System Requirements Specification

Prepared by _____ Date 06/12/19

Sergio Martínez Hamdoun, Requirements Manager

Checked by _____ Date

Xxx, xxx

Approved by _____ Date

Xxx, xxx

Authorized for use by _____ Date

Xxx, xxx

Escuela Técnica Superior de Ingenieros Industriales

Calle de José Gutiérrez Abascal, 2

Madrid, Spain, 28006

Revision Record

Document Issue/Revision Status	Description of Change	Date	Approved
1.0	Initial release	06/12/19	
1.1	Minor correction	29/12/19	
2.0	Requirements correction	17/06/20	

Table of Contents

Definitions

ETSII	Escuela Técnica Superior de Ingenieros Industriales
SE	Systems Engineering
SEPM	Systems Engineering Project Management
UPM	Universidad Politécnica de Madrid
PBL	Project Based Learning
HLO	High Level Objective
RPAS	Remotely Piloted Aircraft Systems
UAS	Unmanned Aircraft Systems
EASA	European Aviation Safety Agency
MII	Master de Ingeniería Industrial
DC	Design Concept
SoI	System of Interest
GCS	Ground Control Station
UGV	Unmanned Ground Vehicle
UAV	Unmanned Aerial Vehicle

1. Introduction

1.1. Purpose

The purpose of this document is to give a deep description of the requirements for the product that is going to be developed. Requirements specification is realized in primarily intended to be proposed to the costumer for its approval. It is also a support document for the SySML requirement model that is going to be used by the developers during the construction of the product.

1.2. Scope

The T-Groundair is a system that tried to improve the training of the tennis players. The principal task is picking the balls of the tennis court. The system is composed by an UAV, an UGV and a server that collaborate.

The UGV and the UAV is connected to the server that receive the values of the sensor, process the data and send the new values for the actuators. The server gives also an user interface for the tennis player to control the behavior of the system.

1.3. Referenced Documents

RD/1	GASSDC19-PM-SEMP-01	Systems Engineering Management Plan
RD/2	ISO/IEC/IEEE 29148	2011 Systems and software engineering
RD/3	GASSDC19-PdM-OpsCon-01	System Operational Concept
RD/4	IEEE 24748-4-2016	International Standard for Systems and Software Engineering

2. Specific requirements

This section has all the requirements of the system with their descriptions and validations.

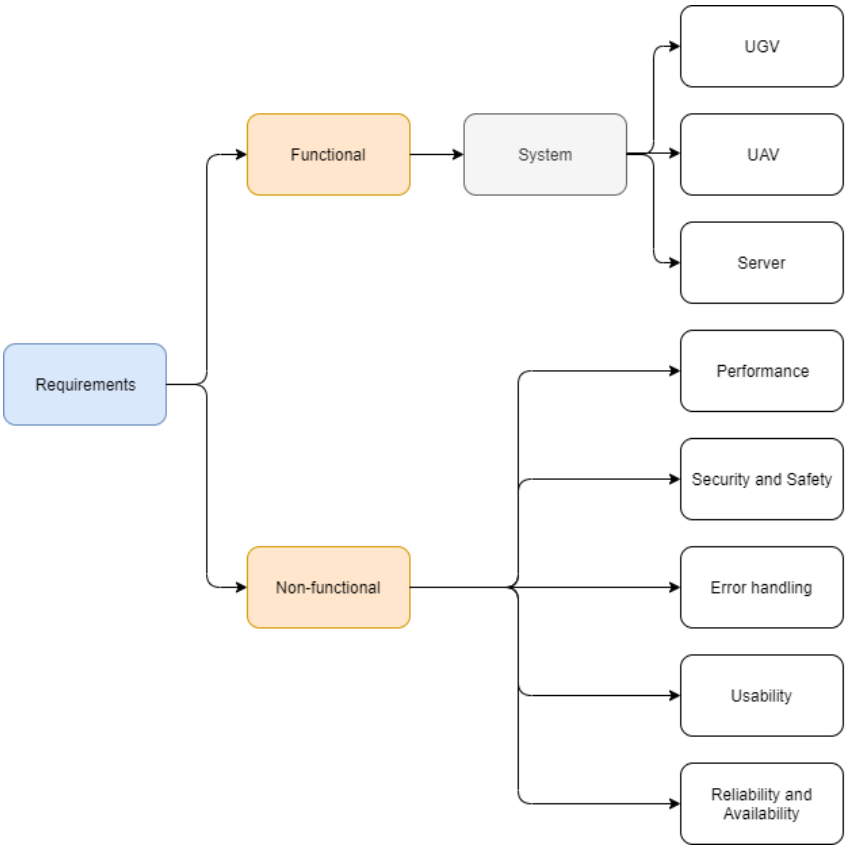

Figure 1. Diagram of the structure of requirements

2.1. ID Code System

All the requirements are associated to a requirement ID that identify it. The ID Code System that will be used is built from the acronym related to the requirement type and a number to differentiate them from the other requirements of the same type.

Example: RFUGXX – **R**equirement **F**unctional **UG**V **XX**

Table 1. Requirements ID

Requirements ID	
Functional requirements	
System requirements	**RFSYXX**
UGV requirements	**RFUGXX**
UAV requirements	**RFUAXX**
Server requirements	**RFSEXX**
Non-functional requirements	
Performance requirements	**RNFPEXX**

Security and safety requirements	**RNFSEXX**
Usability requirements	**RNFUSXX**
Reliability and availability requirements	**RNFREXX**
Error handling requirements	**RNFERXX**

2.2. Requirement format

All requirements will be following the next format:

Requirement ID – Requirement title

Description: Requirement description

Validation: Requirement validation

Example:

RFUA01 – Take Off

Description: The UAV must take off vertically when the systems start working.

Validation: A reference point shall be set in the ground under the UAV before taking off, and once it has reached its height flying point, the alignment of the UAV and the reference point must be checked.

2.3. Functional requirements

Functional requirements include all the necessary requirements of the system and all the basic devices that form part of it.

2.3.1. System's functional requirements

System requirements include all the requirements that stakeholders expected from the entire system.

REFSY01 - Pick up tennis balls

Description: The system must take all the balls of the tennis court.

Validation: When the whole system is ready to be used, throw a ball into the court. The UGV has to take the tennis ball. Moreover, compare the count of tennis balls

drone has detected with count of tennis balls UGV has picked up and verify they're equal.

REFSY02 - Improve training sessions

Description: The system must be able to help the tennis players to improve and optimize the training session

Validation: Demonstration. When the whole system is ready to be used, simulate user operation, throw balls into the court and verify the system is working properly. The court should get empty and balls shall be collected within the same space.

REFSY03 - Help disabled tennis players

Description: The system must be able to help disabled tennis players and facilitate their training.

Validation: Demonstration. When the whole system is ready to be used, simulate user operation, throw balls into the court and verify the system is working properly. The court should get empty and balls shall be collected within the same space.

RFSY04 – User-friendly

Description: The system must be user-friendly for the tennis players and does not require the intervention of a specialised operator.

Validation: We must create a simple interface in order that tennis players can control the system.

2.3.2. UGV functional requirements

UGV requirements specify all the fundamental actions of the UGV that are necessary to the proper functioning of the system.

RFUG01 – Position recognised

Description: The UGV must know its position all the time.

Validation: The UGV has to send its position all the time to the server.

RFUG03 - Avoid obstacles

Description: The UGV must be able to avoid all the obstacles on his trajectory.

Validation: UGV must know in every moment the position so as not to exceed the limits of the track and the controller has to have a function to stop it if he observes that UGV is going to hit an obstacle.

RFUG04 - Turn back to the initial point

Description: The UGV must be able to turn back to the initial point if it will be necessary.

Validation: When UGV is installed on the court and the systems begin to operate, the point at which UGV is located is the origin, this position is stored as a reference in "return function" and the UGV will be sent here when the system restarts.

RFUG05 - Send information

Description: The UGV must be able to send all the information about his position, battery level, speed and sensor lectures to the server.

Validation: Test the algorithm in Gazebo and check that the system sends the information properly.

RFUG06 – Receive information

Description: The UGV must be able to receive the information about its trajectory and new values from the actuators.

Validation: When the UAV sends the information of where the balls are located and therefore the trajectory that has to follow, the UGV must be able to receive this information and also new values from the actuators.

RFUG07 – Acceleration and Velocity

Description: The UGV must be able to control its acceleration and velocity depending on the operation that is performing.

Validation: When the UGV needs, must be able to control its acceleration and velocity depending in the obstacles that has in its trajectory and new values from the actuators.

RFUG08 – Pick up tennis balls

Description: The UGV must be able to take the balls of the court.

Validation: Locate a tennis ball in the simulation and the UGV has to take the tennis ball.

RFUG10 – Ride along a predesigned trajectory

Description: The UGV must be able to ride along a path that the server sent.

Validation: The UGV will sent its position all the time and after following a path we'll check if it has gone for the correct path.

RFUG11 – Reach point not contained in trajectory

Description: The UGV must be able to ride to a point without a trajectory stablished before.

Validation: The UGV must go to a point without a trajectory given.

RFUG12 – Obstacle Detection

Description: The UGV must detect obstacles in its trajectory.

Validation: An obstacle shall be put in the trajectory of the UGV, and if detected, it must send a message to the control center so that we know If the obstacle has been detected.

2.3.3. UAV functional requirements

UAV requirements specify all the fundamental actions of the UAV that are necessary to the proper functioning of the system.

RFUA01 – Take Off

Description: The UAV must take off vertically when the systems start working.

Validation: A reference point shall be set in the ground under the UAV before taking off, and once it has reached its height flying point, the alignment of the UAV and the reference point must be checked.

RFUA02 – Flight height recognised

Description: The UAV must know its flight height all the time.

Validation: The UAV must constantly send the flying height value, recognized by itself to the control center.

RFUA03 – Position recognised

Description: The UAV must know its position all the time.

Validation: define some routes for the drone to follow and store their coordinates. After that, order the drone to take them and keep on sending its current position along all that path, checking whether they are the same as the programmed ones.

RFUA04 – Battery state recognised

Description: The UAV must know its battery state all the time.

Validation: program the drone to send its battery information all the time. It should be shown at the user´s interface in order to validate this requirement.

RFUA05 – Flying along a predesigned trajectory

Description: The UAV must be able to fly along a path that the server sent.

Validation: in a similar way to RFUA03, define some routes for the drone to follow and check whether it follows them or not.

RFUA06 – Reach point not contained in trajectory

Description: The UAV must be able to fly to a point without a trajectory stablished before.

Validation: After setting a destination point, the UAV has to reach it without setting the trajectory beforehand.

RFUA07 – Avoid obstacles

Description: The UAV must be able to avoid all the obstacles that it encounters in this trajectory.

Validation: In the tennis court, a set of obstacles is placed, and a destination is set for the UAV. The UAV must go from its initial position to the destination without colliding with the obstacles.

RFUA08 – Landing

Description: The UAV must be able to land on designated places.

Validation: When the UAV is flying, send an order to make it land. The UAV must land in the designated landing area.

RFUA09 – Auto land

Description: The UAV must be able to realize a safe auto land if it was necessary.

Validation: Test the auto land. In order not to break the UAV, test the auto land in a foam floor and proof that the velocity of landing decreases to zero as the UAV approaches the floor.

RFUA10 – Obstacle detection

Description: The UAV must detect obstacles on his trajectory.

Validation: When the UAV is ready to be used, bring near an obstacle and proof that the UAV's sensor activates and detects the obstacle.

RFUA11 – Tennis ball detection

Description: The UAV must detect tennis balls on the court.

Validation: When the UAV is ready to be used, throw a ball into the court. The UAV must detect the ball with an optical sensor.

RFUA12 – Tennis ball localization

Description: The UAV must obtain the global coordinates of the tennis ball on the court.

Validation: Compare the global coordinates obtained with the UAV, with the ones taken manually.

RFUA13 – Turn back to the initial point

Description: The UAV must be able to turn back to the initial point if it will be necessary.

Validation: When UAV is installed on the court and the systems begin to operate, the point at which UAV is located is the origin, this position is stored as a reference in "return function" and the UAV will be sent here when the system restarts.

RFUA14 – Send information

Description: The UAV must be able to send all the information about his position, battery level, speed and sensor lectures to the server.

Validation: Test the algorithm in Gazebo and check that the system sends the information properly.

RFUG15 – Receive information

Description: The UAV must be able to receive the information about his trajectory and new values for the actuators.

Validation: Send different paths and new actuator values and check if the UAV modifies the path/actuators according to the command.

RFUG16 – Acceleration and velocity

Description: The UAV must be able to control his acceleration and velocity depend of the operation that it is performing.

Validation: Fly the UAV in different situations and the acceleration and speed parameters read by the sensors must be between established values.

2.3.4. Server functional requirements

System requirements include all the requirements that stakeholders expected from the entire system.

RFSE01 – Send information

Description: The server must be able to send all the information necessary for the control of the UAV and the UGV.

Validation: Compare that the information sent by the server and the information received by the UAV match and is necessary for its operation.

RFSE02 – Receive information

Description: The server must be able to receive the information about the status of the UAV and the UGV.

Validation: We must know the UAV and UGV status in a concrete moment and check if these statuses are the same than the ones we receive.

RFSE03 – Start

Description: The server must be able to start all the system.

Validation: We'll send the command to turn on the system.

RFSE04 – Stop

Description: The server must be able to stop all the system if it was necessary.

Validation: We'll send the command to stop the system.

RFSE05 - Restart

Description: The server must be able to restart all the system if it was necessary.

Validation: When the server is all set up, execute the command corresponding to the "restart" function. The command must work, and the server must restart.

RFSE06 - Process the UGV information

Description: The server must be able to process all the information received by the UGV.

Validation: Test the algorithm in Gazebo and check that the system interacts properly. Afterwards, once the system is completed, the validation will be performed experimentally.

RFSE07 - Process the UAV information

Description: The server must be able to process all the information received by the UAV.

Validation: Test the algorithm in Gazebo and check that the system interacts properly. Afterwards, once the system is completed, the validation will be performed experimentally.

RFSE08 - Calculate balls location

Description: The server shall calculate the balls location coordinates from an image received from the UAV.

Validation: Take a picture of the court, process it with the server program and check the results.

RFSE09 – Sort the coordinates list for UGV

Description: The server shall sort the balls coordinates list by proximity for the UGV to follow.

Validation: Enter a list of coordinates to the program and check if the outcome is properly sorted.

RFSE10 - User interface

Description: The server must be including a user-friendly interface to interact with the system.

Validation: Allow someone to use the system in order to check if it is easy to use.

2.4. No functional Requirements

2.4.1. Performance requirements

RNFPE01 - Ball localization precision

Description: The UAV shall detect the ball locate with an accuracy of 3 cm.

Validation: when the UAV system is ready to be used, leave the ball in the court in a random position. The UAV has to detect it location with an accuracy of 3cm.

RNFPE02 - Failure of communications

Description: The communication between the server, UAV and UGV must be constant during the execution of the mission.

Validation: when the whole system is ready to be used, execute a communication for 1 hour without failures.

RNFPE05 – UAV Position and Orientation estimated Error

Description: Control server must estimate UAV position and orientation with less than 3% error.

Validation: Verify and check out, the UAV position and orientation in the simulation program and afterwards, compare the values with the server estimation. Repeat during more than 3 times.

RNFPE06 – UAV position controller

Description: Control server must be able to control UAV with position commands with less than 5% error.

Validation: Repeat for more than three times. Check the distance travelled by the UAV and compare it with the signals commanded by the controller.

RNFPE07 – UAV Velocity Controller

Description: control server must be able to control UAV with velocity commands with less than 5% error.

Validation: UAV velocity may be measured with the onboard sensors and later compared to the given commands.

RNFPE08 – UAV Hovering Action

Description: UAV must be able to keep its current state when no commands were received within 0.5 seconds.

Validation: UAV may take off and be guided to a certain position, and then no further commands may be sent. Then, it's position may be measured to verify its stability.

RNFPE09 – UAV Control Server Communication

Description: UAV must be able to communicate with control server with a bit error rate of 10^-9 or less

Validation: a BERT (Bit Error Rate Test) that uses specific test patterns may be conducted to verify the state of the communication channels.

RNFPE10 – UGV Position and Orientation estimated Error

Description: Control server must estimate UGV position and orientation with less than 3% error.

Validation: check out the UGV position and orientation measuring directly from the simulation program and then compare to the server estimation.

RNFPE11 – UGV position controller

Description: Control server must be able to control UGV with position commands with less than 5% error.

Validation: send a position command to the UGV and compare the reached position to the desired one measuring directly from the simulation program. Repeat this 5 times.

RNFPE12 – UGV velocity controller

Description: Control server must be able to control UGV with velocity commands with less than 5% error.

Validation: send a velocity command to the UGV and see in the real simulation if the UGV moves correctly.

RNFPE13 – UGV Hovering action

Description: UGV must be able to keep its current state when no commands were received within 0.5 seconds.

Validation: Send a basic command to the UGV. After 0.5 seconds, Check that the UGV state does not change its state. Repeat this test with all the basic commands.

RNFPE14 – UGV Control Server Communication

Description: UGV must be able to communicate with control server with a bit error rate of 10^-9 or less.

Validation: Send to the UGV random messages. Check if the messages are well sent and received.

2.4.2. Security and Safety requirements

RNFSE07 – Maintenance

Description: UAV and UGV are properly maintained for safe operation and kept in a secure location when not in use.

Validation: UAV and UGV are stored in a place where only authorized users are allowed access. A log of all the people accessing the system is kept and regularly checked.

RNFSE08 – Legal Standards

Description: System must comply with current legal standards (Real Decreto 1036/2017) and any other commitments that the customer has acquired at all times.

Validation: A detailed list of all requirements set out in Real Decreto 1036/2017 and conditions derived from them is kept. A checklist detailing the fulfilment of each requisite is compiled before project is completed.

RNFSE09 – ODS

Description: System shall pursue sustainable development objectives (ODS) specially the ones regarding health and well-being, sustainable cities, industry and innovation and responsible production and consumption.

Validation: A compiled list of all the ODS objectives and sub-objectives is compiled. System impact on each of the objectives is assessed through the project.

RNFSE11 – Licenses

Description: The user must have all the licenses to use the UAVs within a city.

Validation: The user that will use the drone must show to the PM the licenses.

RNFSE12 – Drone height limits

Description: The flying height of UAV cannot exceed 120 meters, due to the weight of UAV is 500 grams and this height is set by law.

The height will not be necessary to test it because UAV will not pass from the 20 meters height to be able to recognize the balls.

Validation: Validation indoor flight with hovering action at normal height of operation recorded with “Motion Capture Software System” to ensure height limits are not overpassed.

RNFSE13 – Battery recharge

Description: To recharge the battery, the UAV must be completely stopped so as not to damage the user or be damaged.

Verification: Validation on test conditions where the UAV must stop when it is going to recharge the battery.

RNFSE14 – Distance to airports

Description: Fly at a minimum distance of 8 km from any airport or airfield.

Verification: Measure the distance from the tennis court to the nearest airport or airfield.

RNFSE18- Safe Stop

Description: The UGV must be able to execute a safe stop whenever the associated signal is sent by the final user via the server interface.

Validation: When "stop" function is activated the stop will not be abruptly done, it will be done by gradually reducing the speed of the system with the corresponding function, unless the stop is by an emergency and the stop has to be instant.

2.4.3. Error handling requirements

RNFER01 – Connection failure

Description: In case of failure of the connection between the UAV and the server for more than 5 seconds, the drone shall land.

Validation: We'll create a function that will send a command to the UAV ordering it to land if the connection between the UAV and the server is lost for more than 5 seconds.

RNFER03 - Ball position error

Description: UGV must have a maximum pick up time. In case the allowed time is exceeded, the UGV must skip that ball pickup and go for the next one.

Validation: We'll create a function that will send a command to the UAV ordering it to go and pick up the next ball if it takes more than the maximum pick up time to pick up a ball.

2.4.4. Usability requirements

RNFUS04 – Size of the server

Description: The server used to exchange the information between the UAV and the UGV must be easy to use and transport, such as a laptop.

Validation: We must use a laptop as a server in the real simulation to show the most realistic conditions.

RNFUS05 – Indicators

Description: The system should include indicators in order to inform the user about its status.

Validation: We'll have to know the correct status of the indicators and check if the information given by the indicators is the correct one. These indicators will be messages that will show us the correct development of the simulation.

RNFUS06 - Simplified system control

Description: The system controller must have a simple and complete interface.

Validation: We'll create an easy and simple interface to control the system, using several terminals and checking everything easily.

2.4.5. Reliability and Availability requirements

RNFRE01 – Final price of the product

Description: The final price of the simulation should be stablished so that it will be easily affordable by tennis institutions/corporations.

Validation: The final price will be compared with the other solutions of the market and it can't be more than the 5% of the mean price.