

Project Management Plan

Indoor climate control system



CEO: Brice Guayrin

TUTOR: Oswald Figaroa, Xuemei Pu

TEAM MEMBERS (Group #3):

- Victor Covalciuc
- Žana Bašić
- Farros Ramzy
- Sonam Lama
- Andy Verkooijen

TABLE OF CONTENTS

DOCUMENT HISTORY	3
TERMS & ABBREVIATION	3
1. PROJECT DESCRIPTION	4
1.1. CONTEXT	4
1.2. PROJECT GOAL	4
1.3. PROJECT SCOPE	4
1.4. RESEARCH QUESTION	5
1.5. END PRODUCTS & DELIVERABLES	6
2. PROJECT ORGANISATION	8
2.1 TEAM MEMBERS, ROLES AND RESPONSIBILITIES WITH STAKEHOLDERS	8
2.2 COMMUNICATION	10
3. ACTIVITIES AND TIME PLAN	12
3.1 PHASE OF PROJECT	12
3.2 MILESTONES	13
4. RISK MANAGEMENT	14
5. CONFIGURATION MANAGEMENT	16
5.1 TASK MANAGEMENT STRATEGY	16
5.2 SYSTEM MERGING STRATEGY	16
5.3 TESTING STRATEGIES	17
REFERENCE	17

LIST OF FIGURES

Figure 3.1 –	12
--------------	----

LIST OF TABLES

Table 1.1 –	4
Table 1.2 –	6
Table 1.3 –	7
Table 2.1 –	8
Table 2.2 –	9
Table 2.3 –	10
Table 3.1 –	13
Table 4.1 –	14

Document history

Version	Author	Date	Description	Status
1.0	Group #3	26-02-2022	Draft Project Plan	Pending, W.I.P.
1.2	Group #3	05-03-2022	Project Plan	Final
2.0	Group #3	09-03-2022	Draft Revamp Project Plan	Pending, W.I.P.
2.1	Group #3	10-03-2022	Revamp Project Plan	Final

➔ Highlighted in green is the current version on which the document is on.

Terms & abbreviation

SDD	System Design Document
UI	User Interface
MVP	Minimum Valuable Product
TVOC	Total Volatile Organic Compounds
SP	Sprint

1. Project Description

1.1.Context

Airios, founded in 1982, is a systems manufacturer, which produces custom-made electronics for residential climate control. It began as a joint venture with Honeywell, although it is an independent organization today, it specializes in climate control systems, them being the strong suit of the company. Airios has also set their foot into other branches revolving around the control of the environment such as the connectivity, appearances, and inner components of the products. The company strives to create its products like no other on the market by researching customers' needs as well as applying their feedback along the way. The final product is configured with the necessary safety measures as well as the reliability need for the entire system to keep working for a long time.

1.2.Project Goal

The goal of this project is to produce a smart indoor climate control system, which will provide the user with an appropriate environment. The problems the company is being presented with are climate changes which cause the weather to be more unpredictable than ever along with the bad air quality. In this project Airios will attempt to create a system that will automatically regulate the indoor temperature and air quality by comparing it to the preferred user setting and adjusting accordingly.


1.3.Project Scope

Table 1.1 – Showing the features that are inside the scope of the project and features outside the scope of the project that are optional

Inside scope:	Outside scope:
1 Ventilation Box (Simulated App & Physical Fan)	1 Mobile App
2 Room Control UI	2 Scheduler
3 Wireless Communication	3 GPS Sensor
4 Database	4 Damage/Harm Detector
5 Analytic App	
6 SDD	
7 Room Climate Sensors (Temperature, Humidity, CO ₂ , TVOC)	
8 Weather	

1.4. Research Question

Main RQ:

 **How can modern advancements in technology such as new types of sensors and creative thinking can contribute to improving the control of the indoor climate/environment?**

- Information provided by answering this question will show how new technologies and discoveries can improve the quality of life for the end-user of the system. We will manage to answer this question by using such new technologies and observing the result they produce for their intended environment.

Auxiliary RQ:

1. What is it needed for the automatic climate system work properly at a basic level or more?

- For the automatic system to work accordingly and at a decent/basic level it is required that the following function within proper parameters: Detection of CO₂ in the air, automatic ventilation with default presets, UI for the main control panel, Wireless Communication and Temperature/Humidity Sensors. Answering this question will be done by implement all of these sensors together and testing only with the above-mentioned detectors in order to observe the functionality of the system at such level.

2. What is required to make the system adapt to its environment climate?

- In order to adjust to the environment, it is necessary that the Temperature, Humidity and CO₂ sensors are in working parameters i.e., the air quality sensors. Furthermore, the communication between these needs to work in a reliable manner so as not have corrupt or missing data. Answering this question will be done by observing the data from the sensors and finding such manner for the system that it will nicely and autonomously adapt to its surrounding environment.

3. How to make the system communication reliable?

- Creating reliable communication between components will be done by using modern means of data transfer, that being WI-FI communication.

4. How frequently should the control panels receive data?

- The frequency of data supply ought to be in accordance with the importance of the feature to the main functionality of the system.

5. How to make the indoor climate control system responsive?

- This is done by using well implemented manners of working for the systems and high precision sensors for detecting.

1.5.End Products & Deliverables


- ❖ The content of objects delivered at the end of the project will be consisting of the following listed below and in the tables, also below:

MAIN DELIVERABLES

- Indoor Climate Control System
 - Various Sensors such as: CO, CO₂, VOC, Temperature and Humidity.
- Code for The Climate Control Application
 - C Code for the hardware part, C# Code for the control applications.
- Project Plan Document
 - Final Version of the Project Plan
- User Stories
 - Various user stories on how a user would interact with the system.
- System Design Document
 - State Machines
 - Context Diagram
 - Hardware Diagrams
 - Circuit Diagrams of the system.

! MENTION ! - This table will be updated after each meeting with the client where we plan for the next sprint.

Table 1.2 – This tables shows deliverables per SPRINT with their more detailed description.

End Products	Description
 SPO I. Project Plan II. User Stories III. Features	<ul style="list-style-type: none"> • Project description 1.1. CONTEXT 1.2. PROJECT GOAL 1.3. PROJECT SCOPE 1.4. RESEARCH QUESTION 1.5. END PRODUCTS & DELIVERABLES • Project organization 2.1 TEAM MEMBERS, ROLES AND RESPONSIBILITIES WITH STAKEHOLDERS 2.2 COMMUNICATION • Project way of working <ul style="list-style-type: none"> ○ Stand up meeting setups ○ sprint demos ○ communication plan • Timeline & milestones 3.1 PHASE OF PROJECT 3.2 MILESTONES • Deliverables • Risks analysis <ul style="list-style-type: none"> ○ Configuration management: how the team will track the changes in code, files. ○ testing strategies ○ how to branch 5.1 TASK MANAGEMENT STRATEGY 5.2 SYSTEM MERGING STRATEGY 5.3 TESTING STRATEGIES






 SP1 I. Sensors II. MVP III. System Design Document IV. Communication Protocol	❖ Here we intend to deliver working sensor circuits which read and display proper messages according to the environment around them, this should be a MVP. ❖ As for the documentation we intend to begin constructing the documents which describe the design of the system. ❖ We also get a start in creating a communication protocol and maybe finish it.
 SP2 I.	❖
 SP3 I.	❖
 SP4 I.	❖
 SP5 I.	❖

Table 1.3 – Showing What Will Be Delivered (on the left) and When Will it Be Delivered (on the right)

Deliverables	Deadlines
Project plan	14-03-2022
Sprint 0	07-03-2022
System Design Document	20-03-2022
Sprint 1	28-03-2022
Sprint 2	13-04-2022
Sprint 3	11-05-2022
Sprint 4	01-06-2022
Sprint 5 & Final Presentation	20-06-2022

2. Project Organisation

2.1 Team members, roles and responsibilities WITH Stakeholders

Table 2.1 – Table showing all of the people involved in the project together with their contact information and role

Name	Abbreviation	Role and functions	Availability
Brice Guayrin	B. Guaryin	Product Owner of Airios Department	<p>-Monday from 13 pm, Tuesday, Wednesday, Thursday, Friday</p> <p>-Fontys R.10 or b.guayrin@fontys.nl</p>
Victor Covalciuc	V. Covalciuc	Project Team Leader (ScrumMaster)	<p>-Monday & Wednesday, 09.00 -21.00</p> <p>-Tuesday or Thursday, 09.00 -21.00</p> <p>-Friday 09:00 - 16:00</p> <p>-Saturday 10:00 - 12:30</p> <p>-Fontys R.10 or v.covalciuc@student.fontys.nl</p>
Žana Bašić	Z. Basic	Project TeamCo-Leader (Co - ScrumMaster)	<p>-Monday & Wednesday, 13.00 -16.00</p> <p>-Tuesday or Thursday, 16.00 -17.00</p> <p>-Friday & weekends 16.00 - 20.00</p> <p>-Fontys R.10 or z.basic@student.fontys.nl</p>
Farros Ramzy	F. Ramzy	Project TeamMember	<p>-Monday & Wednesday, 13.00 -16.00</p> <p>-Tuesday or Thursday, 16.00 -17.00</p> <p>-Fontys R.10 or f.ramzy@student.fontys.nl</p>



<i>Sonam Lama</i>	<i>S. Lama</i>	<i>Project Team Member</i>	<i>-Monday & Wednesday, 13.00 -16.00 -Tuesday or Thursday, 16.00 -17.00 -Fontys R.10 or s.lama@student.fontys.nl</i>
<i>Andy Verkooijen</i>	<i>A. Verkooijen</i>	<i>Project Team Member</i>	<i>-Monday & Wednesday, 13.00 -16.00 -Tuesday or Thursday, 16.00 -17.00 -Fontys R.10 or a.verkooijen@student.fontys.nl</i>
<i>Pu Xuemei</i>	<i>X. Pu</i>	<i>Tutor</i>	<i>-Fontys R.10 or x.pu@fontys.nl</i>
<i>Oswald Figaroa</i>	<i>O. Figaroa</i>	<i>Tutor</i>	<i>-Fontys R.10 or o.figaroa@fontys.nl</i>

Table 2.2 – Table showing the stakeholders and their roles and reason for being involved in the product

STAKEHOLDERS	IMPLICATION IN PRODUCT	ROLES
CLIENT i.e. CEO	It is the one who is requesting the product in order to bring something new to the market.	PROJECT/PRODUCT OWNER
USERS/CUSTOMERS	They are the ones which profit off recent advancements in technology for their comfort.	PRODUCT END-USER
TUTOR	It is the one who will be our guide into the making of this project.	PROJECT GUIDE
TEAM LEADERS	They are the ones which will manage the team through the course of the project, while also working on it.	PROJECT LEADER & DEVELOPERS
TEAM MEMBERS	They are the one which will be the main builders of the project.	PROJECT DEVELOPERS



2.2 Communication



In order to facilitate the flow of communication in our team we implemented the following means which ought to assure a strong base in our discussions and meetings.

Table 2.3 – Table showing ways of communication in the team and their reasoning

MEANS	REASON
Fontys R10	Meeting with client and physical gatherings for work on project
WhatsApp	Important discussions and essential announcements
Discord	Online gatherings for work and discussion on the project
Microsoft Teams	Backup Platform with the same reasons as Discord
Outlook	Discussions for Sprint Meeting with the Client
Azure DevOps	Version Control Management together with Sprint and Agile Management



The communication within the team and with the client is divided into several different meetings over different mediums as follows:

- **Daily stand-up:**
 - Team meetings are held in person at the University during weekdays to discuss individual hurdles and task progress of each team member. Remote online meetings are organized via discord and Microsoft Teams during weekends.

Attendees:

- Team members

- **Bi-Weekly:**
 - Sprint demo meeting with the client is scheduled every 2 weeks, through emails, depending on the availability of the client. In these meetings a proof of concept for the current sprint is demonstrated followed by backlogs for the next demo to be agreed on with the client.

Attendees:

- Team members
- Client
- Tutor



- **Sprint Review:**

- Pre-demo meetings to prepare for the demo and prioritize backlogs to discuss with the client for the next sprint.
- Post demo meeting to discuss outcomes of the meeting and assign backlogs among team members.
- Remote meetings via Microsoft Teams and Discord server when needed.

Attendees:

- Team members
- Client
- Tutor



3. Activities and Time Plan

3.1 Phase of Project



Figure 3.1 – Picture showing the phases of the project in a summarised and clearer way

✚ Building the product will be divided in 5 phases.

1) PHASE ONE

- This phase consists of preliminary discussions between us. We will be doing mostly brainstorming for any kind of idea regarding the project and planning of it. In addition, we set up the tools for communication in order to facilitate working together in the upcoming planning and execution of the project.
- Moreover, we begin thinking about the more specific requirements of the product, construct user stories and features based on our brainstorming sessions and have an initiatory meeting with the client to get to know each other and set in place our main focus together with the scope and proof of concept to present in the next sprint and phase.

2) PHASE TWO

- In this phase we continue planning of the project, we also begin preliminary work on the base of it in order to construct a solid base for future development to completion.
- We continue focusing more on setting a straight and clear path for future work and development of the project.

3) PHASE THREE

- In this phase, we execute the product as per our client's wishes, together with a good schedule in place so to keep everyone in a nice habitual manner.
- Also, during each sprint, while planning and also executing we have anonymous feedback sessions in order to improve on oneself.
- If execution goes bad at this stage, we have special meetings with the client to discuss a new road for the project from where we would be at that time.
- Also, through the execution of the project our tutor will stand by our side to help us if need be.

4) PHASE FOUR

- In this phase, we focus on finishing the product, having more quality control, and making the aesthetics of it better. Moreover, our tutor will have the final say here as



that will guarantee the functionality of the product together with pleasurable aesthetics.

5) PHASE FIVIE

- This is the last phase where we close the project and deliver the final product approved by the client to his demands. We will hold a presentation where we demonstrate the product together with the client and the tutor.
- At last, all of the documentation, programming text and other files will be made available for everyone one day prior to the presentation.

3.2 Milestones

- ✚ Through the execution of the project separated in sprints we strive and try to reach the following listed points at the end of each phase.

Table 3.1 – Table showing the division of the project across the timespan of it together with the goals for each division

No.	Phase	Milestone
1.	Sprint 0	Project Planning
2.	Sprint 1	MVP of all sensors & SDD
3.	Sprint 2	GUI design & wireless communication
4.	Sprint 3	One room system & cloud/database
5.	Sprint 4	Merged system
6.	Sprint 5	Final product



4. Risk Management

Table 4.1 – Table showing the risks, their probability to happen, impact and measure to try to counteract them if they happen.

Risk	Probability	Impact	Countermeasures
Absent team members due to unforeseeable circumstances <i>E.g:</i> <i>Sickness</i>	HIGH	HIGH	<ul style="list-style-type: none">- Distribute the remaining work package among teammembers.- Git repository, daily stand-up meetings and Kanban board to keep track of assigned tasks and progress.
Technological defects <i>E.g:</i> <i>data loss,</i> <i>Laptop breakdowns,</i> <i>and defective products.</i>	MEDIUM	MEDIUM	<ul style="list-style-type: none">- All code is backed-up in a GIT repository.- Thoroughly test end products before demo or delivery.
Scope Creep <i>E.g:</i> <i>Additional requests of features from the clients existing outside the project scope.</i>	HIGH	HIGH	<ul style="list-style-type: none">- Well defined project scope outlining inside scope and out of scope tasks.- Determine the feasibility of the requests and implement changes, if possible, within the given time frame.



Failure to deliver a complete working end-product	HIGH	HIGH	<ul style="list-style-type: none">- Prioritize important backlogs, highlight bottlenecks, and thorough product tests.
Absent team members due to unforeseeable circumstances <u>E.g:</u> <i>sickness</i>	HIGH	HIGH	<ul style="list-style-type: none">- Distribute the remaining work package among team members.- Git repository, daily stand-up meetings and Kanban board to keep track of assigned tasks and progress.



5. Configuration Management

5.1 Task Management Strategy

- ❖ In this part, the team will split the features into some smaller tasks to work on. Each task should be split equally on their quantities, difficulties, and milestones. The team will use a scrum board to manage this task management per each sprint and will use a repository to organize each update version of their tasks.
- **Scrum Board**
 - The scrum board will have some categories such as: “New”, “Active”, and “Closed” by default, in detail, each of these categories has its own functionalities.
 - In the “New” category, all of the available tasks will be displayed. These are the free tasks that can be picked up by a team member to work on. In “Active”, the board will display every task that is currently being done. There is a developer working on each of their tasks labelled in this category. If a task has been finished, then that task can be just placed in the “Closed” category, so everybody knows that the task is done.
 - However, the scrum board can be extended to have more than these three categories. For example, the team members could add a “Evaluate” category between the “Active” and the “Closed” category, which means someone is still checking and figuring out if something has been missing in a task that was done before it is officially closed.
- **Repository**
 - The repository is a proper place to split, update, and merge a task to work for the industrial project. It is so because of the safe use of it and the easy way to check or go back to the previous version of the task that has been published by multiple versions of the same branch. To control each version for every task, the team will use git for their cloud storage solution.
 - There are two types of branches that the team will use. One of them is the dev branch (<task_name>_dev) where each member does their own tasks for their product features, and the other one is the master branch (<project_name>_master) where the team member should not touch until every smaller task on the active category finished and merged with each other.

5.2 System Merging Strategy

- ✚ During working for this project per task, the team should merge every finished update until these updates become a complete product by the end of this project. There are two ways to do this merging strategy, which are:
- **Merge to Dev**
 - Any developer can merge their dev branches with each other only if these branches are included in the same user story or feature area. The developer must make sure that both of the dev branches which will be merged later have already been reviewed beforehand, and if they want to merge features with completed acceptance criteria of the user story, the dev branch in each task must be completed first.



- **Merge to Master**
 - This part of merging is only allowed when every feature is ready and merged into a one complete product, and that product is already tested before this merging is applied. It is because this merging will include every branch under the master branch into one completed repository.

5.3 Testing Strategies

✚ Testing strategy is a way to ensure the quality of the product before releasing it. The Testing strategy will include:

- **Unit-testing**
 - This basic testing approach is followed by the programmer to test the unit of the program. It helps the developers to know whether the individual unit of their code is working or not.
- **Integration-testing**
 - This type of testing focuses on the construction and the design of the product. The integrated units should be checked if they are working without any errors or bugs before doing this testing strategy.
- **System-testing**
 - This testing strategy checks the main functionality, security, and portability of the product before it goes to the customer. The product should be fully compiled and integration-tested before it is going into this testing part.

Reference

- Airios Webpage
 - <https://www.airios.eu/>
- Google Images & Smartsheet Webpage
 - <https://www.smartsheet.com/blog/demystifying-5-phases-project-management>