras and other security appliances are inside the Staff network. The necessary servers are inside the IT Management network. The servers used in the network are the Domain Controller, The Webserver (Contains the Database too), and the Application Server. The Domain Controller contains an Active Directory.

The used firewall is pfSense version 2.5.2. It is set up before the company router and has external and internal IP addresses.

During the designing of the network, a Tree Tropology has been followed.

DESCRIPTION OF THE SERVICES

1. **Synchronized Cloud Storage:**

Staff members will be able to work on documents simultaneously and changes will be made in real time to prevent different versions of one document existing.

2. **User Management Tools:**

Hotel administration and HR will have the ability to oversee all personnel and their respective permissions and rights over the files and other facilities.

3. **Hotel Booking Services:**

Hotel visitors will be able to make booking requests to the hotel and the staff can verify them online, based on availability.

4. **Event Management Services:**

The guests will be able to see and buy tickets for upcoming events. The staff will be able to track the capacity and number of people attending those events.

5. **Server Monitoring System**

The administrators will be able to monitor the usage of the servers through a python app. It will show graphical data of all the resources being used in the domain controller and the file-server.

6. **Migrating necessary components to cloud**

We planned to migrate the important components only like DC and the webserver. This operation provides a lot of benefits for our company like increased scalability, flexibility, and cost savings.

7. **management dashboard**

We decided to use this tool to help us monitor the performance of our company and make informed decisions about it. It includes visualizations such as charts, graphs, and tables that provide a quick and easy way to understand the data.

8. **A fault management system**

The aim of this software system is to detect, isolate, and resolve faults or errors that occur within a system or network. This system helps us to minimize downtime and ensure that the system or network is operating at its full potential.

9. **Improved website**

10. **Cost management system**

The goal of a cost management system is to ensure that the organization is operating within the budget and making efficient use of its resources. It includes Budgeting, Cost tracking, Cost analysis and Cost control

SYNCHRONIZED CLOUD STORAGE

The storage of the infrastructure is the file server hosted in Hyper V. In this case, all of the users and the admins can use the same server for storing and accessing files. The roles are defined within active directory, therefore, the permissions are restricted and users can only access files that they are permitted to access. The security groups are defined in the active directory and there are folders added in the shared network which individuals can access based on their security groups.

USER MANAGEMENT TOOLS

Park administrators will be able to assess the entirety of their staff through Microsoft's Active Directory. In addition to that, administrators will also be able to manage the permissions set up for the staff members. It's important to note that there are restrictions imposed already on all staff departments and changes should be made with caution, in order not to pose a security risk to the company.

THE WEBSITE

We've opted to create a website using Python by utilizing the flask modules and for the database we've created an ORM database.

THE FRONTEND

The front-page or home is minimalistic, so visitors don't feel overwhelmed by clutter. The first thing you see is the client of the park and then you have options to register or log in, after which you can purchase tickets.

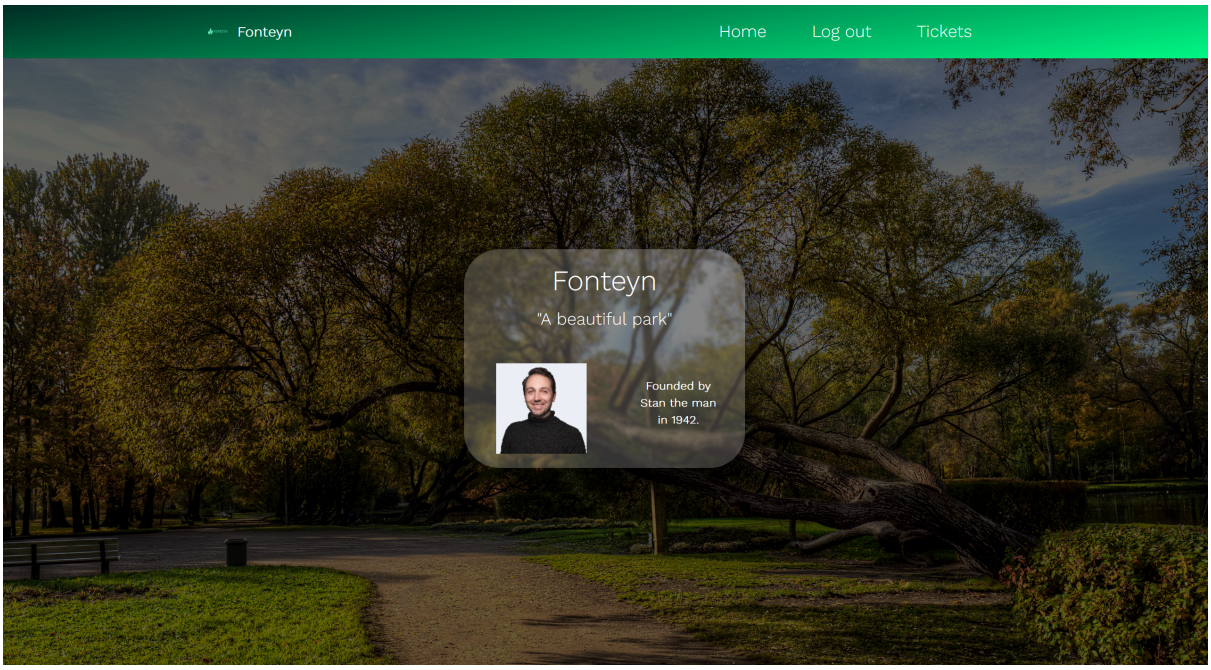

Figure 2 Home page

The ticket purchase page allows the user to choose a type of ticket being either standard, premium or VIP. You can pick how many adults and children you plan to attend with and at which park, this is intended for expansion purposes. After purchase you will be redirected to the home page where you will see a banner confirmation of your purchase, if failed you will see a red banner explaining why it may have failed.

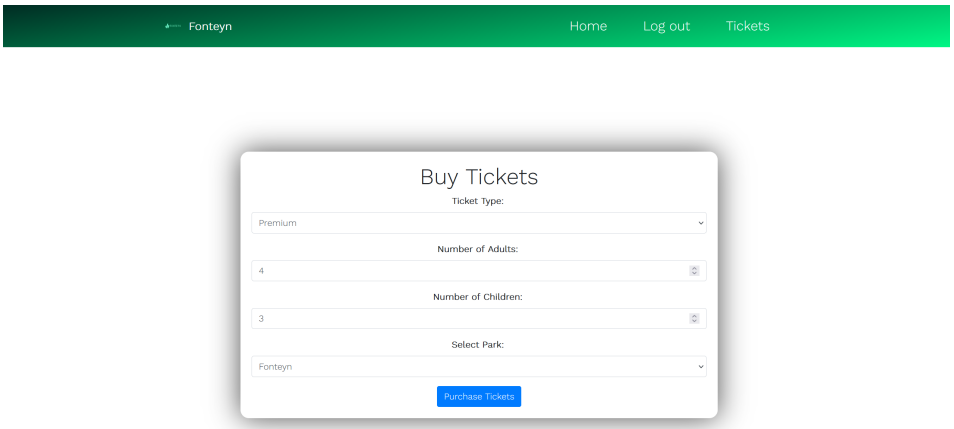

Figure 3 Ticket purchase

The user's credentials are secured using a SHA-256 cryptographic hash function, which is topped off with a salt and pepper. To create an administrative user, a value must be changed in the database itself, this was a design choice for security as accidental administrator users cannot be generated this way.

```python
def __init__(self, email, first_name, password=None, admin=0, salt=None, pepper=None):
    self.email = email
    self.first_name = first_name
    self.admin = admin
    self.salt = salt or os.urandom(32).hex()
    self.pepper = pepper or os.urandom(32).hex()
    if password:
        self.set_password(password)

def set_password(self, password):
    self.salt = os.urandom(32)
    self.pepper = os.urandom(32)
    salted_password = hashlib.sha256(self.salt + password.encode() + self.pepper).hexdigest()
    self.password = salted_password

def check_password(self, password):
    salted_password = hashlib.sha256(self.salt + password.encode() + self.pepper).hexdigest()
    return salted_password == self.password
```

Figure 4 Password hashing

password VARCHAR(120)	salt VARCHAR(120)	pepper VARCHAR(255)	* admin INTEGER
pbkdf2:sha256:260000$F10	c7b7c7e79f1b70889d7a4dc	7b21e1068c861941f574bd7	0
pbkdf2:sha256:260000$tmJ	0092d037bdeeca60ea0b4c9	d30c0ef6f19148d08c36f657	0

Figure 5 Hashing result

The database is created using Object Relational Mapping, this creates, in effect, a virtual object database that can be used from within the programming language which in this case is Python.

```python
You, 2 weeks ago | 1 author (You)
class TicketPrice(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    ticket_type = db.Column(db.String(50), nullable=False)
    price = db.Column(db.Float, nullable=False)

You, 2 weeks ago | 1 author (You)
class TicketPurchase(db.Model):
    __tablename__ = "ticket_purchase"
    id = db.Column(db.Integer, primary_key=True)
    park = db.Column(db.String(100), nullable=False)
    date = db.Column(db.DateTime, nullable=False, default=datetime.utcnow)
    num_adults = db.Column(db.Integer, nullable=False, default=0)
    num_children = db.Column(db.Integer, nullable=False, default=0)
    total_price = db.Column(db.Float, nullable=False, default=0)
    user_id = db.Column(db.Integer, db.ForeignKey("users.id"), nullable=False)
    ticket_price_id = db.Column(db.Integer, db.ForeignKey("ticket_price.id"), nullable=False)
    user = db.relationship("User", backref="ticket_purchases")
    ticket_price = db.relationship("TicketPrice", backref="ticket_purchases")
```

Figure 6 Database models

Link to the website Git repository: https://git.fhict.nl/I506275/cb01-group2-kabv

THE SERVER MONITOR

The server monitoring system is made using python. First, it uses python 3.10 and some additional libraries such as psutil, matplotlib, numpy etc. The psutil library helps to get all the CPU, RAM, Disk and network data usage data. Then, a rest API formats all the data in a JSON format and sends all across the network. This is done using a python script file. Another python file in a host device gets the data from the network as JSON in a loop. Therefore, the host device gets the data periodically. It then uses matplotlib to plot the data in a real time graph that updates every 10-15 seconds and shows in a graphical interface to the user. Therefore, the server resources can be monitored.