**Case Study – Project Plan**

**“Only Flights”**

A picture containing text, clipart

Description automatically generated

Project Members & Student Numbers:

* Francisco Marcó (4467752)
* Sava Vasilev (4663438)
* Kaloyan Andreev (4408020)

Elaboration Date: 15/09/2022

Group 10

Table of content

[1. Project Definition 2](#_Toc117700694)

[1.1. Background 2](#_Toc117700695)

[1.2. Problem definition 2](#_Toc117700696)

[1.3. Project Goal 3](#_Toc117700697)

[1.4. Expected result 3](#_Toc117700698)

[1.5. Way of working 4](#_Toc117700699)

[1.6. Scope 5](#_Toc117700700)

[1.7. Agreements – made with the tutor 6](#_Toc117700701)

[Professional agreements 6](#_Toc117700702)

[Technical agreements 6](#_Toc117700703)

[2. Project Structure 7](#_Toc117700704)

[2.1. Development Team 7](#_Toc117700705)

[2.2. Tutor 7](#_Toc117700706)

[3. Deliverables 8](#_Toc117700707)

[4. Risk assessments 9](#_Toc117700708)

[5. Work estimates 11](#_Toc117700709)

[6. Roll-out plan 12](#_Toc117700710)

# Project Definition

## Background

This project’s purpose and aim is to develop a working infrastructure and service for our Case Study for the third semester of the ICT & Infrastructure class. Our goal in mind is to fulfill the given requirements to provide the service to our client: flight companies who would like to streamline and ease up creating infrastructure for their digital based operations. The project is developed with those prospects in mind, and we believe that the idea for OnlyFlights can easily provide the minimum and recommended requirements of the Case Study.

The following are the minimum requirements we will follow:

* **Develop**:
  + *Web Application*
  + *Database*
  + *Containers for distributed services*
  + *VPN*
* **Use of AWS provided tools such as Lambda, CloudWatch, CloudTrail, EC2 and S3**
* **Use of automation to easily deploy and scale the operations of our service**
* **Follow security measures such as private key encryption, separated subnets, backup policies, monitoring**
* **Ensure High-Availability**

## Problem definition

Currently, we believe that there is an opportunity to provide business in the airflight industry an easy-to-follow and streamlined process to create and deploy IT infrastructure to help these companies get started faster in working through their operations pipelines. In this way, we would be aiding these companies in developing many essential assets for both internal and external operations (like a webserver or storage server) which can be hard to figure out when you are trying to make a jump into implementing Cloud services.

With our service we would be providing the final customer a simple, clear and satisfying experience for booking flights while at the same time by working jointly with aircraft companies we would save costs in providing our own staff, help with expansion and scalability, and work with the latest cutting-edge cloud technologies.

## Project Goal

The goal of our service is to kickstart our client’s journey into Cloud services and setting up their infrastructure in AWS, by means of automation and infrastructure orchestration. We aim to develop tools for the end user to easily handle their infrastructure while taking care of the configuration and initialization of it.

The main goal of this project is to develop and provide the platform while working with Cloud technologies in the AWS environment.

To ensure this is achieved, we expect to dwell into the respective AWS tools needed to provide each of our services.

In the future, we may explore other Cloud platforms if necessary. But at the time of the elaboration of this document, only AWS is planned to be used.

Finally, we expect to fulfil all the indicated requirements in the Student Workbook. We as a group believe this project to be a good opportunity to apply all of our previous knowledge, while also using it as a platform to apply our newly gained skills in AWS management and automation, orchestration, security, networking and monitoring, and programming.

## Expected result

The project aims to develop and provide an IT IaaS that provides flight agencies and companies infrastructure to manage and host their operations. Our platform mainly allows for management and booking of flights, by providing a secure cloud-based infrastructure hosted in AWS. We will also provide Customer Service tools, such as a Chat Client, to ensure that our product not only improves the client company’s internal operations but also is able to improve the service provided.

We expect to successfully develop the already mentioned platform and applications, applying our knowledge from the course lectures. We also expect to develop stable, functioning security implementations to our infrastructure. Naturally, we want to achieve this result by following and applying the minimum requirements for the project.

Our minimum expectations are to provide the full product with most basic features, while having some degree of security compliance. Because we are working with AWS technologies in a Cloud based environment, we have to ensure a bigger degree of effort as these concepts are very new to us.

## Way of working

We will uphold the AGILE methodology, by working in sprints divided in 2 weeks and handing out MVPs, prototypes, and other deliverables on these deadlines. Furthermore, two peer review moments will occur to look at our progress and provide feedback regarding the service, documentation, and more.

The following is the sprint division, plus the deliverables we will be turning in at the end of each sprint:

* *Sprint 1:*
  + - *Develop Project Plan v1*
    - *Start configuring the network*
    - *Develop URS v1*
* *Sprint 2:*
  + - *Develop Design Document v1*
    - *Complete Network Configuration*
    - *Complete Project Plan – Final Version*
    - *Prototype Release*
* *Sprint 3:*
  + - *Complete URS – Final Version*
    - *Complete Design Document – Final Version*
    - *Develop Test Plan*
    - *Peer review v1*
    - *Develop User Manual*
* *Sprint 4:*
  + - *Develop and complete User and Technical manual*
    - *Develop Test report*
    - *Introduce working infrastructure*
* *Sprint 5:*
  + - *Develop Automation Plan*
    - *Develop Test Plan*
* *Sprint 6:*
  + - *Update Design Document*
    - *Update URS document*
* *Sprint 7:*
  + - *Open for further developments*
* *Sprint 8:*
  + - *Evaluation*
    - *Finish Test Report*

We will hold weekly meetings to discuss the work on the project. We plan to have 2-3 meetings per week on Monday, Tuesday and Friday (with our Tutor). On Monday to discuss progress on project, to adapt the schedule if needed and divide the work for the following week. The meeting on Tuesday will be for internal purposes and on Friday will be for a short meeting with our project tutor.

The rest of the work depending on the type and amongst how many people will be done either during the week or during the weekend (depending on a person’s preferred working time)

Meetings with the project tutor will be to share our progress on the project and ask questions if any arise during the week and to receive feedback and guidance.

## Scope

In the following 6 weeks a fully configured server infrastructure and applications will be delivered. Additionally, we will implement two clients to serve as a demonstration for potential clients. For documentation, we will devise a Project Plan, Design Document, Process Report and User Manual.

## Agreements – made with the tutor

### Professional agreements

At the beginning of the project, the team and the tutor make an agreement about the days when the mandatory meetings will take place every week. The exact time of holding and the place are specified as well. Moreover, it is decided that the materials which will be needed for every conference will be determined a week earlier. An essential part of every meeting will be the “agenda” which will be created the day before the talk. It will include the work that is done through the certain week and details of the meeting. The developers will take turns to make it. In addition, for every meeting there will be 3 roles split among the team members: Role 1 – Chairman – the person who is going to present the weekly completed tasks; Role 2 – Note taker – the one who is going to write down the changes that the team needs make (if there are any); Role 3 – Timekeeper – the team member who will keep track of the time. Additionally, the team developers will have in-person meetings to iron out details of the project, plan meetings with the tutor, and work on the network. These meetings will ideally be held every Tuesday and/or every Thursday, but this is subject to change depending on time constraints as the project develops.

|  |  |  |  |
| --- | --- | --- | --- |
| *Meeting calendar* | | | |
| Time | Date | Place | Materials |
| 10:00 | Every Tuesday/Thursday | Room 3.39 / Third floor / R10 building | Agenda |
| 13:00 (Time subject to change according to tutor’s schedule) | Every Thursday – ONLINE  Or  Every Friday – IN PERSON | OIL 3.40 / Third floor / R10 building | Agenda |

### Technical agreements

During each meeting, both sides – the team and the tutor will discuss the completed weekly milestones and define where changes and improvements have to be made. If the team members have any questions about the project, the software or they are stuck at some point, they are free to ask the tutor for advice. The tutor could give the team guidelines for installation of a specific software that will be used. Technical agreements also include questions about the structure of some of the documentation files.

# Project Structure

## Development Team

Our development team consist of our 3 group members:

* **Francisco Marcó (Developer & Project Manager)**
* **Sava Vasilev (Developer & Janitor)**
* **Kaloyan Andreev (Developer & Network Engineer)**

As Infrastructure students, we are looking forward to expanding our skillset by successfully achieving our goals for the project.

We will be working on the project properly as a team. We will be working using GitLab to assign responsibilities and will cooperate and help each other as all three of us possesses very similar skillsets and capacities to successfully develop the project.

We will conduct a rotating system of roles while presenting during the review meetings with our project tutor, consisting of a secretary (in charge of notetaking and keeping track of the agenda), director (in charge of being the spokesperson during the presentation) and assistant (in charge of providing extra assistance if necessary.)

## Tutor

The tutor for our project is Andrius A. Kuprys. We expect that after each meeting for feedback we will receive the guidance needed to fulfill our project.

* **Email:** [**a.kuprys@fontys.nl**](mailto:a.kuprys@fontys.nl)

# Deliverables

Our proposed deliverables are:

* **A Project Plan detailing a rundown of the project.**
* **A Design Document providing our setup, configuration, and more information regarding the project.**
* **User Manual.**
* **Web application.**
* **Chat client application.**
* **A working Infrastructure according to all requirements needed for our applications to work.**
* **Proper Research Report.**

# Risk assessments

|  |  |  |
| --- | --- | --- |
| **Question** | **Impact** | **Details** |
| AWS: Lambda not functioning | High | * Consult AWS documentation * Revise code * Verify network integrity |
| AWS: Gateway not functioning | Medium | * Check database interaction * Check communication protocols * Verify network integrity * Verify back-end |
| Ansible: Playbook scripts not functioning | Very High | * Consult Ansible documentation * Check code: formatting, parsing * Check keys and IPs |
| Docker: Unable to dockerize functionalities | High | * Consult Docker documentation * Verify network integrity * Verify Image, interaction with modules |
| Automation: Scripts not working | High | * Verify syntax, IPs * Check used language documentation * Consider different tool (CloudFormation, PS, Terraform) |
| Monitoring: Zabbix not monitoring | Medium | * Check Zabbix documentation * Re-install agent * Verify network integrity |
| Monitoring, AWS: CloudWatch not monitoring | Medium | * Check AWS documentation * Verify network implementation * Verify Hosts configuration (e.g: firewall rules) |
| AWS: S3 not functioning properly | Medium | * Check AWS documentation * Check Buckets security group * Check Buckets settings and network interaction |
| AWS: Front-end not working | Low | * Verify code, consult HTML/CSS/JS documentation * Simplify frontend * Verify data pulls |
| AWS: Back-end not working | Very High | * Verify code * Verify Cloud Interaction * Verify network integrity * Consult used language documentation |
| Financial: Running out of budget | High | * Develop cashflow diagram * Analyze the costs, consider alternatives * Evaluate necessary costs |
| Hardware: On premises hardware fail | Low | * Consider moving to Cloud based alternative * Attempt to solve issue considering time constraints * Last resort: abandon inclusion of on-premises service |
| Project: Unable to work as a team | High | * Stablish set meeting times * Communicate troubles * In-person working sessions * Provide internal feedback or criticisms |
| Project: Unable to follow deadlines | Medium | * Attempt to provide core and mandatory deliverables only * Focus on the quality of the final product, set different deadlines * Attempt to change way of working to improve productivity |
| Project: Unable to work with required technologies | High | * Follow lectures * See documentation * Research for internet resources * Consult with tutor/teachers about the encountered problem |

# Work estimates

The working process will be separated into several groups:

* ***Group 1 – Documentation***

According to the Product Roadmap, in each sprint there is a certain document(s) that must be prepared. After that, it will be kept up to date throughout the end of the Case Study Project.

* ***Group 2 - cloud infrastructure***

This is the one of the main parts of the project. Each week new components are implemented depending on their function in the project. It is expected to be developed until the end of the final sprint.

* ***Group 3 - website***

The development will start in week 3 (sprint 2) and the final version, with all functionalities implemented, will be dropped in week 15 (sprint 7).

* ***Group 4 - automation processes***

A Python Tkinter application through which the user will be able to select a folder where his/hers terraform scripts are situated. Once the selection is made the user will have three options – to execute the scrips and to build the AWS infrastructure, to delete the AWS infrastructure built by the selected scripts or to make some changes into the scripts and to execute them again (modify infrastructure). In that way, the client can simplify the creation and configuration of their cloud environment. For every MVP release (in sprint 2, sprint 4 and sprint 8) the certain terraform codes will be prepared. An ansible playbook which will automate the setup of flask website on the webserver will be done by the end of sprint 6.

By the end of week 16 (sprint 7), the project is estimated to be finished and the first clients to start using it.

# Roll-out plan

The application will be online by week 10 (sprint 4) and will be accessible to the public by week 17 (sprint 8). In the meantime, various tests will be performed on the security and connectivity of the cloud infrastructure. Tests will be performed on the response of the website as well.

After the website is launch online, every week there will be new updates which purpose is to fix previously occurred bugs and timeouts. Then new tests will be performed again to ensure the changes have fixed the issues and to find other potential problems if there are any.