

COMP2211 Runway Redeclaration

Group 2 - Deliverable 4, Increment 3

Eren Rafet (er10g23)
Bozhang Wu (bw9n21)
Louis Townsend (lmt1n22)
Abdullah Hariry (ah4u22)
Hossameldin Tammam (htft1e22)

May 2025

Contents

1	Response to Feedback	3
1.1	Not all testing strategies are mentioned	3
2	Key Design Artifacts	3
3	Regression Unit Testing	4
4	New JUnit Testing	5
4.0.1	Acceptance Testing	6
5	Planning	7
5.1	Progress for Increment 3	7
5.1.1	Retrospective on Sprint 3:	7
5.1.2	Sprint backlog with actual hours and integration	7
5.1.3	Definition of "done" for the tasks	8
5.1.4	Complete Burndown Chart for Increment 3	9
A	Appendix A: Acceptance testing evidence	10
A.1	Visualisation of the runways and the calculations are put together in a PDF	10
A.2	Each logged in has a specific role and access is granted based on the role	13
A.3	Each field where the user can enter a value has input validation	15
A.4	Visual and sound notifications after placing obstacles and new airport addition	20
A.5	Both top-down and side-on views are available and are updated with obstacle placement	22
A.6	User can access the help documentation from both screens	25

1 Response to Feedback

1.1 Not all testing strategies are mentioned

Firstly, as we always divide our tasks into small subtasks that can be handled independently, they are well-suited for unit testing. We perform unit testing on these subtasks to both verify that they work as intended and validate that they meet the expected requirements.

While boundary and partition testing are also useful at higher levels of testing, we apply these strategies at the unit level to catch issues early. This reduces the time spent debugging in higher-level tests, as we can be confident that any problems are not coming from the individual units.

For example, consider the registration function. We test each required field independently. For instance:

- **Boundary testing:** Checking the minimum number of characters allowed for a password.
- **Negative testing:** Deliberately leaving out symbols in a password to ensure it triggers an error.

By testing fields like this individually, we ensure that when it's time for integration testing, we already know that these components are reliable.

However, apart from testing positive and negative values, partition testing wasn't very applicable in our case. This is because our tool mostly deals with specific numerical values rather than distinct categories of data. This is why, for this increment, we did not encounter many scenarios where partition testing was relevant.

A similar reasoning goes for regression testing, as, in increment 1, we implemented the basis for everything and did not really have the time to develop them further, we started increment 2 with minimal testable components in hand, which meant that the overall regression testing we could perform was minimal.

We also performed acceptance testing to validate that we followed the specification and met user expectations. This testing was done for user stories that were fully completed and ready for review. Since we had another sprint remaining, not all user stories were ready to be tested at this stage.

Lastly, scenario testing was left for the last increment, for a similar reason, as not all the user stories were complete and we decided to wait to use the final version of the tool to perform scenario testing.

2 Key Design Artifacts

With the latest updates made to our UML diagram, and the additions of the database design and use case diagram, we believe that all the design artifacts included so far accurately cover the features that we have implemented, therefore no more design artifacts were included.

3 Regression Unit Testing

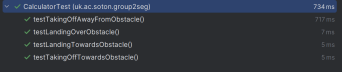
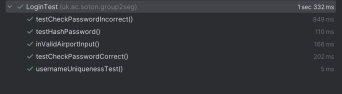
Category	What we were testing	Method of Test	Result of Test	Evidence
Calculation Tests	Testing Calculations According to Project Specs	JUnit tests for LDA, TORA, TODA, ASDA under different obstacle placements	As expected	
Password/Login Tests	Password Methods for Database	JUnit test for password hashing/matching/verifying methods	As expected	

Table 1: Regression JUnit Test Table

4 New JUnit Testing

Category	What we were testing	Method of Test	Result of Test	Evidence
Auto/Fill Tests	Autofilling Distances According to Project Spec Using Only 1 Distance	Comparing Values to Project Spec	As Expected	Appendix Figure 23

Table 2: New JUnit Test Table

4.0.1 Acceptance Testing

Previously covered user stories are as follows, evidences listed in the previous report:

User Story	Test	Result
As ATC I want to be able to check the validity of calculations to maintain safety	Calculation breakdowns can be viewed and are valid	Pass
As a user I want to see obstacles on the runway when they cause a redeclaration	Obstacles are rendered around the runway	Pass
As ATC I want to see available distances on the screen	Distance lines are rendered to scale on the screen	Pass
As ground crew I want to be able to report obstructions on the runway	An obstacle can be added to the runway strip	Pass
As a user I want to be able to add a new airport	Airports can be added from within the application	Pass
As a regulator I want an error log so I can check system integrity	Error logs are generated and accessible	Pass
As a regulator I want calculation logs so I can check system validity	Calculation logs are generated and accessible	Pass
As ground crew I want predefined obstacles list so I can quickly add an obstacle to the system	A predefined list of obstacles is available for selection	Pass
As ATC I want to see distances for each runway separately to ensure I only see relevant information	Distances are displayed separately for each runway	Pass

Table 3: Acceptance testing from the previous increment

The following are the user stories we have covered in this increment:

User Story	Test	Result
As ground crew I want to export reports so I can save them for future reference	Visualisation of the runways and the calculations are put together in a PDF	Pass
As a regulator I want a PDF export feature so I can view and share logs from the system in case of investigation or safety audit	Visualisation of the runways and the calculations are put together in a PDF	Pass
As a regulator I want only authorised people to be able to make changes to the system to ensure system integrity	Each logged in has a specific role and access is granted based on the role	Pass
As a user I want error notifications to prevent me from inputting incorrect data and inform me of calculation errors	Each field where the user can enter a value has input validation	Pass
As a user I want notifications of new obstacles and redeclarations so I can stay informed	Visual and sound notifications after placing obstacles and new airport addition	Pass
As ATC I want to visualise the airfield from multiple angles	Both top-down and side-on views are available and are updated with obstacle placement	Pass
As a user I want detailed documentation so I can learn the system	User can access the help documentation from both screens	Pass

Table 4: Acceptance testing for this increment. See appendices for evidence

5 Planning

5.1 Progress for Increment 3

5.1.1 Retrospective on Sprint 3:

Since this was the final sprint, we focused on adding features that depended on the core ones being finished. For example, exporting visualizations as PDFs required us to first complete the runway visualization.

We also aimed to reorganise the tool, moving some features to more logical locations, which made it easier to implement role-based access. For example, we moved the registration feature to the admin dashboard, since new airport employees wouldn't register themselves. Instead, they would simply receive credentials to use the tool.

Lastly, since we were also working on other projects, we generally did not have the time to implement all of the extensions, so we prioritised the tasks that "must" be done.

5.1.2 Sprint backlog with actual hours and integration

Task	Owner	Estimation	Actual Hours
1: Admin dashboard back-end	Eren (Scrum master)	10h	9h
1.1: SQL queries to edit and add users		5h	6h
1.2: Modifying access levels based on roles		5h	3h
2: Finalise views	Louis	10h - 14h	10h
2.1: Debug top down view		2h	3h
2.2: Implement side on view		8h	7h
2.3: (EXTENSION) Pan and zoom views		4h	-
3: Help documentation	Andy	10h	10h
3.1: Writing documentation		5h	6h
3.2: Implementing documentation within application		5h	4h
4: Information exporting	Abdullah	14h	15h
4.1: Exporting airport and obstacle details as XML		7h	9h
4.2: Exporting visualisations and calculations in PDF		7h	6h
5: General graphics and UI improvements	Hossam	9h-11h	12h
5.1: Improve usability		2h	3h
5.2: User(role)-specific application views		7h	6h
5.3: (EXTENSION) Colour blindness views		3h	3h
6: Integration	All	4h	5h

Table 5: Task Ownership Table for Increment 3, with actual hours

5.1.3 Definition of "done" for the tasks

We ensured that the following criteria were met to count a task as "done":

- Comprehensive test coverage that take into account many scenarios, with all the tests passing.
- The development team agrees that any code written for related tasks (if applicable) can be easily integrated.
- The customer is satisfied with the implementation(product) and believes it delivers value.

5.1.4 Complete Burndown Chart for Increment 3

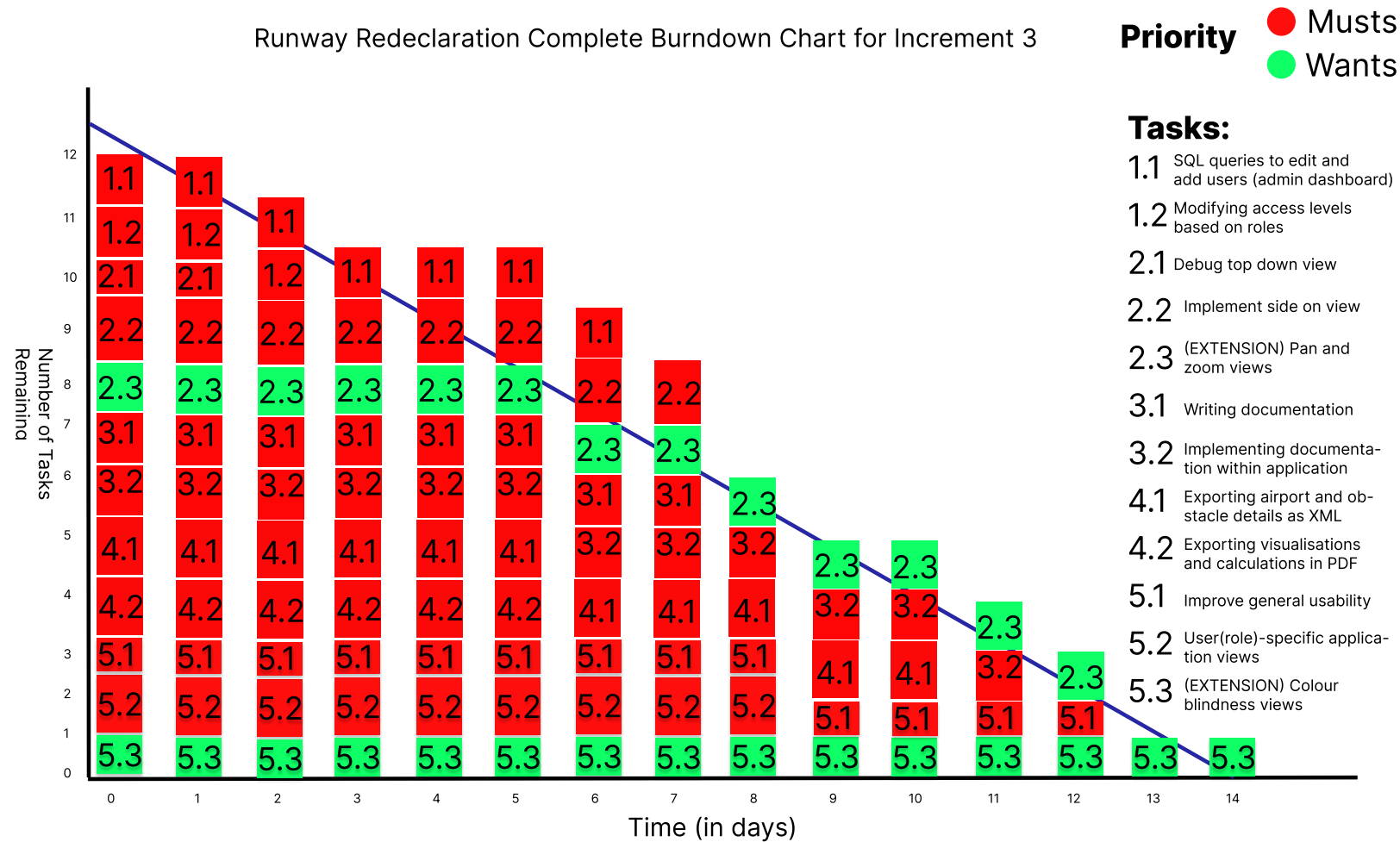


Figure 1: Increment 3 Complete Chart

A Appendix A: Acceptance testing evidence

A.1 Visualisation of the runways and the calculations are put together in a PDF

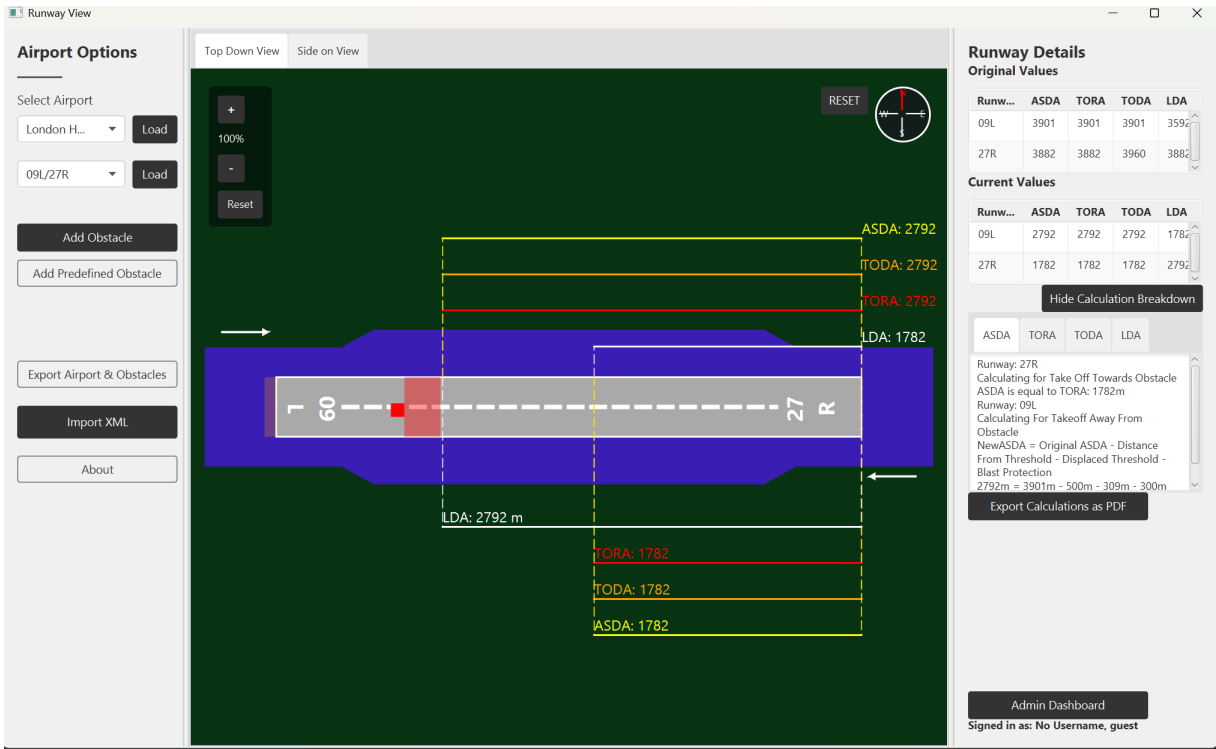


Figure 2: Obstacle placed on runway

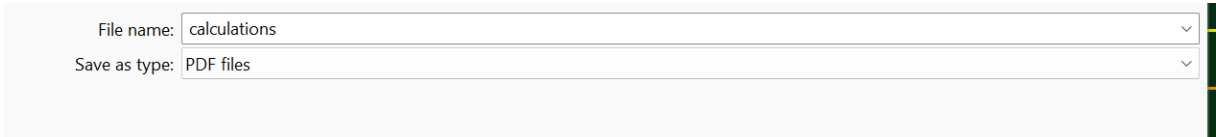


Figure 3: PDF saving

ASDA:

Runway: 27R

Calculating for Take Off Towards Obstacle

ASDA is equal to TORA: 1782m

Runway: 09L

Calculating For Takeoff Away From Obstacle

NewASDA = Original ASDA - Distance From Threshold - Displaced Threshold - Blast Protection

2792m = 3901m - 500m - 309m - 300m

TORA:

Runway: 27R

Calculating for Take Off Towards Obstacle

NewTORA = Distance From Threshold + Displaced Threshold - MAX(Slope*Height, RESA) - Strip End

1782m = 3092m + 0m - 1250m - 60m

Runway: 09L

Calculating For Takeoff Away From Obstacle

NewTORA = Original TORA - Distance From Threshold - Displaced Threshold - Blast Protection

2792m = 3901m - 500m - 309m - 300m

TODA:

Runway: 27R

Calculating for Take Off Towards Obstacle

TODA is equal to TORA: 1782m

Runway: 09L

Calculating For Takeoff Away From Obstacle

NewTODA = Original TODA - Distance From Threshold - Displaced Threshold - Blast Protection

2792m = 3901m - 500m - 309m - 300m

LDA:

Runway: 27R

Calculating For Landing Towards Obstacle

NewLDA = Distance From Threshold - RESA - Strip End

2792m = 3092m - 240m - 60m

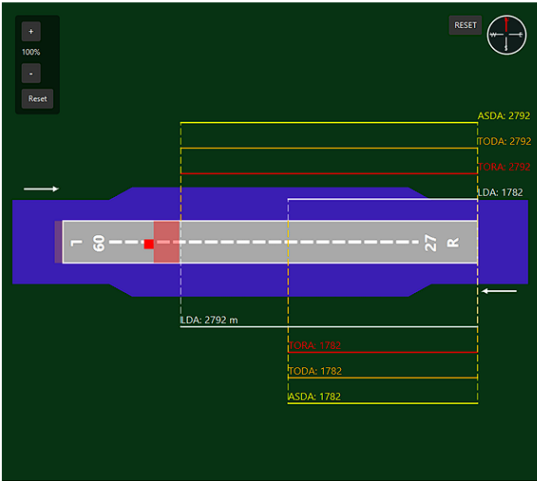
Runway: 09L

Calculating For Landing Over Obstacle

NewLDA = Original LDA - Distance From Threshold - MAX(Slope * Height, RESA) - Strip End

1782m = 3592m - 500m - 1250m - 60m

Top Down View



Side On View

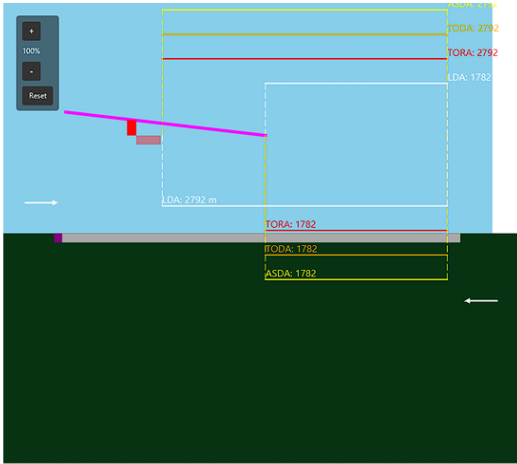


Figure 5: PDF contents second page

A.2 Each logged in has a specific role and access is granted based on the role

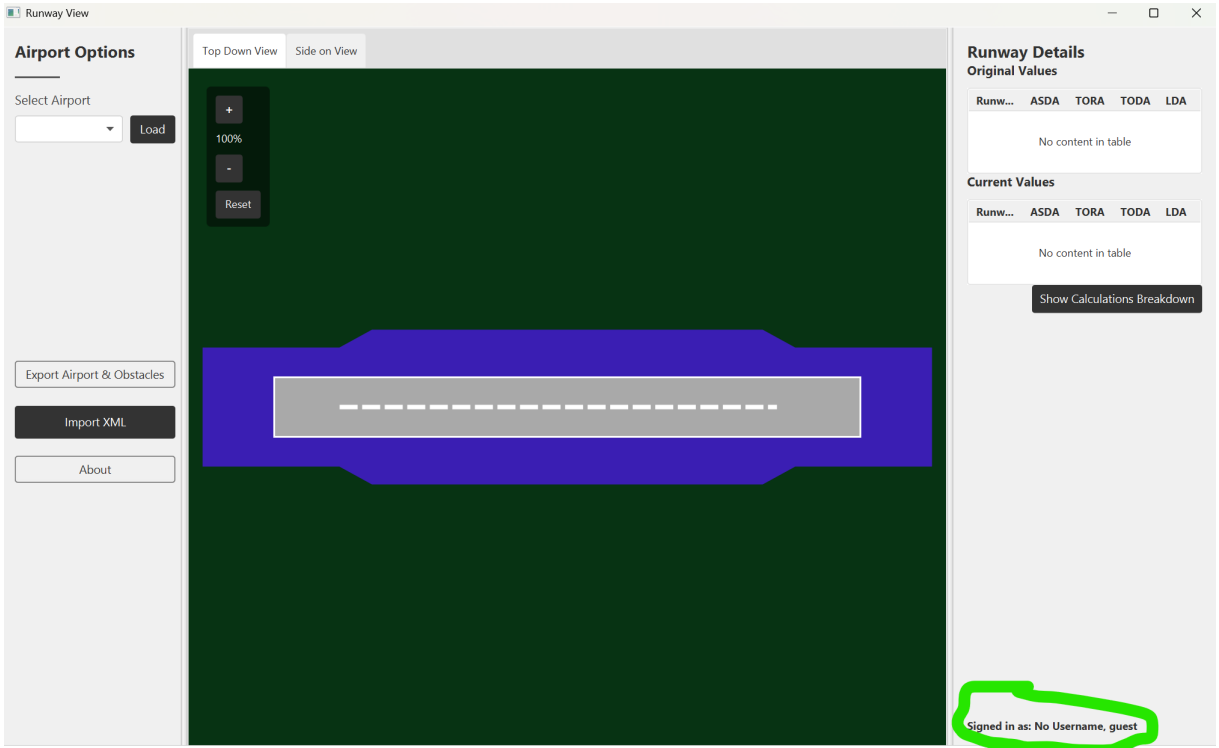


Figure 6: Features available for Guests

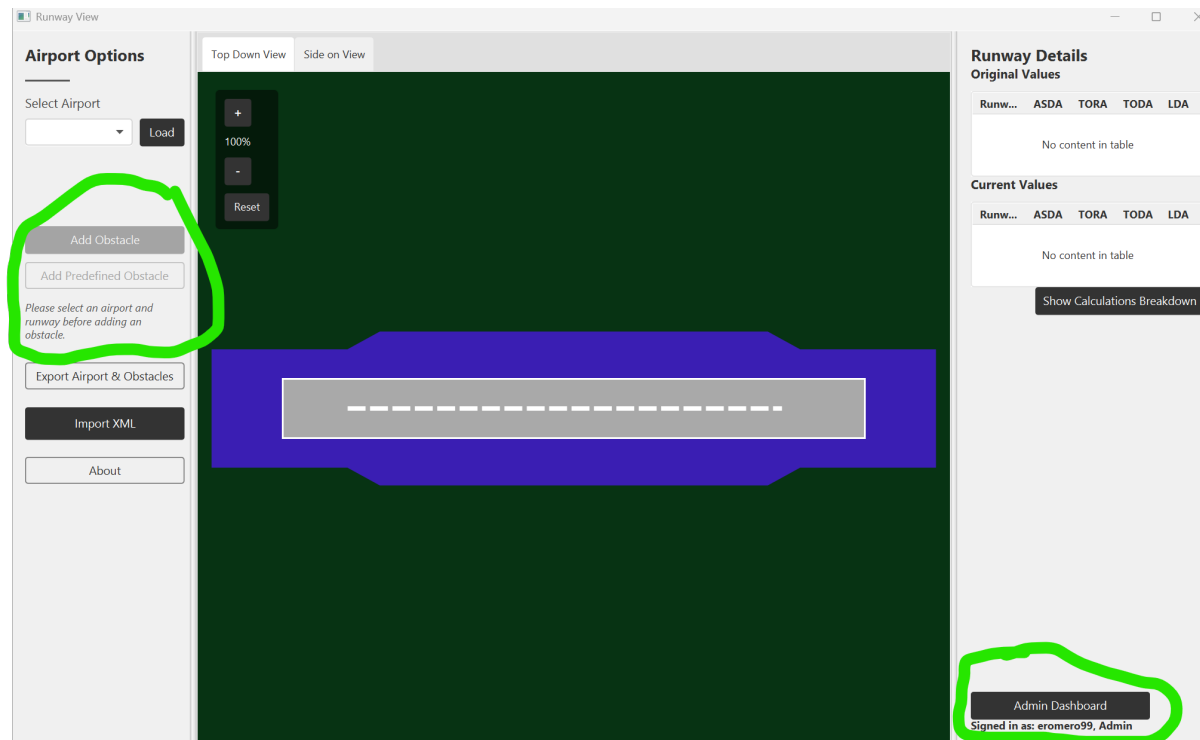


Figure 7: Features available for Admins

A.3 Each field where the user can enter a value has input validation

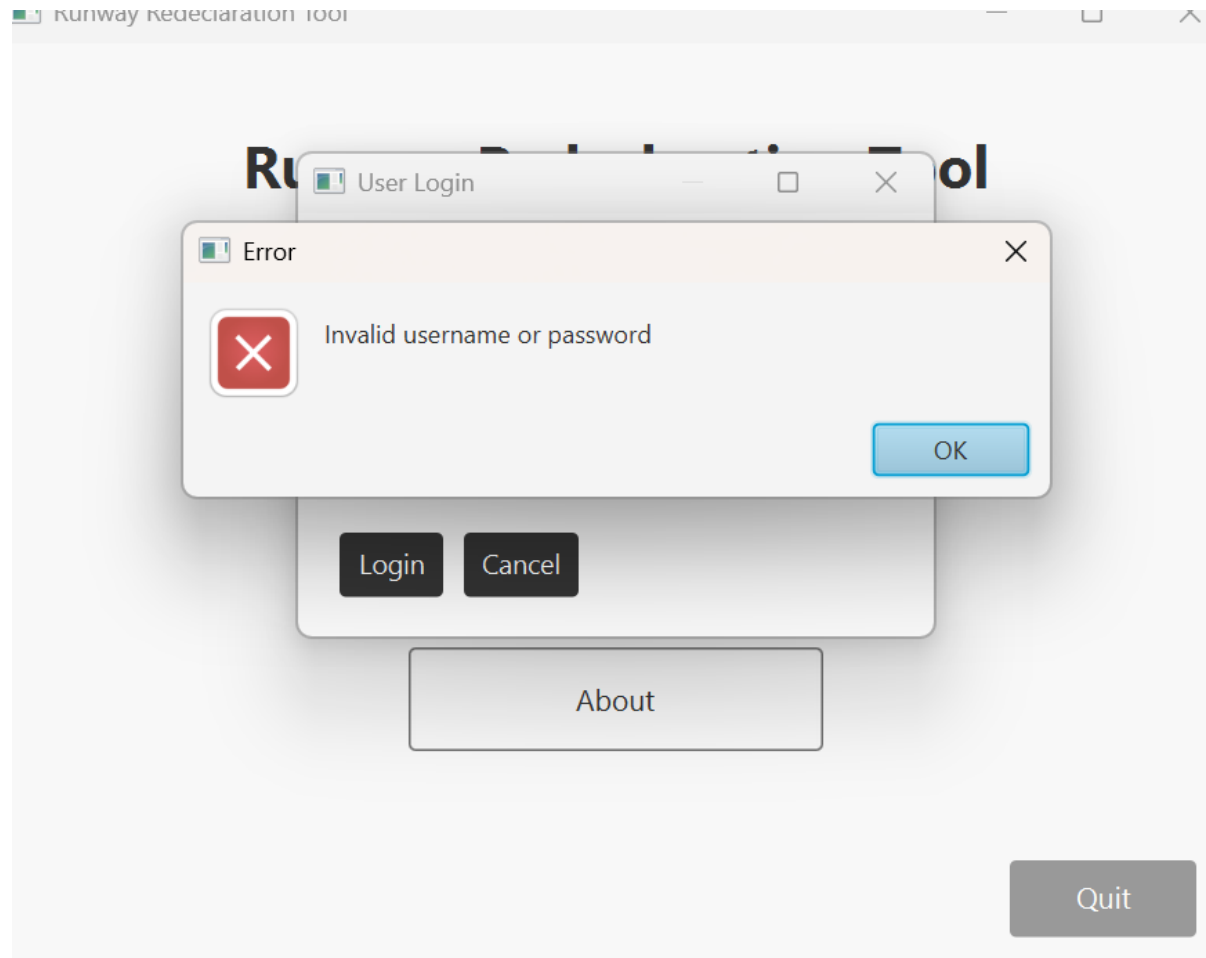


Figure 8: Entering wrong username for login

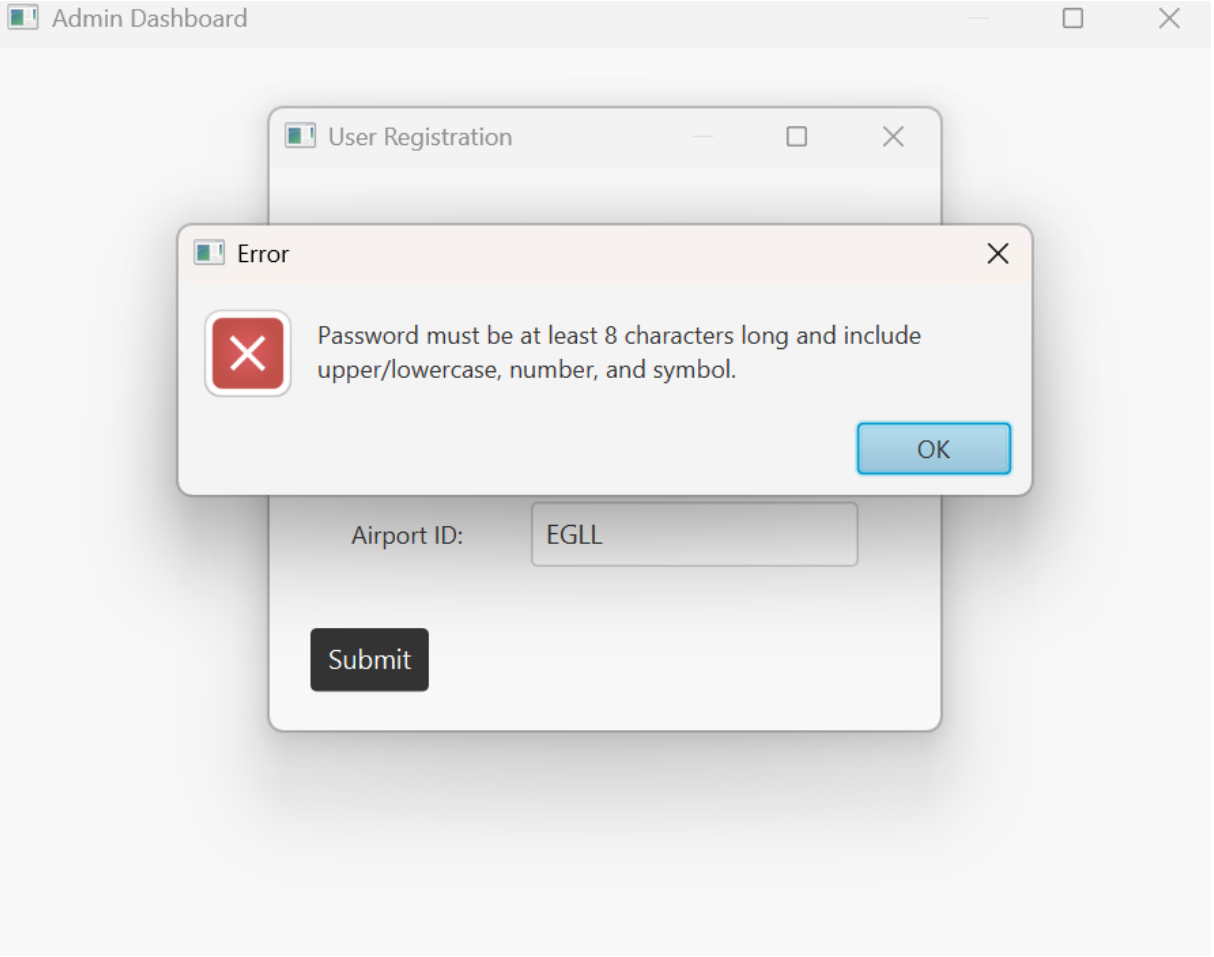


Figure 9: Password incorrect format

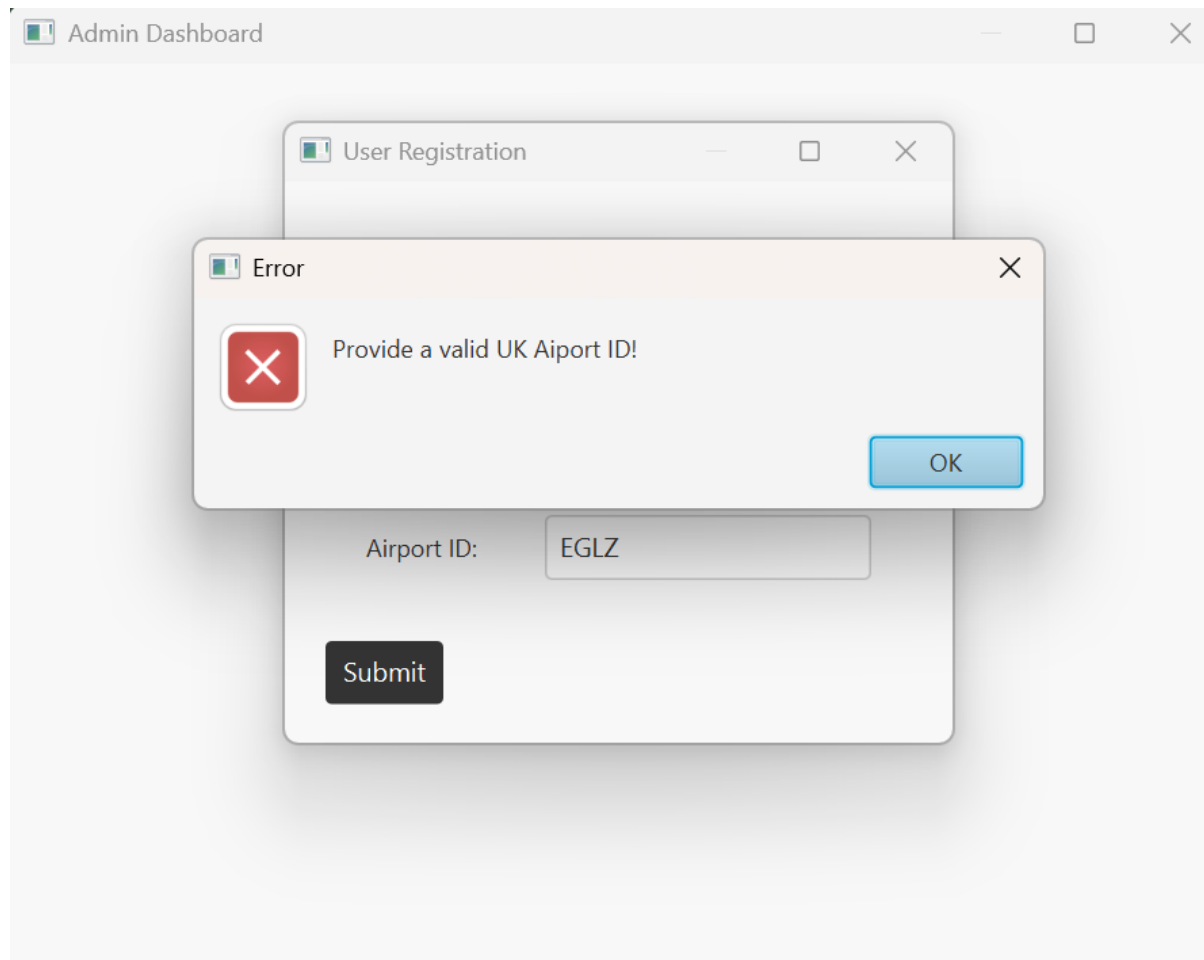


Figure 10: Non existing airport ID

Add Obstacle

×

Declare Obstacle

Distance to threshold 01-18:

m

Distance to threshold 19-36:

m

Distance to centreline:

m

Obstacle height:

m

Auto-fill: ON

Cancel

Generate

Figure 11: Leaving a field blank in obstacle adding

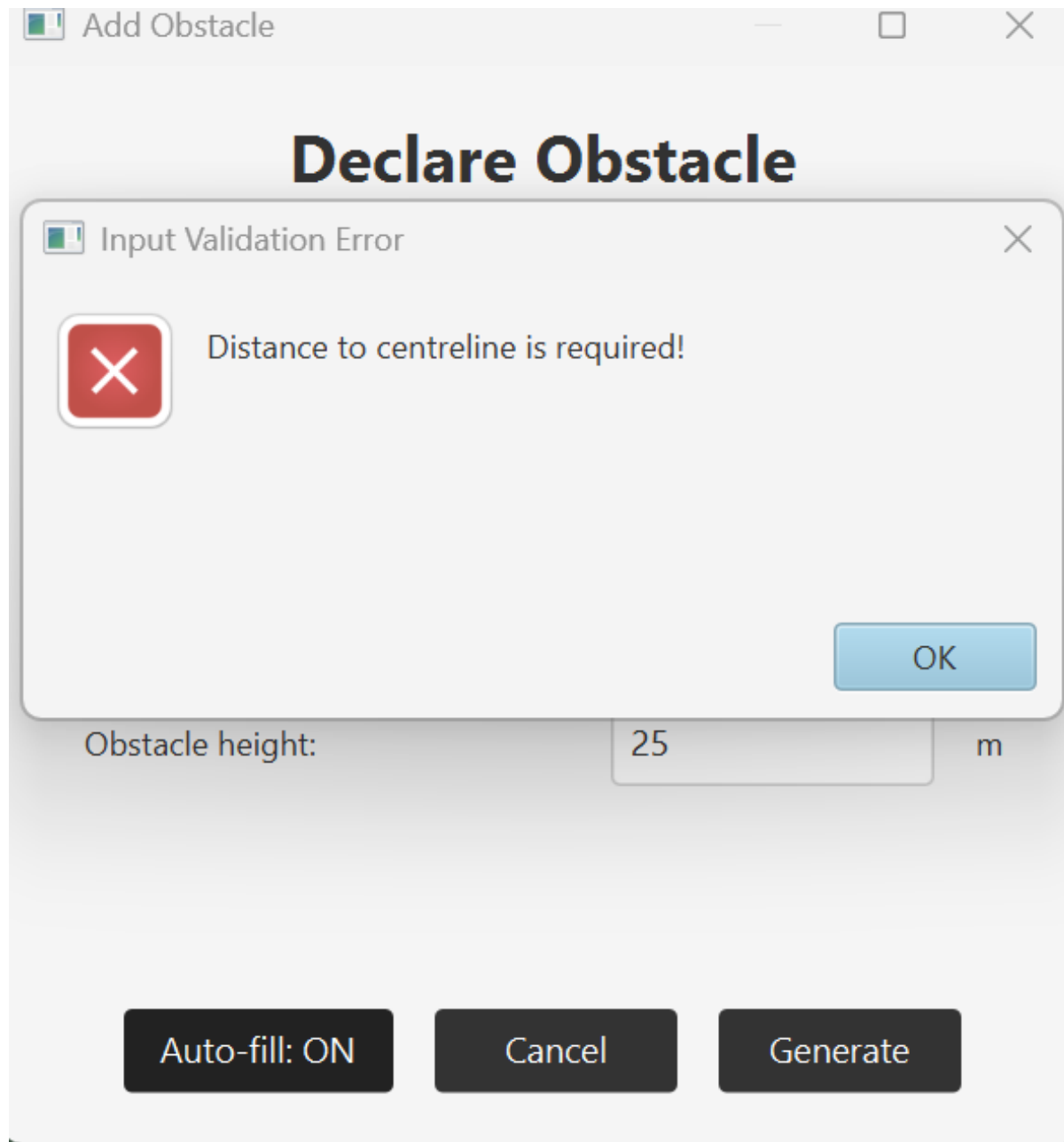


Figure 12: Error Generated by obstacle field

A.4 Visual and sound notifications after placing obstacles and new airport addition

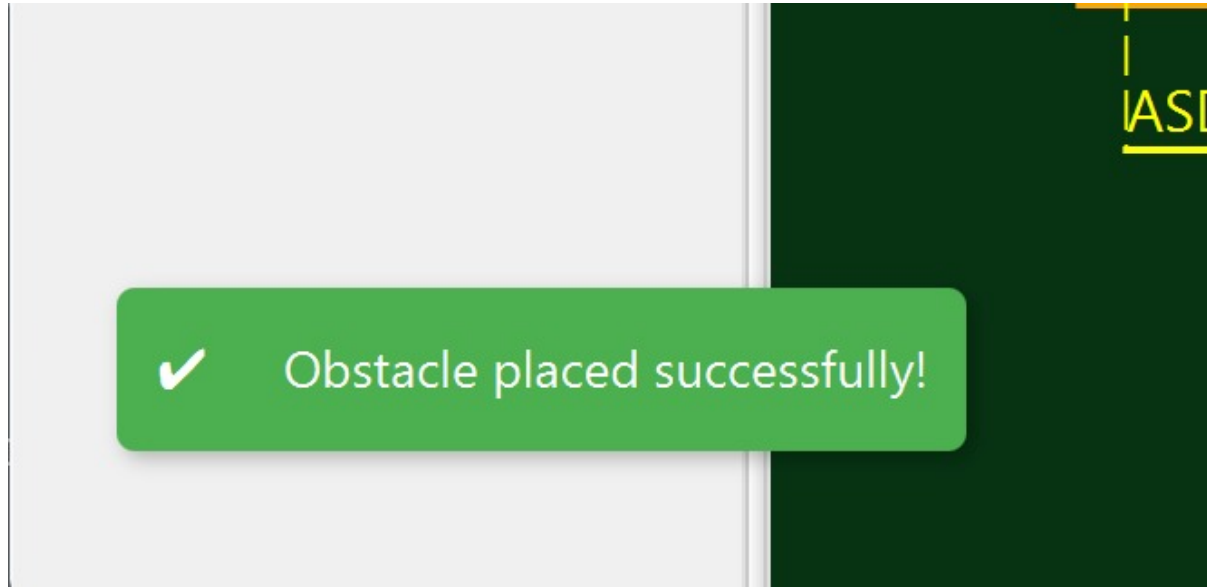


Figure 13: Successful Notification for Obstacles

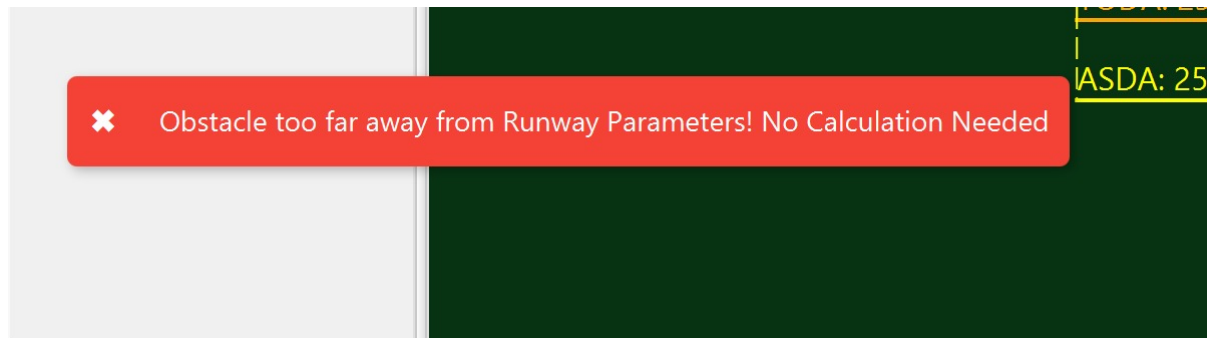


Figure 14: Unsuccessful Notification for Obstacles - runway parameters

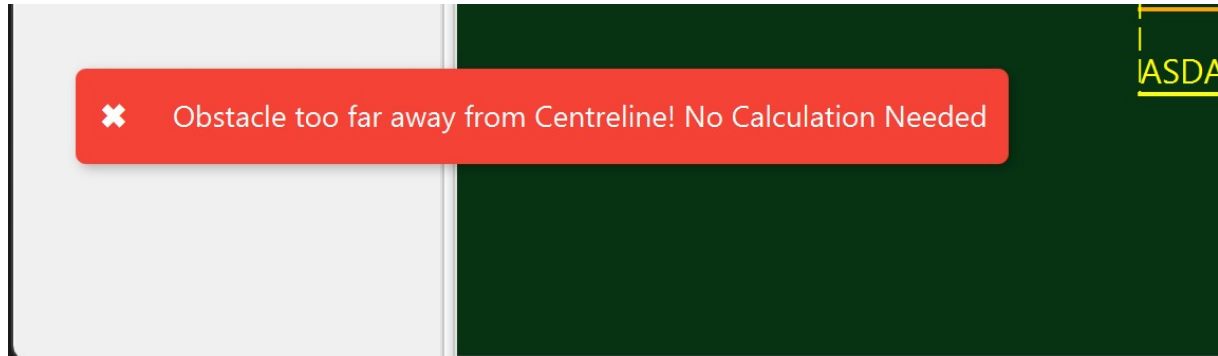


Figure 15: Unsuccessful Notification for Obstacles - centre line

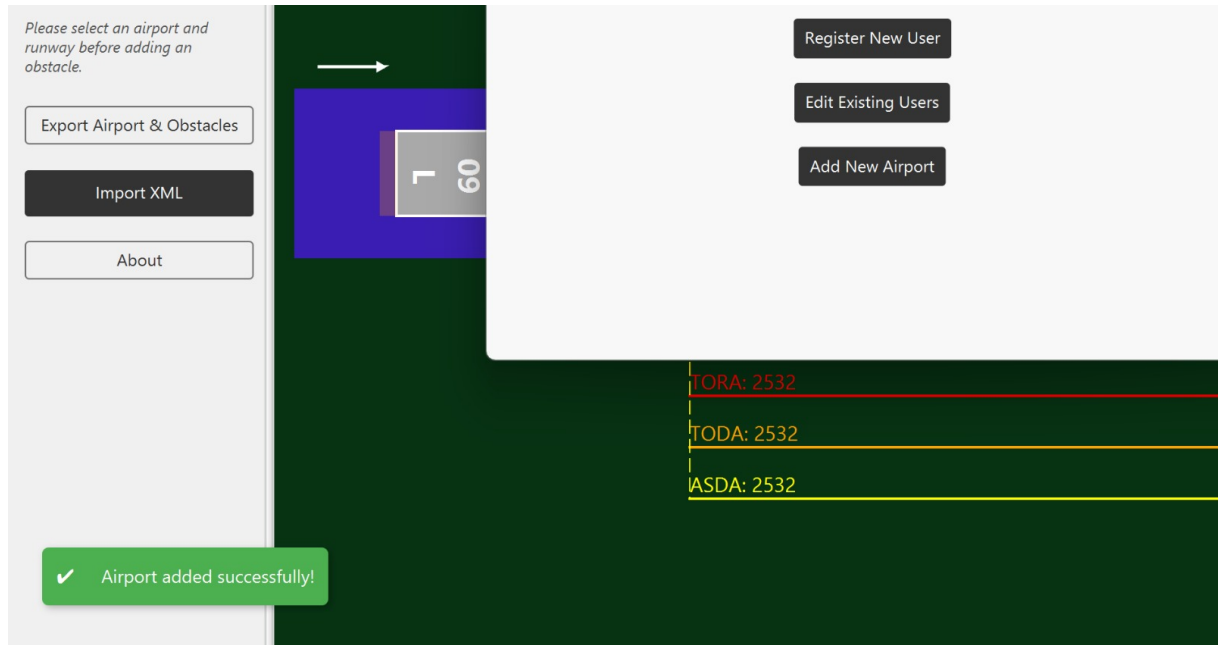


Figure 16: Successful Notification for Obstacles

A.5 Both top-down and side-on views are available and are updated with obstacle placement

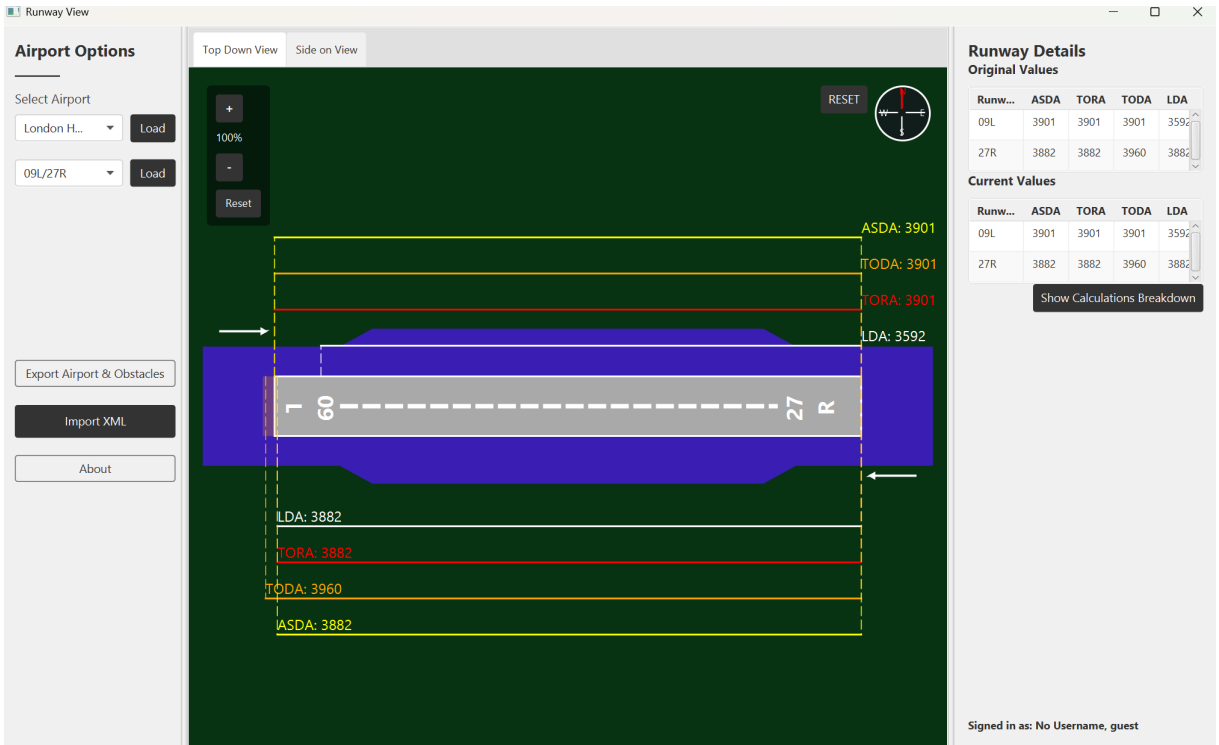


Figure 17: Top Down View for the Airport - blank runway

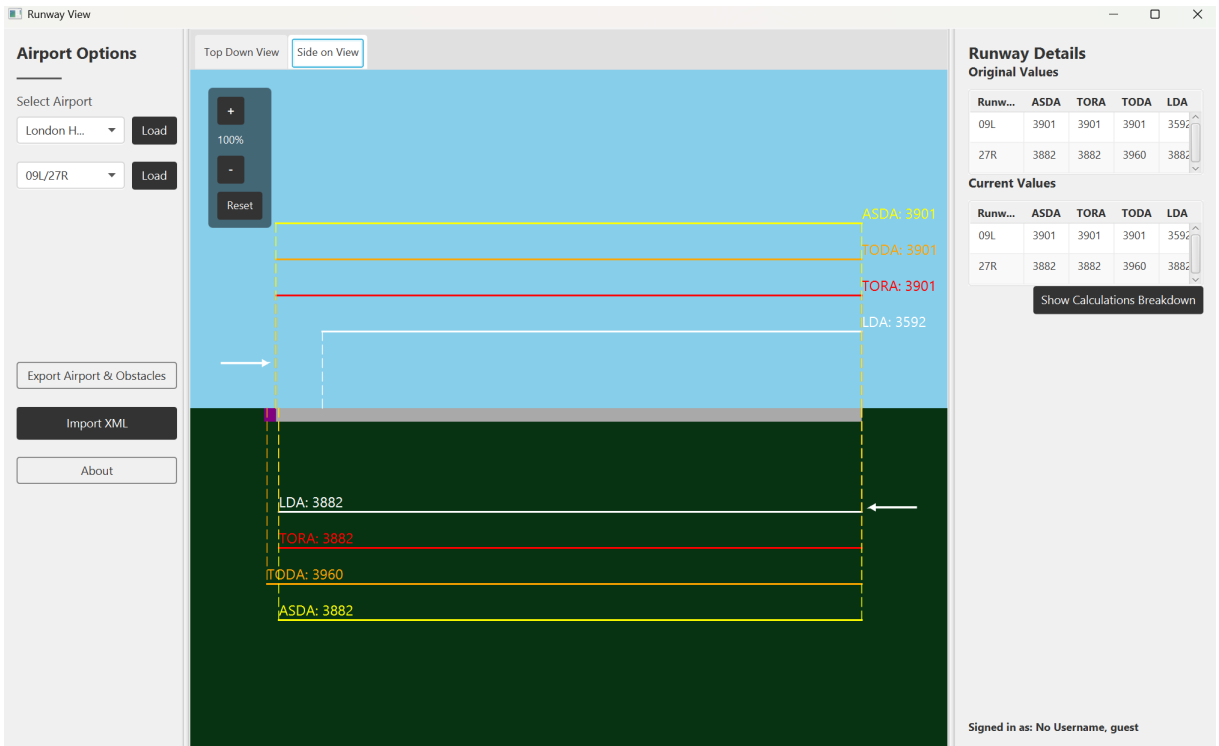


Figure 18: Side on View for the airport - blank runway

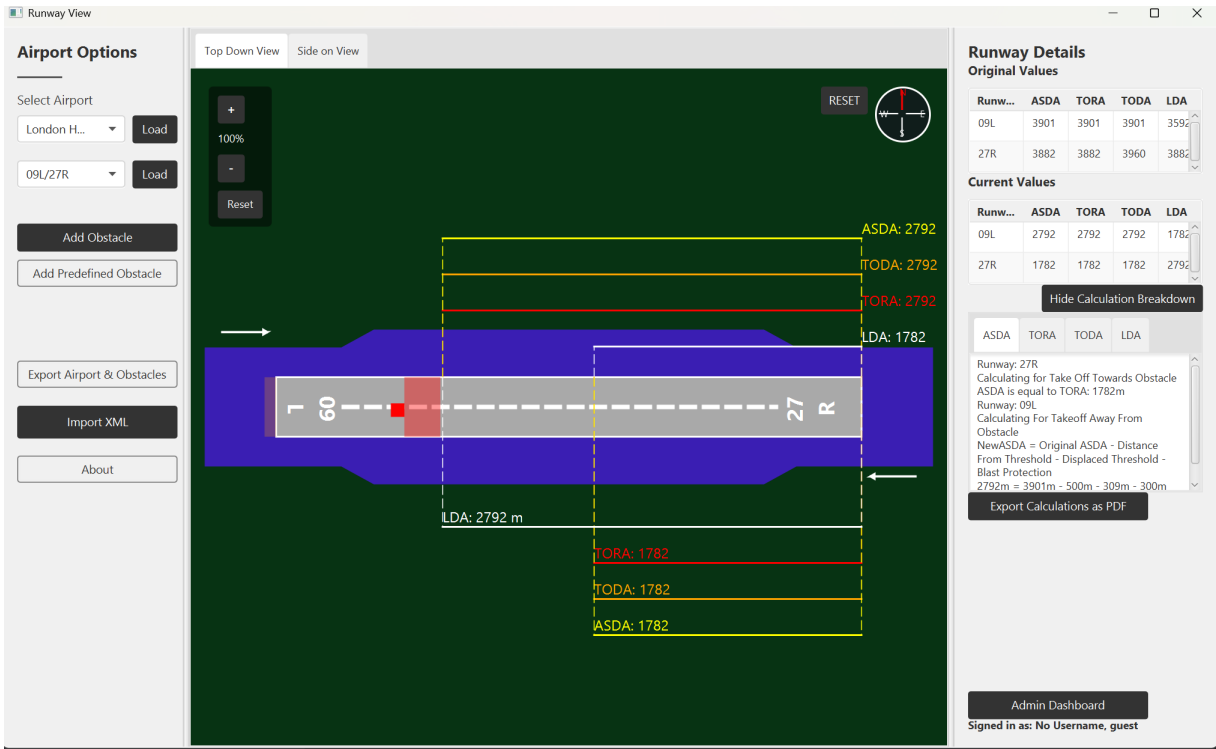


Figure 19: Top down View for the airport - obstacle placed

A.6 User can access the help documentation from both screens

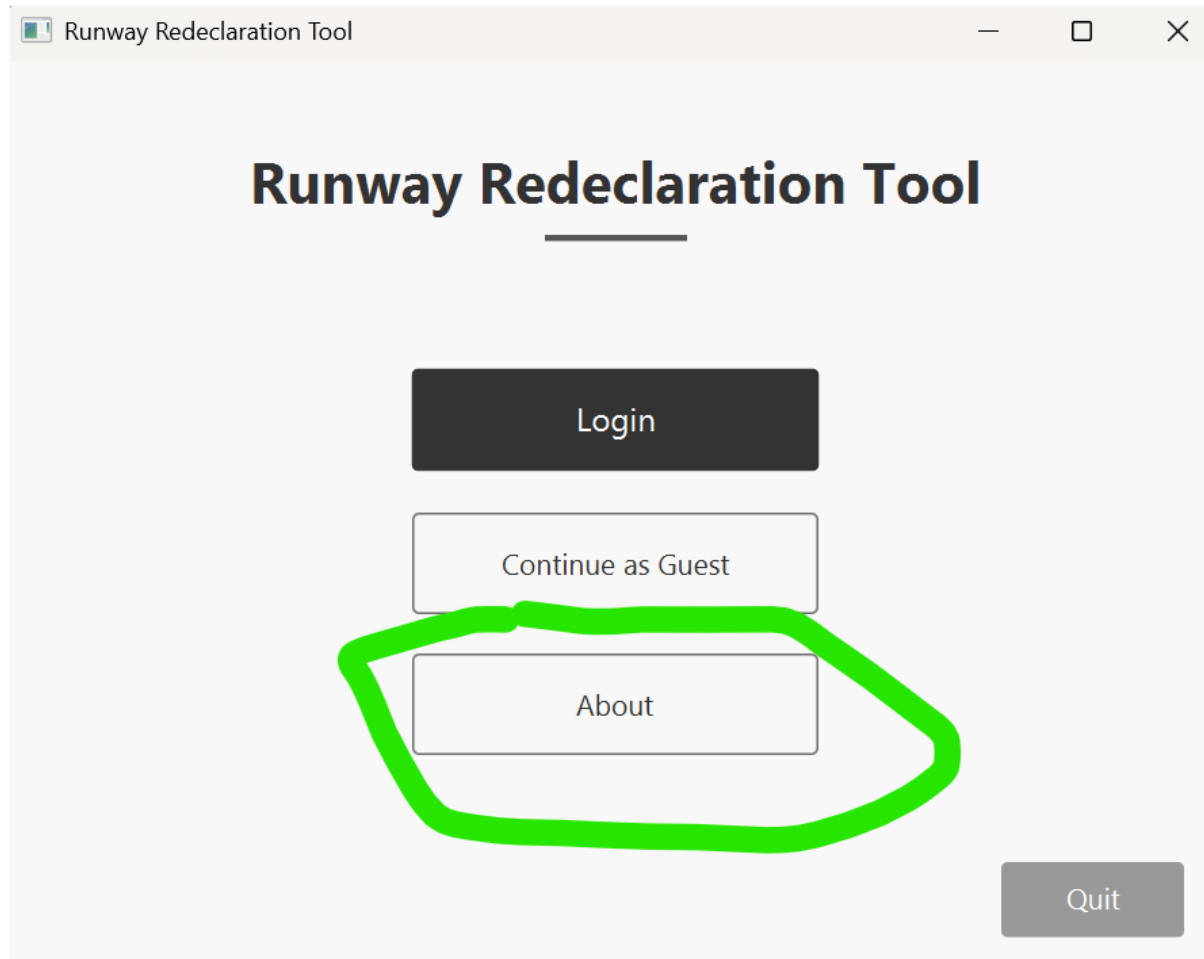


Figure 20: Help documentation button in the login page

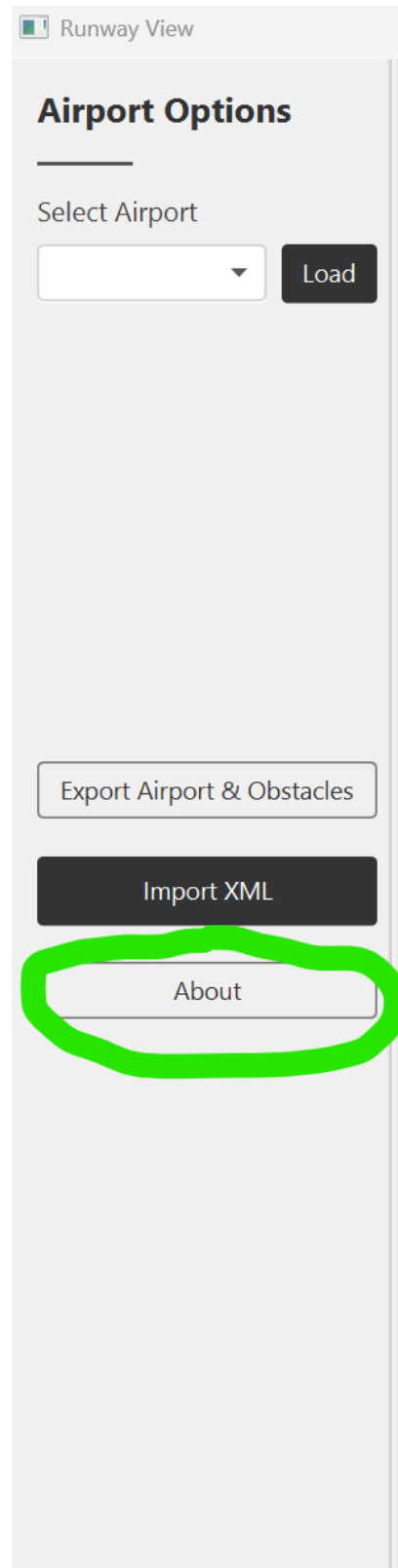


Figure 21: Help documentation button in the main page

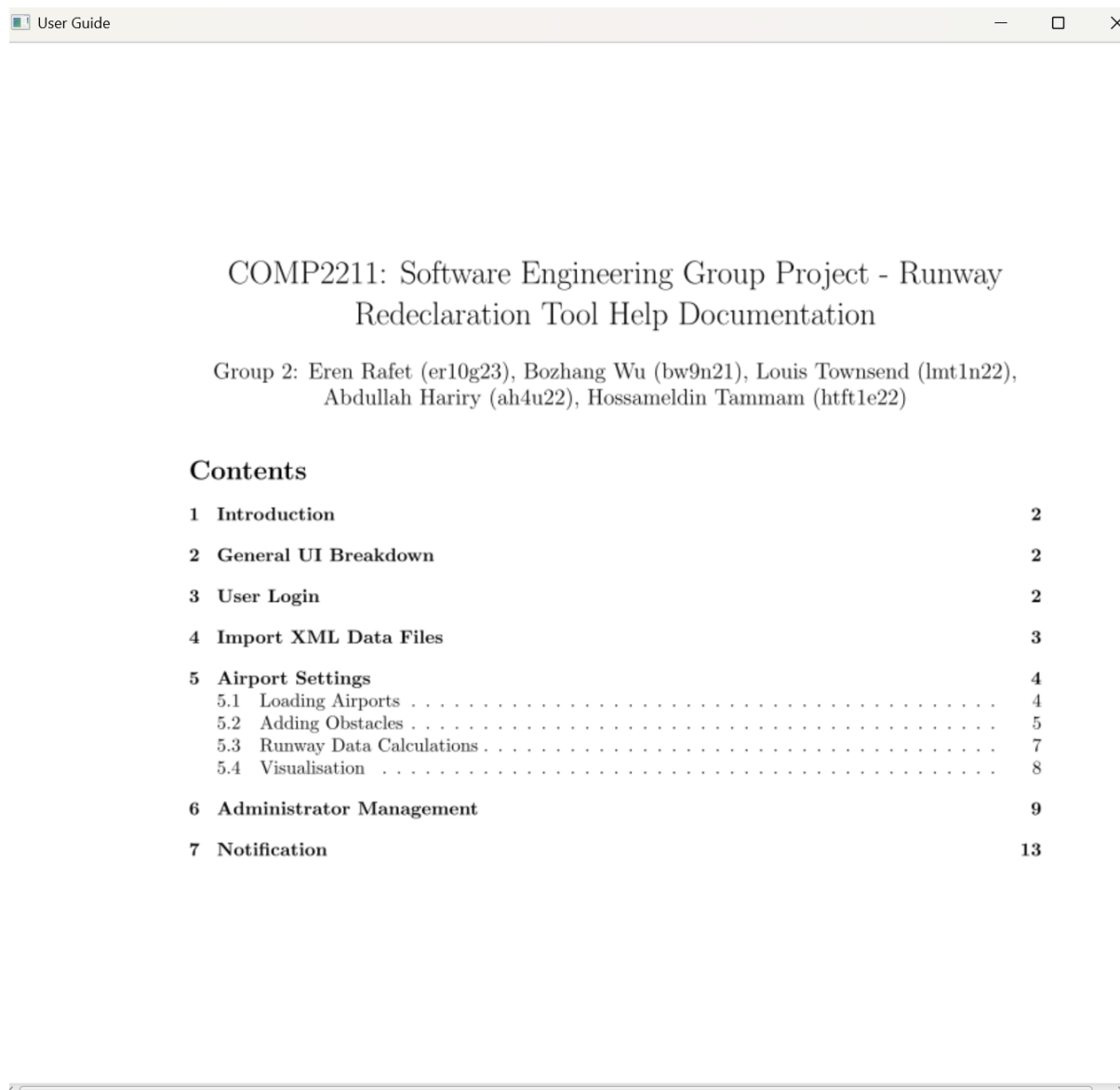


Figure 22: Help documentation content

✓	✓	AutoFillTest (uk.ac.soton.group2seg)	1 sec 82 ms
	✓	Test distance calculation with non-numeric input	1 sec 45 ms
	✓	Test with runway 09L/27R - obstacle 50m before 09L threshold	15 ms
	✓	Test with runway 09R/27L - obstacle 500m from 27L threshold	5 ms
	✓	Test when auto-fill is disabled	3 ms
	✓	Test with runway 09L/27R - obstacle 50m from 27R threshold	3 ms
	✓	Test with runway 09R/27L - obstacle 150m from 09R threshold	5 ms
	✓	Test distance calculation with null input	3 ms
	✓	Test with runway 09R/27L - obstacle 50m before 09R threshold	3 ms

Figure 23: AutoFill Tests