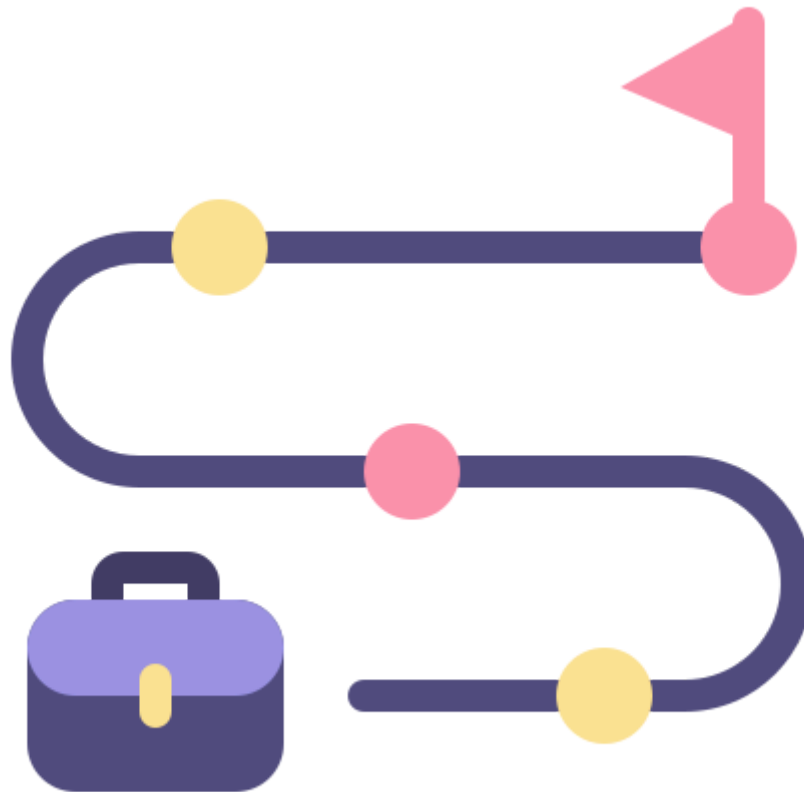




Student name:	Student 1: Oleksii Babii Student 2: Igor Tryndey Student 3: Ruslan Tleubergen				
Student number:	Student 1: 3104904 Student 2: 3105023 Student 3: 3091317				
Faculty:	Computing Science				
Course:	BSCH/BSCO/EXCH		Stage/year:	2	
Subject:	Software Development 2				
Study Mode:	Full time	<input checked="" type="checkbox"/>		Part-time	
Lecturer Name:	Gemma Deery				
Assignment Title:	Review 2				
Date due:	17/04/2024				
Date submitted:					
<p>Plagiarism disclaimer:</p> <p><i>I understand that plagiarism is a serious offence and have read and understood the college policy on plagiarism. I also understand that I may receive a mark of zero if I have not identified and properly attributed sources which have been used, referred to, or have in any way influenced the preparation of this assignment, or if I have knowingly allowed others to plagiarise my work in this way.</i></p> <p><i>I hereby certify that this assignment is my own work, based on my personal study and/or research, and that I have acknowledged all material and sources used in its preparation. I also certify that the assignment has not previously been submitted for assessment and that I have not copied in part or whole or otherwise plagiarised the work of anyone else, including other students.</i></p> <p>Signed: _____ Date: _____</p>					
<div style="border: 1px solid black; padding: 5px;"> <p>Please note: Students MUST retain a hard / soft copy of ALL assignments as well as a receipt issued and signed by a member of Faculty as proof of submission.</p> </div>					

Milestone Report



Student 1 name:Oleksii Babii	Student 1 number:3104904
Student 2 name:Igor Tryndey	Student 2 number:3105023
Student 3 name:Ruslan Tleubergen	Student 3 number:3091317

VERSIONING APPROACH	4
DEVELOPMENT APPROACH	4
MILESTONE 1	4
Goals	4
Junit Tests	4
Commit Logs	4
Full Log Details	4
MILESTONE 2	4
Goals	4
Junit Tests integration	4
Commit Lists and Branches Trees	4
Full Log Details	4
COLLABORATION	4
APPENDIX	4

Versioning Approach

We continued using git for version control. As we were going to change a significant amount of the code, we made a new branch called “feature”, where we’ll test and work on this iteration of the project until we’re satisfied with it, after which we’ll push it to the main branch.

Later on in the development we made a “development” branch as the team faced some issues with implementing some dependencies. This allowed us to start from scratch and waste less time on trying to fix the broken code.

Development Approach

New tests were created for the newly implemented weather and timezone APIs.

```
//testing the cityToCoordinate function
void testCityToCoordinate(){
    double[] dublinCoordinates = {53.350140, -6.266155};
    double[] coords = skyTalk.cityToCoordinate(city:"Dublin");

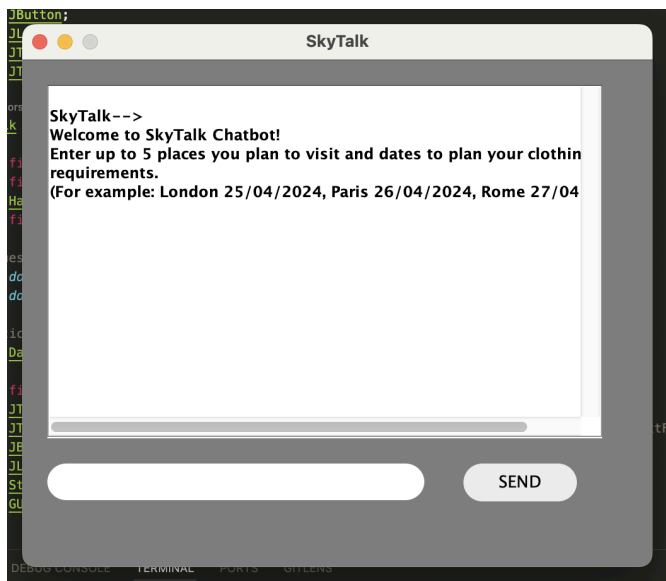
    //check that the coordinates the function returned are close enough to the actual coordinates
    //there will be slight differences depending on which exact point on the map the API picks as Dublin's "location", so we use a delta of 0.1
    assertEquals(dublinCoordinates[0],coords[0],0.1);
    assertEquals(dublinCoordinates[1],coords[1],0.1);
}
```

```
@Test
//testing the timezoneToCoordinate function
void testTimezoneFromCoordinate(){
    double[] dublinCoordinates = {53.350140, -6.266155};
    String dublinTimezone = "Europe/Dublin";
    String timezone = skyTalk.timezoneFromCoordinate(dublinCoordinates);

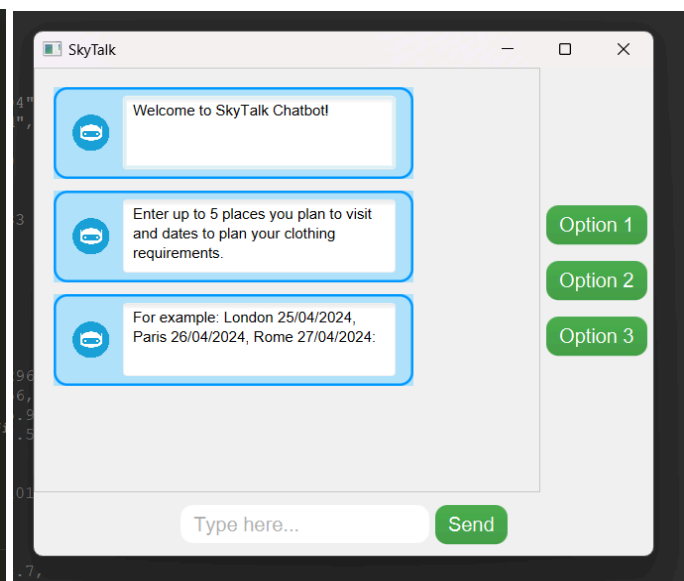
    //compare expected and result
    System.out.println("Expected timezone: " + dublinTimezone);
    System.out.println("Timezone received: " + timezone);
    assertEquals(dublinTimezone,timezone);
}
@Test
```

The Skytalk class was revamped, with the old methods used for getting forecast data and clothing recommendations being replaced.

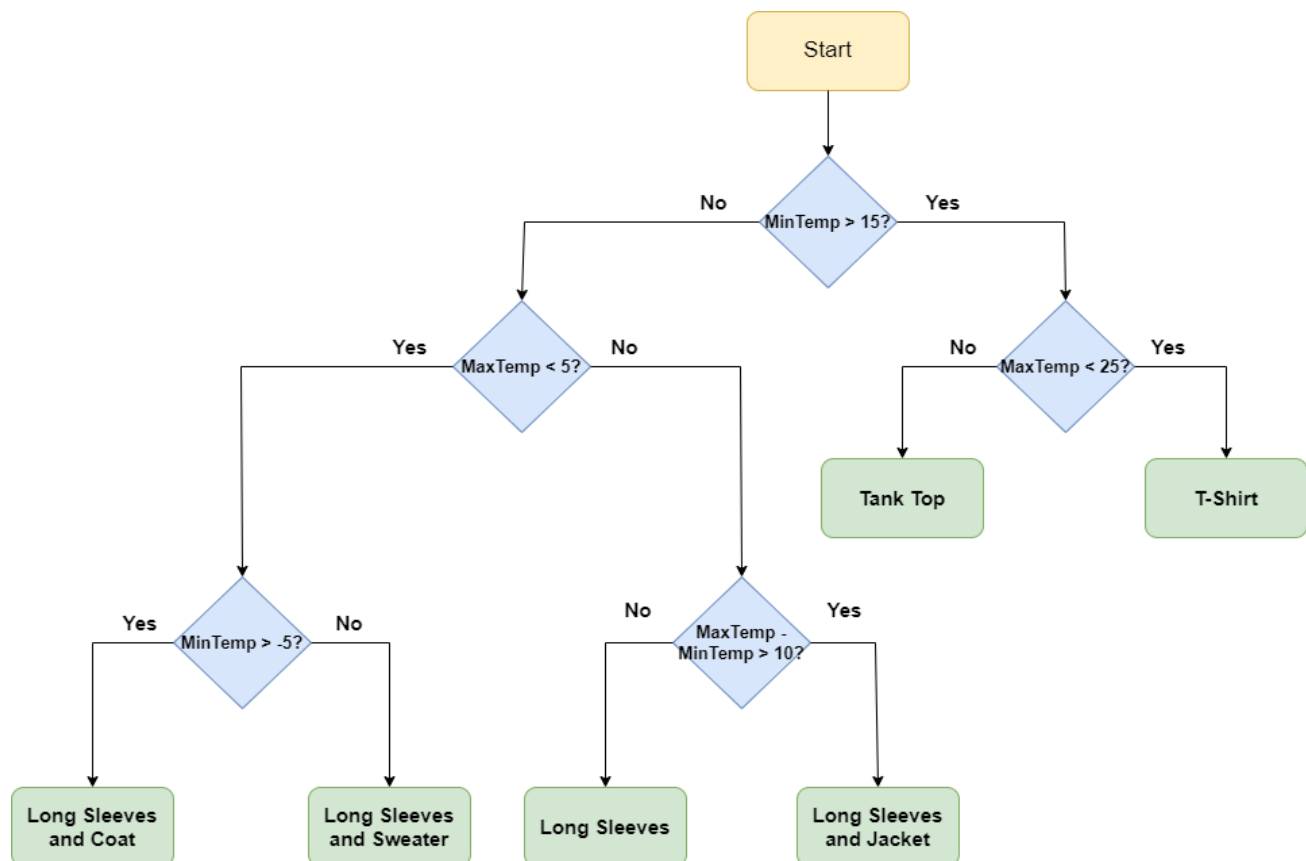
The GUI was updated to make for a smoother user experience:



The GUI as of the first review



The new GUI

















Milestone 2

Goals

The goals for this milestone were to:

1. Update the GUI to make it more intuitive for users by updating the visuals and providing an options menu, and
2. Simplify the process of getting a forecast for a given city and date. We rewrote the weather-related methods in Skytalk.java and added more comments so that there would be less confusion as to what each bit of the code does.

Branch Structure:

Default				
Branch	Updated	Check status	Behind	Ahead
main 	 3 days ago			Default
Your branches				
Branch	Updated	Check status	Behind	Ahead
development 	 5 minutes ago		0	10
feature 	 12 hours ago		0	14
Active branches				
Branch	Updated	Check status	Behind	Ahead
development 	 5 minutes ago		0	10
feature 	 12 hours ago		0	14
api 	 2 weeks ago		30	0
gui 	 2 weeks ago		30	0

main - the branch for finished and ready to deploy versions of the chatbot.

development - the branch where all current development is being pushed to

feature - the old development branch that we had to abandon due to dependency issues

Log Details

Log details are shown in each member's contribution sections.

Collaboration

Please list the contributions made by each student for this iteration.

- Each student is expected to make equal contributions
- Each student will be marked according to his/her contribution
- You must have relevant comments on all code and messages on all commits

Student 1: Oleksii

In this review, my attention was focused on developing the design and functionality of the GUI using a modern approach that includes the JavaFX software platform. Interface design, colours and placement of fields and buttons. My work also includes the implementation of animations and the development of user-friendly designs. Setting up the project and adding dependencies, and plugins. Customizing buttons and popups for input and output. Creating a modern design using CSS.

```
Home@DESKTOP-QL2S8TT MINGW64 ~/Desktop/sky/SkyTalk (feature)
$ git log --author="oleksii.babii"
commit a2c75d3c413d5e36c8ff6180d1dcc1b0244034a1
Author: oleksii.babii <oleksii.babii@student.griffith.ie>
Date:   Wed Apr 17 19:37:29 2024 +0100

    feat: add external css resources and photos

commit 9b7032e6a50304d9e32b493683f98d37bd0406d8
Author: oleksii.babii <oleksii.babii@student.griffith.ie>
Date:   Wed Apr 17 04:03:00 2024 +0100

    fix: fix the bug in start() method

commit c4dd4e89007b23884989c11ae1c902245e88763f
Author: oleksii.babii <oleksii.babii@student.griffith.ie>
Date:   Tue Apr 16 22:59:18 2024 +0100

    feat: add 'send' button in start() method

commit 0495bce7e48351ab4f57860fbff2f48014717397
Author: oleksii.babii <oleksii.babii@student.griffith.ie>
Date:   Tue Apr 16 22:54:15 2024 +0100

    feat: implement start() method to display frame

commit 5976379e34cd520563621d77b5b47f162f9ec73b
Author: oleksii.babii <oleksii.babii@student.griffith.ie>
Date:   Tue Apr 16 22:47:25 2024 +0100

    feat: add a ChatBot class to represent the logic

commit 3fda4312de5bddcf64a4efccd1909884b8820b2e
Author: oleksii.babii <oleksii.babii@student.griffith.ie>
Date:   Tue Apr 16 20:24:55 2024 +0100

    feat: set up default javaFX Maven project
```

Student 2: Igor

In this project, I take part in the development of the GUI using the JavaFX framework. Especially I implemented the addMessage and sendMessage methods. Also I take part in the GUI design and make it more user friendly. The GUI creation is important to implement, because it allows people, who are not friendly with programming and command prompt to work with your application code. Also I made an animation for messages to appear in the chat.

Last but not least, I take a part in concatenation of collaborators work to make it look like as a whole project.

```
commit 3dd53f0e6bbc053f0078112b7d93bf8eefbfe522
Author: Ihor Tryndey <igortryndey@gmail.com>
Date: Thu Apr 18 02:54:08 2024 +0100

    troubles shooting

commit 3dd53f0e6bbc053f0078112b7d93bf8eefbfe522
Author: Ihor Tryndey <igortryndey@gmail.com>
Date: Thu Apr 18 00:30:07 2024 +0100

    worked version

commit db295687fe2eb7fd3c4df2262d03b1c27b056776
Author: Ihor Tryndey <igortryndey@gmail.com>
Date: Thu Apr 18 00:15:25 2024 +0100

    format changes

commit d6fd40743abf345cc9797751320e6066ecd8bfd0
Author: Ihor Tryndey <igortryndey@gmail.com>
Date: Thu Apr 18 00:11:44 2024 +0100

    implementation of API

commit e0f51e82c16c4fcc7b6bea7e24678a85fc66500a
Author: Ihor Tryndey <igortryndey@gmail.com>
Date: Thu Apr 18 00:10:23 2024 +0100

    welcome message and input method
```



```
Home@DESKTOP-QL2S8TT MINGW64 ~/Desktop/sky/SkyTalk (feature)
$ git log --author="Ihor Tryndey"
commit 08d578dfe92164e67cf034ddc677e70191b06045
Author: Ihor Tryndey <igortryndey@gmail.com>
Date: Wed Apr 17 18:18:25 2024 +0100

    comments for addMessage method

commit 0871fb7b91be9a37e6c2d0aca4568ad05409c8e9
Author: Ihor Tryndey <igortryndey@gmail.com>
Date: Wed Apr 17 18:14:15 2024 +0100

    animation optimization

commit 23acac9f42ecc6b5c4c75fcd25d71936693b65a2
Author: Ihor Tryndey <igortryndey@gmail.com>
Date: Wed Apr 17 18:07:49 2024 +0100

    addMessage method implementation

commit b21dc2fd03e1c5710e0e4240045b680ebc826ec8
Author: Ihor Tryndey <igortryndey@gmail.com>
Date: Wed Apr 17 14:20:29 2024 +0100

    fixing PauseTransition inside sendMessage

commit 543bfb2ad85e2aed76308764cdcfa9634f42c2fa
Author: Ihor Tryndey <igortryndey@gmail.com>
Date: Wed Apr 17 06:16:11 2024 +0100

    sendMessage method implementation
```

Student 3: Dessie (Ruslan)

I wrote most of the documentation for this milestone report and worked on implementing OpenWeatherAPI into this project, in order to streamline the forecasting process, as the previous iteration of this chatbot relied on methods that were difficult to decipher. Using OpenWeatherAPI, it's very easy to get forecast data just from a pair of coordinates and a date. This forecast data can then be filtered to show only the most important information like the weather conditions and the temperature. To aid the forecast API, I made use of another API called GeoTimeZone that returns the timezone for a given set of coordinates, so that the date that's passed to the forecast method is offset accordingly. There were a lot of different coordinate-to-timezone APIs to choose from. I went with GeoTimeZone as it's very simple in its functionality and the only call it has is exactly what I needed.

Once the forecast data is ready, the temperature and weather conditions will be passed to a clothing recommendation method that'll return the appropriate string response.

Based on recommendations given on the previous review, I made sure to be more consistent in adding comments on what each part of the methods I wrote do.

update methods with OpenWeatherAPI, complete implementation of feature branch methods into development branch
desmondbergen committed 2 minutes ago

5fa821c

```
MacBook-Air:Skytalk dessie$ git log
commit c4471b5eef5d4509a8bb710a838bc06c8d967122 (HEAD -> development, origin/development)
Author: dessie <lynxes.desmond@gmail.com>
Date: Thu Apr 18 10:05:44 2024 +0100

    fix issues with JSON dependency, rewrite methods
```

```
MacBook-Air:Skytalk dessie$ git log
commit a4de260c0f426a7e9f8c6bc5be4bfd93db7120cf (HEAD -> feature, origin/feature)
Author: dessie <143977574+desmondbergen@users.noreply.github.com>
Date: Wed Apr 17 22:02:55 2024 +0100

    Delete obsolete src directory

commit 47d62730ea7cf009aa081398b2d68b6e011f764c
Author: dessie <lynxes.desmond@gmail.com>
Date: Wed Apr 17 21:53:28 2024 +0100

    transfer new methods to new version of bot

commit 5140aba233348b9031386ada0a6c72341b2c7823
Author: dessie <lynxes.desmond@gmail.com>
Date: Wed Apr 17 21:48:12 2024 +0100

    implement new weather forecasting methods and tests for those methods
```

Links to APIs used

OpenWeatherAPI: <https://openweathermap.org/api>

GeoTimeZone: <https://www.geotimezone.com/>

Appendix

link to repo: <https://github.com/ihortry/SkyTalk/tree/feature>