Daily Doom

Contents

[I. Initial Idea 2](#_Toc82691176)

[I. Diagram 2](#_Toc82691177)

[II. Intended Program Flow 3](#_Toc82691178)

[III. Intended Classes 4](#_Toc82691179)

[IV. Other Initial Items 5](#_Toc82691180)

[i. Auto-Generated Database 5](#_Toc82691181)

[ii. Text Files 5](#_Toc82691182)

[II. Modifications 5](#_Toc82691183)

[I. Design Changes 5](#_Toc82691184)

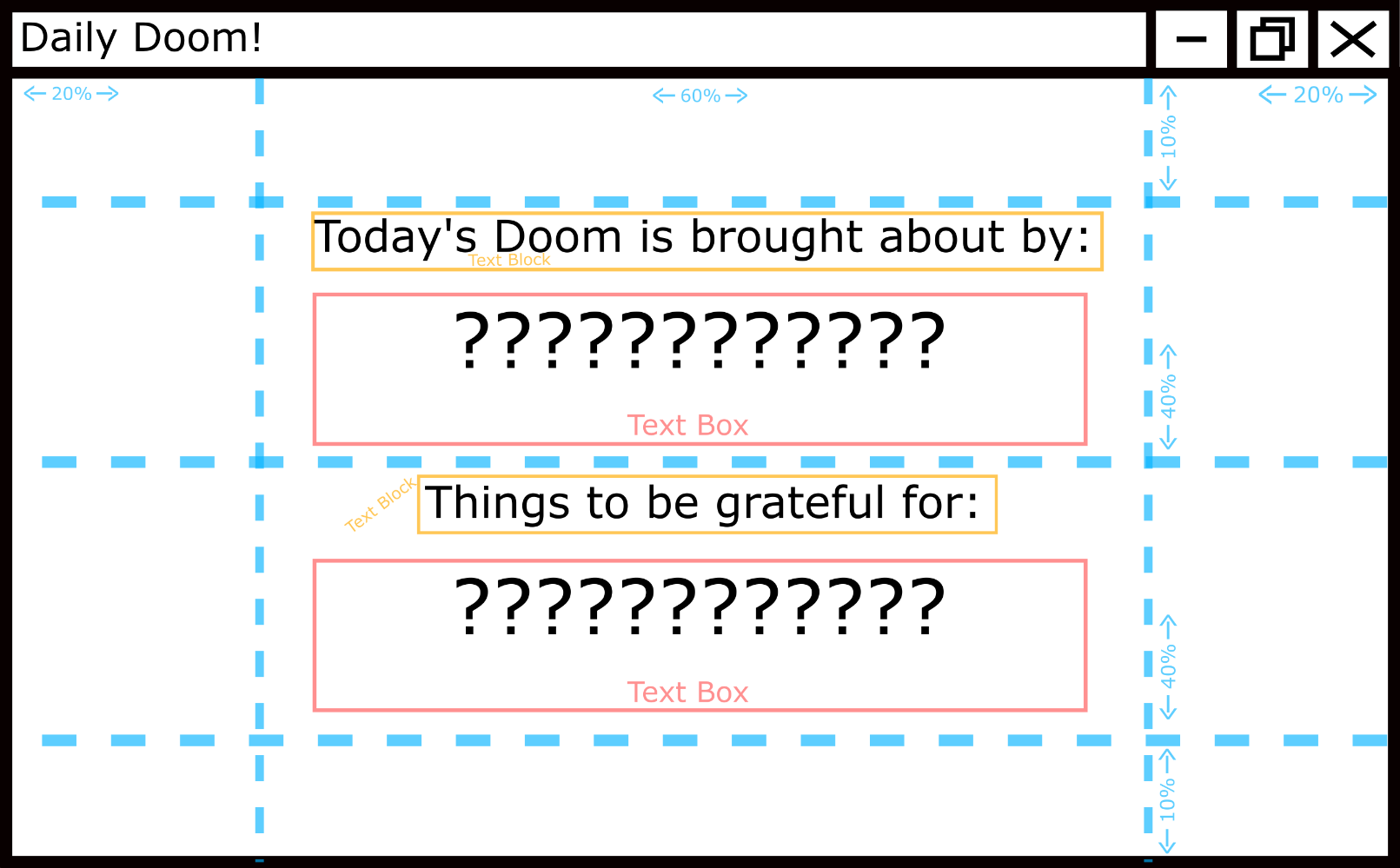
[II. Issues and Fixes 5](#_Toc82691185)

[III. Finishing Touches 5](#_Toc82691186)

# Initial Idea

* Create a fun little program that attempts to predict a user’s doom, or what would cause them hassle that day.
* To balance things out, it should also contain a lucky item and / or grateful phrase the user can keep with them for the predicted doom.
* The design should be something simple to underline the unserious nature of the program.
* As it is a ‘daily’ doom prediction application, there should be some controls in place to prevent the program from running twice in the same day (most likely done with the help of a linked database).

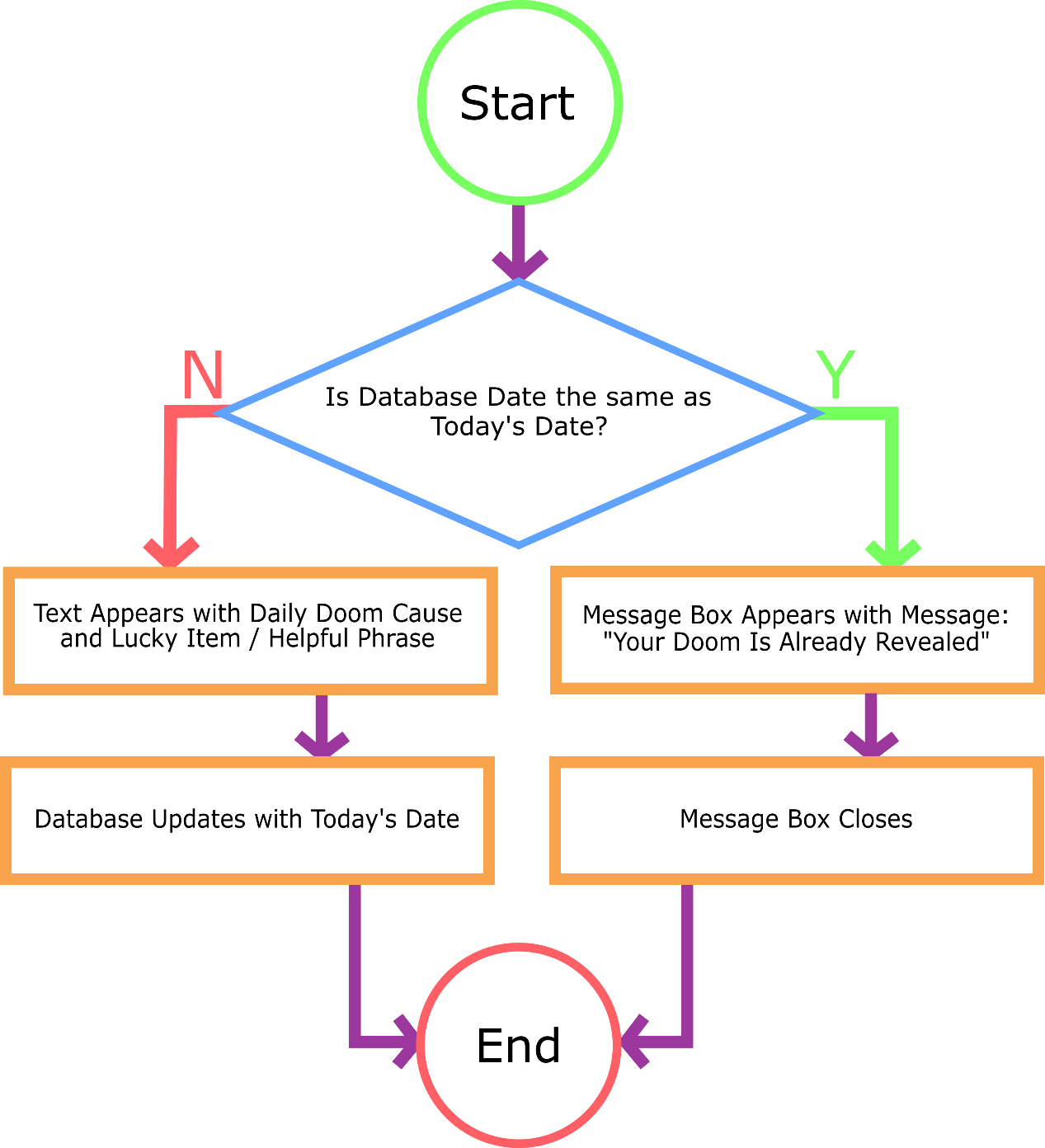
## Diagram



***Figure 1: DailyDoomDiagram.png***

* The main window should be split into a grid of 3 columns (20%, 60% and 20% respectfully) and 4 rows (10%, 40%, 40% and 10% respectfully).
* Both the end columns (left and right) and end rows (top and bottom) are to remain empty for neatness and future additions, should there be any further development.
* Both Column 2, Row 2 and Colum 2, Row 3 will contain a centred text block that is top aligned with a margin of 10, 3 or 5, 10, 0 pixels, as well as a centred text box that is bottom aligned with a margin of 10, 7 or 5, 10, 0 pixels.
* In Column 2, Row 3, the text block may change depending on whether the program has decided to pick a lucky item or grateful phrase to display (this will be done with a Math.Random() object).

## Intended Program Flow



***Figure 2: DailyDoomProgramFlow.png***

## Intended Classes

|  |  |
| --- | --- |
| *Prediction Class* | |
| *Properties* | |
| DoomNum | Int |
| ItemNum | Int |
| GratitudeNum | Int |
| DoomPhrase | String |
| ItemName | String |
| GratitudePhrase | String |
| *Methods* | |
| GetDoom | * Selects a random number and sets it as the DoomNum * Uses number to FileStream() / StreamReader() a line from the DoomPhrases.txt and sets it as the DoomPhrase |
| GetItem | * Selects a random number and sets it as the ItemNum * Uses number to FileStream() / StreamReader() a line from the Items.txt and sets it as the ItemName |
| GetGratitude | * Selects a random number and sets it as the GratitudeNum * Uses number to FileStream() / StreamReader() a line from the GratitudePhrases.txt and sets it as the GratitudePhrase |

**Note:** Although this can be done in the MainWindow object without the need for a class, the Prediction class will be created and implemented to make any further development easier.

## Other Initial Items

### Auto-Generated Database

Using VisualStudio, a database will be generated to keep track of the last date the application ran in order to ensure that there aren’t two predictions read in one day.

### Text Files

A text file for doom phrases, items and gratitude phrases (called DoomPhrases.txt, Items.txt and GratitudePhrases.txt respectfully) will be created and placed in the RandomBits\DailyDoom\bin\Debug\TextFiles directory. As there is a .gitignore file for VisualStudio in the GitHub repository, a copy of these files will also be saved to the RandomBits\DailyDoom\TextFiles directory for full disclosure.

# Modifications

## Design Changes

## Issues and Fixes

# Finishing Touches