UNIVERSITY OF Southampton

# Group 23: Deliverable 2: Increment 1

COMP2211: Software Engineering Group Project

Jury D'Alessio (jd3n18), Mikolaj Kolybko (mk2u19), Rowan Kettle (rgk1g19), Velimir Nikolaev Anastasov (vna1u19), Charles Powell (cp6g18), Amir Abbasgholi Ghafghazi (aag1u18)

Electronics and Computer Science
University of Southampton

# Contents

# Introduction

This document discusses the results of the first increment of COMP2211: Software Engineering Group Project, for Group 23. We start by considering the feedback given for the Project Envisioning followed by design details of the first sprint, a product showcase and conclude with product testing, sprint reviewing and planning for the second increment.

# 1 Response to Envisioning Feedback

On March 2nd, Group 23 met with the group supervisor and an additional examiner to discuss the project Envisioning (Deliverable 1) that was submitted in the week prior. The supervisors provided the group with comprehensive feedback on the submitted work, and this section of the report details the response to this feedback, with respect to each of the sections within the Envisioning, as well as some additional comments that were given relating to the report as a whole.

## 1.1 Stakeholder Analysis

### 1.1.1 Feedback

The supervisors found that the group had provided very detailed and thorough descriptions for all stakeholders within the system, and particularly liked that a distinction had been made between the different types of passenger that could be affected by the system (Airline Passenger and Airport Passenger).

### 1.1.2 Response to Feedback

No response necessary.

## 1.2 Persona Analysis

### 1.2.1 Feedback

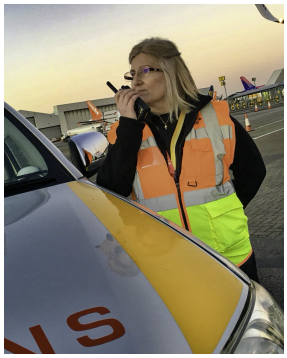
The supervisors were impressed with the level of detail provided in the Personal Analysis descriptions, as well as the range of personas that were discussed. However, the supervisors noted that the group had provided personas for tertiary stakeholders, in favour of providing additional personas for primary stakeholders. Namely, the group had provided a persona for an 'Airline Passenger' , instead of providing an additional persona for an Airline Operations Manager.

The group has acknowledged this feedback, and now appreciates the insight that the persona analysis provides into how the primary stakeholders use/interact with the product. While it is important to consider how other individuals (such as tertiary stakeholders) may be affected by the system, it is more favourable to focus on the primary stakeholders, so that a greater understanding of the product can be gained, leading to a more focused development

### 1.2.2 Response to Feedback

In response to this feedback, another persona was written for the Airfield Operations Manager stakeholder, found below.

**Additional Persona: Camilla: Airfield Operations Manager**



Camilla is a 42 year old Airfield Operations Manager with ten years experience. Camilla has only recently started at her current job, having moved across country to be closer with her extended family.

Camilla takes immense pride in her work and is highly praised by her peers for it. Camilla has excelled by making use of technology in her work whenever possible, and would welcome any new tools that could assist her, provided that they meet the regulations laid out by the CAA. Camilla finds the runway declaration process to be overly time-consuming, as the calculations must be performed by hand by two independent employees. Therefore, she would greatly appreciate a system that is able to assist her in the re-declaration process, by performing the preliminary calculations for her.

One factor that concerns Camilla however, is the over-dependence on external tools and systems. Camilla understands that, while technology can be useful, it is vital to be able to perform the work independently, so that she can go unaffected in the event of a system failure.

## 1.3 Requirements Plan

### 1.3.1 Feedback

The supervisors commented that the requirements plan was structured well, and included effective prioritisation for the product backlog items.

### 1.3.2 Response to Feedback

No response necessary.

## 1.4 Project Plan

### 1.4.1 Feedback

The supervisors were pleased with the provided breakdown of product backlog items across the Increment Plan, but felt that it would be useful to also show the prioritisation of the items within the plan. This should be done for convenience, so that the customer does not have to repeatedly move between the two sections to properly understand how the product backlog has been broken down.

The supervisors also noted that no backlog items marked as "WON'T" had been listed in the increment plan, and that some of these should be considered. However, upon discussion, it was agreed that these backlog items would be left out of the increment plan, as they are considered "extensions" and thus it is likely the group will not have time to complete them. Despite this, as the group is following Agile Methodologies, it is completely reasonable for the backlog items to be developed during a sprint, should time allow for it.

Finally, the supervisors were impressed with the group's use of story points within the 'First Increment Sprint Plan', but noted that a 'Completion Criteria' for each item within the plan would be a useful addition. Completion criteria allow for both the development team and the customer to agree on the conditions upon which the backlog item has been completed. Therefore, when testing the product and presenting it to the customer, the "success" of the product can be directly determined using the defined completion criteria.

### 1.4.2 Response to Feedback

In response to the above feedback, the tables that were shown in the 'Project Plan' have been edited to provide show the prioritisation of the backlog items within them. In addition, a new table has been made as part of the 'First Increment Sprint Plan' that details the completion criteria for each of the backlog items in the sprint.The team shall also provide completion criteria for any further increment plans.

All of these items can be found in the below tables, titled accordingly.

#### Updated Increment 1 Plan

The below table shows the updated increment plan for the first increment.

| **Increment 1 Backlog Items** | **Priority** |
|---|---|
| Declare New Airport (1) | **MUST** |
| Declare New Runway (2) | **MUST** |
| Predefined Obstacle List (3) | **SHOULD** |
| Declare New Obstacle (4) | **MUST** |
| Obstacle Placement (5) | **MUST** |
| Input Error Checking (6) | **SHOULD** |
| Updated Runway Parameters (7) | **MUST** |
| Parameter Comparison (13) | **MUST** |
| Calculation Breakdown (14) | **SHOULD** |

#### Updated Increment 2 Plan

The below table shows the updated increment plan for the second increment.

| **Increment 2 Backlog Items** | **Priority** |
|---|---|
| Runway Side-on View (8) | **MUST** |
| Runway Side-on View TOCS/ALS Slope (9) | **SHOULD** |

| | |
|---|---|
| Runway Top-down view (10) | **MUST** |
| Runway Top-down View Clear and Graded Area (11) | **SHOULD** |
| Simultaneous Runway View (12) | **COULD** |
| Change Thresholds (15) | **SHOULD** |
| Runway Rotation (16) | **COULD** |
| Obstacle Placement Notification (21) | **SHOULD** |
| Updated Runway Parameters Notification (22) | **SHOULD** |

**Updated Increment 3 Plan**

The below table shows the updated increment plan for the third increment.

| **Increment 3 Backlog Items** | **Priority** |
|---|---|
| XML Airport Import (17) | **COULD** |
| XML Obstacle Import (18) | **COULD** |
| XML Airport Export (19) | **COULD** |
| XML Obstacle Export (20) | **COULD** |

**First Increment Completion Criteria**

The below table details the completion criteria that have been generated for the first increment.

| ID | Identifier | Priority | Completion Criteria |
|---|---|---|---|
| **1** | Declare New Airport | **MUST** | The user is able to add a new airport into the system, which then persists in memory until removed. |
| **2** | Declare New Runway | **MUST** | The user is able to add a new runway into the system, which then persists in memory until removed. |
| **3** | Predefined Obstacle List | **SHOULD** | The application provides the user with a pre-defined list of obstacles that can be used in calculations, without the user having to manually input their details. |
| **4** | Declare new obstacle | **MUST** | The user is able to add a new obstacle into the system, which then persists in memory until removed. |
| **5** | Obstacle Placement | **MUST** | The user is able to enter the necessary information to define thee placement of an obstacle on a runway, so that a revision can be performed. |
| **6** | Input Error Checking | **SHOULD** | The application is able to alert the user when invalid/erroneous information is input into the system, and this information is discarded. |
| **7** | Updated Runway Parameters | **MUST** | The user is able to select a runway, obstacle and position, to then receive the updated runway parameters according to this obstacle placement. |
| **13** | Parameter Comparison | **MUST** | When performing runway calculations, the user is shown the original runway parameters with the updated runway parameters, side-by-side in the same view/screen. |
| **14** | Calculation Breakdown | **SHOULD** | When performing receiving updated runway parameters calculations, the user is shown a step-by-step breakdown of how the updated parameters were calculated. |

## 1.5 Risk Assessment

### 1.5.1 Feedback

The supervisors found that the group had included a good range of risks, that were appropriately balanced with effective mitigation strategies. However, the supervisors noted that the 'Risk Assessment' did not include a risk related to communication issues that could develop between the production team (Group 23) and the customer (supervisor).

When developing software for a client, communication is vital. In order for the production team to ensure the customer receives the product they want, the production team and customer must be able to communicate effectively to establish what this is. Therefore, it is important that the risk of communication issues between the group and the customer is discussed, and that appropriate mitigation strategies are considered.

### 1.5.2 Response to Feedback

In response to the above feedback, another risk has been added to the risk assessment that was conducted in the Envisioning. This risk explores the issues surrounding miss-communication between the group and the customer. The new risk can be found in the below table.

**Risk of Miss-communication Between Group and Customer**

| Risk Description | P | S | R.E | Mitigation |
|---|---|---|---|---|
| Miss-communication between production team and customer. | 2 | 4 | 8 | In accordance with Agile Methodologies, regular meetings will be held with the customer during sprints to gain consistent feedback on the product as it is being developed. The team will then be able to reflect on this feedback during the sprint, and ensure that the product being developed meets the customer's expectations. |

## 1.6 Agile Methodologies and Software Tools

### 1.6.1 Feedback

The supervisors were impressed with the level of detail in the descriptions of the Agile Methodologies and software tools that will be used by the team, finding them to be very thorough.

### 1.6.2 Response to Feedback

No response necessary.

## 1.7 Additional Comments

### 1.7.1 Feedback

In addition to the feedback listed above, the supervisors also made some comments on the report/project as a whole. The supervisors mentioned that it would be useful for the group to provide a description of the value that the customer will receive from an increment within the hand-in of that increment. The value that the customer receives from the product at each increment is reflective of the way in which the group has structured the development process (i.e., which backlog items are completed in which increment). Therefore, providing the customer with an explanation of this value could help them to better understand the development process, and thus better appreciate the product that they receive at each stage, enhancing the quality of the feedback they can give as a result.

Furthermore, the supervisors suggested that, at the end of each sprint (i.e., after each hand-in), the team should produce a presentation (i.e., PowerPoint) that demonstrates the work that has been completed in that sprint. While the team had already planned to give a live demonstration of the product, a presentation can help to re-iterate the key features of the application, and gives the group a chance to further explain the decisions that were made during development. Once again, this helps to further the customers understanding of the product, which leads to higher quality feedback, and ultimately a higher quality product.

### 1.7.2 Response to Feedback

In response to the above feedback, the group will now provide a description of the product value within each increment hand-in. The description of product value for this increment can be found in **Product Value**. Furthermore, the group will now create presentations at the end of each sprint to demonstrate the key product features and design decisions to the customer, as to improve their understanding.

# 2 Design Choices

## 2.1 Model-View-Controller Architecture

MVC (Model-View-Controller) is a software pattern mostly used for designing user interfaces that divides its program logic into three connected elements. The idea is to decouple the domain data and its visual representation. The model is usually a one-to-one mapping to real-world objects and concepts. It contains relevant domain or business logic and responds to queries and operations to and on its state. The view represents the model graphically and is usually passive, responding to update operations. It receives user input and passes it on to the controller. The controller translates user actions into operations on the model, which can then be displayed on the view.

In the group's project, the Model contains representations of airports, obstacles, physical runways, logical runways and their associated parameters, obstacle positions, revised logical runways, and flight methods.

The View contains a main dashboard with a calculation panel, runway view and calculation breakdown, and a toolbar that allows the user to manage the persistent storage within the application.

The Controller comprises of controllers for runway displaying, reviewing, error handling, XML handling and one for all other forms in the view. For now, the XML controller is used for saving obstacles, but by sprint, three users will be able to import and export their XML files.

# 3 Design Artifacts

During the first development sprint, The group made use of UML techniques to effectively plan and design the product that would be produced. The resulting artifacts from this design process are presented in this section of the report.

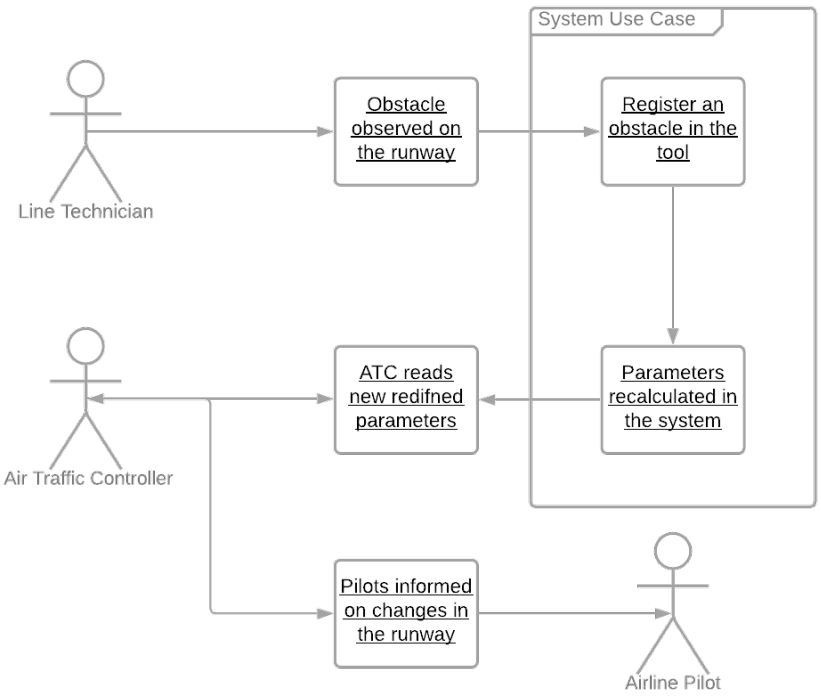
## 3.1 User Scenarios

Several user scenarios were developed during the design process to illustrate how the primary stakeholders could interact with the system, and the value that they gain from using the product. These user scenarios are shown below.

The user scenarios explore how both Line Technicians and Airfield Operation Managers will interact with the system, and consider how the product can be used during their work in order to perform runway re-declarations.

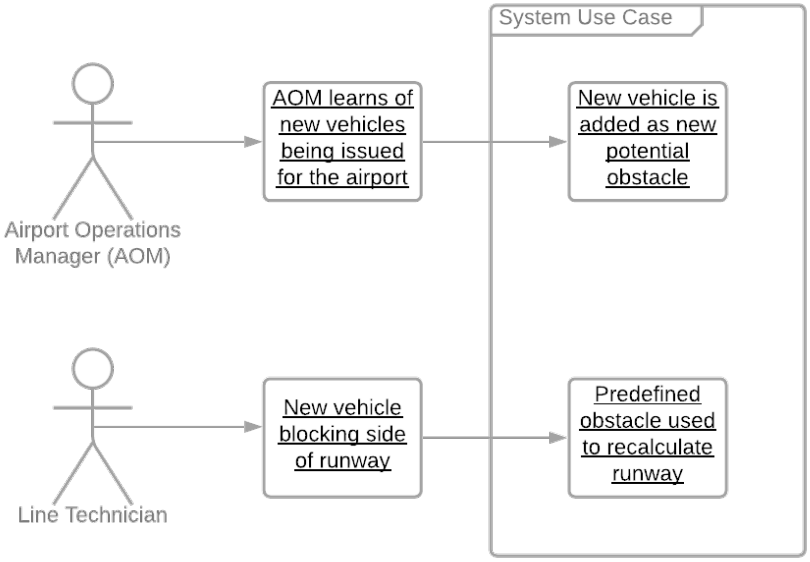
### 3.1.1 Scenario 1: Line Technician Observes Obstacle on a Runway

1. Line Technician notices obstacles on the runway.
2. Line Technician logs the obstacle into the Re-declaration system.
3. The change is updated and changes the runway parameters in the system.
4. ATC is notified of the changes.
5. ATC informs all relevant parties (i.e. Airline Pilots) of the changes to the runway.



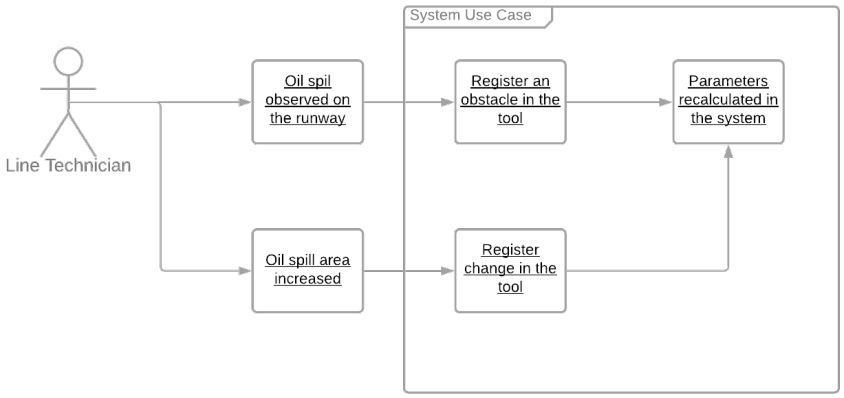
### 3.1.2 Scenario 2: Airport Operations Manager Creates New Obstacle

1. Airport Operations Manager is informed of a new potential obstacle (i.e. new Line Tech vehicle).
2. Airport Operations Manager logs the new vehicle dimensions (width, length, height) as an obstacle.
3. Line Technician sees the new vehicle could affect the runway.
4. Line Technician can easily use the predefined obstacle in recalculation.




### 3.1.3 Scenario: Line Technician Observes Rapidly Changing Obstacle

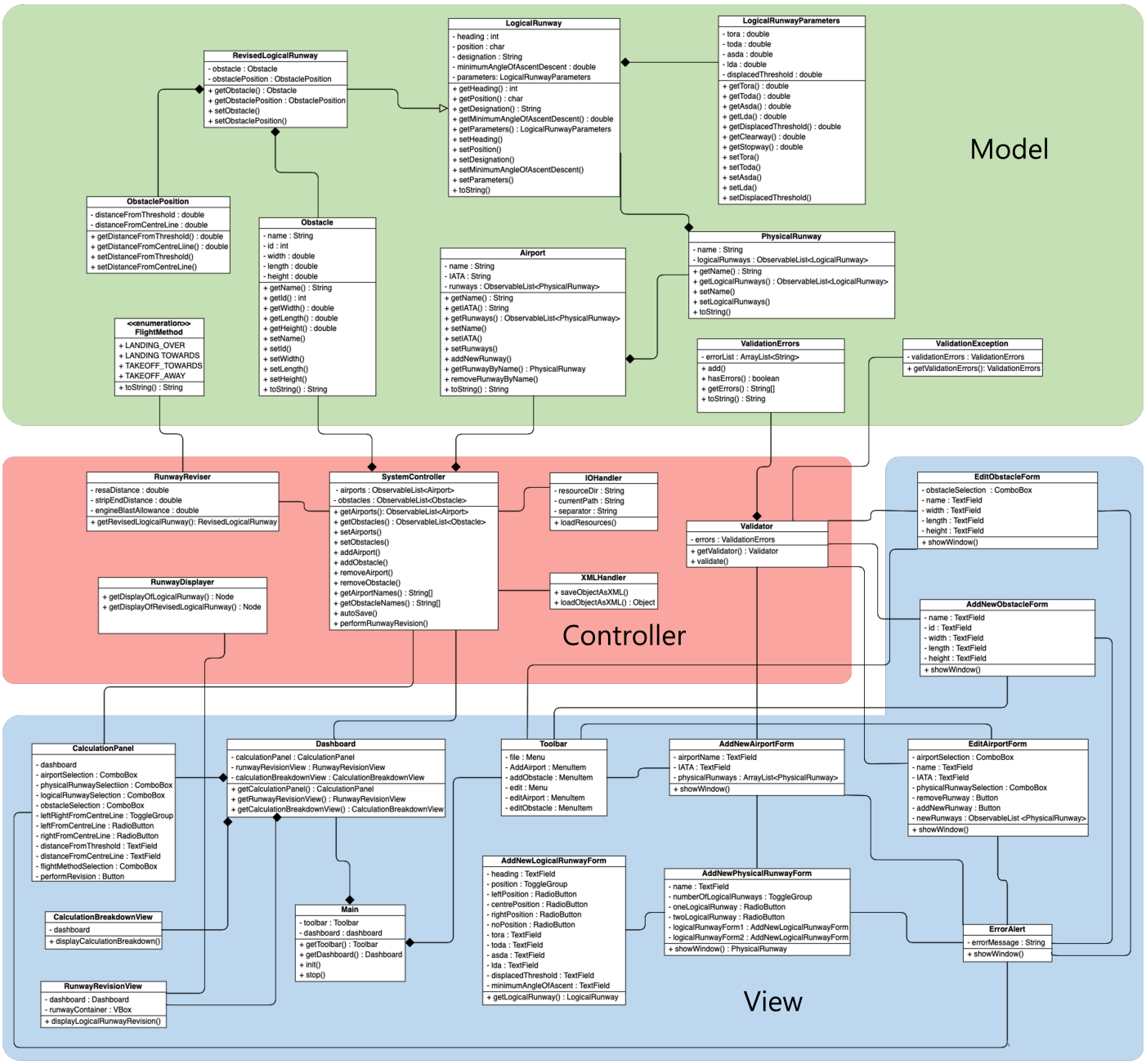
1. Line Technician notices a rapidly changing obstacle on the runway (i.e. an oil spill).
2. Line Technician logs the obstacle into the Re-declaration system.
3. The Re-declaration system calculates new runway parameters based on the obstacle.
4. Line Technician notices a change in the obstacle (spill has increased in size)
5. Line Technician logs the change as a change in the obstacle he just added to the tool.
6. The Re-declaration system calculates the change again based on the obstacles new dimensions.



## 3.2 Class Diagram

A Class Diagram was created during the design process to provide a clear definition of the system structure. This class diagram is shown below.

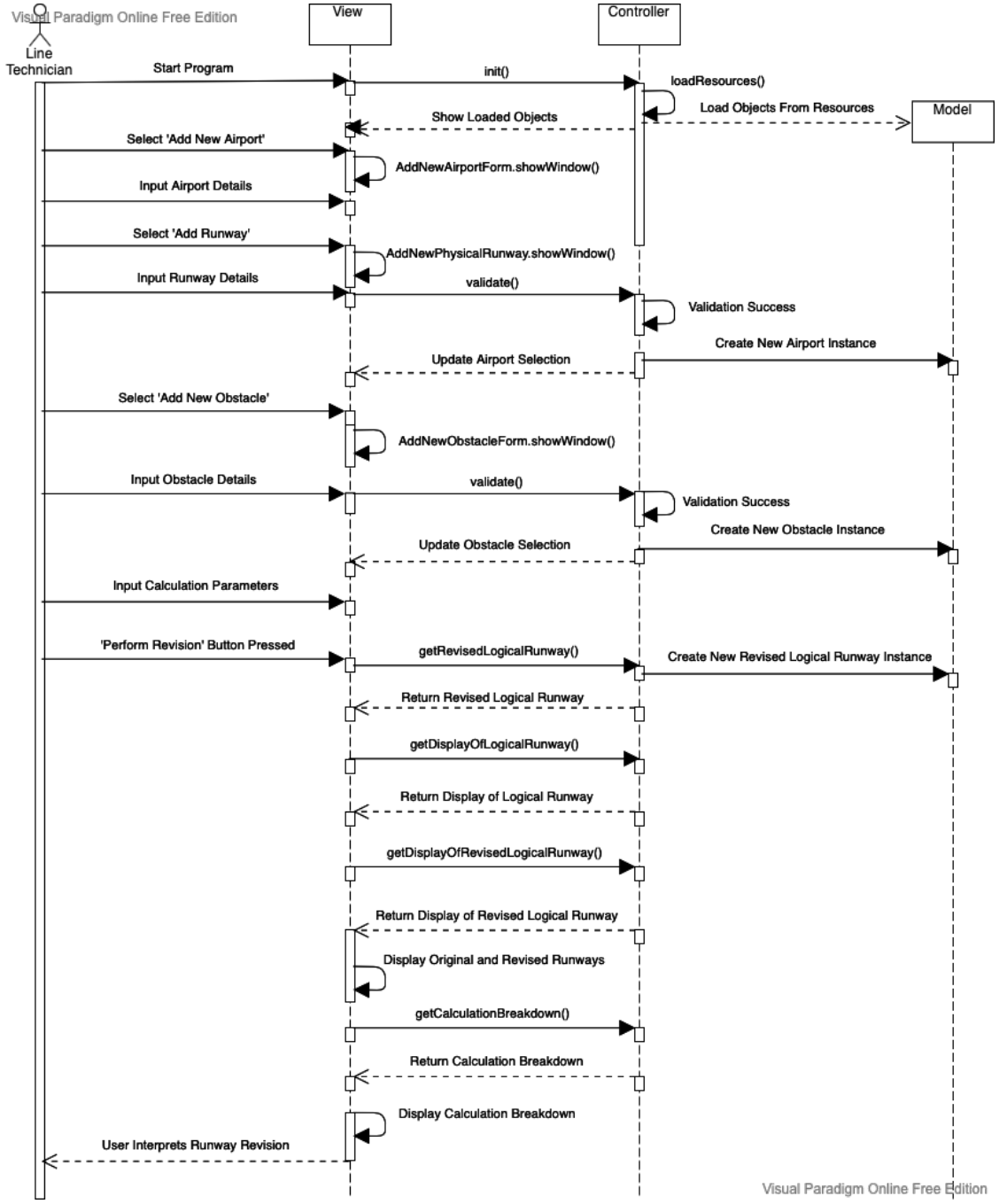
The diagram shows all of the classes within the system, along with their associated attributes and (public) methods. The diagram has also been divided into three sections, to highlight the Model-View-Controller design of the system.



## 3.3 Sequence Diagram

A Sequence Diagram was created during the design process to demonstrate the workflow of the system, and the interaction that may occur between the system and a primary stakeholder. This sequence diagram is shown below.

The diagram depicts a use case scenario, in which a Line Technician opens the application, defines a new airport, defines a new obstacle, and performs a runway revision calculation. The scenario then ends with the Line Technician being displayed the results of the given runway revision. The diagram has been divided according to the Model-View-Controller design approach, as to show how the different sections of the system interact with each other during standard use.




## 3.4 Product Storyboards

Product Storyboards were devised during the design process to provide a clear indication of the ideal system user-interface (View). These story boards are shown below and a short description is provided with each to explain its context and use.

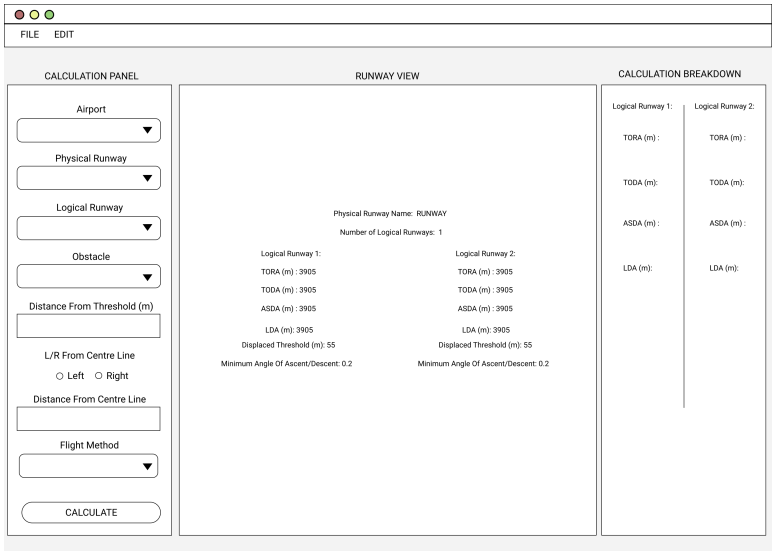
The story boards depict "mock" user-interfaces for each of the different screens within the program. Each "mock" user-interface showcases a core functionality of the program, and provides an insight into how a primary stakeholder would be able to interact with the system.

### 3.4.1 System Dashboard

The system dashboard should operate as the home screen of the application, and present all of the functionally of the product to the user in a clean and understandable way.

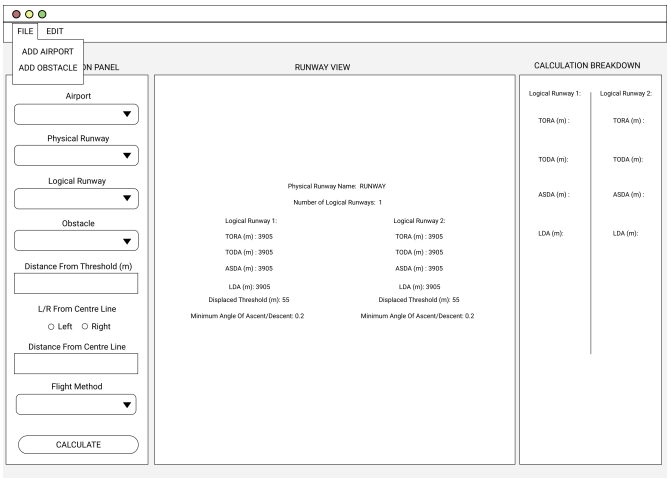
**Dashboard : Main**

The main view of the Dashboard should allow the user to input data and perform calculations, with the results of these calculations being shown in the 'Runway View', and a step-by-step breakdown shown within the 'Calculation Breakdown'.

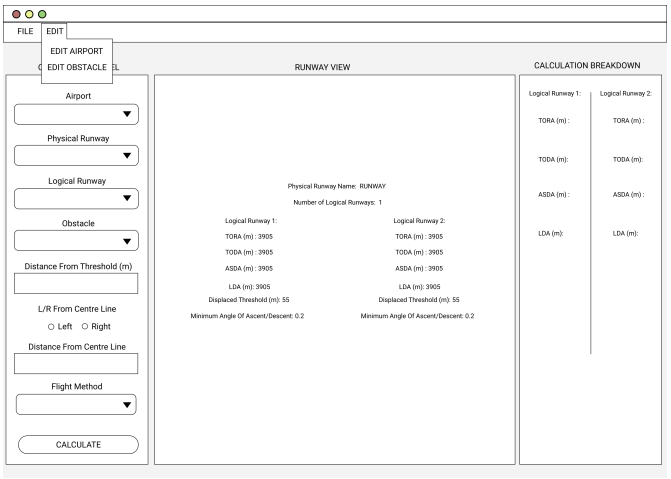



**Dashboard: Toolbar**

Found in the upper portion of the system Dashboard is the Toolbar. The Toolbar contains two menus that should provide management for the persistent storage within the application. The 'File' menu should allow the user to create new airports and obstacles, while the 'Edit' menu should allow the user to edit existing airports and obstacles.




(a) Toolbar: File Menu



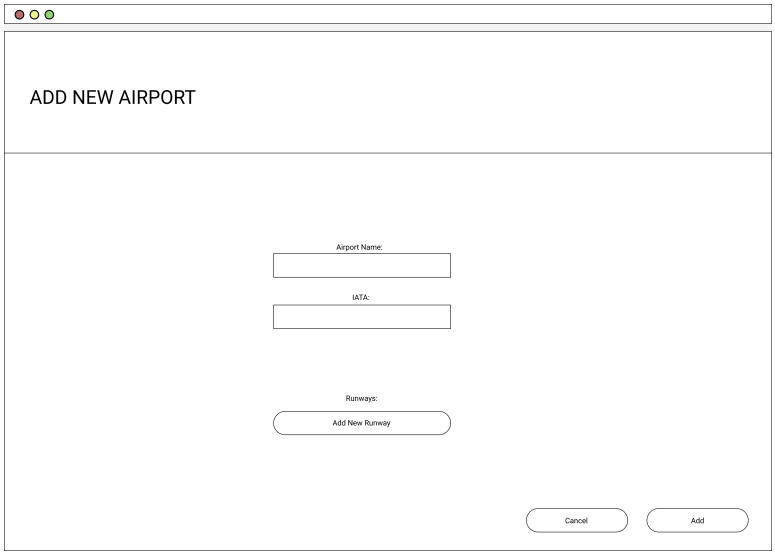

(b) Toolbar: Edit Menu

### 3.4.2 File Menu

As mentioned above, through the File Menu, the user should be able to create new airports and obstacles to use within the application. The user should be shown different forms/screens upon selecting to add either a new airport or a obstacle.

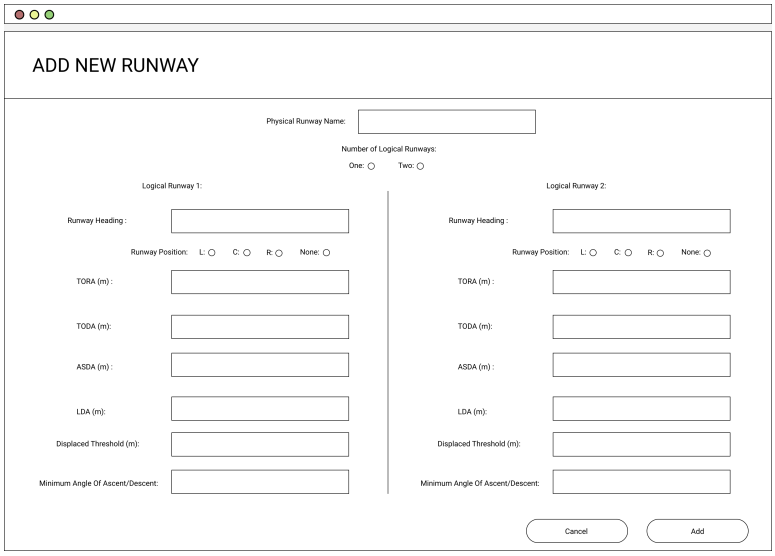
**File Menu: Add New Airport**

Upon selecting the 'Add New Airport' menu item, the user should be shown a form that allows for them to input the details of an airport, including the runways that are contained within it.



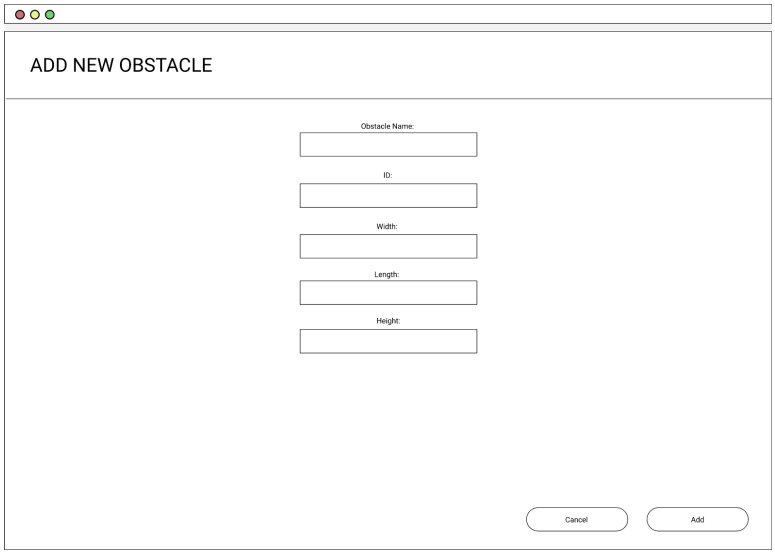

### File Menu: Add New Airport : Add New Runway

Upon selecting to 'Add New Runway' from within the 'Add New Airport' form, the user should be shown a form that allows for them to input the details of an airport runway (both physical and logical).




### File Menu: Add New Obstacle

Upon selecting the 'Add New Obstacle' menu item, the user should be shown a form that allows for them to input the details of an obstacle.

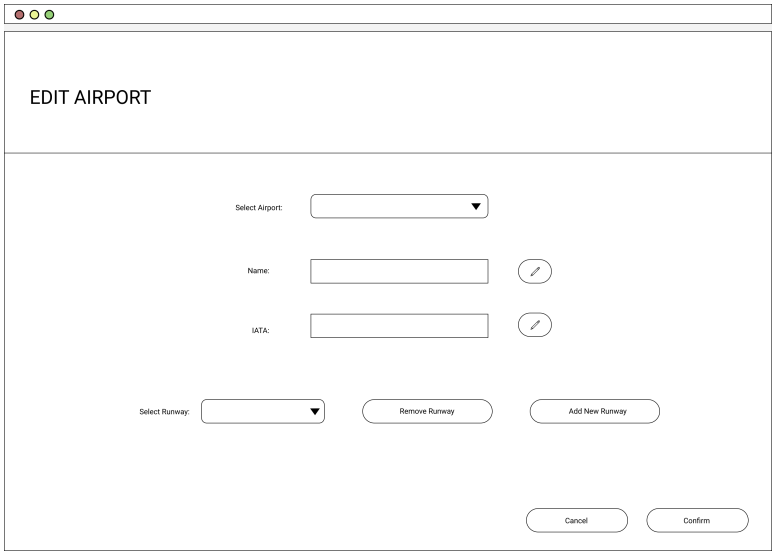


### 3.4.3 Edit Menu

Through the Edit Menu, the user should be able to edit existing airports and obstacles that are contained within the system. The user should be shown different forms/screens upon selecting to edit either an airport or an obstacle.

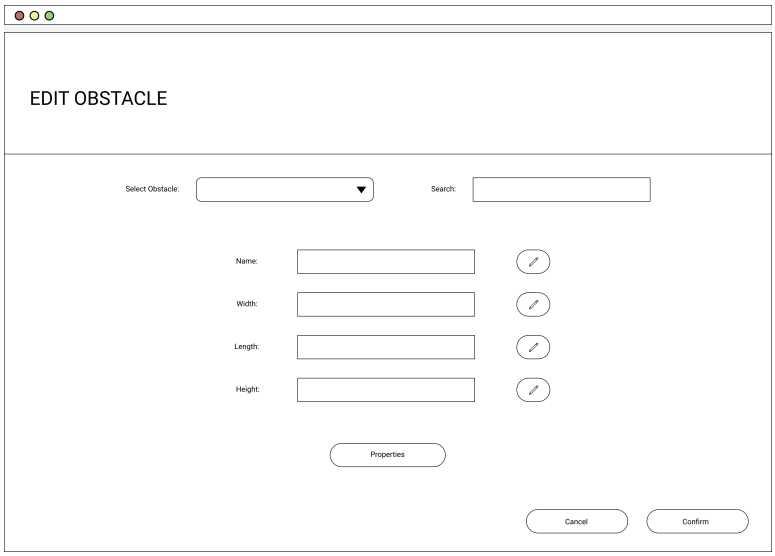
#### Edit Menu: Edit Airport

Upon selecting the 'Edit Airport' menu item, the user should be shown a form that allows for them to select an existing airport, edit it's details, and configure it's runways. When selecting to 'Add New Runway' to the airport, the user should again be shown the form to enter in the details of a runway.




#### Edit Menu: Edit Obstacle

Upon selecting the 'Edit Obstacle' menu item, the user should be shown a form that allows for them to select an existing obstacle and edit it's details.



# 4 Product Showcase

## 4.1 Product Value

As described in the Project Envisioning, the focus of this first sprint was to develop the main system functionality, and the product that has been produced should reflect this. The "main system functionality" refers to the core operations that the system must permit, such as creating and configuring new airports, or performing runway revision calculations, but does not refer to the ability of the program to display this information. Despite focusing on core system operations, the group was still confident that the product needed to have some form of graphical application at this stage, and so has produced a basic GUI that a user can interact with to make use of the system. This GUI will then be updated in future increments to improve the presentation of information, and user experience as a whole.

The Product Demonstration showcases the current system and details it's full capabilities, providing an insight into the product value.

When reviewing the increment 1 product, the focus should be placed on what information is presented to the user, and not on the methods used to present this information. For example, when considering runway revisions, the program currently displays the results of the calculations simply by listing the updated runway parameters along with a textual calculation breakdown. As such, the value of the product should be determined by the fact that the program is able to perform the calculations, and not by the way in which the calculation results are displayed to the user.

The result of this first increment is a product that presents the core system operations to the user, laying a solid foundation for future increments that will focus on the display of information, and in particular, the visual representation of runway calculations.

## 4.2 Product Demonstration

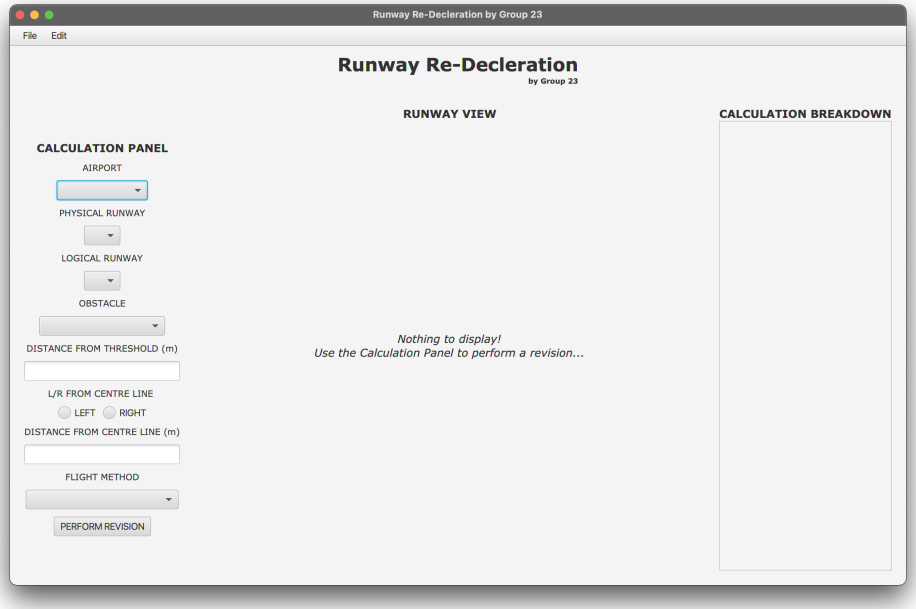
We will now showcase the increment 1 product and demonstrate its capabilities. When demonstrating the system functionality, we will often refer to the product backlog items listed in the increment plan, providing evidence that each item has been successfully implemented during the sprint. This evidence is presented in the form of screenshots, and the completion of a product backlog item can be determined from that said item's completion criteria. Note that, in some cases, the running of the application may be required to ensure that the backlog item has been implemented correctly (e.g., persistence of objects in memory cannot be demonstrated through product screenshots).

### 4.2.1 Dashboard

The dashboard is presented to the user when the application opens, and functions as the system home screen. From the dashboard, the user is able to perform runway revisions, review their results, and configure the persistent storage of the application (i.e., airports and obstacles).

The dashboard can be divided into a 'Toolbar', a 'Calculation Panel', 'Runway View' and 'Calculation Breakdown'. Descriptions of each of these components can be found within Product Storyboards

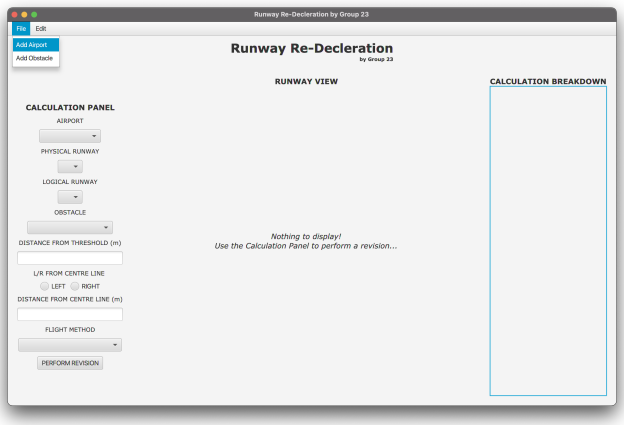
Shown below is a screenshot of the dashboard that the user is shown upon application startup.



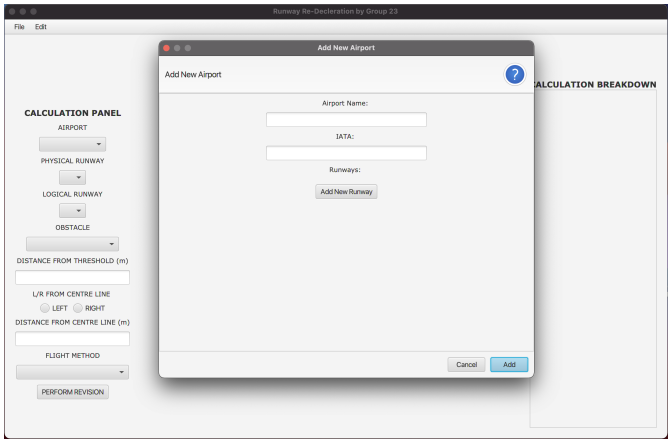
### 4.2.2 Declaring New Airports

| ID | Identifier | Priority | Completion Criteria |
|---|---|---|---|
| 1 | Declare New Airport | MUST | The user is able to add a new airport into the system, which then persists in memory until removed. |

The below screenshots shows the system allowing the user to declare a new airport. The user is able to open the form to configure a new airport by navigating to the 'File' menu within the toolbar. The persistence, of a newly created airport in memory can be seen when running the application.




(a) Toolbar: 'File' Menu

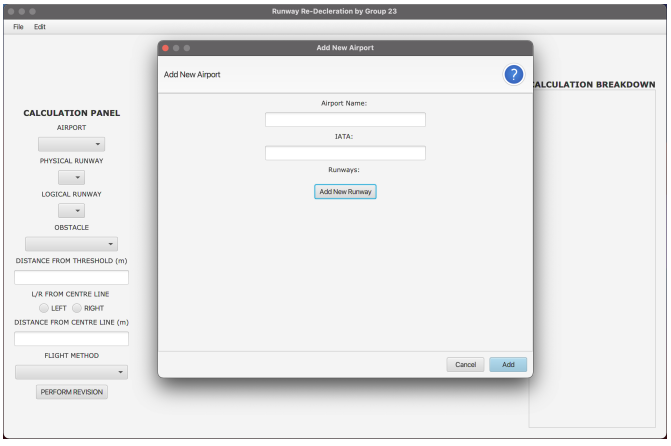



(b) 'Add New Airport' Form

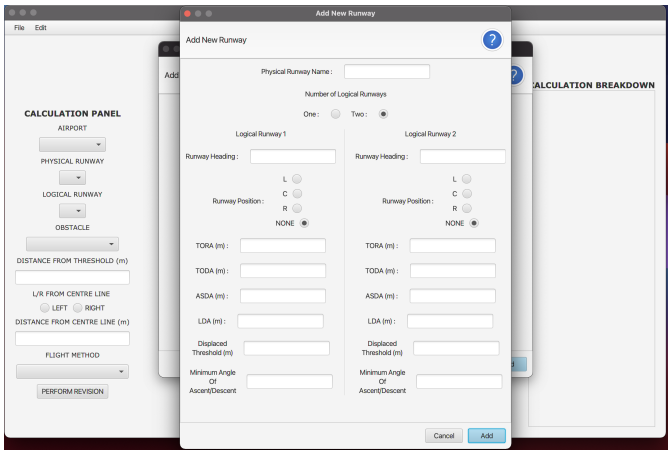
### 4.2.3 Declaring New Runways

| ID | Identifier | Priority | Completion Criteria |
|---|---|---|---|
| 2 | Declare New Runway | MUST | The user is able to add a new runway into the system, which then persists in memory until removed. |

The below screenshots shows the system allowing the user to declare a new runways. The persistence, of a newly created runway in memory can be seen when running the application.



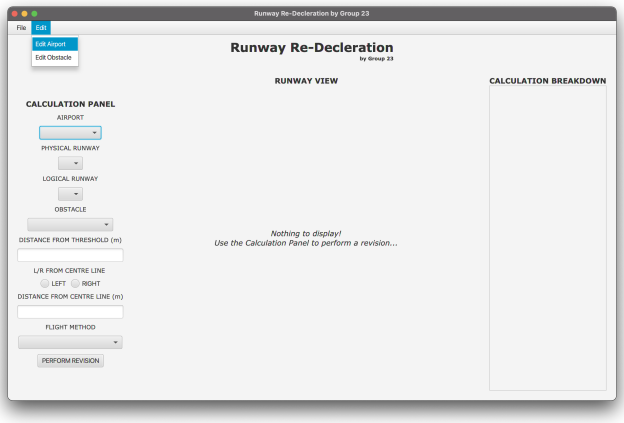

(a) 'Add New Airport' Form



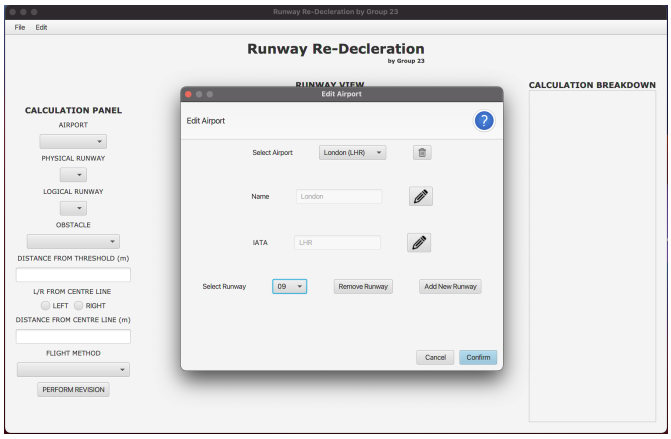

(b) 'Add New Runway' Form

### 4.2.4 Editing Existing Airports

The below screenshots shows the system allowing the user to edit the details of an existing airport. The user is able to open the form to configure an existing obstacle by navigating to the 'Edit' menu within the toolbar. From within the form, the user is able to select an existing airport, and can then configure the airport parameters, add/remove runways from the airport and remove the airport from the system entirely.




(a) Toolbar 'Edit' Menu

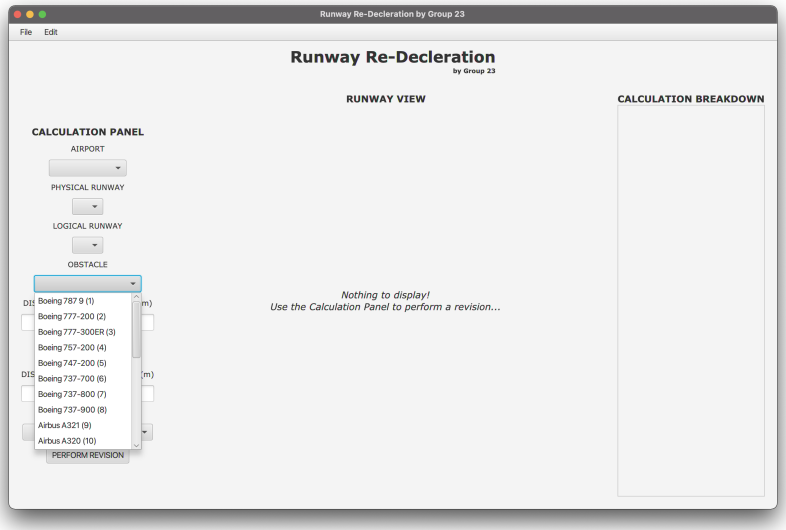



(b) 'Edit Obstacle' Form

### 4.2.5 Predefined Obstacle List

| ID | Identifier | Priority | Completion Criteria |
|---|---|---|---|
| **3** | Predefined Obstacle List | **SHOULD** | The application provides the user with a pre-defined list of obstacles that can be used in calculations, without the user having to manually input their details. |

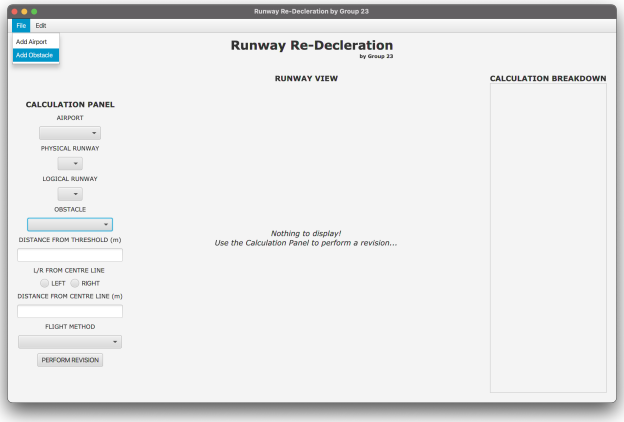
The below screenshot show the some examples of the selection of obstacles that are predefined within the system, and ready for use in runway revision calculations.



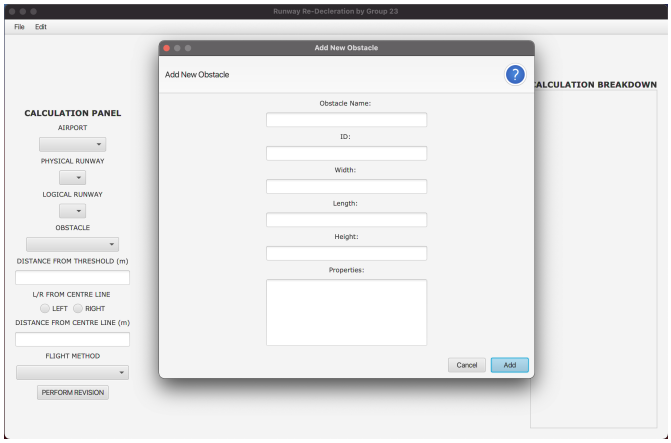
### 4.2.6 Declaring New Obstacles

| ID | Identifier | Priority | Completion Criteria |
|---|---|---|---|
| 4 | Declare new obstacle | **MUST** | The user is able to add a new obstacle into the system, which then persists in memory until removed. |

The below screenshots shows the system allowing the user to declare a new obstacle. The user is able to open the form to configure a new obstacle by navigating to the 'File' menu within the toolbar. The persistence, of a newly created obstacle in memory can be seen when running the application.



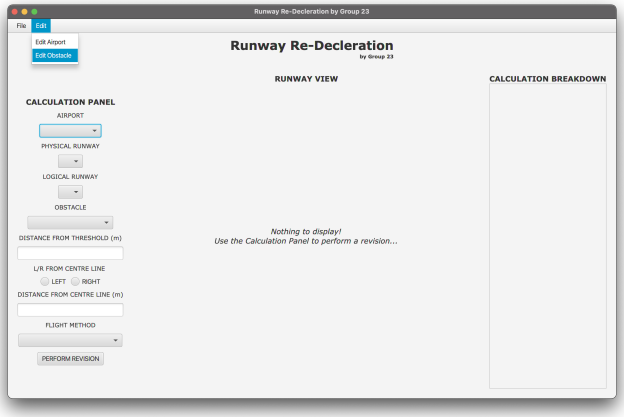

(a) Toolbar 'File' Menu



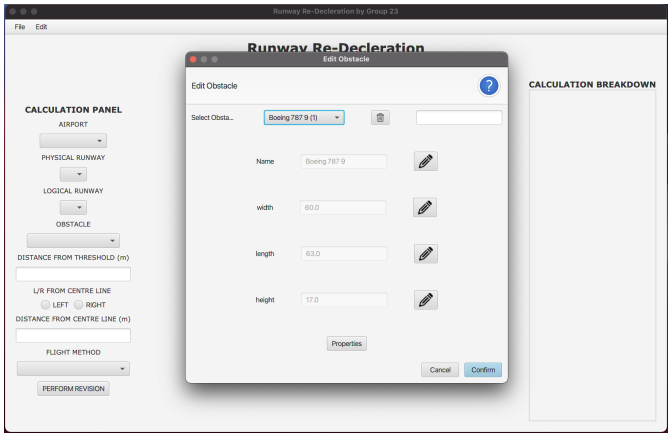

(b) 'Add New Obstacle' Form

### 4.2.7 Editing Existing Obstacles

The below screenshots shows the system allowing the user to edit the details of an existing obstacle, runways. The user is able to open the form to configure an existing obstacle by navigating to the 'Edit' menu within the toolbar. From within the form, the user is able to select an existing obstacle, and can then configure the obstacle parameters, or remove the obstacle from the system entirely.




(a) Toolbar 'Edit' Menu

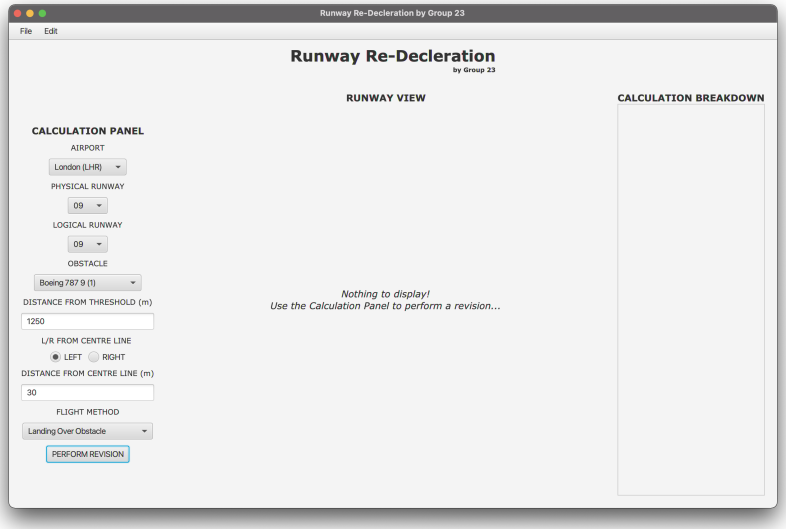



(b) 'Edit Obstacle' Form

### 4.2.8 Positioning Obstacles for Runway Revisions

| ID | Identifier | Priority | Completion Criteria |
|---|---|---|---|
| **5** | Obstacle Placement | **MUST** | The user is able to enter the necessary information to define thee placement of an obstacle on a runway, so that a revision can be performed. |

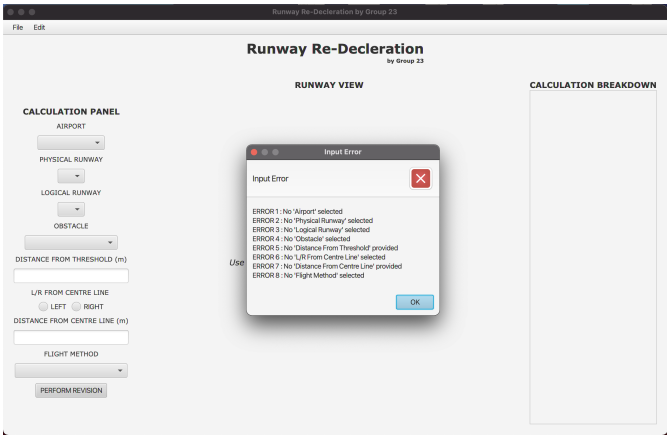
The below screenshot shows the system allowing the user to declare the placement of an obstacle within the 'Calculation Panel' of the dashboard. The user can then submit the runway for revision by selecting the 'Perform Revision' button.



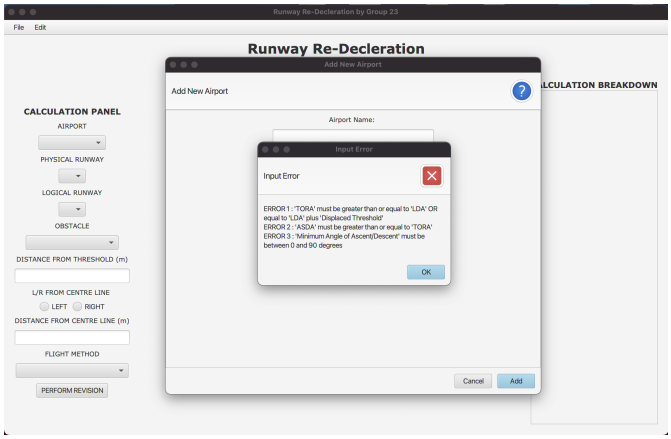

### 4.2.9 Input Error Checking

| ID | Identifier | Priority | Completion Criteria |
|---|---|---|---|
| **6** | Input Error Checking | **SHOULD** | The application is able to alert the user when invalid/erroneous information is input into the system, and this information is discarded. |

The system will validate all user input, ensuring that it is of the correct format, and that it follows all necessary constraints. The below screenshots show examples of error alert windows that are shown to the user when invalid information is submitted.




(a) Input error while performing a calculation

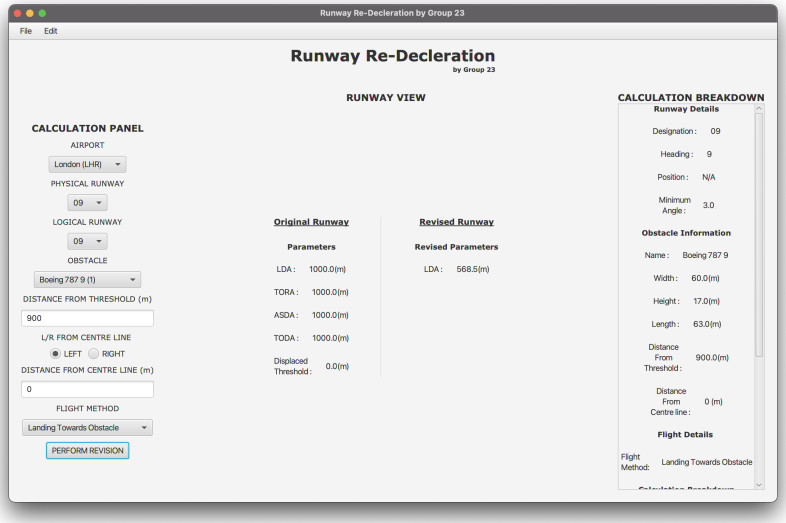



(b) Input error while adding a new runway

### 4.2.10 Runway Revision Calculations

| ID | Identifier | Priority | Completion Criteria |
|---|---|---|---|
| **7** | Updated Runway Parameters | **MUST** | The user is able to select a runway, obstacle and position, to then receive the updated runway parameters according to this obstacle placement. |
| **13** | Parameter Comparison | **MUST** | When performing runway calculations, the user is shown the original runway parameters with the updated runway parameters, side-by-side in the same view/screen. |

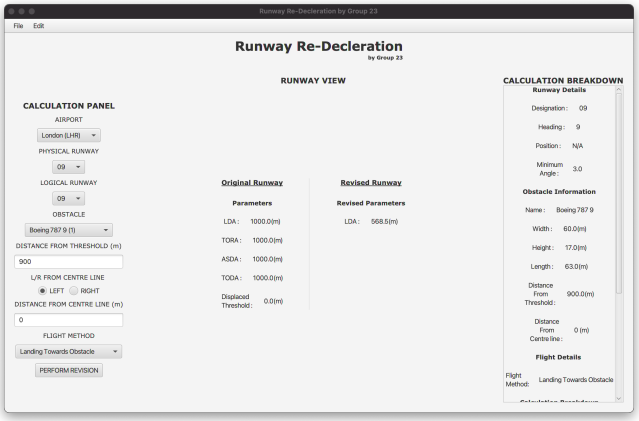
Once the user has selected/provided the information for a runway revision within the 'Calculation Panel', the user can select the 'Perform Revision' button to be shown the updated runway parameters, as well as a side-by-side comparison of the updated parameters with the original. This information is displayed to the user in the 'Runway View', and the results of an example calculation are shown in the below screenshot.



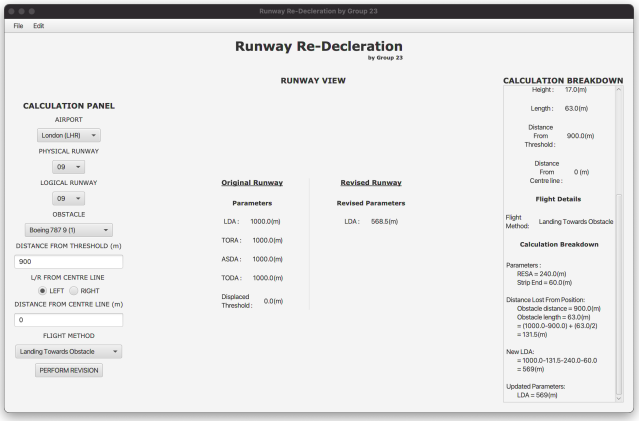

### 4.2.11 Runway Revision Calculation Breakdown

| ID | Identifier | Priority | Completion Criteria |
|---|---|---|---|
| **14** | Calculation Breakdown | **SHOULD** | When performing receiving updated runway parameters calculations, the user is shown a step-by-step breakdown of how the updated parameters were calculated. |

After a runway revision has been performed, the user is able to see a breakdown of the calculations that took place in the 'Calculation Breakdown' panel. The breakdown is contained within a scroll-able window, and provides the user with information on the runway being revised, the obstacle and its respective placement and a step-by-step description of the updated parameters were calculated. The below screenshots show the full contents of the calculation panel, following an example revision.




(a) First portion of the calculation breakdown



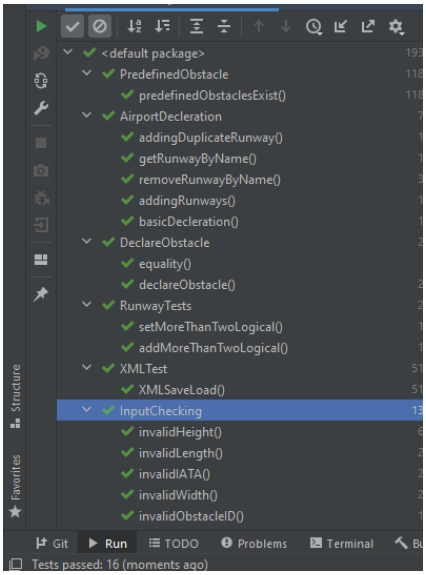

(b) Second portion of the calculation breakdown

# 5 Product Testing

## 5.1 Scenario Testing

The system was tested using scenarios according to the backlog items within the increment plan. That is, a test was developed for each of the backlog items that represents a scenario use case of that said item. The use of scenario testing ensures that the system demonstrates the correct behaviour, as defined in the User Scenarios, and that all of the backlog items set out in the increment plan have been completed.

The test cases that were written can be found within the code submission for this increment, and shown below is a screenshot taken from the IntelliJ IDEA IDE, that provides the results of the test suite when run on the product.




## 5.2 Future Testing

The group has acknowledged that further testing must have been performed on the system to validate its correctness (e.g., correctness of calculations), and plans to perform this testing in the future increments.

Unit tests will be written to test the correctness of the application, and TestFX will be used to perform extensive scenario testing on the application. The TestFX framework performs testing by taking control of the mouse pointer and keyboard to operate the application and evaluate it's behaviour. The framework carries out the tasks defined within the test cases, and compares the final state of the application to the provided expected state.

The use of both unit tests and the TestFX framework will ensure the product operates with the intended behaviour.

# 6 First Increment Sprint Review

## 6.1 Sprint Review

Following the completion of this development sprint, the group met to conduct a sprint review. The group shared their opinions on the success of the sprint, and considered areas that could be improved for further sprints. This review allowed group members to reflect on their work, and ensures an efficient and agile development process.

### 6.1.1 Successes

The greatest strength of the group during the sprint was the effective time management that the was demonstrated. The group set out to complete the sprint before the two-week deadline, as to leave enough time to review the work, receive feedback from the customer, and make changes where appropriate - and were successful in doing so. Finishing before the sprint deadline also allowed the group to begin the development of backlog items from other sprints, following agile methodologies to reduce the future workload. The group has agreed that the ability allocate and manage suitable amounts of time to the development process is a quality that must be maintained throughout the remaining sprints, to ensure that the final product is of the highest quality.

The effective time management of the group can be attributed to the sprint plan that was established in the product envisioning. The plan was able to allocate an appropriate number of product backlog items to the sprint, whilst still ensuring that the developed product would provide the customer with significant value. In addition to the importance of the plan itself, the group showed great discipline to follow this plan, and this discipline must be carried forward into the future sprints.

Another key aspect of the sprint was the extensive design work that was completed before development began (**Design Artifacts**). Design is a crucial part of software development, and the group dedicated significant portions of time at

the start of the sprint to effectively plan and structure the system that was to be developed. The result of this was an organized and efficient development process, and thus the same focus on design must be applied to future increments.

### 6.1.2 Challenges

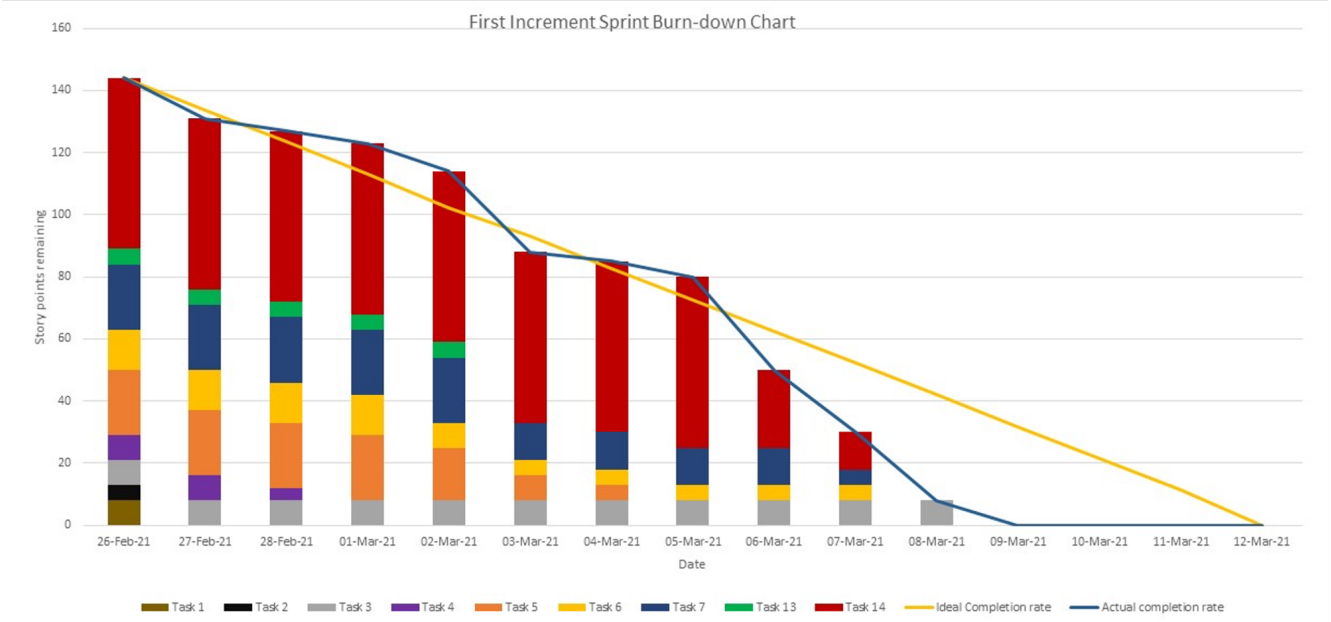
Despite the group being pleased with the overall outcome of the sprint, there remains some room for improvement.

The group encountered some difficulties while using with the resource management tool 'git', resulting in the generation of numerous conflicts (as described in the Envisioning Risk Assessment). This can mainly be accredited to the inexperience of some of the group members with the use of 'git' collaboratively, and the issue was swiftly resolved by those members with more experience.

Upon discussion, the group concluded that one concept to give more focus to in future increments is the system scope of a sprint. That is, the group had not clearly defined the system that was to be produced before the sprint began, leading to confusion among group members about what was actually being developed. The group now acknowledges that more attention should be given to the scope of the system before the sprint begins, as so to give a more focused and efficient development process.

## 6.2 Sprint Burn-down Chart

The below chart shows the timeline for the completion of product backlog items for the first increment, across the course of the sprint. A slower completion rate is present at the start of the sprint, as the group focused on project design, which became worthwhile further into the sprint, as the group as able to finish ahead of schedule. The remainder of the sprint was then focused on testing, responding to customer feedback, before finally performing the sprint review.




# 7 Second Increment Plan

During the second increment, the team will be primarily focusing on the visual components of the system. This involves developing the graphical representations of the airfield scenarios. Extensions of the functionality of the system will also be preformed by the team based upon the new visual components. The Airfield Operations Manager/Line Technician will now be able to use graphical representations of the airfield to aid their decision making process when Runway Re-Declaration needs to be performed.

## 7.1 Product Backlog Items

The Table below outlines all of the items that the team set out to complete in the upcoming increment. The number in brackets signifies the ID assigned to the task from the original product backlog created in the envisioning stage of the development. Item 15, Change Thresholds, does not appear here because the team felt it was easier to include this in the first increment and so it was completed then.

| Increment 2 Backlog Items | Priority |
|---|---|
| Runway Side-on View (8) | **MUST** |
| Runway Side-on View TOCS/ALS Slope (9) | **SHOULD** |
| Runway Top-down view (10) | **MUST** |
| Runway Top-down View Clear and Graded Area (11) | **SHOULD** |
| Simultaneous Runway View (12) | **COULD** |
| Runway Rotation (16) | **COULD** |
| Obstacle Placement Notification (21) | **SHOULD** |
| Updated Runway Parameters Notification (22) | **SHOULD** |

## 7.2 Sprint Plan

The table below breaks down each of the backlog items to be developed in Increment 2, including an estimation as to the number of story points that's required for each item. The team has has stuck with the original choice of using the Fibonacci Series as the story point range, which maintains consistency throughout the project. The total estimation of story point required for this upcoming increment is 141.

| ID | Identifier | Sub-tasks | Story Points |
|---|---|---|---|
| **8** | Runway Side-On View | • Develop Back-End<br>• Develop Middle-tier<br>• Delevop UI<br>• Testing | **34** |
| **9** | Runway Side-on View TOCS/ALS Slope | • Develop Back-End<br>• Develop Middle-tier<br>• Develop UI<br>• Testing | **21** |
| **10** | Runway Top-down view | • Develop Back-End<br>• Develop Middle-tier<br>• Develop UI<br>• Testing | **34** |
| **11** | Runway Top-down View Clear and Graded Area | • Develop Back-End<br>• Develop Middle-tier<br>• Develop UI<br>• Testing | **13** |

| | | | |
|---|---|---|---|
| **12** | Simultaneous Runway View | • Develop Back-End<br>• Develop Middle-tier<br>• Develop UI<br>• Testing | **8** |
| **16** | Runway Rotation | • Develop Back-End<br>• Develop Middle-tier<br>• Develop UI<br>• Testing | **21** |
| **21** | Obstacle Placement Notification | • Develop Back-End<br>• Develop Middle-tier<br>• Develop UI<br>• Testing | **5** |
| **22** | Updated Runway Parameters Notification | • Develop Back-End<br>• Develop Middle-tier<br>• Develop UI<br>• Testing | **5** |

### 7.3 Completion Criteria

The table below outlines the criteria that needs to be met to deem an item from the product backlog complete. It also includes the priority of the item from the backlog which also helps outline the importance of meeting the given criteria for each specific item.

| ID | Identifier | Priority | Completion Criteria |
|---|---|---|---|
| **8** | Runway Side-on View | **MUST** | The runway can be viewed side on, and displays the runway itself, the obstacle and the updated runway parameters. |
| **9** | Runway Side-on View TOCS/ALS Slope | **SHOULD** | When the runway is being viewed from the side, the TOCS/ALS slope is clearly shown. |
| **10** | Runway Top-Down View | **MUST** | The run-way and any obstacles and updated parameters can be viewed from top-down. |
| **11** | Runway Top-Down View Clear and Graded Area | **SHOULD** | When using the top-down viewing method, the clear and graded area is marked out and clearly visible. |
| **12** | Simultaneous Runway View | **COULD** | When using side on and top-down views, the runway, its obstacles and updated parameters are all shown at the same time. |
| **16** | Runway Rotation | **COULD** | The runway rotates to match its compass heading. |
| **21** | Obstacle Placement Notification | **SHOULD** | When an obstacle has successfully been placed on a runway, a notification appears to say inform the user the obstacle has been placed. |
| **22** | Updated Runway Parameters Notification | **SHOULD** | When updated runway parameters have been calculated, a notification confirming this appears for the user to read. |

## 7.4 Day-Zero Burn-down Chart

The below chart describes the Day Zero Burn-down for the second increment. The chart shows the 'Ideal Completion Rate' of tasks throughout the sprint in terms of their corresponding story points, with all tasks yet to be completed on day-zero of the sprint.