Integrative Project of the 4th Semester

Licenciatura em Engenharia Informática, ISEP - 2024/2025 Porto,

June, 2025

Version 3c

Course coordinators

Angelo Martins (AMM) – LAPR4
Ana Madureira (AMD) - LPROG
André Moreira (ASC) - RCOMP
Luis Nogueira (LMN) - SCOMP
Paulo Sousa (PAG) – EAPLI

Other contributors

Margarida Afonso (AFO)

1 Context

In the 2024-2025 academic year, the fourth semester (i.e. 2nd year, 2nd semester) of the Degree in Informatics Engineering (LEI) of the Instituto Superior de Engenharia do Porto (ISEP) adopts a Project-Based Learning (PBL) learning process based on the development of a single project integrating all units of the semester: Applications Engineering (EAPLI), Laboratory and Project IV (LAPR4), Languages and Programming (LPROG), Computer Networks (RCOMP) and Computer Systems (SCOMP). The project, common to all course units, consists of developing the system described in this document in accordance with the general procedures also described in this document.

To develop the project, students must form groups/work teams common to all UC. The objective is to obtain an integrated solution, i.e., a single system encompassing the various components developed.

The operating rules indicated in this document on working methodology and the formation of work teams are guidelines that cover the main existing scenarios. Situations other than those mentioned in this document must be presented to the regent (RUC) of LAPR4, that together with the RUC of the remaining UCs, will decide how to proceed in that specific scenario.

It is important to highlight that the development of project should follow the best practices acquired in the previous semesters, as well as best practices acquired/deepened throughout this semester. The application of these best practices is an integral part of the assessment criteria of the final project.

2 Rules

In the development of project it is mandatory to apply the knowledge and skills acquired in all the semester's UC in an integrated and phased manner as presented in Table 1~~Table 1~~ (the "X" indicates that the UC participates in the respective project sprint).

The development of project will take place primarily in the PL classes of each UC (including LAPR4), which are dedicated the resolution of problems and applications of techniques related with the contents of each UC. Students must follow the indications of their UC teachers regarding the specificities of each UC in the context of the project. In addition, parts of the project must be developed outside of class time.

	EAPLI	LPROG	RCOMP	SCOMP	LAPR4
Sprint 1	X				X
Sprint 2	X	X		X	X
Sprint 3	X	X	X	X	X

Table 1 – UC participating in each sprint

Deadlines (commit on GitHub before):

- Sprint 1 – 6th April 2025, 20:00
- Sprint 2 – 18th May 2025, 20:00
- Sprint 3 – 15th June 2025, 20:00

2.1 Teams

The project is carried out by teams of 4 students of the same PL class having all the units of the semester, as depicted in Table 2~~Table 2~~. Each team will operate as an independent development team and the focus should always be on the project as a whole, not in each UC individually.

	EAPLI	LPROG	RCOMP	SCOMP	LAPR4
Student 1	X	X	X	X	X
Student 2	X	X	X	X	X
Student 3	X	X	X	X	X
Student 4	X	X	X	X	X
Total per UC	4	4	4	4	4

Table 2 – Usual scenario for team composition

However, considering the diversity of LEI students enrolled in the 4th semester, other team compositions may be accepted. If the team members aren't enrolled in all courses, the number of team members can go up to 5 students and should be ensured is that there are at least 3 students enrolled in each of the UCs.

Special situations must be presented to the LAPR4 RUC, which together with the RUCs of the remaining UCs, will decide how to proceed in those specific scenarios.

The students should create the teams during practical classes in the first two weeks (especially in LAPR4), request to the LAPR4 PL teacher to pre-validate/accept the group and then submit it in Moodle. The team is responsible for the accuracy of the information submitted in Moodle.

There will be a repository in Github for each team. These repositories will be created in the context of a Github classroom. The teams will receive information regarding the creation of the repository. Failure to comply with the instructions may negatively affect the team's assessment and the grades. Regarding repository maintenance, as per best practices, Github should be updated on a regular basis as work progresses (e.g. daily updates). Failure to comply – such as doing only 1 or 2 submissions per week, will negatively affect the team's assessment and the grades.

2.2 Work methodology

At the beginning of each sprint, a set of user stories indicating the work to be carried out by the team during the sprint is made available. Thus, the team's responsibility (per sprint):

- analyse this information and, if necessary, request clarifications in a timely manner. Functional questions (which allow, for example, to clarify acceptance criteria) should be asked in sessions with the Client (LAPR4 RUC);
- correctly apply the engineering process and its different activities (e.g., analysis, design, implementation, tests, implantation), leaving the necessary evidence;
- carry out a planning of the tasks to be carried out that takes into account both the sprint objectives and the existence of possible dependencies between tasks and, whenever necessary, review and change the planning carried out previously;
- carry out a balanced division of tasks (e.g., in terms of effort, knowledge and complexity) among all team members in order to ensure that any element from the team is able to describe and explain how a certain requirement has been implemented/satisfied in the system and also be able to implement requested changes (e.g., in an assessment).

Please note that it is not intended nor acceptable for each team member to dedicate themselves just the realization of tasks that require the application of knowledge of a single UC (e.g., EAPLI or SCOMP). On the contrary, each element must participate in tasks that require the application of knowledge from all the UCs in which he/she is enrolled. Students enrolled in LAPR4 must implement requirements from all UCs of the semester.

Students not enrolled in LAPR4 may not be involved in the implementation of requirements from UCs they are not enrolled in, but they must be able to discuss the whole project.

It is also important to emphasize that it is everyone's responsibility to ensure the smooth running of the team. In the event of conflict, it should first be addressed by the team. If the team fails to solve the problem, help should be asked to the LAPR4 PL teacher responsible for monitoring the team.

The team must work regularly (several times a week) and provide evidence of this in Github via commits and issues. These will be used in the assessment.

Some courses may have specific requirements about user story assignment. That is the case of EAPLI in the second and third sprints (Table 3).

Sprint	US	1	2	3	4	5	6
2	US 221 Add a customer representative						X
2	US 222 List customer representatives						X
2	US 223 Edit a customer representative						X
2	US 224 Disable a customer representative						X
2	US 231 List public figures in Figure catalogue		X				
2	US 232 Search Figure catalogue		X				
2	US 233 Add Figure to the catalogue		X				
2	US 234 Decommission Figure		X				
2	US210 Authentication and authorization	X					
2	US211 Register users	X					
2	US212 Disable/enable users	X					
2	US213 List users	X					
2	US220 Register customer	X					
2	US230 Register Show request			X			
2	US235 List show requests of client			X			
2	US236 Edit a show request			X			
2	US240 Drone model creation					X	
2	US241 Add drone to the inventory					X	
2	US242 Remove drone from the inventory					X	
2	US243 List drones in the inventory					X	
2	US245 Add figure category				X		
2	US246 Edit figure category				X		
2	US247 List figure categories				X		
2	US248 Inactivate/Activate a figure category				X		
3	US 310 Create Show Proposal	X					
3	US 315 Add video of simulation to the proposal	X					
3	US 316 Send show proposal to the customer		X				
3	US 317 Mark show proposal as accepted			X			
3	US 318 Templates for show proposals		X				
3	US311 Add drones to a proposal			X			
3	US312 Add figures to a proposal				X		
3	US321 Add maintenance type					X	
3	US322 List maintenance types					X	
3	US323 Edit a maintenance type					X	
3	US325 List maintenance history of a drone						X
3	US326 Add maintenance record to a drone						X
3	US327 Register drone usage time						X
3	US328 List drones needing preventive maintenance						X
3	US340 DSL Plugin	X					
3	US341 Validate figure description		X				
3	US344 Generation of a drone program			X			
3	US345 Drone language Plugin				X		
3	US346 Validation of a drone program					X	
3	US370 Analyse a proposal			X			
3	US371 Accept/reject proposal		X				
3	US372 Check shows dates				X		
3	US373 Get show info	X					

Table 3 User stories per group member in EAPLI (sprint 2 and 3)

2.3 Professional and academic conduct

LEI's integrative projects are essential components in the preparation and training of students for their future profession as software engineers. A software engineer assumes very important responsibilities today, where a large part of activities are supported directly or indirectly by software solutions that

are designed, implemented and maintained by software engineers. In addition to technical skills, the software engineers must follow one global set of principles that are part of the professional code of ethics.

Students must know and apply principles of ethics and professional conduct. Below are some points taken from the Code of Ethics and Deontology of the Portuguese Engineers Association[1]:

- Engineers are aware of the implications of the integration of technical systems in social, economic and environmental context, which is why, in the development of new technologies, they are concerned with criteria and values such as:
  - the sustainability of the technical systems, throughout their life cycle;
  - fitness for use and safety;
  - the contribution to the health and well-being of the citizens;
  - personal development contributing to common good.
- The fundamental guideline in the creation of new technological solutions will be to maintain open the possibility of free and responsible action, both to current and future generations.
- The engineer must oppose to the fraudulent or harmful use of its work.
- The engineer must seek the best technical solutions, considering the economic and quality constraints.

The practice of the principles of good conduct begins in the academic training process itself. Therefore, principles such as accountability and integrity are of utmost importance. Academic integrity principles published by MIT [2] can be considered general and applicable to our academic environment.

On this subject, it is also important that all students are aware of the Regulamento Disciplinar dos Estudantes do Instituto Politécnico do Porto[3].

2.4 Assessment

Each UC participating in the project will carry out, at the end of each sprint, an individual and/or team assessment (formative and/or summative) of the work carried out.

The criteria used in these assessments are defined exclusively by each UC, and they apply to the technical aspects of the project that fit into the its programmatic objectives of each UC. As such, the criteria applied vary and are distinct from one UC to another UC. The results (grades) obtained in a UC do not directly influence the results of the others UC. It's very important that the students apply a suitable work methodology and divide the task taking the assessment in account. Failure to do so may result in negative consequences in the assessment of some UCs.

The assessment of each UC will be in accordance with the respective Curricular Unit Form (FUC) and may include other assessment moments. If these additional assessment moments exist, they will tend to occur before the UC in question begins its participation in the project or after the end of the project development (i.e. during the examination period).

LAPR4 assessment will be focused on the correct application of the scrum methodology and of a iterative and incremental development process. Testing is an important component of this process.

[1] https://www.ordemengenheiros.pt/fotos/editor2/regulamentos/codigo_ed.pdf
[2] http://integrity.mit.edu
[3] https://www.ipp.pt/ensino/informacoes-academicas/regulamentos/Despacho41032013_RegulamentoDisciplinarEstudantesIPP.pdf

3 System specifications

Shodrone it's a company dedicated to providing customized drone multimedia shows. It has a library of figures and allows clients to select sets of them for their personalized shows. It also allows clients to submit proposals for new figures to be used in their super exclusive shows (e.g., a company may ask for the drones to represent its logo in the sky). When submitting a figure, the client may decide whether this will be exclusive - i.e. only available to that client – and if so, for how long would the exclusivity period extend - or it the new figure would be available for all clients.

Shodrone may use dozens to hundreds of drones in a single show. Each figure has a script for each drone involved, i.e. a program routine run by the drone to implement that figure.

The software to be developed aims at supporting most of Shodrone's back office operation:

- Manage clients (register, update, etc.)
- Manage clients' requests (create, analysis and validation, acceptance/rejection, cancelling, exclusivity, etc.). Pricing and billing are excluded.
- Manage library of figures.

To avoid vendor lock-in, Shodrone wants to use a DSL (Domain Specific Language) to describe the figures. The DSL code it's not generated in the system, but in an outside specialized tool. The code is then imported into the system and the generation of each drone's actual code is done using a specific module/plugin.

Finaly, there is also the need to test/validate figures and complete shows:

- Simulation of the code of the figure, which means simulation the operation of all drones involved and check if there are no collisions. A collision is when two drone are in the same place at the same time. If you create a 3D matrix (e.g., 1 cell = 1 cubic meter) and use time-based simulation, it is possible to verify approximately that no collisions will occur.
- Finally, there is also the need to test a full show, which is a composition of multiple figures. In this one, the simulation involves the coordination with a central orchestrator/maestro server, that gives the order to initiate a figure and receives feedback from each drone regarding the completion of a figure.

Shodrone thinks big and aims to become a leader in the drone air show business. One important requirement is to support large, very large air shows. Hundreds of drones flying simultaneously. Thus, the simulation of a large number of drones may not be possible using just one machine. The system should be scalable, so that the simulation can be parallelized. A possibility is to divide the air space of the show in several subareas and use a server for each subarea. The possibility of a drone moving from a subarea to another must be addressed.

3.1 Use cases

The **main use cases**[4] agreed with the client are depicted in Figure 1~~Figure 1~~ and Figure 2. Six main actors were identified:

- Admin
- CRM Manager
- CRM Collaborator
- Show Designer
- Drone Tech

[4] Other use cases might be necessary and added throughout the project implementation. Missing use cases that the team deems necessary must first be validated with the customer.

- Customer (Customer Representative)

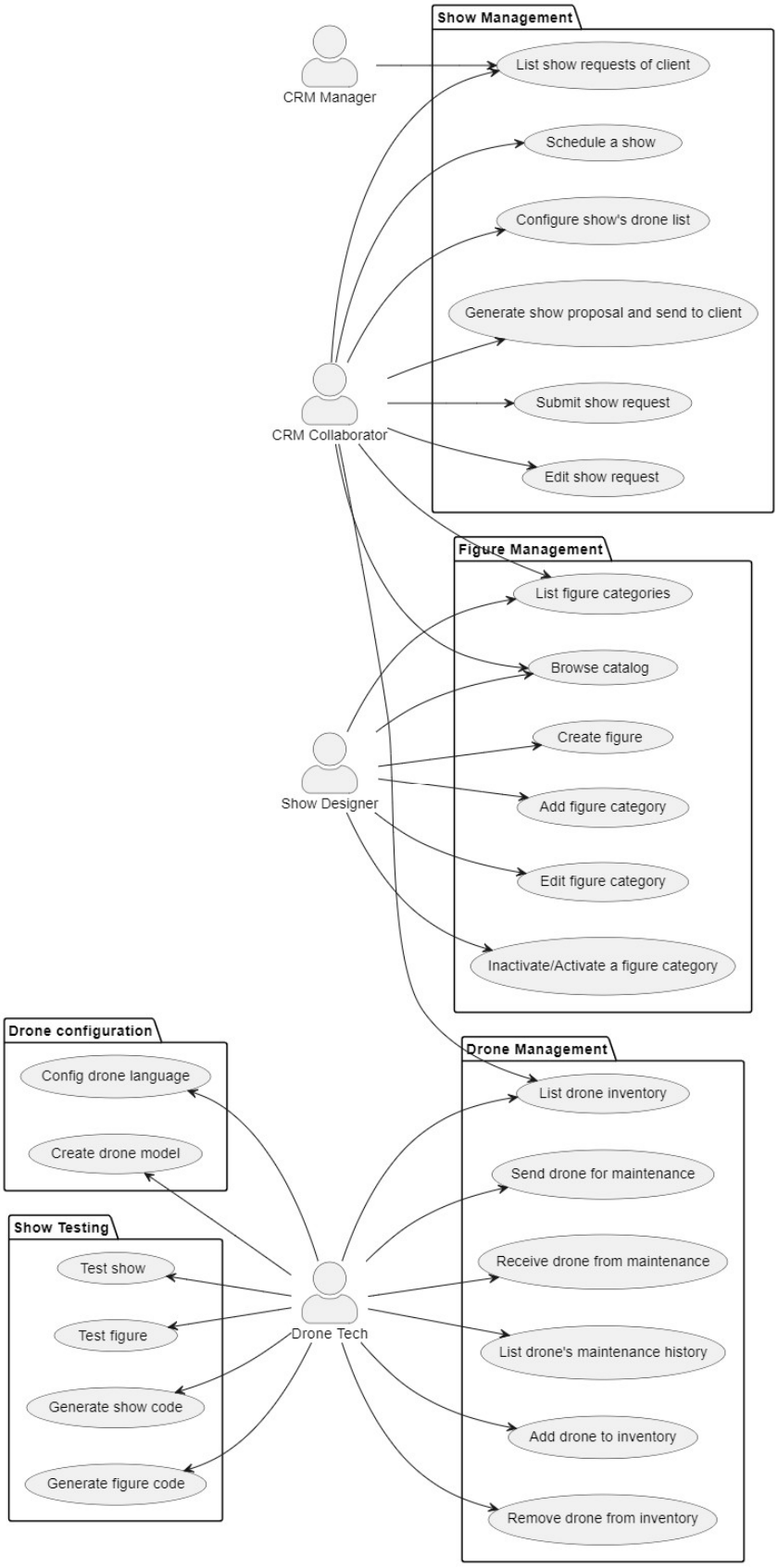

Figure 1 Main use cases

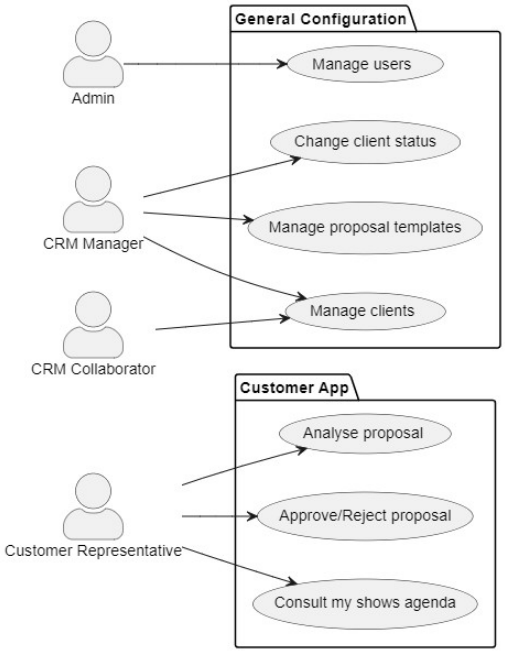

Figure 2 Main use cases (cont.)

3.1.1 Users and roles

A user is someone with access to the system. It is identified by a unique valid email from Shodrone's domain (showdrone.com). A user also has a name and phone number. Users must authenticate into the system to do anything.

At the moment, Shodrone's management hasn't yet decide if a user can have multiple roles in the system. Right now, none has, but that may happen it the future. The number of roles is poised to rise in the future.

3.1.2 Customers

Shodrone works only in the European corporate and public markets. In the system, a customer has the usual information for an enterprise, public entity, etc., including a unique European VAT number, an address, etc.

A customer may have several representatives/employees interacting with the company, thus a list of these representatives must be stored for each customer. Email (valid company email), name and position will probably be enough for the purpose of the system. The set of active representatives may change over time.

A customer will also have status. A CRM collaborator can only submit show request of active customers and "VIP" customers have always priority over "Regular" customers. Only CRM Mangers can register customers in the system and change their status (Deleted, Infringement, Created, Regular, VIP).

3.1.3 Figures and shows

Shodrone is a leading provider of large drone shows. A show is a sequence of figures, and a figure is a composition of a large set of drones, each one running a specific routine. In a static figure, the drones maintain their position in the air, while in a dynamic one they move in the air performing a routine with a fix duration.

Shodrone has a strong commitment to continuous improvement, thus it's creative department is always designing new figures and updating existing ones. The current trend is "the bigger the better", so shows are increasing the number of drones used. But Shodrone also receives specific requests from customers, the most common being to display the logo of the client company in the sky.

Shodrone uses a special tool created in-house to design figures and shows and wants it to generate a figure/show description in a high-level DSL. This neutral description should be used in order to avoid vendor lock in and to be scalable, i.e. to allow its use with different number of drones. To be tested and run, the actual code for each drone must be generated from the high-level description (given the drone's model and its role in the show).

The DSL description of a figure includes sets of geometric figures or 3D bitmaps, but not both in the same figure, and movements applied on them. Currently, Shodrone only supports two types of movements: rotation on an axle and translation. The DSL may evolve in the future to support more advanced features. Each DSL version is identified by an ID. Shodrone believes, in the future, it will to be able to automatically migrate DSL code from previous versions.

Each element of a figure, i.e. a geometric figure or a 3D bitmap, is implemented using a drone type. Please note that the same drone type may be used by more than one element. To generate the drones' code for simulating a figure, one has to supply the actual drone model for each drone type in the figure. The same model can be supplied for all drone types, i.e. use a single drone model in the simulation. One must not forget that, currently, all figures in a show use the same drone set. No drone can be left out. The mapping of the drone types in the figure to drone models used in the simulation is always a surjective function.

Nevertheless, not all drone models have the same capabilities and an error will be generated if an unsupported feature is used when generating a drone's code.

In the system, a figure has a code, a description, a version and, if it is an exclusive one, the client. The figure designer should also be stored as well as the actual figure DSL code and DSL version. Please note that different versions of the figure can be in use. A technical document describing the modelling of figures and shows will be available.

The system must provide the tools to test/simulate figures and full shows.

3.1.4 Show request

A customer contacts a Shodrone's CRM Collaborator to submit a request for a show. The CRM Collaborator creates the show request into the system. If it is a new customer, it must be created in the system first.

A show request includes the customer, place, time, number of drones requested (tentative) and duration. It also should include the show's description, i.e. a document with the sequence of figures from the Shodrone's catalogue and/or request of new figures, as well as customer's exclusivity requirements. As usual, basic workflow information should also be kept (author of the request, history, etc.)

Whenever new figures have to be created, the CRM Manager assigns each request to a Show Designer that, based on the show request, designs the new figures and add them to the system. The CRM Collaborator is then able to generate a show proposal with the figures the customer desires. If the customer accepts the proposal, the CRM Collaborator updates the status of the request and the proposal, and it goes into production.

Upon acceptance of the show proposal by the customer, the show is scheduled by the CRM team. This probably involves some negotiation with the customer. The date and time are stored in the system.

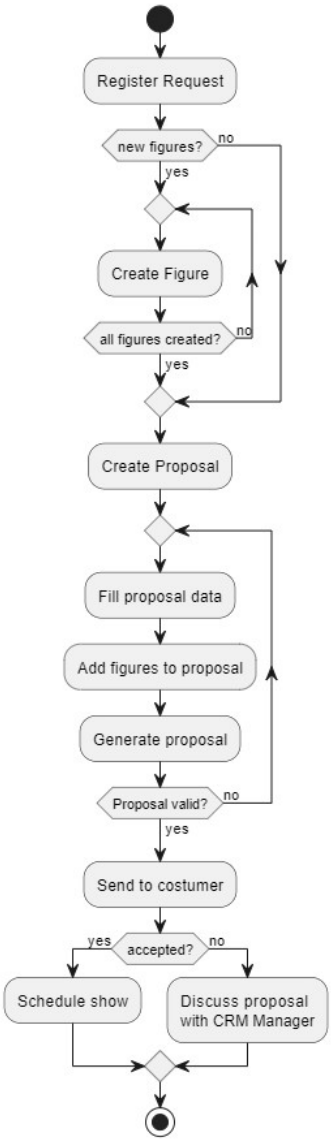

Figure 3 Request and Proposal workflow

3.1.5 Figure and show testing

Shodrone has a strong commitment to quality and safety, thus all figures and shows must be simulated and tested by the Tech department before being deployed. In order to test a show or a figure, a drone technician has to generate actual code of the show/figure from the DSL code and deploy it on a testing environment. The actual drone programming language may be drone brand/type specific.

A show/figure must be tested in a 3D space, a rectangle prismatic volume divided in cubic volumes. The dimensions of the volume must be the same for the whole show, as well as the size of the elemental volume.

In a simulation, the code of all drones involved is run “in parallel” to verify that there are no collisions. An orchestrator is used to synchronize all drones. To ensure safety, two drones cannot be in the same elemental volume at the same time. If this happens during a simulation, it must be halted and the collision reported so that the show/figure is redesigned.

4 User stories

4.1 Sprint 1

US101 Technical constraints

As Project Manager, I want the team to follow the technical constraints and concerns of the project.

These constraints and concerns are described in section 5.

US102 Project repository

As Project Manager, I want the team to use the defined project repository in GitHub and setup a GitHub tool for project management.

US103 Project structure

As Project Manager, I want the team to configure the project structure to facilitate/accelerate the development of upcoming user stories.

Define the structure of the project to support the envisioned architecture, such as presented in Figure 4, including support for adopted technologies (e.g., ANTLR)

US104 Continuous integration server

As Project Manager, I want the team to setup a continuous integration server.

GitHub Actions/Workflows should be used

US105 Automated deployment

As Project Manager, I want the team to add to the project the necessary scripts, so that build/executions/deployments... can be executed effortlessly.

Include scripts for all the major tasks and execution of applications

US110 Domain model

As Project Manager, I want the team to elaborate a Domain Model using DDD.

4.2 Sprint 2

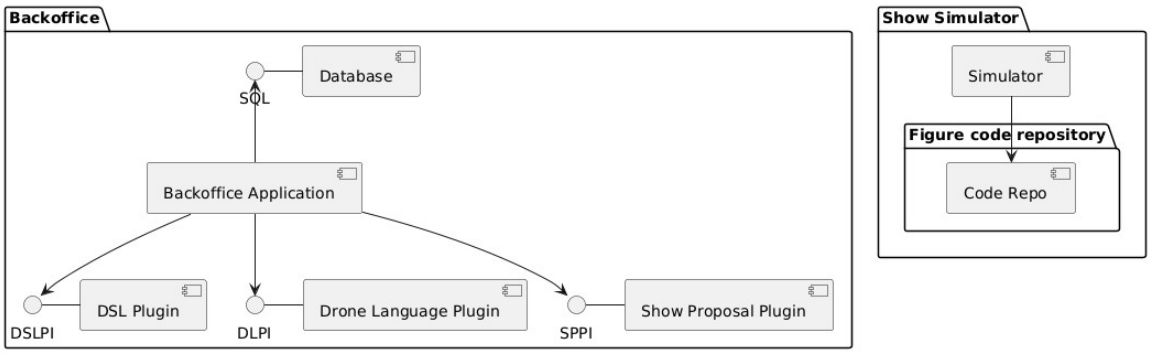

Figure 4 System components in sprint 2

US210 Authentication and authorization [5]

As a Project Manager, I want the system to support and apply authentication and authorization for all its users and functionalities.

US211 Register users

As Administrator, I want to be able to register users of the backoffice.

This must also be achieved by a bootstrap process.

US212 Disable/enable users

As Administrator, I want to be able to disable/enable users of the backoffice.

US213 List users

As Administrator, I want to be able to list the users of the backoffice, including their status.

US220 Register customer

As a CRM Collaborator, I want to register a customer, and that the system automatically creates a customer representative for that customer.

This must also be achieved by a bootstrap process. The customer representative will also be a user of the system (Customer App).

US 221 Add a customer representative

As a CRM Collaborator, I want to register a new representative of a customer.

The customer representative will also be a user of the system (restricted to the Customer App), thus each representative will be a distinct system user. There is no need to verify that customer representative's email in the customer's domain. Actually this could be extremely difficult or even impossible, and many users have emails outside the company's domain.

US 222 List customer representatives

As a CRM Collaborator, I want to list all customer representatives of a given customer. It should not list disabled representatives.

US 223 Edit a customer representative

As a CRM Collaborator, I want to edit the information (email and phone number) of a customer representative. All other information cannot be changed.

US 224 Disable a customer representative

As a CRM Collaborator, I want to disable a customer representative so that they are not contacted in the future.

US230 Show request

As a CRM Collaborator, I want to register (create) a show request.

US 231 Figure catalogue

As CRM Collaborator, I want to list all active public figures in the catalogue so that I can select them during the design of a show request proposal.

[5] You may, and are advise to, use existing frameworks to handle these authentication and user management aspects. They are a necessity but are not the main business focus of the system. The team should work primarily on use cases that have meaningful business value.

US 232 Search Figure catalogue

As CRM Collaborator, I want to search the figure catalogue by category and/or keyword.

The search should ignore accents and shouldn't be case sensitive.

US 233 Add Figure to the catalogue

As Show designer I want to add a figure to the public catalogue.

Figures are classified with a category and a set of keywords. If a figure is custom-made to a customer's request it is not public and can only be used in shows for that customer.

US 234 Decommission Figure

As CRM Manager, I want to decommission a figure from the catalogue so that it will not be used anymore.

US 235 List show requests of client

As CRM Manager or CRM Collaborator, I want to list all show requests of a client.

The show request status information should be provided.

US 236 Edit show requests

As CRM Collaborator, I want to edit a show requests of a client.

Only show requests without a proposal can be edited.

US240 Drone model creation

As a Drone Tech, I want to create a drone model in the system.

This must also be achieved by a bootstrap process. A drone model includes its behaviour under wind as a tolerance of the drone's position (x, y, z) as a function of the wind speed (x, y, z). In a first crude approach, this can be done in steps. Here is an example in a given direction (wind in m/s):

- wind <= 5 -> 0,
- 5 < wind <= 7 -> 0.3 m,
- 7 < wind <= 10 -> 0.5 m,
- 10 < wind <= 15, 0.8 m,
- 15 < wind – not safe to fly.

US241 Add drone to the inventory

As a Drone Tech, I want to add drones of an existing type to inventory. For each drone the serial number has to be stored.

This must also be achieved by a bootstrap process.

US242 Remove drone from the inventory

As a Drone Tech, I want to remove a specific drone from the inventory. The reason for removal and the date must be stored.

US243 List drones in the inventory

As a Drone Tech, I want to list active drones of a given model in the inventory.

US245 Add figure category

As a Show Designer, I want to add a figure category to the figure category catalogue.

The category name must be unique (not case sensitive).

US246 Edit figure category

As a Show Designer, I want to edit an existing figure category in the figure category catalogue.

US247 List figure categories

As a Show Designer or as a CRM Collaborator, I want to list all figure categories in the catalogue.

The category status information should be provided.

US248 Inactivate/Activate a figure category

As a Show Designer, I want to inactivate/activate an existing figure category in the figure category catalogue.

Inactivated categories cannot be used in new figures.

US251 Specification of the language for figure and show description

As a PO, I want a high-level neutral language to describe figures to be specified.

This language will have to support the requirements in section 3.1.3.

US253 Configuration of a drone's language

As a Drone Tech, I want to specify in the system the programming language for a given drone model, so that drones of this model can be used in figures/shows.

A given programming language can be supported by several drone models.

US255 Configuration of proposal templates

As a CRM manager, I want to define a language to specify proposal templates.

US261 Initiate simulation for a figure

As a system user, I want to start a simulation process for a figure so that I can check for collisions before approving it.

Acceptance criteria:

- This component must be implemented in C and must utilize processes, pipes, and signals.
- The system should fork a new process for each drone in the figure
- Each drone process should execute its designated movement script.
- Pipes should facilitate communication between the main process and each drone process.
- The main process should track drone positions over time using an appropriate data structure.

US262 Capture and process drone movements

As a simulation process, I want to receive movement commands from drone scripts so that I can track drone positions over time.

Acceptance criteria:

- Each drone process must send position updates to the main process via a pipe.
- The main process should maintain a time-indexed 3D matrix to track drone positions.
- The system must store past positions to anticipate and detect potential collisions.

US263 Detect drone collisions in real time

As a simulation system, I want to continuously monitor drone positions for overlaps so that I can identify and report collisions.

Acceptance criteria:

- The system must detect when two or more drones occupy the same position at the same time.
- Upon detecting a collision, the system should log the event and notify the involved drones via signals.
- Each drone must handle the received signal and notify the system user with a message.
- When a drone receives a SIGUSR1 (collision detected), it should block other signals while handling it.
- The system should allow early termination if collisions exceed a predefined threshold by sending termination signals to drones.

US264 Synchronize drone execution with a time step

As a simulation engine, I want to synchronize drone movements based on time steps So that I can accurately simulate real-world execution.

Acceptance criteria:

- The simulation must progress step by step.
- Each drone process should send position updates at defined intervals.
- The main process must ensure all updates for a given time step are processed before advancing to the next step.

US265 Generate a simulation report

As a system user, I want to receive a summary of the simulation results so that I can determine if the figure is safe to use.

Acceptance criteria:

- The system must generate a report and store it in a file.
- The report should include the total number of drones and their execution status.
- If collisions occur, the report must list timestamps and positions.
- The report should indicate whether the figure passed or failed validation.

US266 Generate a simulation report

As a simulation engine, I want to incorporate environmental factors such as wind into the drone simulation so that the drones' flight paths become more realistic and adapt to dynamic conditions.

Acceptance criteria:

- The main process must load environmental configuration data (e.g., wind speed/direction) from an external file at simulation start.
- Each drone process receives periodic updates about current environmental conditions via pipes.
- Drone movement scripts are adapted to factor in environmental influences, modifying their paths accordingly.
- The collision detection mechanism is enhanced to consider deviations in drone trajectories caused by environmental factors.

4.3 Sprint 3

One scenario for the system components in sprint 3 is depicted in Figure 5.

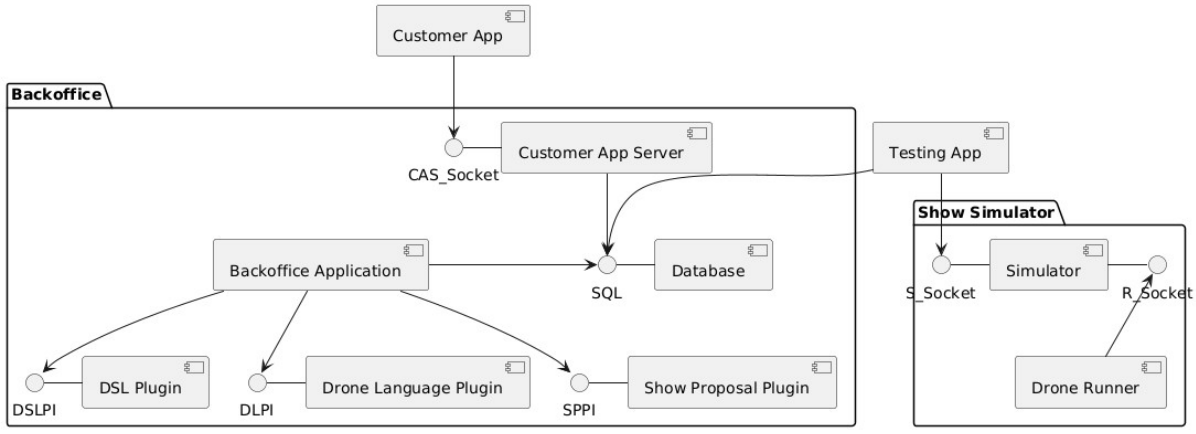

Figure 5 System components in sprint 3

US 310 Create Show Proposal

As CRM Collaborator I want to start the process for creating a show proposal so that we can reply to the customer.

The show proposal includes the total number of drones to be used in the show. Currently, all figures in a show must use all drones.

The show proposal must follow a predefined template.

US311 Add drones to a proposal

As a CRM Collaborator, I want to configure the list of drone models (number of drones and model) of a show proposal.

The drones in the proposal must be compatible with the drones in the Shodrone's inventory. As such, the number of drones of a given type in a proposal cannot exceed the total number of active drones of that type in the inventory. There is no need to verify if these drones are used in another show on the same date.

US312 Add figures to a proposal

As a CRM Collaborator, I want to add one of the available figures to a show proposal.

Any active figure can be added to the show proposal. A figure may have more than one occurrence in a show, but never in two consecutive positions.

For each figure, it must be established the relation between each drone type in the figure and the drone models in the show.

US 315 Add video of simulation to the proposal

As CRM Collaborator, I want to add a video[6] of the simulated show so the customer can have a preview of the show.

US 316 Send show proposal to the customer

As CRM Collaborator, I want to send the show proposal to the customer.

[6] In the scope of LAPR4, the team does not need to actually generate the video and can use any suitable video file.

The proposal to be sent is a properly formatted document with the show details and a link to the video preview. The format of the proposal must be one supported by the system and has to be generated using the correct plugin.

No proposal can be sent to the customer without prior successful testing of its show.

US 317 Mark show proposal as accepted

As CRM Collaborator, I want to mark the proposal as accepted by the customer after it has been accept by a Customer Representative in the Customer App.

US 318 Templates for show proposals

As CRM Manager, I want to be able to configure the template that formats the document to be sent to the customer. The plugin used to validate the proposal document ~~template~~ must be previously registered in the system.

US321 Add maintenance type

As a Drone Tech, I want to add a maintenance type.

The maintenance types apply to all drone models.

US322 List maintenance types

As a Drone Tech, I want to list all maintenance types in the system.

US323 Edit a maintenance type

As a Drone Tech, I want to edit an existing maintenance type.

Only maintenance types without stored maintenance records can be edited.

US325 List maintenance history of a drone

As a Drone Tech, I want to list a drone's maintenance history between two dates.

US326 Add maintenance record to a drone

As a Drone Tech, I want to add a maintenance record to a given drone.

A maintenance record is of an existing maintenance type. Some maintenance operations reset the usage time counter of the drone.

US327 Register drone usage time

As a Drone Tech, I want to record the flying/usage time of a given drone.

US328 List drones needing preventive maintenance

As a Drone Tech, I want to list all drones requiring preventive maintenance in a given date.

A drone is said to require preventive maintenance when its usage exceeds the maintenance threshold defined for that model.

US340 DSL Plugin

As a Drone Tech, I want to deploy and configure a plugin to be used by the system to analyse the figure high-level description.

US341 Validate figure description

As a Show Designer, I want to validate the syntax of the figure description (DSL), so that I can register the figure in the system.

US344 Generation of a drone program

As a Drone Tech, I want the system to use the figure/show high-level description code to generate the code of the drones to be used in the simulation/test.

US345 Drone language Plugin

As a Drone Tech, I want to deploy and configure a plugin to be used by the system to analyse/validate a drone program. There must be a plugin for each different drone language.

US346 Validation of a drone program

As a Drone Tech, I want to validate the syntax of the code for a specific drone in a given show/figure, so that I can later test the figure/show.

US347 Proposal generation

As CRM Manager, I want the system to generate and to validate ~~the proposal template and to generate~~ the proposal document for the show proposal.

US348 Show generation

As a Drone Tech, I want the system to generate the show high-level description from its set of figures, so that I can later generate the actual code for each drone.

US361 Initialize hybrid simulation environment with shared memory

As a Drone Tech, I want to start the simulation with a multi-threaded parent process and multiple child drone processes communicating through a shared memory area, so that the system efficiently coordinates simulation data across processes.

Acceptance criteria:

- The parent process spawns dedicated threads for its functionalities.
- Each drone is launched as an independent process.
- A shared memory segment is allocated and properly initialized for inter-process communication.
- Drone processes are configured to use semaphores for synchronization.
- This component must be implemented in C and must utilize threads, mutexes, condition variables, and signals.

US362 Implement function-specific threads in the parent process

As a PO, I want the simulation controller parent process to have at least two dedicated threads (one for collision detection and one for report generation), so that each functionality operates concurrently and independently.

Acceptance criteria:

- The parent process creates a collision detection thread responsible for scanning the shared memory for drone position conflicts.
- The parent process creates a collision detection thread responsible for scanning the shared memory for drone position conflicts.
- A report generation thread is created to compile simulation results and respond to collision events.
- Any additional thread that you deem appropriate for any of the required functionalities.
- Threads are managed using mutexes and condition variables for internal synchronization.

US363 Notify report thread via condition variables upon collision

As a PO, I want the simulation system collision detection thread to notify the report generation thread through condition variables when a collision occurs, so that the report is updated in real time with accurate collision information.

Acceptance criteria:

• The collision detection thread monitors the shared memory for overlapping drone positions.

• Upon detecting a collision, the thread signals the report generation thread using condition variables.

• The report generation thread, waiting on the condition variable, immediately processes the collision event and logs it.

• Proper mutex locking is used to ensure thread-safe notification.

US364 Enforce step-by-step simulation synchronization

As a PO, I want the simulation engine to synchronize the simulation's step-by-step progression using semaphores, so that all drone processes and parent threads advance in lockstep through each simulation time step.

Acceptance criteria:

- Semaphores are used to control the progression of each simulation time step.

US365 Generate and store final simulation report

As a Drone tech, I want a comprehensive report that details the simulation outcomes — including drone execution statuses, collision events (with timestamps and positions), and overall validation results —, so that I can assess the safety and performance of the figure post-simulation.

Acceptance criteria:

- The report generation thread aggregates data from the shared memory once the simulation concludes.
- The report includes the total number of drones, individual execution statuses, and detailed collision events.
- The final validation result (pass/fail) is clearly indicated.
- The complete report is saved to a file for future reference.

US366 Integrate environmental influences into simulation

As a PO, I want the simulation to incorporate environmental factors such as wind into the drone simulation, so that the drones' flight paths become more realistic and adapt to dynamic conditions.

Acceptance criteria:

- The parent process spawns an additional "environment" thread at simulation start
- This thread loads environmental configuration (wind speed/direction) from a file
- Environment data is written into the shared memory segment at each time step
- The collision-detection thread factors in possible drift caused by wind when scanning for overlapping positions

US370 Analyse a proposal

As a Customer Representative, I want to have access to a show proposal of mine in the App. I received a link/code to download the file.

US371 Accept/reject proposal

As a Customer, I want to accept/reject a proposal using the Customer App. I may provide feedback.

US372 Check shows dates

As a Customer, I want to list my scheduled shows.

US373 Get show info

As a Customer, I want to get the details of a show (scheduled or in the past), including the drone models, figures, duration, etc.

US376 Show testing

As a Drone Tech, I want to select a show to be tested in the Show Simulator, among any of those available at the Backoffice Server.

US378 Running test

As a Drone Tech, I want to connect a Remote Drone running at a remote network node to the Show Simulator.

5 Non-Functional Requirements (NFR)

This section presents some specific non-functional requirements. This includes some constraints and concerns that should be considered when designing and implementing the solution.

NFR01 - Project management using scrum

Scrum should be used for project management. The group's LAPR4 PL teacher will be the Scrum Master and there will be one weekly scrum meeting with the scrum Master during the LAPR4 PL+OT class.

The Development Process document, available at Moodle[7], provides detailed information regarding this requirement.

NFR02 - Technical Documentation

Project documentation should be always available on the project repository ("docs" folder, markdown format) and, when applicable, in accordance to the UML notation. The development process of every US (e.g.: analysis, design, testing, etc.) must be reported (as part of the documentation). Whenever possible, the PlantUML tool shall be used to generate diagrams. The diagrams' source files and the actual diagrams in a vector format (PNG) must be included in the repository.

NFR03 - Test-driven development

The team should aim to adopt a test-driven development approach.

NFR04 - Source Control

The source code of the solution as well as all the documentation and related artifacts should be versioned in a GitHub repository to be provided to the students. Only the main (master/main) branch will be used (e.g., as a source for releases)

NFR05 - Continuous Integration

The Github repository will provide night builds with publishing of results and metrics.

NFR06 - Deployment and Scripts

The repository should include the necessary scripts to build and deploy the solution in a variety of systems (at least Linux and Windows). It should also include a readme.md file in the root folder explaining how to build, deploy and execute the solution.

NFR07 - Database by configuration

The system must support that data persistence is done either "in memory" or in a relational database (RDBMS). Although in-memory database solutions can be used during development and testing, the solution must include a final deployment where a remote persistent relational database is used. The system should have the ability to initialize some default data.

NFR08 - Authentication and Authorization

The system must support and apply authentication and authorization for all its users and functionalities.

NFR09 - Programming language

The solution should be implemented using Java as the main language. Other languages can be used in accordance with more specific requirements.

[7] https://moodle.isep.ipp.pt/mod/resource/view.php?id=258685

NFR10 – Network sockets APIs

The network sockets APIs may either implement a new application protocol developed for this purpose or use an existing standard application protocol, like HTTP.

NFR11 – High-level language (DSL) and drones' language analysis/validation

The support for this functionality must follow specific technical requirements provided in LPROG. The ANTLR tool should be used (https://www.antlr.org/).

NFR12 – Simulation system in sprint 3

Shodrone wants the simulation system to use a multithreaded parent process and child drone processes communicating via shared memory, with functionalities separated into dedicated threads (collision detection and report generation), so that the simulation runs step-by-step with robust synchronization using semaphores and can be terminated via signals when a collision threshold is exceeded.