

**Software Modeling**  
**Modeling and Implementation Phase**

---

**“Radio Simulation”**

**- Diagrams -**

---

By ESSAFSYFY Abdelkrim and MELUZOLA KIMFUTA Gingu

Academic year 2019-2020

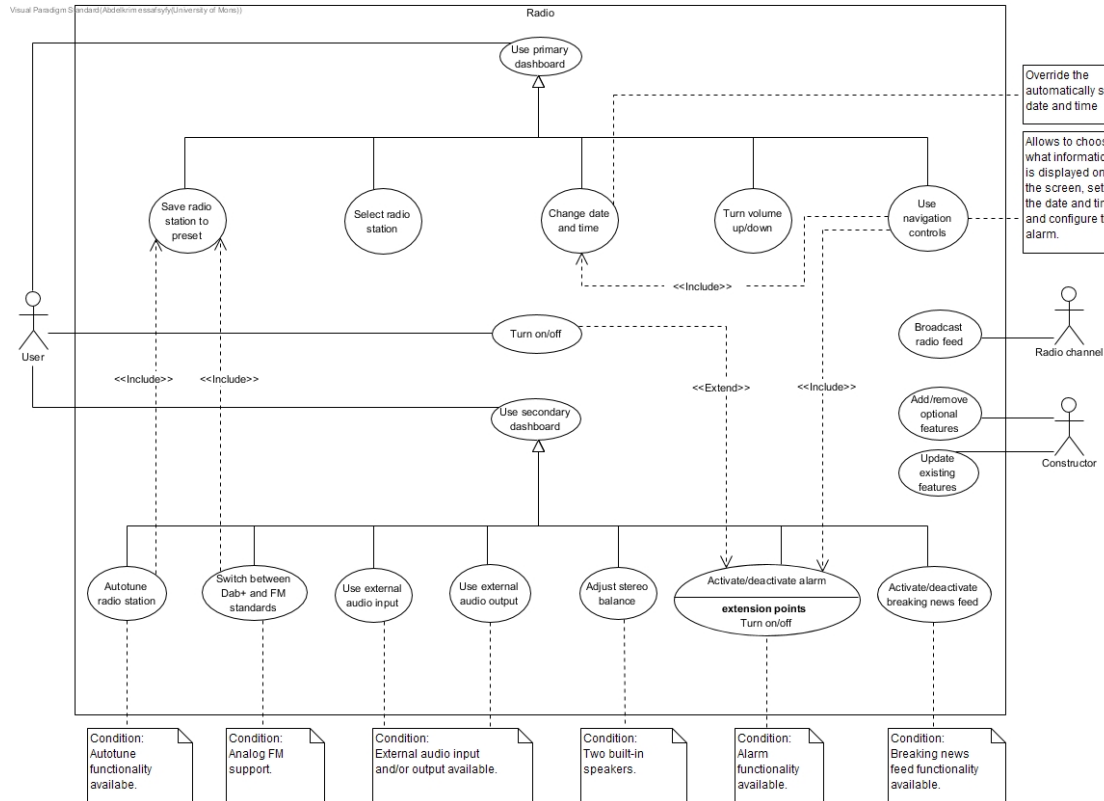
# Contents

<b>1</b>	<b>Use case diagram</b>	<b>1</b>
1.1	Diagram . . . . .	1
1.2	Diagram description . . . . .	1
1.3	Individual use cases' description . . . . .	2
1.3.1	Turn on/off . . . . .	2
1.3.2	Use primary dashboard . . . . .	2
1.3.3	Select radio station . . . . .	3
1.3.4	Turn volume up/down . . . . .	3
1.3.5	Use navigation controls . . . . .	4
1.3.6	Change date and time . . . . .	4
1.3.7	Save radio station to preset . . . . .	5
1.3.8	Use secondary dashboard . . . . .	5
1.3.9	Autotune radio station . . . . .	6
1.3.10	Adjust stereo balance . . . . .	6
1.3.11	Use external audio input . . . . .	7
1.3.12	Use external audio output . . . . .	7
1.3.13	Activate/deactivate alarm . . . . .	8
1.3.14	Activate/deactivate breaking news feed . . . . .	9
1.3.15	Switch between DAB+ and FM . . . . .	9
1.3.16	Broadcast radio feed . . . . .	10
1.3.17	Add/remove optional features . . . . .	10
1.3.18	Update existing features . . . . .	10
<b>2</b>	<b>Class diagram</b>	<b>12</b>
<b>3</b>	<b>Object diagram</b>	<b>13</b>
<b>4</b>	<b>State diagrams</b>	<b>14</b>
4.1	Diagram 1 . . . . .	14
4.2	Diagram 2 . . . . .	14
4.3	Diagram 3 . . . . .	14

<b>5</b>	<b>Sequence diagrams</b>	<b>15</b>
5.1	Diagram 1 . . . . .	15
5.2	Diagram 2 . . . . .	15
<b>6</b>	<b>Activity diagram</b>	<b>16</b>
<b>7</b>	<b>Interaction Overview diagram</b>	<b>17</b>

# 1 Use case diagram

## 1.1 Diagram



## 1.2 Diagram description

The use case diagram for this project (radio simulation) has been made with the idea of two separate functionality categories:

Primary: the functionalities that are present in each and every radio.

Secondary: the optional features that the manufacturer (in this case the radio configurator<sup>1</sup>) chooses to add to the radio. In the diagram, we notice these two categories in form of two abstract use cases named "Use primary dashboard" and "Use secondary dashboard". There are also four separate use cases; "Turn on/off" which will allow the user to manually turn the radio on and off, "Broadcast radio feed" by the radio stations, "Add/remove optional feature" and "Update existing features" by the radio constructor.

It is noteworthy that the use case "Use navigation controls" allows the user not only to choose what

<sup>1</sup>See Models' report

the radio's screen displays but also configure two use cases which are: "Change date and time" and "Activate/deactivate alarm", for the latter it is for the purpose of setting the time of the alarm, the activation/deactivation is done by another button.

### 1.3 Individual use cases' description

#### 1.3.1 Turn on/off

**Use case name:** Turn on/off

**Summary:** Allows the user to turn the radio on or off.

**Actors:** User.

**Assumptions:** Radio has access to a power source.

**Preconditions:** Radio is plugged in.

**Basic course of action:**

1. User presses power button on.
2. LED indicator switches from red to green.
3. Radio is powered on.

**Postconditions:** The power button is functional.

**Alternate courses :**

- 1a. Alarm (if set) turns the radio on, on the programmed time.
  - 3a. Radio is powered off.
- 

#### 1.3.2 Use primary dashboard

**Use case name:** Use primary dashboard

**Summary:** An abstract use case that allows the user to use the primary radio functionalities. Every specialization of this use case inherits its actors, assumptions, preconditions and postconditions.

**Actors:** User.

**Assumptions:** Radio is plugged in.

**Preconditions:** Radio is powered on.

**Basic course of action:**

1. User chooses to use one of the available functionalities.
-

### 1.3.3 Select radio station

**Use case name:** Select radio station

**Summary:** Allows the user to select a radio station using a turn knob

**Actors:** User

**Assumptions:** Same as parent

**Preconditions:** Radio stations are available.

**Basic course of action:**

1. User turns knob until signal is good enough to produce sound
2. Radio plays radio station according to frequency
3. Radio automatically synchronization of date and time using the received signal

**Alternate courses:**

- 1a. User clicks one of the preset buttons
    - 1a1. No station is saved into the pressed preset
    - 1a2. Signal received is not good enough
    - 1a3. Preset button is damaged, the radio doesn't play
  - 2a. No radio station with good signal found
  - 3a. The date and time are unsynchronized, the radio proceeds to automatically set the date and time.
- 

### 1.3.4 Turn volume up/down

**Use case name:** Turn volume up/down

**Summary:** Allows the user to adjust the radio's volume from 0 to 20 in discrete steps.

**Actors:** User.

**Assumptions** Same as parent.

**Preconditions:** Built-in speaker.

**Basic course of action:**

1. User turns the knob left or right to adjust the volume.
2. Radio changes the volume accordingly.

**Alternate courses:**

- 2a. The knob is damaged, the radio is unable to change the volume.

### 1.3.5 Use navigation controls

**Use case name:** Use navigation controls

**Summary:** Allows the user to change the display's content depending on its size.

**Actors:** User

**Assumptions:** Same as parent

**Preconditions:** Radio has a functioning screen

**Basic course of action:**

1. User uses navigation controls (Up, down, left, right) to choose was type of information is displayed on the screen. This includes displaying of date and time and alarm activation.
2. Radio changes the display's content accordingly

**Alternate courses:**

- 1a. The navigation controls are damaged, the user is unable to choose the type of information the screen displays.
- 

### 1.3.6 Change date and time

**Use case name:** Change date and time

**Summary:** Allows the user to change the date and time of the radio

**Actors:** User

**Assumptions:** Same as parent

**Preconditions:** Radio is turned on

**Basic course of action:**

1. User changes date and time using navigation controls (see use case: navigation controls)

**Alternate courses:**

- 1a. Navigation controls are damaged, the user is unable to change the date and time.
-

### 1.3.7 Save radio station to preset

**Use case name:** Save radio station to preset

**Summary:** Allows the user to save the currently playing radio station to one of the free available presets (ranging from 3 to 10)

**Actors:** User

**Assumptions:** Same as parent

**Preconditions:** Radio station is playing

**Basic course of action:**

1. User holds the preset button during at least 3 seconds
2. Radio saves the currently playing station to the selected preset

**Alternate courses:**

- 1a. The selected preset already has a saved station
    - 1a1. The radio overrides the old station with the new one
  - 1b. The selected preset is damaged, the station cannot be saved
  - 1c. No free presets are available, the user chooses not to override any
- 

### 1.3.8 Use secondary dashboard

**Use case name:** Use secondary dashboard

**Summary:** An abstract use case that allows the user to use the secondary (optional) radio functionalities. Every specialization of this use case inherits its actors, assumptions, preconditions and postconditions.

**Actors:** User.

**Assumptions:** Radio is plugged in.

**Preconditions:** Radio is powered on.

**Basic course of action:**

1. User chooses to use one of the available functionalities.
-



### 1.3.9 Autotune radio station

**Use case name:** Autotune radio station

**Summary:** Allows the user to use autotune functionality which will automatically search for all available radio stations and will store them in its internal memory in the order they were found until all memory locations are filled or until no more radio stations are found

**Actors:** User

**Assumptions:** Same as parent

**Preconditions:** Radio has autotune functionality

**Basic course of action:**

1. User activates autotune functionality using a button
2. Radio scans for all the stations available with good quality
3. Radio stores the fetched stations into its internal memory
4. User browses the stations found

**Alternate courses:**

- 1a. Autotune button is damaged, the functionality is unavailable
- 2a. No stations with good enough quality are found
- 2b. Radio's search functionality is damaged, it is unable to search for any station
- 3a. Radio's internal memory is damaged, no stations are stored
- 4a. User saves the playing station into a preset (see "Save radio station to preset" use case)
- 4b. User saves all found stations into all available presets.
  - 4b1. Radio saves stations into presets until no more stations are found or no free presets remain.

---

### 1.3.10 Adjust stereo balance

**Use case name:** Use external audio output

**Summary:** Allows the user to adjust stereo balance by turning a knob which controls the output volume between the left and right speaker

**Actors:** User

**Assumptions:** Same as parent

**Preconditions:** Radio has two built-in speakers

**Basic course of action:**

1. User turns the knob left or right
2. Radio adjusts the stereo balance accordingly

**Alternate courses:**

- 2a. Stereo balance knob is damaged, no audio change will result
- 

**1.3.11 Use external audio input**

**Use case name:** Use external audio input

**Summary:** Allows the user to plug in an external audio input allowing the radio to receive sound from an external device.

**Actors:** User

**Assumptions:** Same as parent

**Preconditions:** External audio input functionality available

**Basic course of action:**

1. User establishes connection (USB, audio jack or bluetooth) with the radio
2. Radio detects the external audio input
3. Radio starts playing audio from external device

**Alternate courses:**

- 2a. External audio input connector is damaged, the radio is unable to detect the device.
  - 3a. External device has no audio stored within, the radio doesn't play any sound.
- 

**1.3.12 Use external audio output**

**Use case name:** Use external audio output

**Summary:** Allows the user to plug in an external audio output allowing the radio to redirect the sound to external speakers

**Actors:** User

**Assumptions:** Same as parent

**Preconditions:** External audio output functionality available

**Basic course of action:**

1. User establishes connection (audio jack or bluetooth) with the radio
2. Radio detects the external audio output
3. Audio is played in the external device from the radio

**Alternate courses:**

- 2a. External audio output connector is damaged, the radio is unable to detect the device.
- 3a. External device is damaged, unable to play audio from the radio

---

**1.3.13 Activate/deactivate alarm**

**Use case name:** Activate/deactivate alarm

**Summary:** Allows the user to set, activate and deactivate the alarm functionality

**Actors:** User

**Assumptions:** Same as parent

**Preconditions:** Radio has alarm functionality

**Basic course of action:**

1. User sets alarm time
2. User activate the alarm
3. Alarm rings at the programmed time, when the radio is on
4. User deactivates the alarm

**Alternate courses:**

- 1a. Alarm functionality is damaged, the user cannot set, activate or deactivate it
- 2a. Same as '1a'
- 3a. Radio turns on at the programmed time, when the radio is idle
  - 3a1. Last selected radio station starts playing
  - 3a2. Normal alarm sound is played

---

### 1.3.14 Activate/deactivate breaking news feed

**Use case name:** Activate/deactivate breaking news feed

**Summary:** Allows the user to enable or disable current station's breaking news display

**Actors:** User

**Assumptions:** Same as parent

**Preconditions:** Breaking news functionality available

**Basic course of action:**

1. User enables/disables breaking news functionality
2. Radio stations is playing and sudden breaking news are received
3. Radio displays breaking news if the user has set it to do so

**Alternate courses:**

- 2a. No breaking news are received
  - 3a. Breaking news functionality encounters an issues, thus the information is not received
  - 3b. Breaking news functionality encounters an issues, thus the information is not displayed
- 

### 1.3.15 Switch between DAB+ and FM

**Use case name:** Use external audio output

**Summary:** Allows the user to switch between DAB+ and FM radio standards

**Actors:** User

**Assumptions:** Same as parent

**Preconditions:** Radio has analog FM support

**Basic course of action:**

1. User switches between DAB+ and FM standards using a switch
2. Radio switches to the selected standard

**Alternate courses:**

- 1a. The switch is damaged, the user is stuck with the default (or last used) standard
-

### 1.3.16 Broadcast radio feed

**Use case name:** Broadcast radio feed

**Summary:** Allows the radio channel to broadcast its radio feed

**Actors:** Radio channel

**Preconditions:** Radio channel is active

**Basic course of action:**

1. Radio channel broadcasts its feed via a certain frequency
2. All radios tuning into said frequency receive the radio broadcast

**Alternate courses:**

- 1a. The chosen frequency is unavailable (already reserved by another radio station, ...)
  - 2a. The signal is weak, few to no radios receive the signal
- 

### 1.3.17 Add/remove optional features

**Use case name:** Add/remove optional features

**Summary:** Radio constructor adds or removes radio optional features

**Actors:** Constructor

**Preconditions:** Current radio version has not been abandoned

**Basic course of action:**

1. Constructor adds/removes one or more optional features
2. Constructor test the changes and verifies radio integrity
3. New radio version is commercialized

**Alternate courses:**

- 2a. No testing is done, resulting in issues within the radio
  - 3a. Radio has flaws and is then submitted to testing (back to 2.)
- 

### 1.3.18 Update existing features

**Use case name:** Update existing features

**Summary:** Radio constructor updates radio firmware, updating functionalities in the process.

**Actors:** Constructor

**Preconditions:** Current radio version has not been abandoned

**Basic course of action:**

1. Constructor updates firmware
2. One or more features are updated
3. Constructor test the change and verifies radio integrity
4. New radio update is uploaded
5. Radios automatically update to the latest firmware

**Alternate courses:**

- 2a. No features have been updated (beta testing, improved radio's quality of life, ...)
- 3a. No testing is done, resulting in issue within the radio
- 3b. Radio has flaws and is then submitted to testing (back to 3.)

## 2 Class diagram

DESCRIPTION HERE.

### 3 Object diagram

DESCRIPTION HERE.



## 4 State diagrams

INTRODUCTION HERE

### 4.1 Diagram 1

DESCRIPTION HERE.

### 4.2 Diagram 2

DESCRIPTION HERE.

### 4.3 Diagram 3

DESCRIPTION HERE.

## 5 Sequence diagrams

INTRODUCTION HERE.

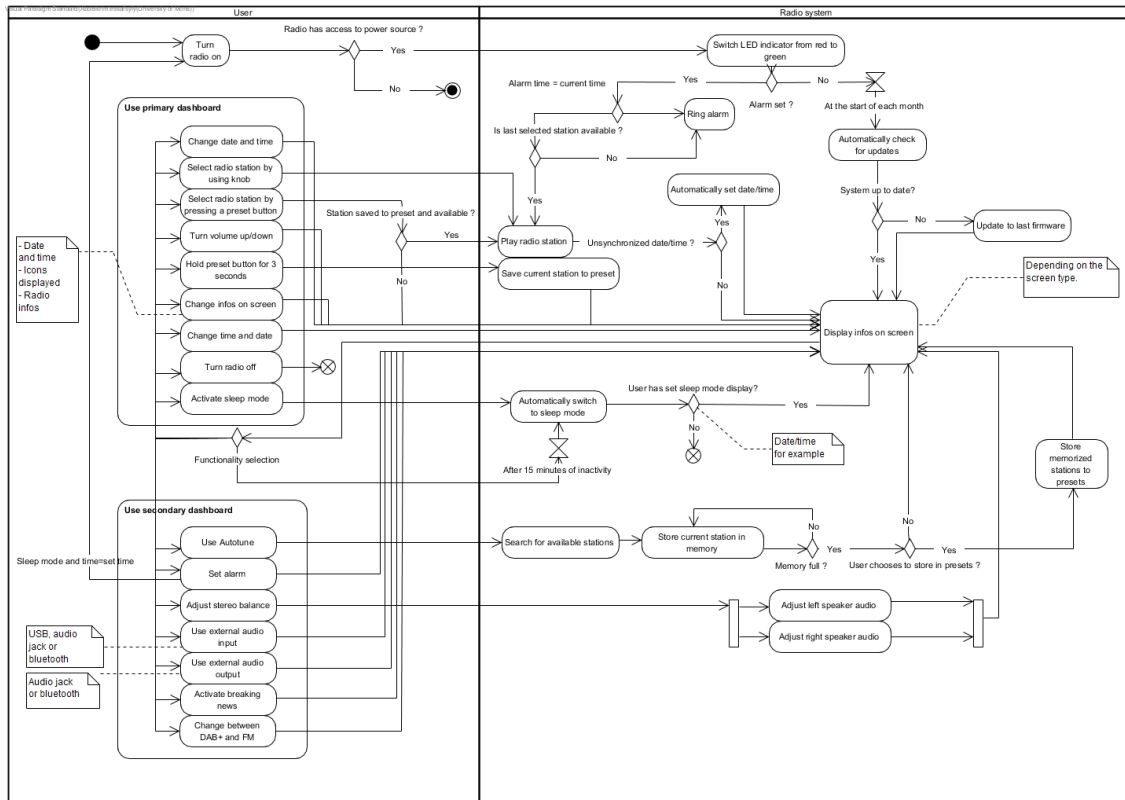
### 5.1 Diagram 1

DESCRIPTION HERE.

### 5.2 Diagram 2

DESCRIPTION HERE.

## 6 Activity diagram

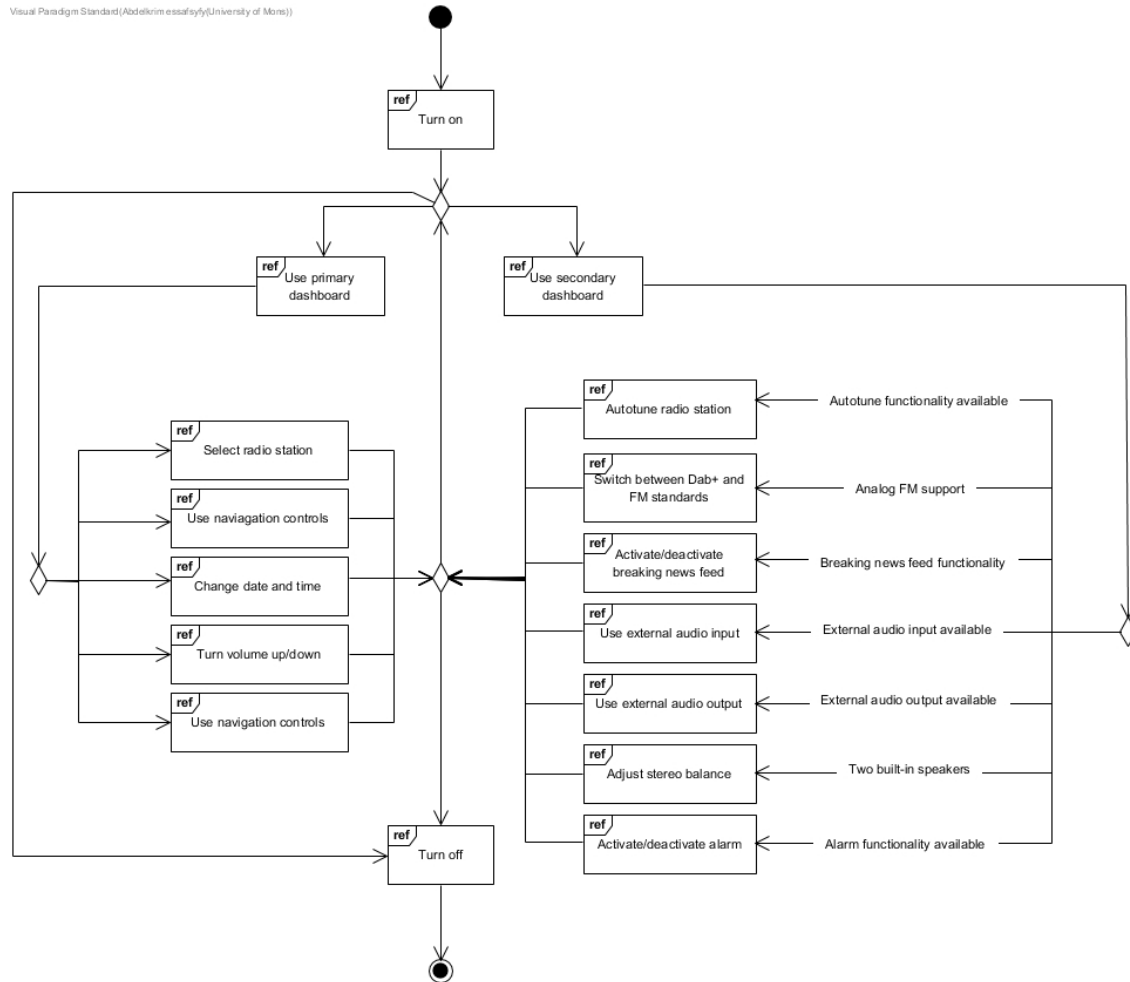


While remaining consistent with the concept of two separate dashboards, we created the activity diagram following another -realistic- concept; the user can freely choose between functionalities, which means that the user can, for example, turn the volume up or down before choosing a station. In other words, most of the time, there are no mandatory sequences to follow.

In order to achieve this, we use a "hub" activity that will allow the user to switch from one action to the other. This "hub" activity is "Display infos on screen", which makes sense since the user has to constantly receive feedback following their interaction with the radio. Doing so, we are able to start from any of the two dashboards and move on to whatever other action the user desires to do. In order to make the diagram as legible as possible, some arrows (Control flows) are overlapping in a manner that avoids confusion as much as possible. Take secondary dashboard for example, it is clear that every functionality, except "Adjust stereo balance", once over, is redirected to the "hub" activity.

Finally, if the user has been inactive for 15 minutes, the radio automatically switches to sleep mode.

## 7 Interaction Overview diagram



In this interaction overview diagram, we specify the steps taken by the user when manipulating the radio. Please note that there aren't as many sequence diagrams as referenced in each interaction. If done so, most of the sequence diagrams would have been uninterestingly similar and repetitive. The steps taken by the user are as follows:

1. The user turn the radio on
2. They choose to use either the primary or secondary dashboard (or turn the radio off)
3. Which leads them to choose on of the available functionalities in each category
4. Once the action is performed, the user can once again choose between the two categories
5. The user can also turn the radio off

Please note that every diagram present in this report will be available for viewing in the "Diagrams" folder.

If need be, a link to the *Git* repository used for this project:

<https://github.com/Virtuosek/Radio>

Since the repository is private, you might get a 404 error if you're not a collaborator, if you wish to be one, please send an email to [karim.essaf@gmail.com](mailto:karim.essaf@gmail.com) with your *Git* username in order to be added as a collaborator<sup>2</sup>.

---

<sup>2</sup>Be aware that we are only able to use the basic version of *Git*, so only one collaborator can be added.