Gmail Fauzan Armia <fauzanarmia@gmail.com>

We've processed your Assignment extension request FORM-AEX-133592
1 message

no-reply@qut.edu.au <no-reply@qut.edu.au> Thu, Jun 9, 2022 at 9:37 PM
To: fauzan.armia@connect.qut.edu.au

Hi Fauzan,

Thank you for your assignment extension request **(FORM-AEX-133592)**.

We have approved your request and the due date for your assignment **Software Development Project**, for unit CAB302 has been extended by 48 hours from the original due date. If your unit outline does not specify that your assignment is eligible for an extension, this confirmation email is not valid and unless you submit by the original due date, the late assessment policy will apply.

You are responsible for ensuring that this assignment is eligible for extension before submitting it after the original due date. Check your unit outline for eligibility.

As you indicated this is a group assignment, you are also responsible for informing other members of your group of this extension.

Be aware that a copy of this email is kept on file. You should not alter this email in any way. Email notifications that have been altered or differ in any way from the original may result in an allegation of student misconduct as set out in the Student Code of Conduct.

Need extra support? You can access free, confidential counselling with qualified professionals. We also offer planning and support if you have a disability, injury or health condition that affects your ability to study. If you are struggling to meet your university commitments, academic support is also available.

Have a question? You can contact us by email or phone. We're also available to help you on campus or via online chat. Visit the HiQ website for our contact details and opening hours.

Email us | +61 7 3138 2000

HiQ is your place to go for **student services, support** and **general enquiries:** qut.to/abouthiq

hi, how can we help you?

You have received this email because you have submitted an assignment extension request. View our Privacy and Security statement.

CAB302 Detailed Design Report

Group 51 – Harry Fry (n10215611), Fauzan Armia (n10010921), Ryan Hafizh Indrananda (n10852565), Neeta Vijaya Kumar (n11262109)

Table of Contents

User Stories **4**

Project Burndown Graph 7

Software Deployment Instructions **7**

UML Class Diagram **8**

GUI Design **9**

JavaDoc Documentation **19**

User Stories

The team has allocated priorities and story points to the provided user stories to help us to understand what implementations are the most essential to the project. This can be seen in the table below. ‘Must have’ priorities will be necessary implementations, ‘Should have’ are implementations that are important, and which a great deal of effort should be expended to implement them, but which are not integral to the project and its success, and ‘Nice to have’ will be implementations which would be beneficial to implement, but are not essential. The team will aim to implement everything included in the table below, however due to complexity of the project, time constraints, and potential unforeseen circumstances, not all of it may be implemented. This is why the team decided to create user stories.

The story points represent the amount of effort required to implement a user story, where 8 represents the greatest amount of effort, and 1 the least. Story points are utilised in the release plan to assist in planning how much work the team should aim to complete for each release so as to keep them on track.

Best Interest of	User Story	Priority	Story Points	Progress
Maze designer	Creating an automatically generated maze that can be edited post creation. Maze has a maximum size of 100x100 units.	Must Have	8	Successfully Completed
Maze designer	Maze having the option of inserting a logo within it which is not in the path of the maze itself	Must Have	4	Successfully Completed
Maze designer	Change the entrance and exit of the maze to a logo/picture that is targeted towards children	Must Have	4	Successfully Completed

Maze designer	Be able to give the maze a title, also providing credentials of who created the maze and finally including date and time the maze was first created and last edited	Must Have	1	Successfully Completed
Maze designer	Have a button to show that at any point of starting the maze that it can be completed. This includes a % to show how much of the maze is left to explore.	Nice To Have	8	Successfully Completed
Maze designer	A toggle button to show the optimal/fastest path for the maze that's being created to be solved	Should Have	4	Successfully Completed
Maze designer /User	Requires a user-friendly GUI interface for both the maze designer and the user	Must Have	2	Successfully Completed
Publisher	Be able to see a list of created mazes, along with who created them and the date and time.	Must Have	2	Successfully Completed
Publisher	Any maze that has been exported should have the option to show the solvable version of the maze as well	Should Have	1	Successfully Completed
System Administrator	Maze must be stored in a database on the server. To ensure that it is stored in one place and can be backed up	Must Have	4	Successfully Completed

System Administrator	Be able to log in with a username and password to access stored files	Must Have	2	Successfully Completed

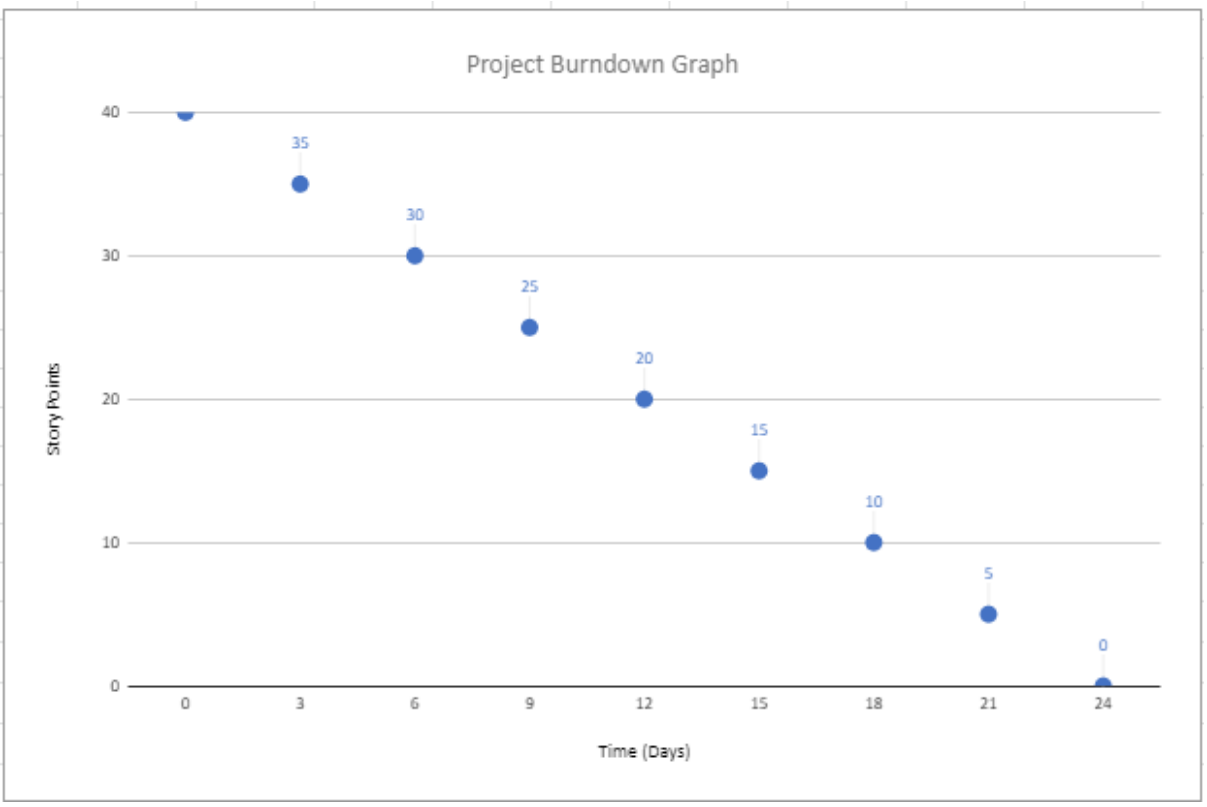
Statement of Completeness

The maze requirements met are as follows:

- The automatic generation of a maze, with a maximum size of 100x100 cells and a minimum of 5x5 cells for a standard maze, and a maximum size of 8x8 and a minimum of 5x5 for a child maze. Anything lower would barely be a maze, not providing a challenge, and anything higher than 8x8 for a child maze would be too complicated.
- Users can customize the size of the maze within the defined size range mentioned above. Therefore, the user has the option to either having the maze generated automatically, where the rows and columns are determined randomly within the defined size range, or they can manually input the numbers themselves. We decided not to do manual cell-by-cell maze creation as it would be tedious for the maze user.
- A percentage of the cells with dead ends is displayed, as well as a percentage of the cells explored in the maze solution is displayed when the ‘Solve’ button is pressed.
- The maze logo was not implemented.
- The maze image can be downloaded as a JPEG image file in the directory of the user’s choosing.
- The mazes are solvable and a user can download the image of the solved maze as a JPEG image file in the directory of the user’s choosing.
- Mazes are saved into a database, and the user can retrieve the existing mazes’ details from the database. Details stored are: maze title, maze length, maze height, author name, creation date, and updated date. These are all manually specified by the user after creation by clicking the ‘Save Maze’ button, but for the creation date and updated date, which are automatically added.
- The user cannot see the maze image when interacting with the database.
- The user cannot properly edit the maze details when interacting with the database - it instead creates a new field.
- The user cannot sort the mazes via criteria in the database.
- Unit testing was done for classes.

Project Burndown Graph

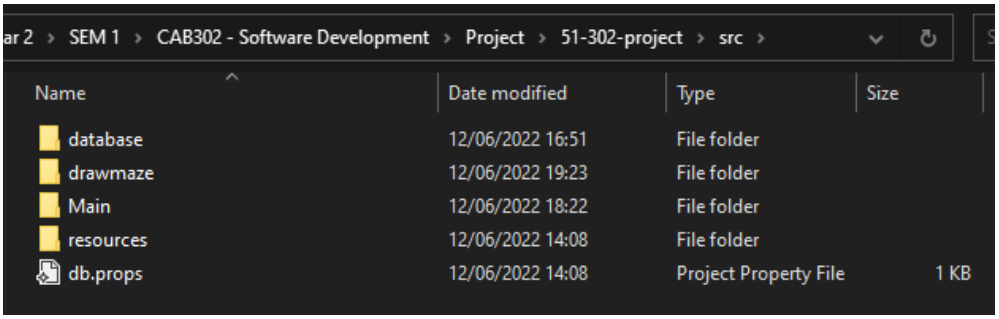
The graph below shows the successful progress of the project according to the set timeline.



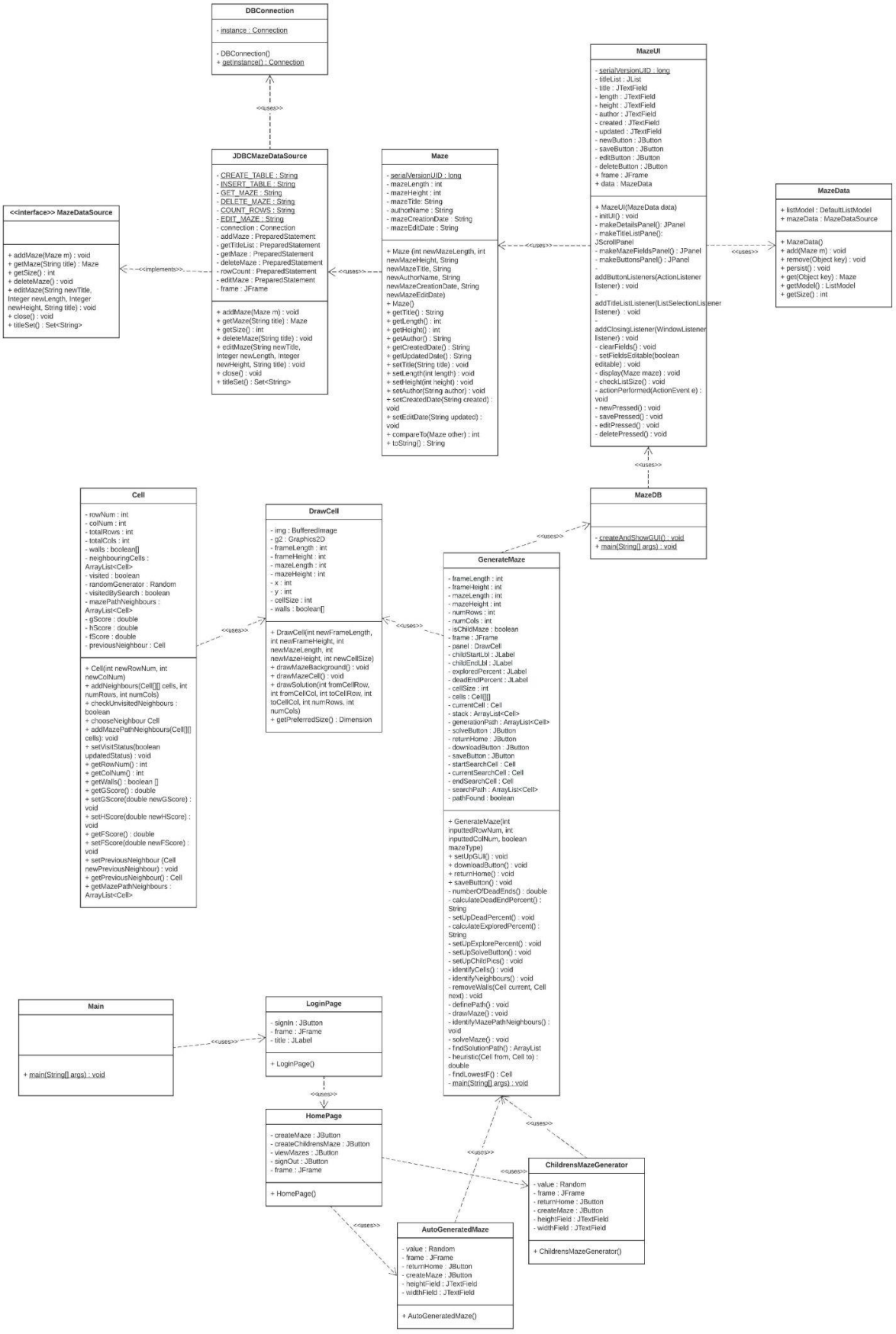
Software Deployment Instructions

ENSURE TESTSUITE FOLDER IS EXCLUDED WHEN NOT CONDUCTING UNIT TESTING.

To set up the database, ensure that the db.props file is inside the src folder along with the database, drawmaze, and Main packages. Any time the database is called within the application, such as with the 'View Existing Mazes' or 'Save Maze' buttons, the DBConnection class will be called which will open a connection relying on the db.props file being located within the src folder. When a connection is opened, a Maze.db file will be created, the database file which will store all the mazes and its details. No external Java frameworks were used.

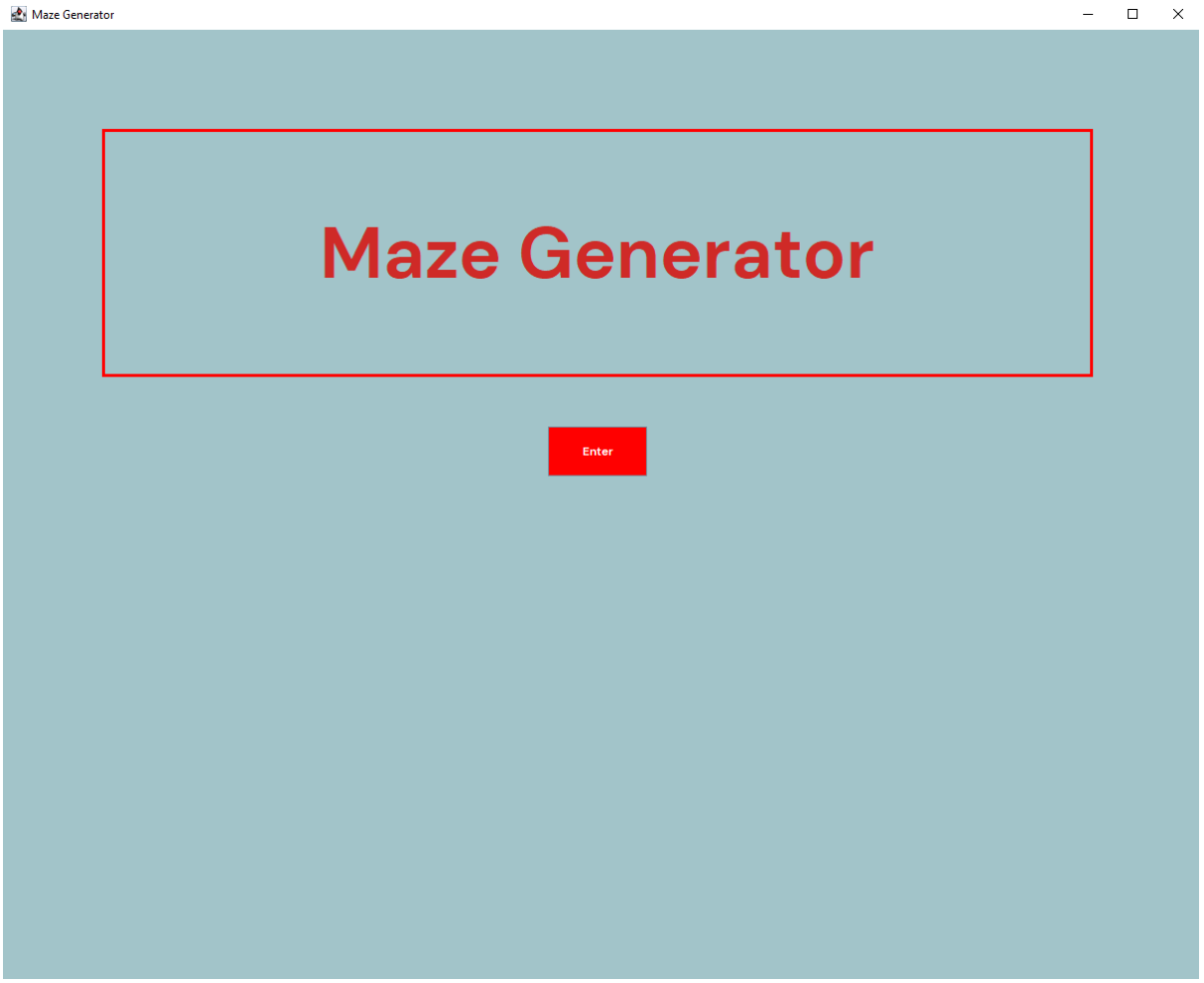


UML Class Diagram

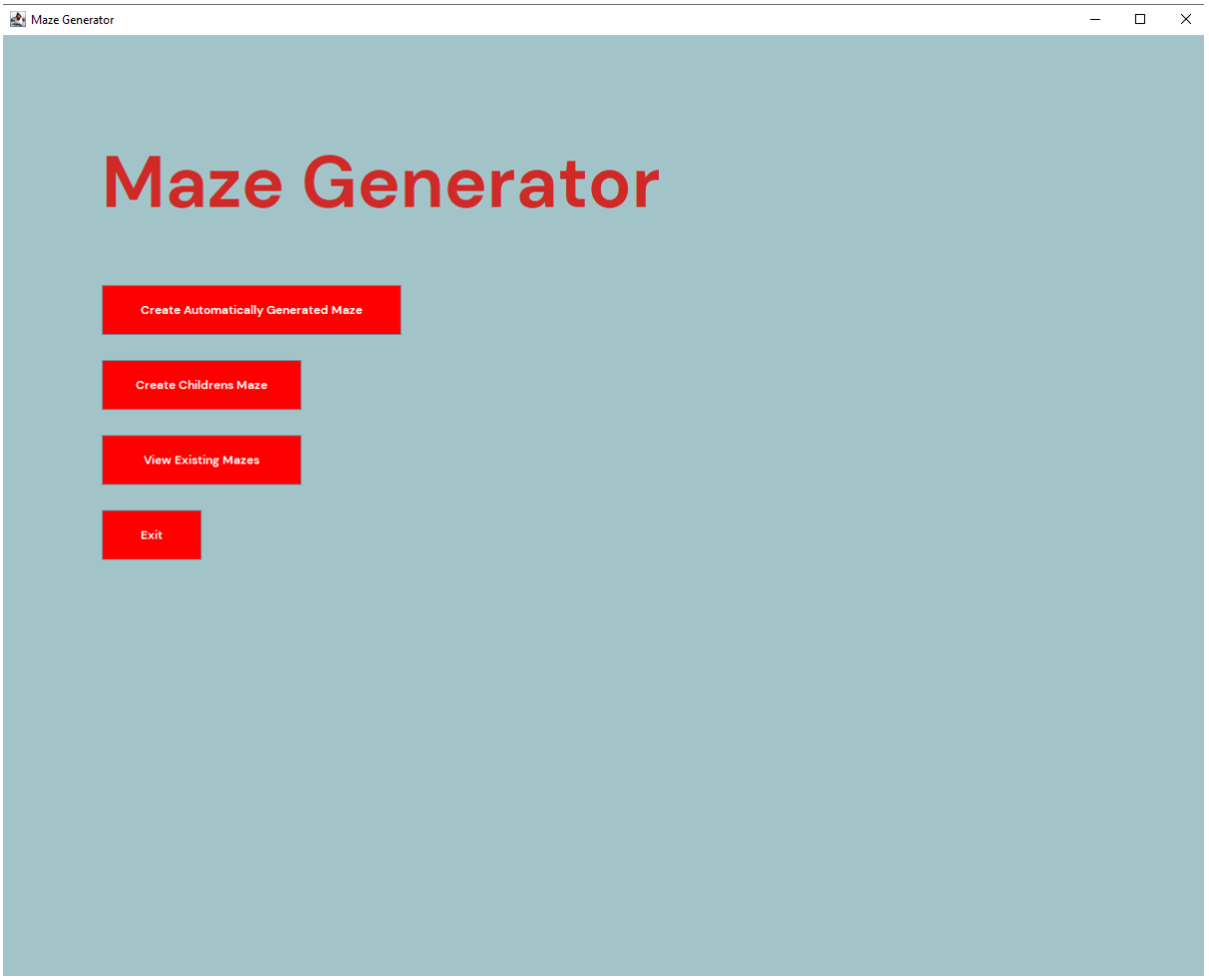


GUI Design

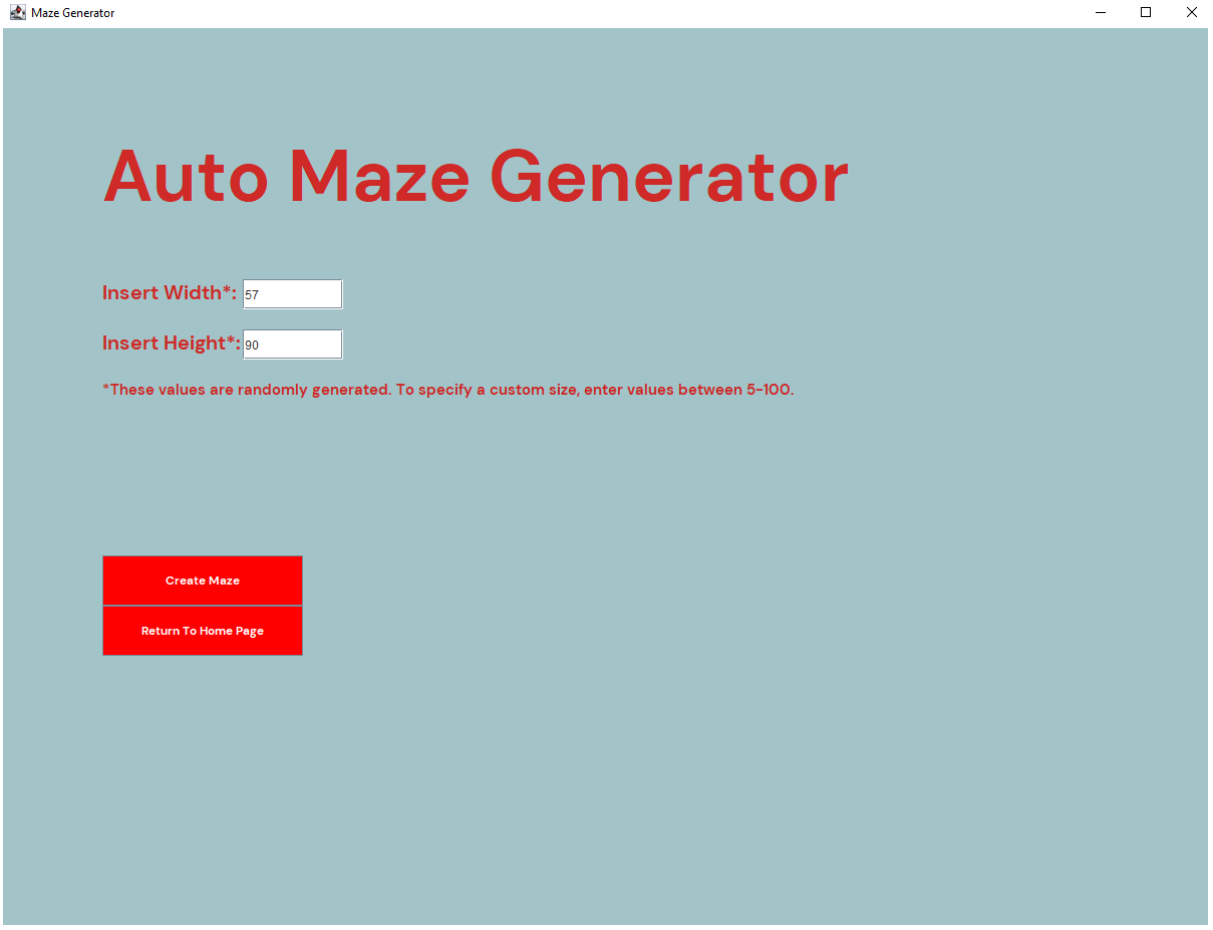
Starting Page:



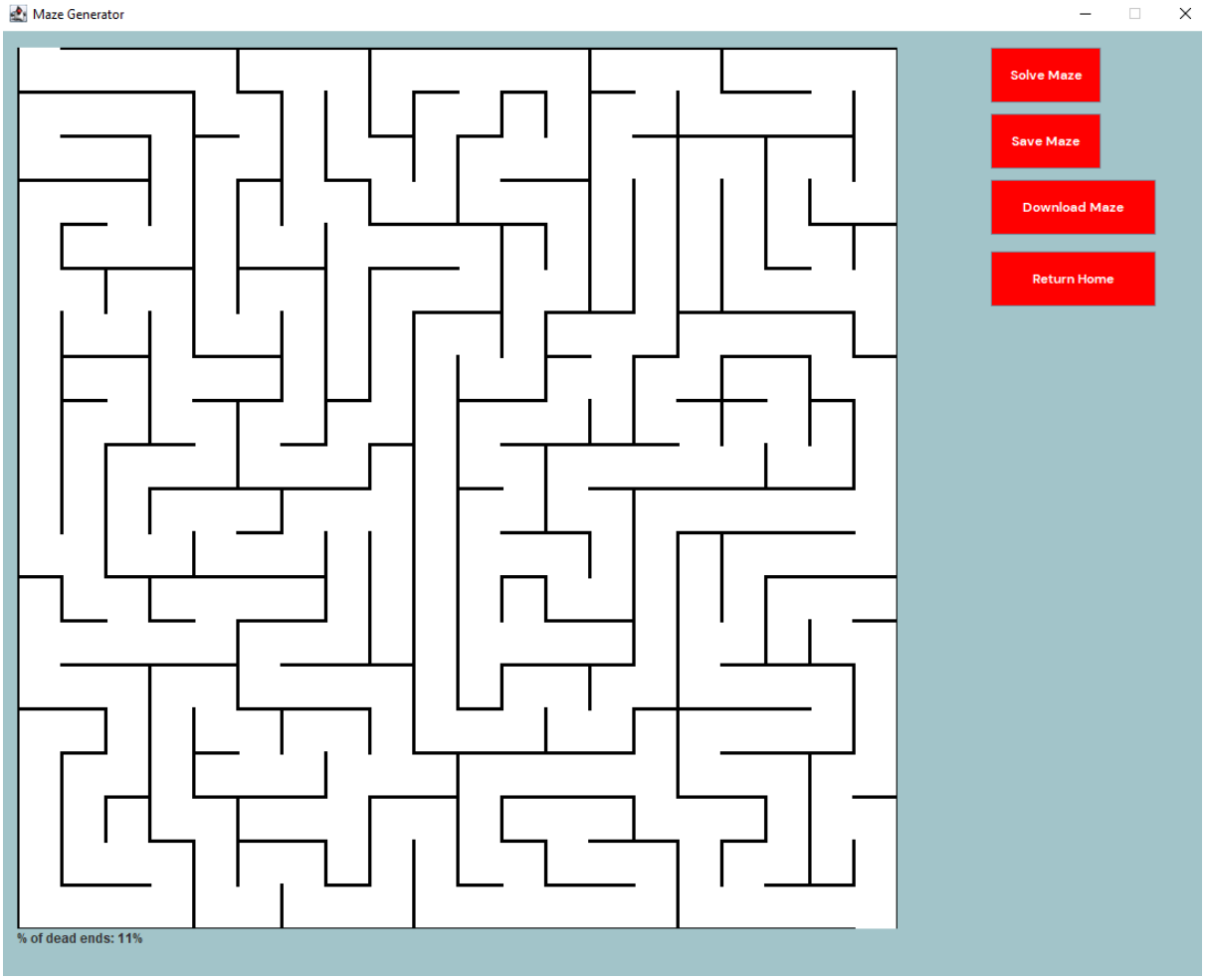
Home Page:


Maze Generator
Maze Generator
Create Automatically Generated Maze
Create Childrens Maze
View Existing Mazes
Exit

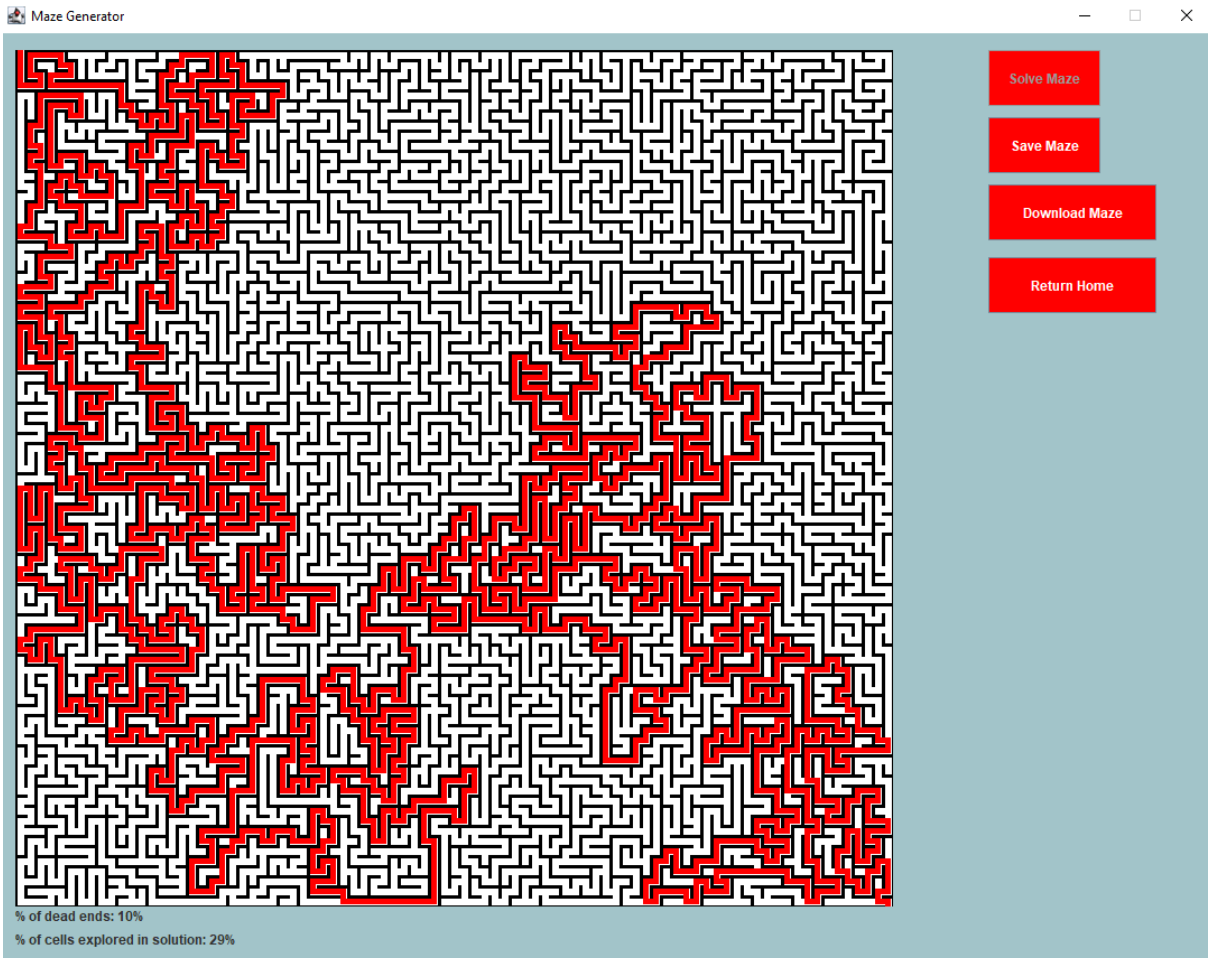
Auto Maze Generator Page:



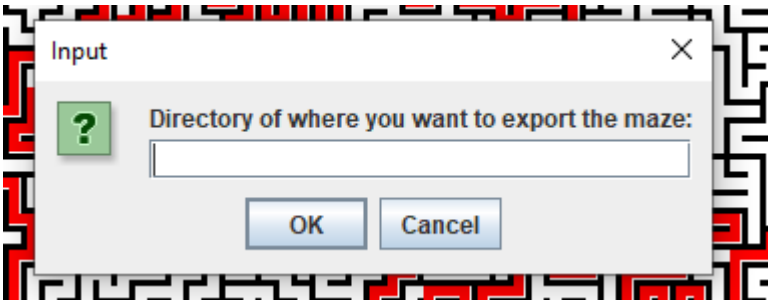
Created Auto Maze:



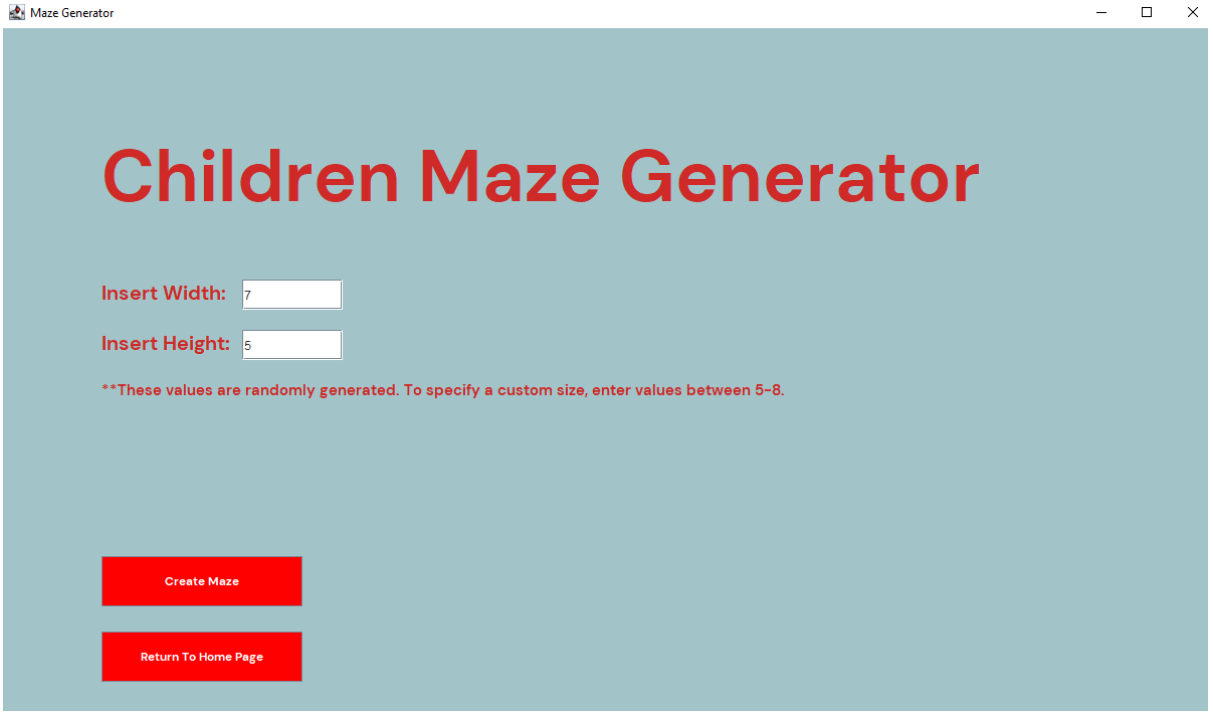
Solved Created Auto Maze:



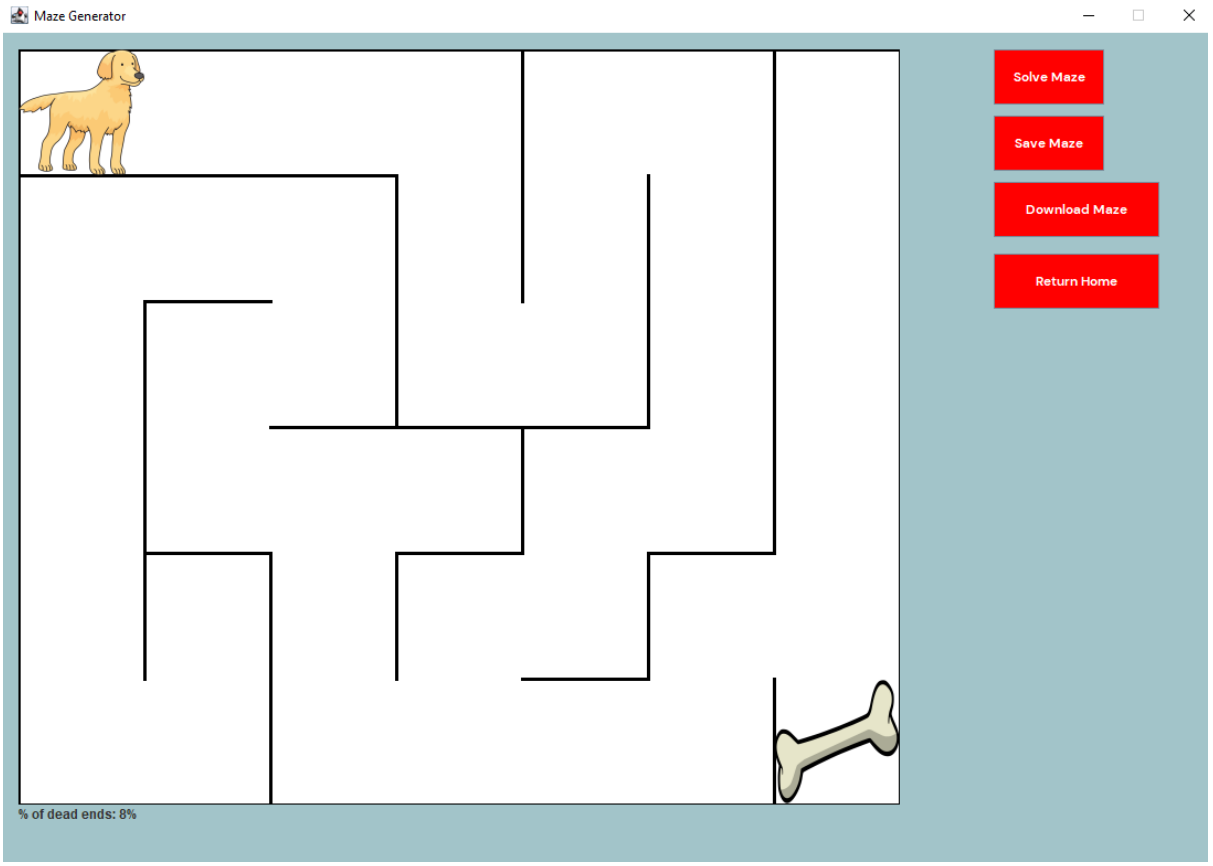
Popup Box when download maze



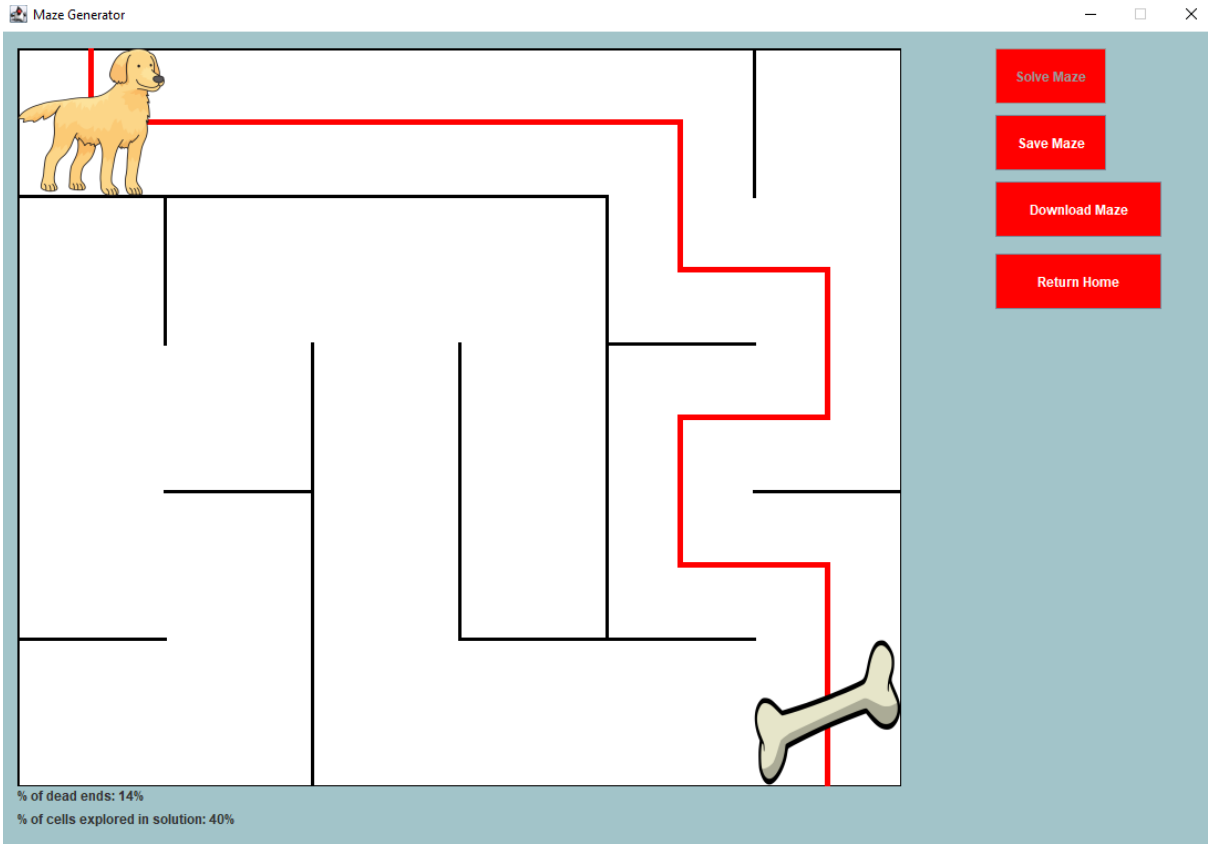
Child Maze Generator Page:



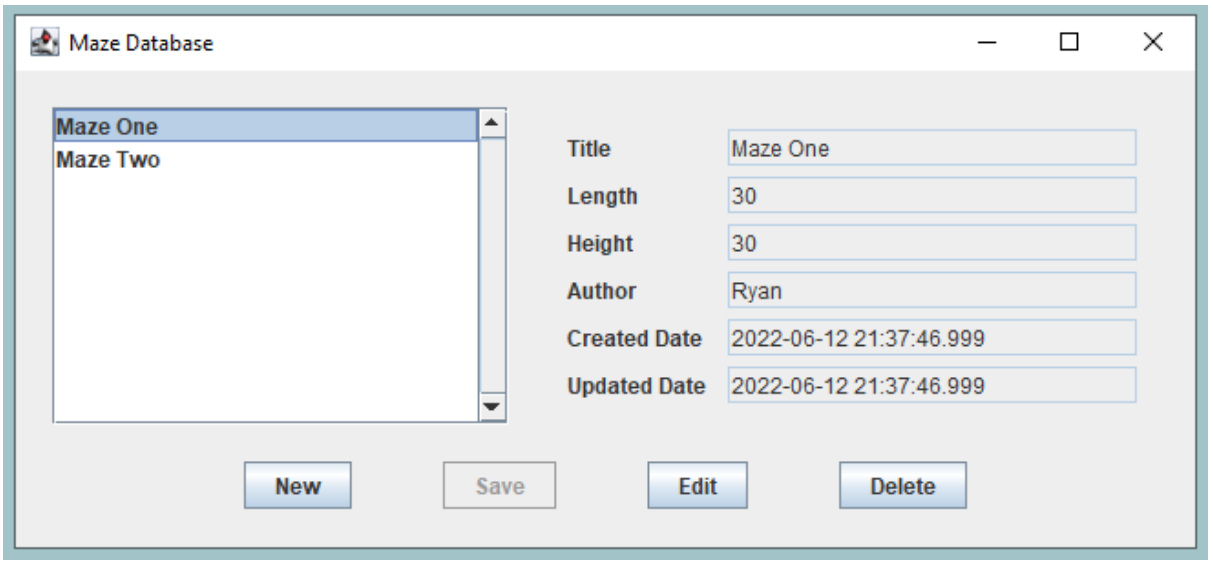
Auto Child Maze:



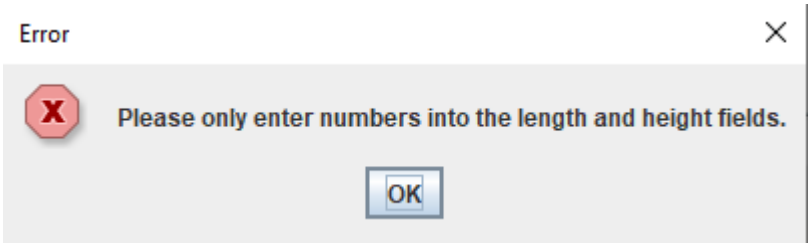
Solved Auto Child Maze:



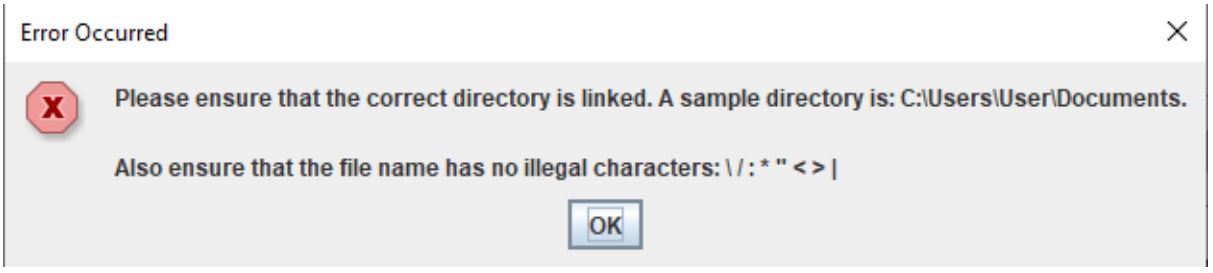
Maze Database:



Error message if user enters letters instead of numbers:



Error message if user doesn't enter correct directory or illegal characters when exporting maze:



Unit Testing

The unit testing was carried out using JUnit5 to ensure the project quality. The table below is a test plan that was utilized throughout the project as each sprint was completed. The test case type was mainly for functionality.

Task	Expected Outcome	Result
Creating an automatically generated maze	Once the button is clicked, a maze is automatically generated.	PASS
Maze can be edited post creation.	User is able to edit the maze after the maze is automatically generated.	PASS
Maze has a maximum size of 100x100 units.	Maze size does not exceed 100x100 units or an error occurs.	PASS
Change the maze to a targeted towards children maze.	Changes the entrance and exit of the maze to picture that is targeted towards children	PASS
Be able to give the maze a title.	User is able to input characters for the title.	PASS
Date and time the maze was first created and last edited is shown.	An accurate date and time the maze is first created and last edited is shown.	PASS
A button to show the percentage of how much of the maze is left to explore.	When the button is clicked, an accurate percentage of how much of the maze left to be completed is shown.	PASS
A toggle button to show the optimal path for the maze that's being created to be solved	When the button is clicked, the optimal solution for the maze is shown.	PASS
Export maze generated to image file.	User is able to export the maze generated to an image file.	PASS
Export maze generated with solution to image file.	User is able to export the maze generated with its solution to an image file.	PASS

Maze can be saved and stored in a database.	User saves and stores the maze created into the database.	PASS

JavaDoc Documentation

The designs of the Java classes used that comprise our program have been compiled using the JavaDoc tool and implementation. Access the JavaDoc documentation via the ‘index’ html file in the ‘docs’ directory in the submitted project folder (51-302-project\doc).