

Group 23: Deliverable 4: Increment 3

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	2
1 Response to Increment 2 Feedback	2
1.1 Application	2
1.1.1 Feedback	2
1.1.2 Response to Feedback	2
1.2 Design	2
1.2.1 Feedback	2
1.2.2 Response to Feedback	3
1.3 Testing	3
1.3.1 Feedback	3
1.3.2 Response to Feedback	3
1.4 Planning	3
1.4.1 Feedback	3
1.4.2 Response to Feedback	3
2 Design Artifacts	3
2.1 Class Diagram	3
2.2 Product Storyboards	4
2.2.1 Dashboard	4
2.3 User Scenarios	6
3 Product Adjustments	7
3.1 Response to Feedback	7
3.1.1 Cleaning Up Application Dashboard	7
3.1.2 Top-Down View Legend	8
3.1.3 Disabling Controls That Have No Functionality	8
3.1.4 Displaying Alerts on the Correct Screen	9
3.1.5 Top-Down View: Improving Controls	9
3.1.6 Simultaneous View: Improving Top-Down View Button Display	9
3.1.7 Cleaning Up Testing Code	9
3.2 Improvements	9
3.2.1 Top-Down View Obstacle Graphic	10
3.2.2 Coloured and Detailed Notifications	10
3.2.3 Persistent Notifications	11
3.2.4 Clearing Notifications	11
4 Product Showcase	11
4.1 Product Value	11
4.2 Product Demonstration	12
4.2.1 Dashboard	12
4.2.2 Importing Obstacles and Airports	12
4.2.3 Exporting Obstacles and Airports	13
4.2.4 Exporting Runway Revision Graphics	14
4.2.5 Exporting Runway Revision Calculation Breakdown	14
4.2.6 Exporting System Notifications	15
4.3 Un-implemented Product Features	15
5 Product Testing	15
5.1 Scenario Testing	15
6 Third Increment Sprint Review	17
6.1 Sprint Review	17
6.1.1 Successes	17
6.1.2 Challenges	17
6.2 Sprint Burn-down Chart	17

Introduction

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

1 Response to Increment 2 Feedback

On April 26th, Group 23 met with the group supervisor and an additional examiner to discuss the second increment that was submitted in the week prior. The supervisors provided the group with comprehensive feedback relating to the different aspects of the Increment as well as general comments discussing the product as a whole. A summation of this feedback, along with the Group's response is provided below.

1.1 Application

1.1.1 Feedback

Overall, the supervisors noted that the application is intuitive and brings real value to the customer; however, some quality of life features could make the user experience more enjoyable. The supervisor and examiner suggested that the Group:

- Remove confirmation boxes for non-destructive actions (such as adding airports or obstacles).
- Ensure error dialogues are displayed on the correct screen.
- Disable user controls that the user is not supposed to click at a given time (e.g., a choice box with no options).
- Remove the title that is present in the application Dashboard.
- Simplify the buttons present in Top-Down view.
- Ensure the size of the Top-Down View buttons in the Simultaneous View tab is the same as in the actual Top-Down View tab.
- Provide a key for the Top-Down view to demonstrate what each of the different colours mean.

1.1.2 Response to Feedback

The below list summarises the changes made to the application in response to the above feedback. Detailed descriptions of these adjustments can be found in the Product Adjustments section of the report.

- Error dialogues have been adjusted to display on the correct window.
- User controls that should not be accessible have been disabled.
- The application Dashboard has been changed to remove the title and unnecessary spacing.
- The Top-down view buttons have been adjusted to present a simpler interface.
- The Top-Down view buttons within the Simultaneous View tab have been adjusted so that they are the same size as in the actual Top-Down View tab.
- A key has been added to the Top-Down view to demonstrate what the different colours mean.
- Confirmation boxes on non-destructive items remain in the application, as their removal would require a large re-factoring of the application.

1.2 Design

1.2.1 Feedback

The supervisors described the updated class diagram's layout as "simplistic", "well fitted", and "intuitive to use". It was mentioned that the simultaneous view panel is a good addition, as it is no longer required to switch between tabs to access both views.

According to the feedback, all design artefacts are present, and the UML diagram is very well done. However, there is a criticism concerning the miss-use of types of lines, causing indications of associations between classes that were supposed to be indications of usage.

It was mentioned that some classes might be becoming too big, which hinders readability; however, that should not be a priority over other criticisms.

The supervisors also commented that the previous feedback has been received well and the changes made are meaningful.

1.2.2 Response to Feedback

The Group has taken on board the provided feedback and will apply it to future projects.

1.3 Testing

1.3.1 Feedback

The supervisor feedback stated that the testing using automated tools was rigorous and allowed for proof of correctness with quantifiable data. While the amount of automated testing was impressive, the feedback notes that there could have been more JUnit testing. It was also noted that the Group could improve the formatting of tests by splitting them into three blocks: arrange, act and assert with a blank separating each.

The acceptance criteria were present, although the feedback states that it could have been more descriptive.

1.3.2 Response to Feedback

The Group has adjusted the testing code so that it is divided into three sections as suggested. Time has also been taken to research into frameworks that provide more comprehensive testing as recommended. The Group understands the importance of effective testing and the value that it can provide to the customer.

1.4 Planning

1.4.1 Feedback

The supervisor feedback stated the Group continues to show good planning and prioritization, and that the burn-down chart accurately reflects the Group's performance during the sprint.

The supervisors agreed the Group had demonstrated effective prioritization by only having 'COULD' and 'WON'T' tasks remaining.

It is mentioned that the completion criteria are mostly a repetition of the user story's descriptions. The supervisors agreed that planning stages would benefit from more specific and detailed completion criteria.

1.4.2 Response to Feedback

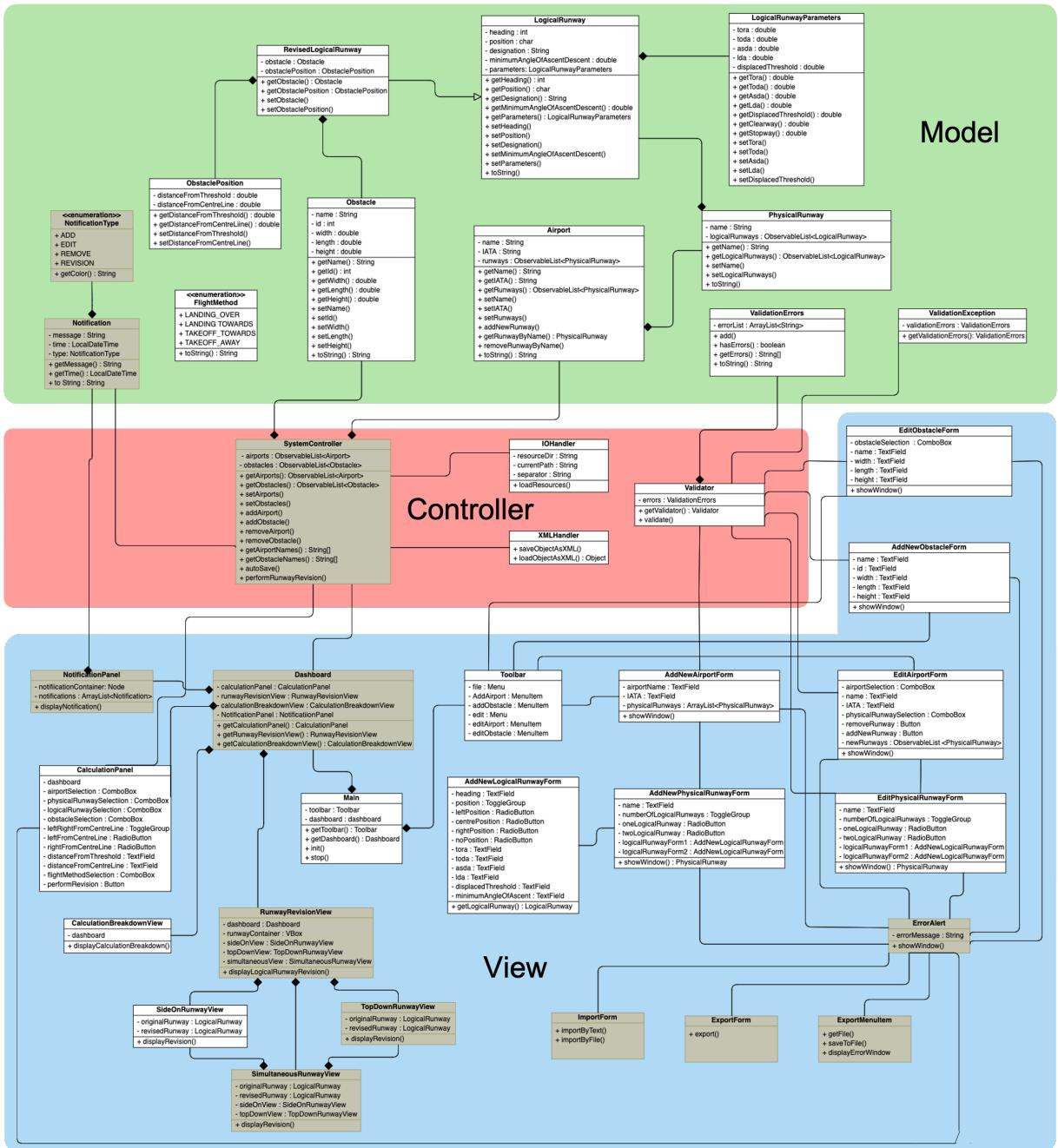
Although there is no outstanding planning for the project, the Group will apply the provided feedback to future development.

2 Design Artifacts

During the second 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. The presented artifacts should represent the product being developed in this sprint, and reflect how it is an extension of the Increment 2 product.

2.1 Class Diagram

The Class Diagram produced in the second Increment has been extended to support the functionality being developed in this third sprint. This diagram is shown below and showcases the classes within the system, along with their associated attributes and (public) methods. For convenience, the classes new to or that have been changed during this increment are highlighted.



2.2 Product Storyboards

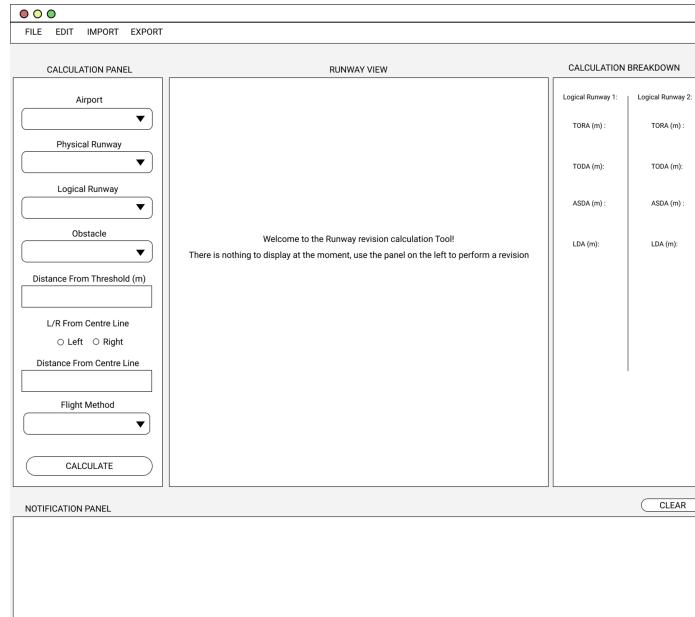
Product story boards were devised during the design process to provide a clear indication of how the user-interface would be extended in this Increment. These story boards are shown below and a short description is provided with each to explain its context and use.

The story boards shown 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 make use of the system.

2.2.1 Dashboard

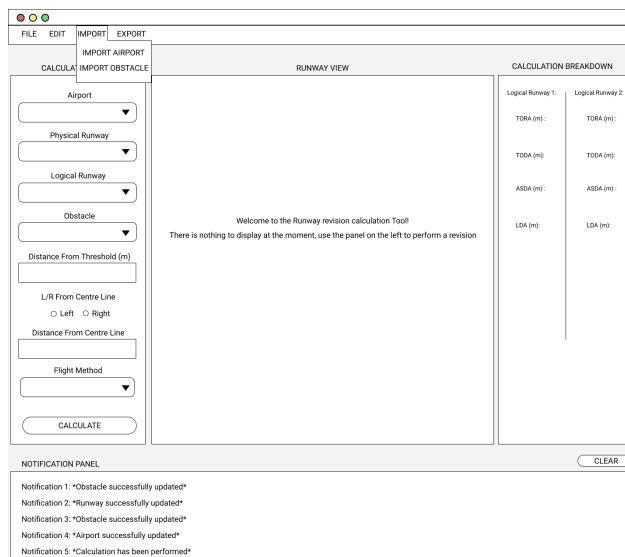
Dashboard : Main

The Dashboard should be extended to support the exporting and importing of system data (i.e., airports and obstacles) and revision data (e.g., side-on view of runway revision). This functionality will be provided by two new menu items in the toolbar - 'Import' and 'Export'. The notification panel will also be improved to allow for the system notifications to be cleared (i.e., removed).

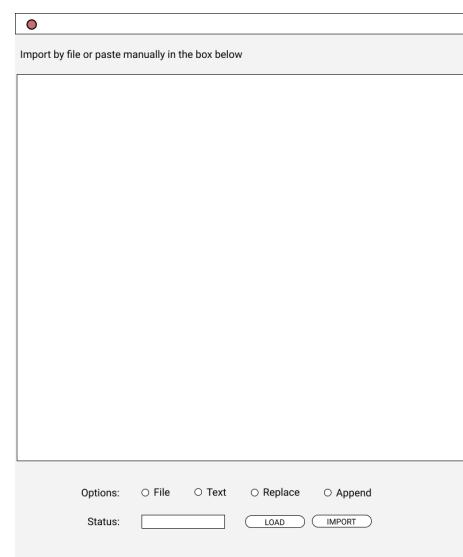


Import Menu

The import menu will allow for the user to import an obstacle or airport into the system. By navigating to the 'Import Menu' in the toolbar, the user can select to import an obstacle/airport into the system, and will be presented with a dialog that allows for them to either import the object from a text file, or enter the text manually.



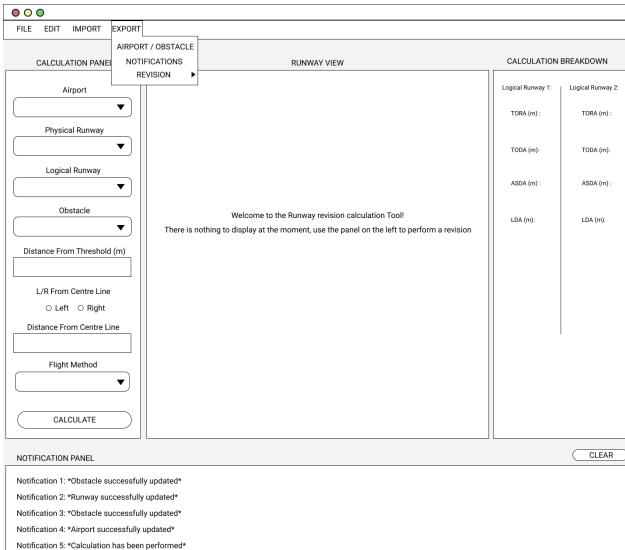
(a) 'Import' menu within the dashboard toolbar.



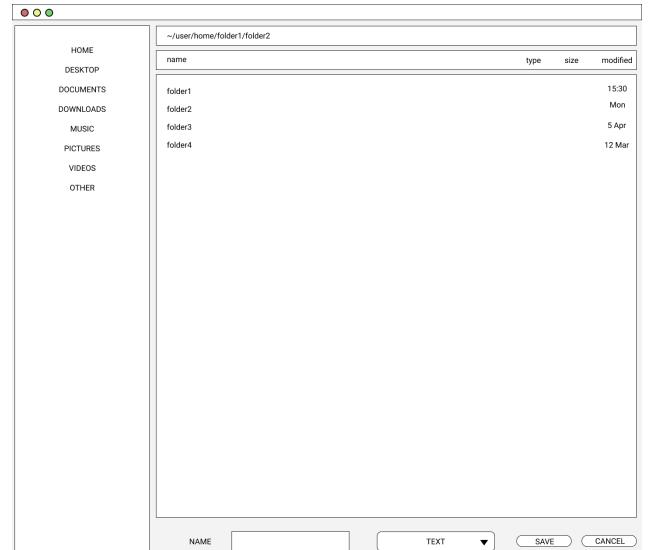
(b) Window displayed when importing an obstacle/airport

Export Menu

The export menu will allow for the user to export system objects, information relating to runway revisions and system notifications outside of the application. The Export Menu will be located in the toolbar, and after selecting an export option, a file chooser dialog will be opened that will allow for the user to select the location of the export.



(a) 'Export' menu within the dashboard toolbar.



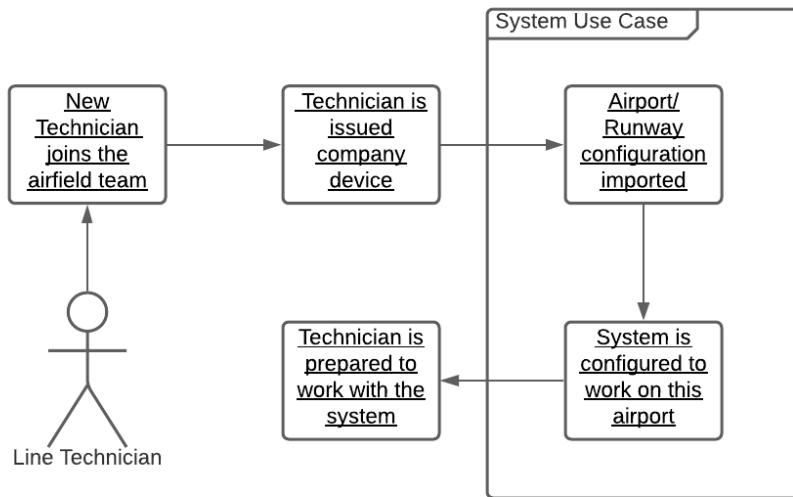
(b) Example file-chooser dialog shown when exporting data

2.3 User Scenarios

A number of user scenarios were developed during the design process to illustrate how the primary stakeholders would interact with the system, and the value that they gain from using the product. These user scenarios are described below

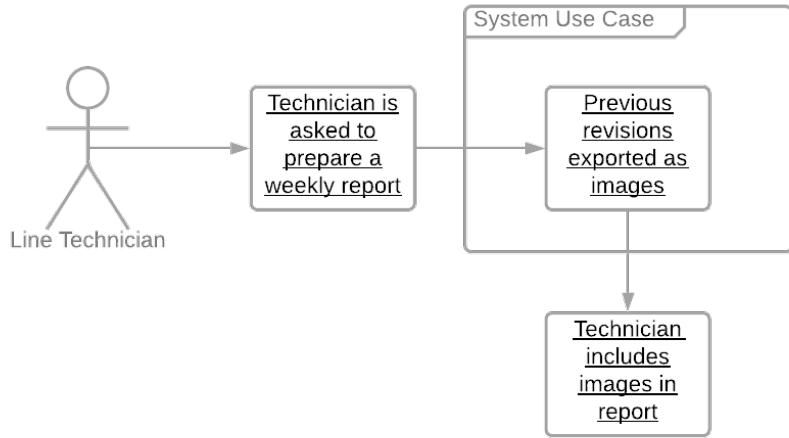
Scenario 1: Airport/Runway Import and Export

1. A new Line Technician joins the team
2. Line Technician is issued a device with the Redeclaration Software
3. The setup of the airport and runways is imported onto the Technician's device
4. Technician is able to use the software without any manual airport/runway transcription



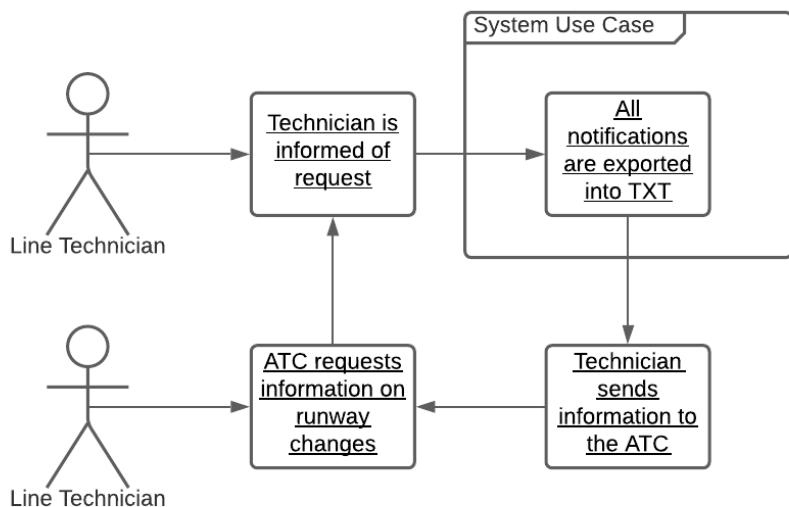
Scenario 2: Exporting Runway Revisions as images

1. Line Technician asked to perform a weekly report on their work
2. During the week Line Technician compiled all runway revisions as images using the 'Export to PNG' feature
3. The setup of the airport and runways is imported onto the Technician's device



Scenario 3: Airport/Runway Import and Export

1. ATC requests technicians to inform off all recent changed to the runways
2. Line Technician was previously informed of all changes in form of notifications
3. Line Technician exports notifications using the "Export to TXT" feature
4. All the information from Line Technicians is sent to the ATC



3 Product Adjustments

This section of the document details the adjustments that were made to the second increment product following the sprint review meeting and feedback from the customer. These adjustments were made before development began on increment 3 backlog items, and aim to improve the overall usability of the program and ensure that the customer's requirements are met.

3.1 Response to Feedback

The following adjustments to the product were made following feedback from the customer, and aim to provide the customer with an application that better suits their requirements.

3.1.1 Cleaning Up Application Dashboard

The application Dashboard has been adjusted to remove the title present in the header and some of the unnecessary spacing surrounding the individual panels. This presents the user with a less 'cramped' and more usable application.



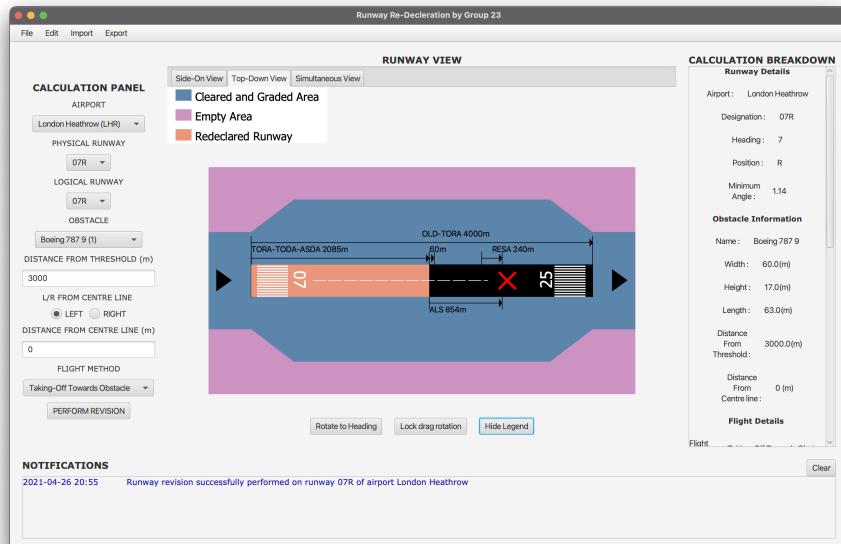
(a) Old application Dashboard



(b) Updated application Dashboard

3.1.2 Top-Down View Legend

The top-down view now contains a 'legend' that describes the meaning of the different colours used in the display. The user is able to show/hide the legend using the 'Display Legend' toggle button.

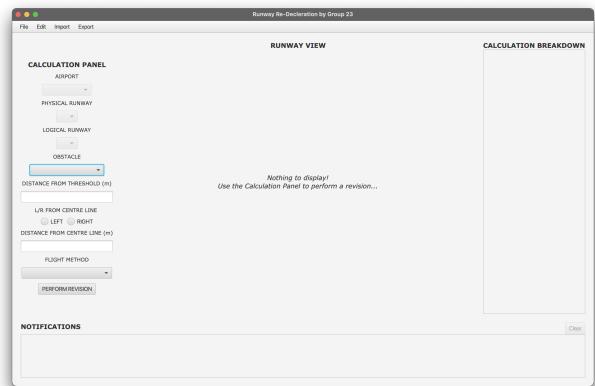


3.1.3 Disabling Controls That Have No Functionality

The controls present in the application have been edited so that they are only enabled when they have functionality to provide. For instance, previously, the user could select a choice box even if that choice box had no content to display. In the updated application, controls are only enabled when the user should be interacting with them, improving the usability of the application.



(a) Old Calculation Panel with no airports in the application



(b) Updated Calculation Panel with no airports in the application

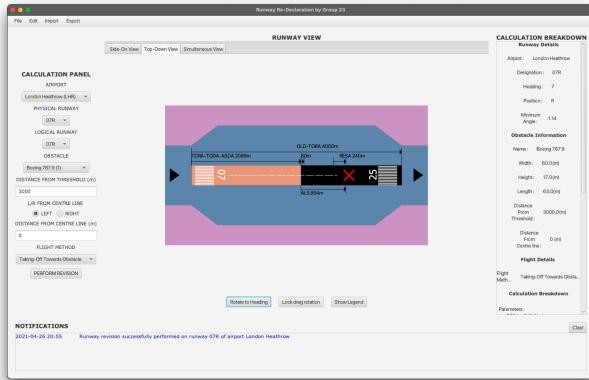
3.1.4 Displaying Alerts on the Correct Screen

The alert windows (confirmation and error) have been adjusted so that they always open on the same screen that the application is being ran on (i.e., in the case where the user is using multiple monitors). Previously, the dialog would be displayed on the system's main screen, regardless of the location of the application.

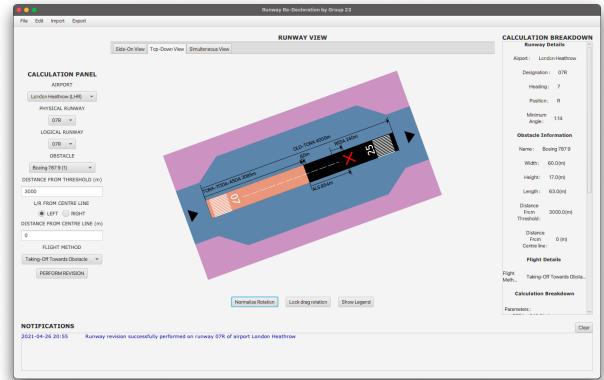
This feature cannot be demonstrated through screenshots, but will be showcased to the supervisors in the next review meeting.

3.1.5 Top-Down View: Improving Controls

The Top-Down View tab contains three buttons that allow the user to control the displayed graphic. Two of these buttons are 'Rotate Heading' and 'Normalize Heading' that rotate the graphic to match the runway heading and normalize the rotation of the graphic respectively. These two buttons have now been combined into a single toggle button that alternates its label between the two forms.



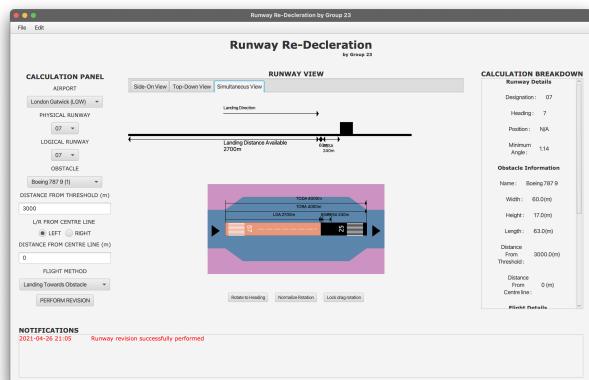
(a) Old Top-Down View controls



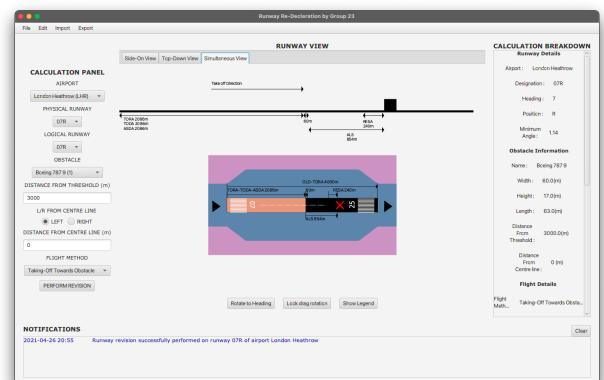
(b) Updated Top-Down View controls

3.1.6 Simultaneous View: Improving Top-Down View Button Display

The display of the top-down perspective of a revision within the Simultaneous View tab has been adjusted so that the buttons for the top-down view are the same size as they are in the actual Top-Down View tab. Previously, the buttons were smaller, and thus the application was inconsistent.



(a) Old Simultaneous View button display



(b) Updated Simultaneous View button display

3.1.7 Cleaning Up Testing Code

The code written for the application tests has been improved to present more structured and logical test cases. To do this, the test cases have been divided into three components of Setup, Actions and Assertions. While this adjustment does not directly affect the user, developing more structured test cases allows for a more structured and therefore thorough testing process. This ultimately presents the customer with a more correct application.

3.2 Improvements

The following adjustments were made in order to improve the application's usability and extend its basic functionality.

```

@Test
public void scenario10Runway09TakeOffAway() {
    // Obstacle
    Obstacle obstacle = new Obstacle();
    obstacle.setHeight(12);

    // distance from threshold
    double distanceFromThreshold = -50;
    double distanceFromCentreline = 0;

    // Obstacle Position
    ObstaclePosition obstaclePosition = new ObstaclePosition(distanceFromThreshold, distanceFromCentreline);

    // Revised Logical Runway 09L
    RevisedLogicalRunway revisedRunway = RunwayReviser.getRevisedLogicalRunway(logicalRunway09L,
        obstacle, obstaclePosition,
        FlightMethod.TAKEOFF_AWAY);

    // expected values
    double expectedTORA = 3345;
    double expectedASDA = 3345;
    double expectedTODA = 3345;

    // actual values
    double actualTORA = revisedRunway.getParameters().getTora();
    double actualASDA = revisedRunway.getParameters().getAsda();
    double actualTODA = revisedRunway.getParameters().getToda();

    // Testing the TORA, ASDA and TODA
    assertTrue("condition: actualTORA == expectedTORA", message: "Expected TORA = " + expectedTORA + " Actual TORA = " + actualTORA);
    assertTrue("condition: actualASDA == expectedASDA", message: "Expected ASDA = " + expectedASDA + " Actual ASDA = " + actualASDA);
    assertTrue("condition: actualTODA == expectedTODA", message: "Expected TODA = " + expectedTODA + " Actual TODA = " + actualTODA);
}

```

(a) Example old test case

```

@Test
public void scenario10Runway09LTakeOffAway() {
    // SETUP //
    // Obstacle
    Obstacle obstacle = new Obstacle();
    obstacle.setHeight(12);

    // distance from threshold
    double distanceFromThreshold = -50;
    double distanceFromCentreline = 0;

    // Obstacle Position
    ObstaclePosition obstaclePosition = new ObstaclePosition(distanceFromThreshold, distanceFromCentreline);

    // ACTIONS //
    // Revised Logical Runway 09L
    RevisedLogicalRunway revisedRunway = RunwayReviser.getRevisedLogicalRunway(logicalRunway09L,
        obstacle, obstaclePosition,
        FlightMethod.TAKEOFF_AWAY);

    // ASSERTIONS //
    // expected values
    double expectedTORA = 3345;
    double expectedASDA = 3345;
    double expectedTODA = 3345;

    // actual values
    double actualTORA = revisedRunway.getParameters().getTora();
    double actualASDA = revisedRunway.getParameters().getAsda();
    double actualTODA = revisedRunway.getParameters().getToda();

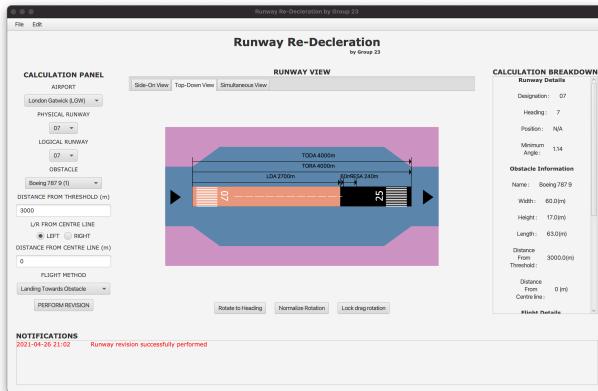
    // Testing the TORA, ASDA and TODA
    assertEquals("condition: actualTORA == expectedTORA", message: "Expected TORA = " + expectedTORA + " Actual TORA = " + actualTORA);
    assertEquals("condition: actualASDA == expectedASDA", message: "Expected ASDA = " + expectedASDA + " Actual ASDA = " + actualASDA);
    assertEquals("condition: actualTODA == expectedTODA", message: "Expected TODA = " + expectedTODA + " Actual TODA = " + actualTODA);
}

```

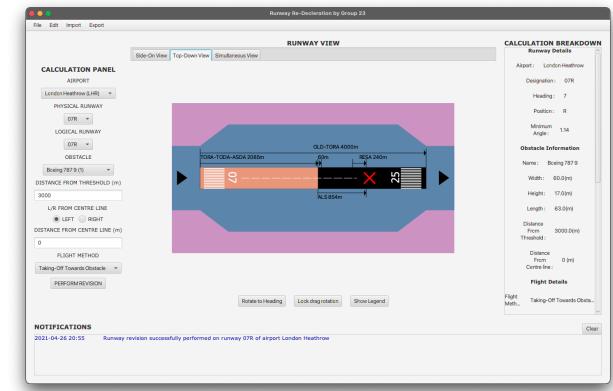
(b) Example updated test case

3.2.1 Top-Down View Obstacle Graphic

A graphic has been added to the top-down view of the runway revision to represent the obstacle currently involved in the revision.



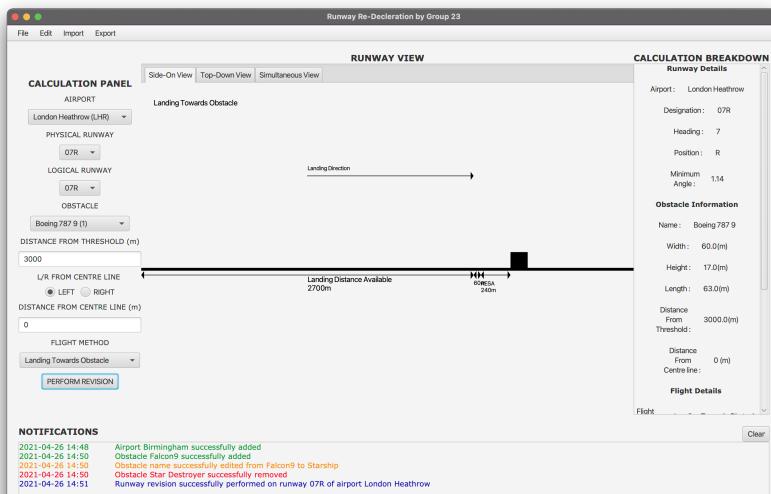
(a) Old Top-Down View graphic



(b) Updated Top-Down View graphic

3.2.2 Coloured and Detailed Notifications

The system notifications have been improved so that different types of notification are displayed in different colours, emphasising the action being performed by each notification. The notifications will also display action-specific information instead of action-generic information. For instance, after adding an airport to the application, instead of displaying “An airport has been added.”, the updated notification system will display “Airport NAME has been added.”, where “NAME” is the name of the recently added airport.



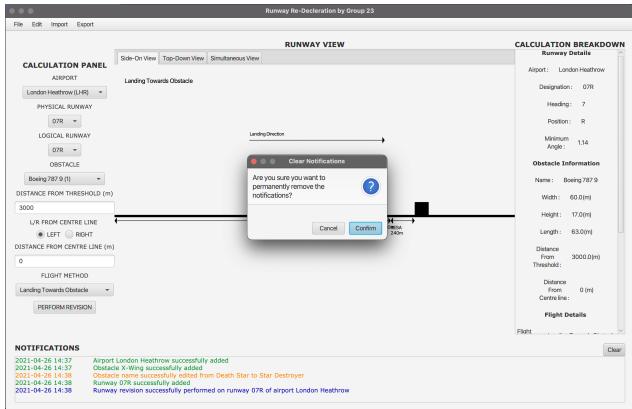
3.2.3 Persistent Notifications

The notification system has been improved so that the notifications persist after application close. Any notifications displayed to the user at the time of application close will be saved into the system storage, and reloaded when the application is restarted. This was done to provide the user with a persistent record of application use, and more specifically, of runway revisions.

This persistence cannot be demonstrated through screenshots, but will be shown to the customer in a future review meeting.

3.2.4 Clearing Notifications

As the system notifications are now stored after application close, a button has also been added to the 'Notifications Panel' that allows for the user to clear the notifications, permanently removing them from the system.



(a) Confirmation window shown when clearing notifications



(b) Application state after clearing notifications

4 Product Showcase

4.1 Product Value

The focus of this third sprint was to allow for the user to export/import information out of/into the application, and to extend the application beyond the initial customer requirements. The user is now able to import and export Airports and Obstacles via XML files, export information relating to runway revisions, and export system notifications. The following list details the operations that the user is now able to perform within the application.

- Import an airport from an XML file/from XML text.
- Export an airport to an XML file.
- Import an obstacle from an XML file/from XML text.
- Export an obstacle to an XML file.
- Export the system notifications to a text file.
- Export the side-on view of a runway revision to a PNG, JPEG or GIF file.
- Export the top-down view of a runway revision to a PNG, JPEG or GIF file.
- Export the simultaneous view of a runway revision to a PNG, JPEG or GIF file
- Export the calculation breakdown a runway revision to a PNG, JPEG or GIF file

Being able to import/export airports/obstacles to/from the system allows for interoperability between different instances of the application and for system backups to be performed. Allowing for runway revisions and notifications to be exported provides the user with a permanent record of system activity and revision details. The exporting of runway revisions also allows for a user to share a runway revision with another user, outside of the application - an incredibly useful feature.

It should also be noted that the adjustments made to the first increment product, as listed in Product Adjustments, provide the customer with a more functional and user-friendly application that better meets their requirements, increasing the value they receive as a result.

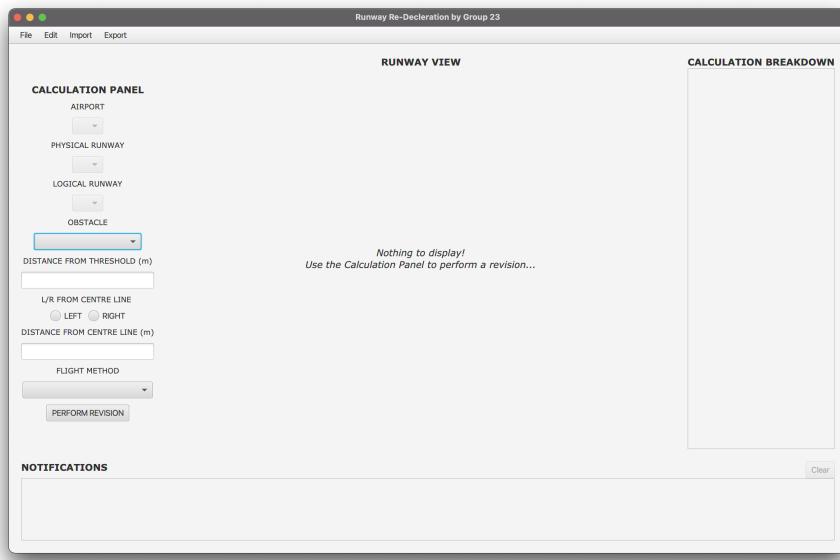
4.2 Product Demonstration

The Increment 3 product will now be showcased to demonstrate its capabilities. The different aspects of the system are presented in terms of this increment's product backlog items, with a description and image of the application being provided for each.

Note that, in some cases, the running of the application may be required to verify that an aspect of the system functionality has been correctly implemented.

4.2.1 Dashboard

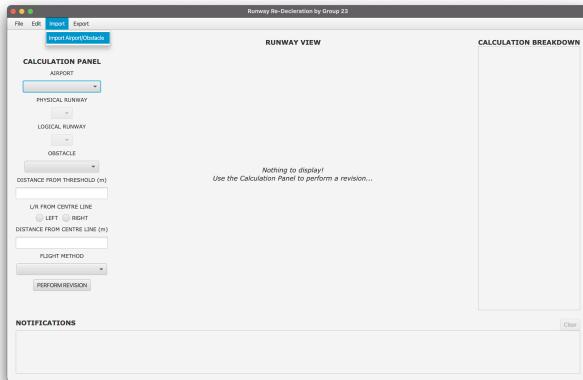
The 'Dashboard' functions as the home screen of the application, and has been updated to include 'Import' and 'Export' menus. These menu's can be found within the application toolbar. A button has also been added to the notification's panel that allows for the user to clear the currently stored notifications from the system (as the notifications now persist on application close).



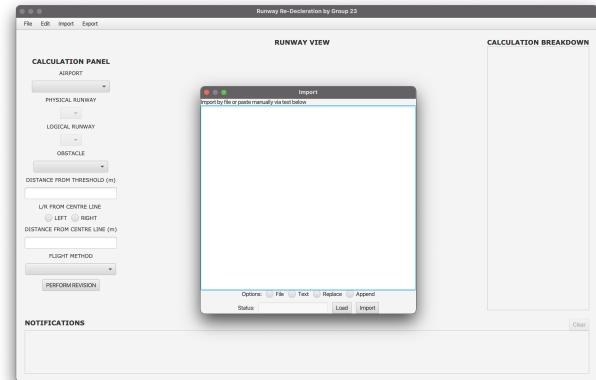
4.2.2 Importing Obstacles and Airports

ID	Identifier	Priority	Completion Criteria
17	XML Airport Import	COULD	Airports and their runways will be able to be imported into the system via XML.
18	XML Obstacle Import	COULD	Predefined obstacles will be able to be imported into the system via XML.

By navigating to the 'Import' menu within the toolbar, the user is able to select to import an obstacle or airport. After selecting this menu item, a dialog is opened that allows for the user to either input the XML text for an airport/obstacle, or select an XML file to import into the system. Note that the user can import multiple obstacles/airports in one import operation. The dialog also provides the user with two loading options - "Replace" and "Append". "Replace" will replace the current system objects with the imported ones, whilst "Append" will append the imported objects to the existing ones.



(a) Import 'Airports/Obstacles' menu item within 'Import' menu



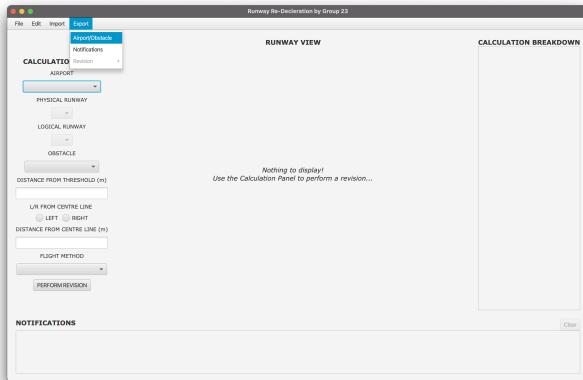
(b) The 'Import Airports/Obstacles' window

4.2.3 Exporting Obstacles and Airports

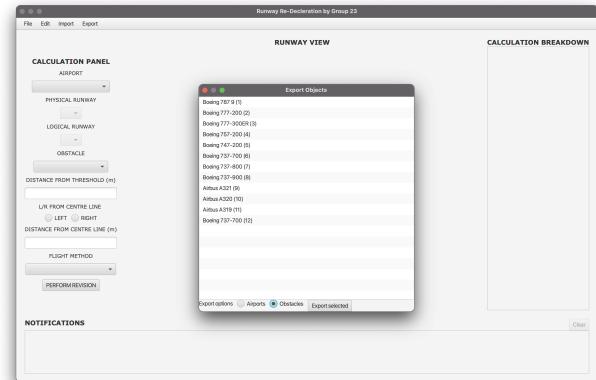
ID	Identifier	Priority	Completion Criteria
19	XML Airport Export	COULD	Airports and it's runways can be exported into XML for use in other systems.
20	XML Obstacle Export	COULD	Obstacles can be exported into XML for use on other systems.

By navigating to the 'Export' menu within the toolbar, the user can select to export an obstacle or airport. After choosing this menu item, the user is presented with a dialog that allows for them to select the type of object to export (i.e., airport or obstacle), and which instances of this object type they would like to export. The user is able to select multiple objects to be exported at one time by holding the 'SHIFT' key whilst selecting.

Note that the format of the exported XML objects is the same format that is expected when importing objects from XML files, allowing for object instances to easily be transferred between different applications, or re-loaded in the case of a crash/from a backup.



(a) Export 'Airports/Obstacles' menu item within 'Export' menu

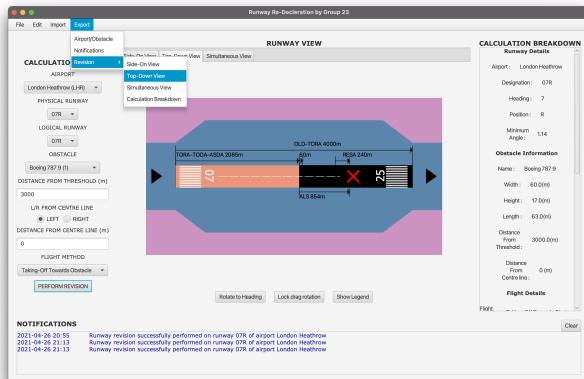


(b) The 'Export Airports/Obstacles' window

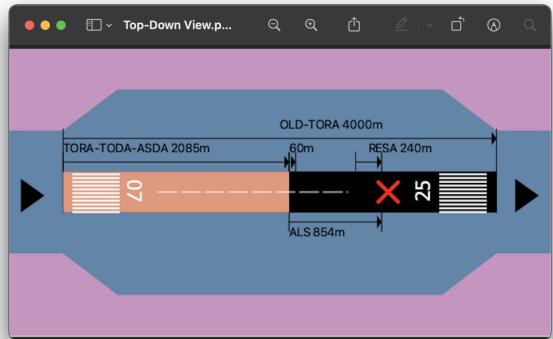
4.2.4 Exporting Runway Revision Graphics

ID	Identifier	Priority	Completion Criteria
26	JPEG Export	WONT	The views shown on the screen can be exported into a JPEG image file.
27	PNG Export	WONT	The views shown on the screen can be exported into a PNG image file.
28	GIF Export	WONT	The views shown on the screen can be exported into a GIF image file.

After performing a revision, and after navigating to the 'Export' menu of the toolbar, the user is presented with an 'Export Revision' sub-menu that allows for them to export the results of the revision outside of the application. The user can chose to export the side-on view, top-down view, or simultaneous view of the revision, and can export the chosen view as a JPEG, PNG or GIF file. A file-chooser dialog is displayed to allow the user to select the location of the exported file.



(a) Export 'Revisions' menu item within 'Export' menu

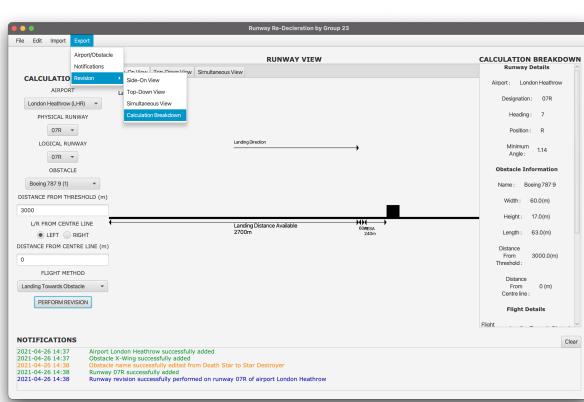


(b) The top-down view exported from the application

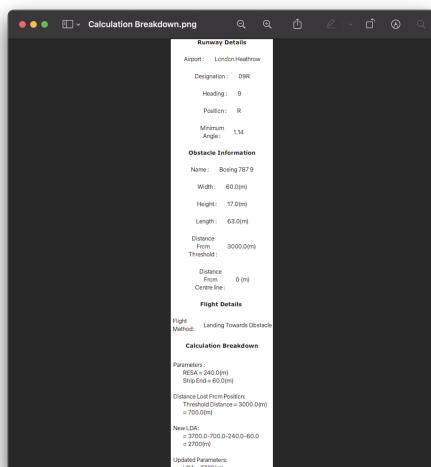
4.2.5 Exporting Runway Revision Calculation Breakdown

ID	Identifier	Priority	Completion Criteria
31	Print Results	WONT	The results of the current scenario can be printed for use externally.

After performing a revision, and after navigating to the 'Export' menu of the toolbar, the user is presented with an 'Export Revision' sub-menu that allows for them to export the calculation breakdown of the revision outside of the application. The user can chose to export the calculation breakdown as a PNG, JPEG or GIF file. A file-chooser dialog is displayed to allow the user to select the location of the exported file. This exported calculation breakdown image can then be printed by the user.



(a) Export 'Revisions' menu item within 'Export' menu

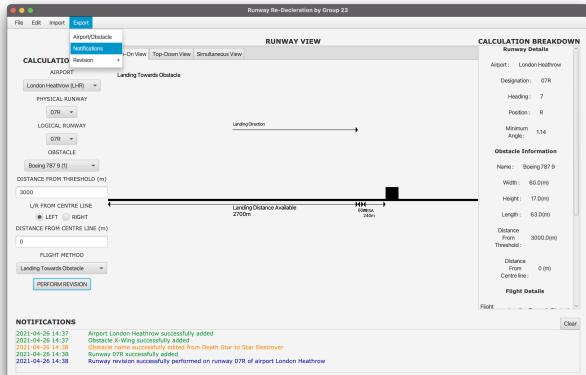


(b) The calculation breakdown exported from the application

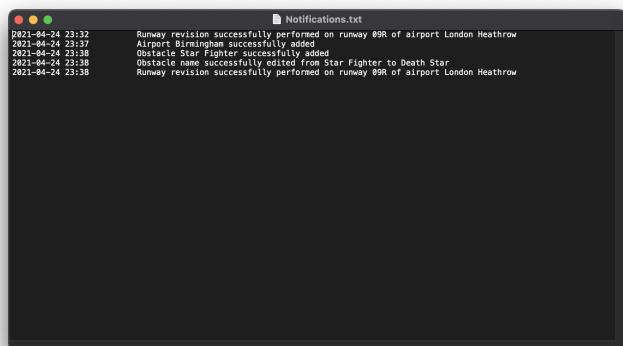
4.2.6 Exporting System Notifications

ID	Identifier	Priority	Completion Criteria
33	Notifications Export	WONT	The notifications from the current session can be exported to show an accurate log of system activity.

When notifications are present in the system, and after navigating to the 'Export' menu within the toolbar, the user can select to export the system notifications. The notifications are then exported as a text file into the chosen location.



(a) Export 'Notifications' menu item within 'Export' menu



(b) The notifications exported from the application

4.3 Un-implemented Product Features

Unfortunately, two of the product backlog items that the Group had planned to develop during this third sprint have not been implemented - items 24 and 32. Given the time constraint placed on this development sprint, the group decided to focus on the higher priority backlog items (XML import and export classified as 'COULD'), which ultimately did not leave enough time for all of the remaining, lower priority items. As a result, and in accordance with agile methodologies, the Group chose to implement only items 26, 27, 28, 31 and 33, as it was concluded that these items provide would provide the customer with the most value.

ID	Identifier	Priority	Completion Criteria
24	Zoom and Pan	WONT	The view of the runway can be zoomed in on and panned to get alternate views of the runway.
32	Custom Object Images	WONT	The user can choose an image for a certain obstacle which will be then scaled to give an accurate representation of the current scenario.

5 Product Testing

Following the extensive testing carried out in the second increment, the Group has chosen to only conduct scenario testing for this increment - Unit tests were not included as the functionality implemented in this increment is best assessed through scenario testing.

5.1 Scenario Testing

The Group defined a series of scenarios for the product, which were then manually carried out and the state of the application was assessed. While the 'TestFX' framework was used to automate the scenario test cases during the second increment testing, the framework was not used in this increment as the Group were sufficiently able to assess the application behaviour through only manual testing.

The below table describes each of the tests carried out, providing the actions performed in the scenario, the expected outcome of the application after these actions, and the result of the test when performed manually.

ID	Identifier	Scenario	Expected Outcome	Pass/Fail
1	Importing Airport from XML	The user opens the 'Import Airport/Obstacle' window from the 'Import' menu, selects an import method, provides XML data and submits.	The airports defined in the selected XML file are now present in the system, and can be used for runway revisions.	PASS

2	Importing Airport from XML: Invalid Input	The user opens the 'Import Airport/Obstacle' window from the 'Import' menu, selects an import method, provides XML data and submits. The provided XML data is invalid.	No new airports are added to the system, and a window is displayed to the user explaining why the provided XML file was invalid.	PASS
3	Importing Obstacle from XML	The user opens the 'Import Airport/Obstacle' window from the 'Import' menu, selects an import method, provides XML data and submits.	The obstacles defined in the selected XML file are now present in the system, and can be used for runway revisions.	PASS
4	Importing Obstacle from XML: Invalid Input	The user opens the 'Import Airport/Obstacle' window from the 'Import' menu, selects an import method, provides XML data and submits. The provided XML data is invalid.	No new obstacles are added to the system, and a window is displayed to the user explaining why the provided XML file was invalid.	PASS
5	Exporting Airport to XML	The user opens the 'Export Airport/Obstacle' window from the 'Export' menu, selects to export Airports, selects the airports they would like to export, and submits.	The user is presented with a dialog that allows for them to select the location of the export. After responding to this dialog, a new XML file is present at the selected location that contains the chosen airports.	PASS
6	Exporting Obstacle to XML	The user opens the 'Export Airport/Obstacle' window from the 'Export' menu, selects to export Obstacles, selects the obstacles they would like to export, and submits.	The user is presented with a dialog that allows for them to select the location of the export. After responding to this dialog, a new XML file is present at the selected location that contains the chosen obstacles.	PASS
7	Exporting Notifications	The user opens the 'Export Notifications' window from the 'Export' menu, selects a location to export to and submits.	A new text file is created at the selected location that contains the system notifications in text format.	PASS
8	Exporting Side-on View	The user performs a revision, opens the 'Export Revision' sub-menu from the 'Export' menu, selects to export the side-on view, selects a location and file type and submits.	A new file is created at the chosen location that contains the side-on view of the previously performed runway revision in the chosen file format.	PASS
9	Exporting Top-Down View	The user performs a revision, opens the 'Export Revision' sub-menu from the 'Export' menu, selects to export the top-down view, selects a location and file type and submits.	A new file is created at the chosen location that contains the top-down view of the previously performed runway revision in the chosen file format.	PASS
10	Exporting Simultaneous View	The user performs a revision, opens the 'Export Revision' sub-menu from the 'Export' menu, selects to export the simultaneous view, selects a location and file type and submits.	A new file is created at the chosen location that contains the simultaneous view of the previously performed runway revision in the chosen file format.	PASS
11	Exporting Calculation Breakdown	The user performs a revision, opens the 'Export Revision' sub-menu from the 'Export' menu, selects to export the calculation breakdown, selects a location and file type and submits.	A new file is created at the chosen location that contains the calculation breakdown of the previously performed runway revision in the chosen file format.	PASS

6 Third 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 team made great progress in this increment, and there were many things that the team did well. One of those things that was done well was that given the time constraints of this increment, the workload of spread out well to give an efficient development process and to stay on track with the time constraints. Ensuring that the team sticks to the time constraints is an essential part of what will make this project successful, as failure to not complete certain aspects of the project will lead to an incomplete product at the end of the development stage.

The improvements made to the product that were developed in previous increments was a key aspect in what made this increment a success. Following on from feedback given by the teams supervisor, the team were able to effectively make adjustments and improvements to different aspects of the product. The team also has a keen eye for spotting things that could be done better, which also helps to ensure that a high quality of product is completed.

The effective use of manual testing was used in this increment to ensure that certain aspects of the system that had been developed worked in the correct way. Manual testing was used in certain areas of this increment because , as mentioned in Scenario Testing, the team were able to clearly assess the applications behaviour using manual testing. The use of effective and efficient testing was a huge success in this increment and allowed the team to fully maximise the time available.

A final success of the sprint was the Group's ability to execute agile methodologies and effectively prioritize tasks. Despite being unable to implement two product backlog items during the sprint, the Group agrees that the product value has been maximised as a result of the selected features.

6.1.2 Challenges

Despite a successful third increment, there was always going to be some challenges to overcome in the development of the increment.

One of the main challenges the team faced in this increment, was managing to complete all tasks in the given space of time. This is because the team was given just one week to complete this increment, which is not a lot of time given the tasks that are required to be done once the project is complete, such as the User Guide. despite the team managing their time very well, the extremely short time frame for this third increment made getting all the relevant aspects done on time a huge challenge for the team to overcome.

Another challenge that the team had to face was ensuring that the User Guide if fit for purpose and explains clearly how to use the system. It needed to be easy to interpret and understand so that the end user can use the system for its intended purpose easily and effectively. This was challenging because it mean the team had to think about every little detail in how the system worked to ensure end users knew how to use it correctly.

6.2 Sprint Burn-down Chart

The below chart shows the timeline of completion for product backlog items across the course of the sprint. The group began the sprint by tackling issues raised in the customers' feedback and designing the second increment product. A focus on design in the early stages of the sprint allowed for an efficient development process.

