CSCI217

Documentation

For Study Partner Project

Team Members:

Khadija Hatem

Rawan El – Farmway

Sama Okasha

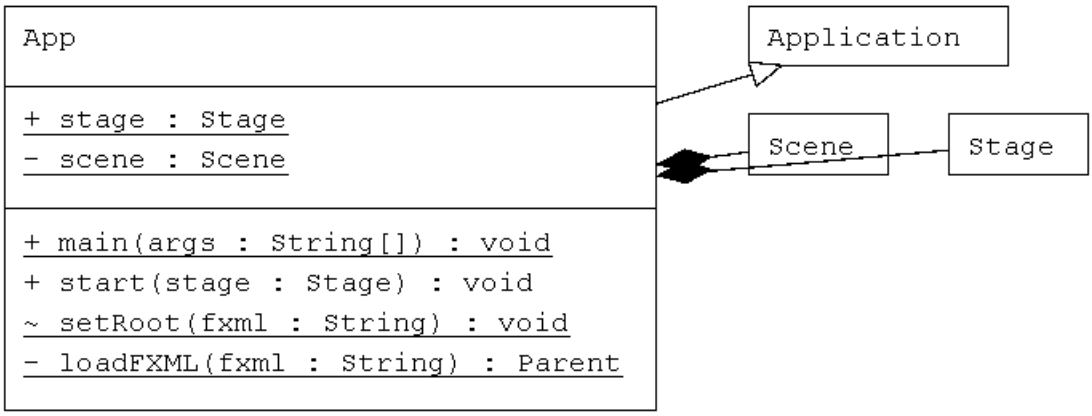
Mariam Barakat

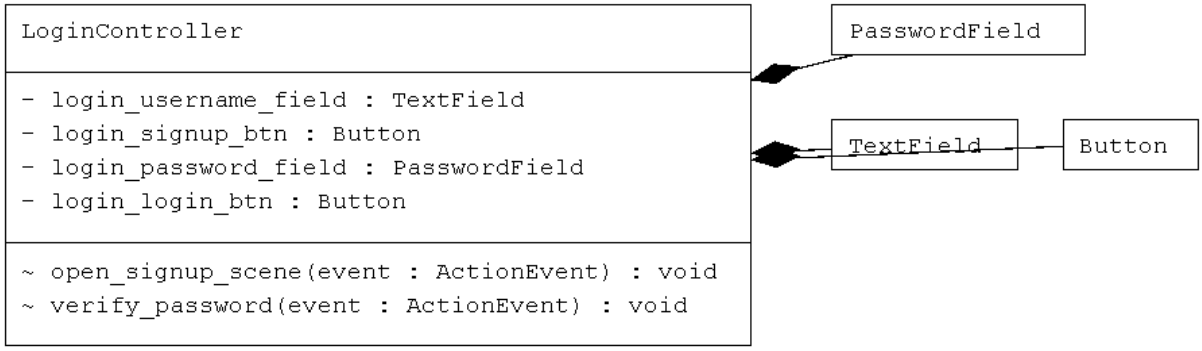
Description about the project and the way it works:

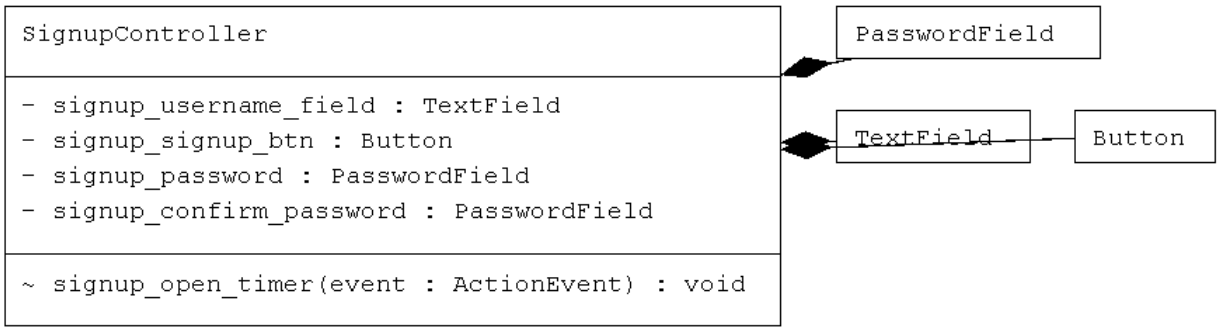
The study partner project is the name of our project. As the user first logs into his account, if he has one, or registers for one. Once the user is in, he can set the timer for studying and select the music he wants to listen to while learning. Furthermore, he can use the to do-list button to store any information he like, although he can use the reset button to restart the timer or the music.

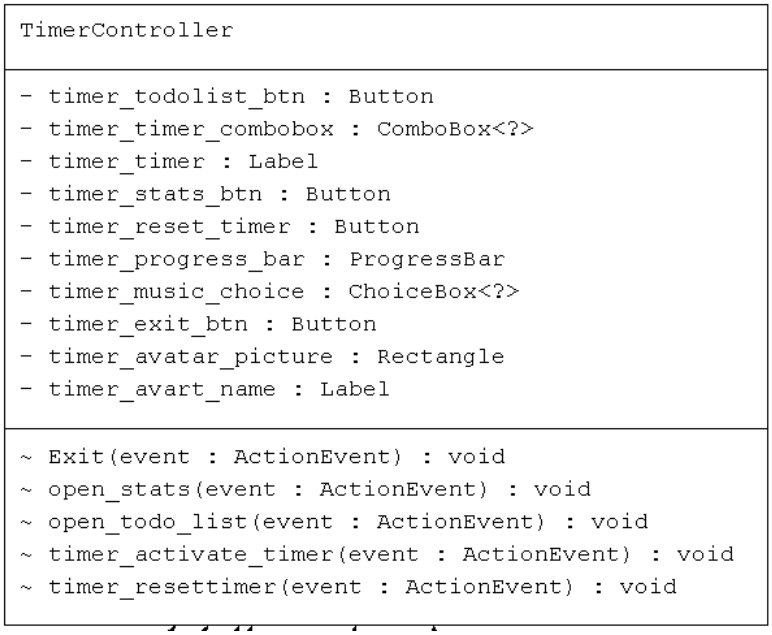
At first, we utilized Scene Builder to create the outline of our project and all of the fonts we needed to make it look more interesting. Furthermore, we used JAVAFX in our functions to bring all of the scene builder implementation to life. We used a variety of imports. We used five separate classes. One for the main app, one for the signup process, one for the start controller, one for the timer controller, and one for the to do controller. We then implemented all of our functions in those various classes while also utilizing the database. Because the database contains a large procedure from the application. It is in charge of the login and sign-up processes, as well as playing a role in the to do function. We tested our application after we completed all of our functions.

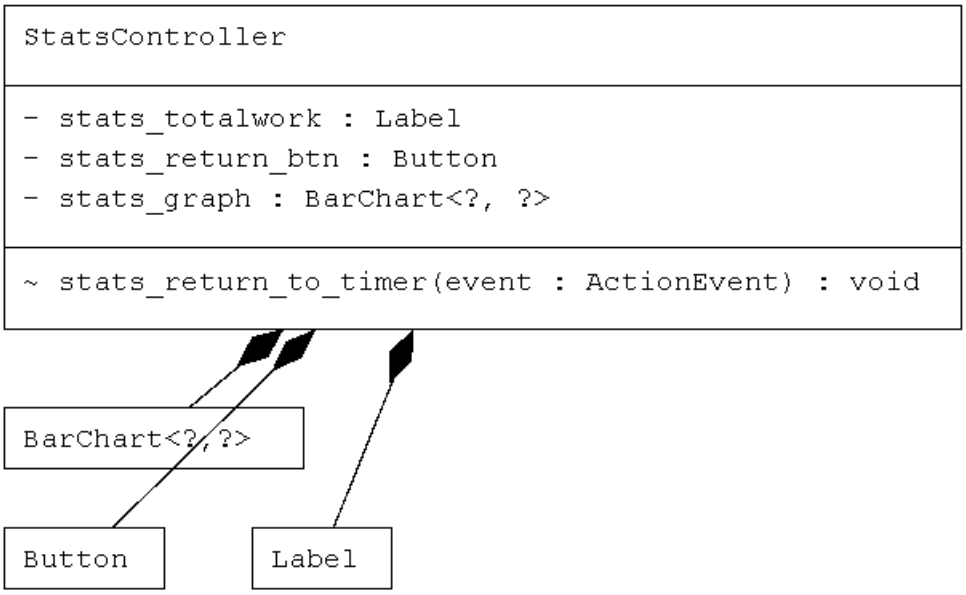
UML of our Project:

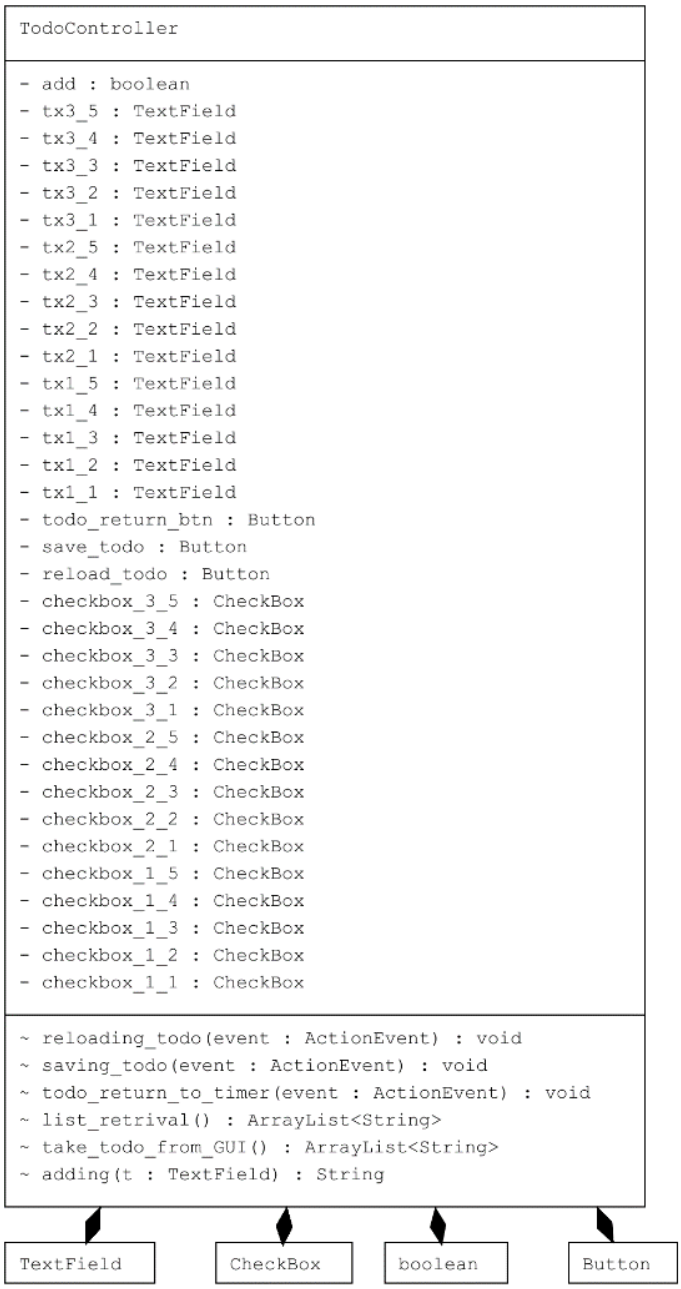




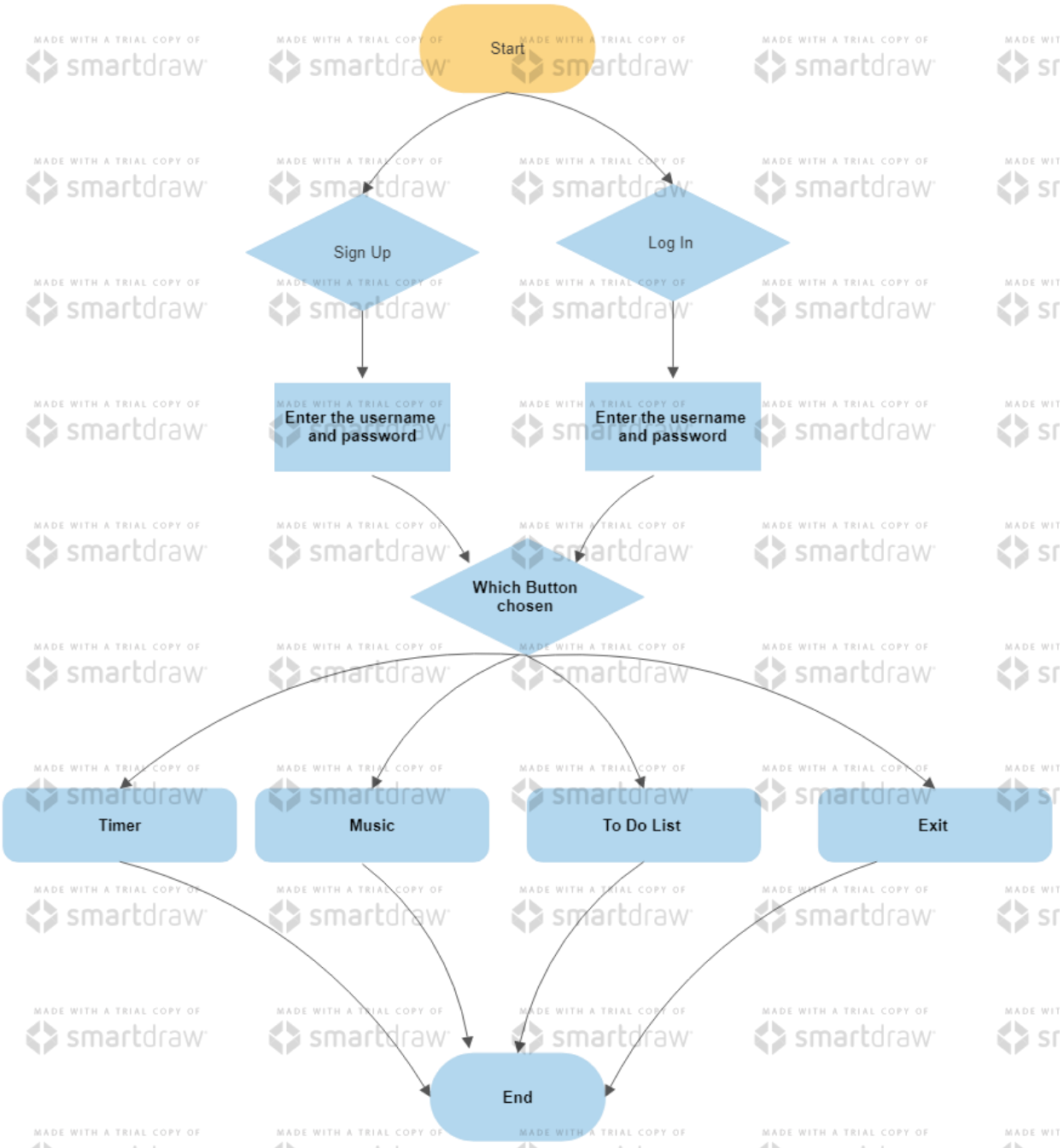



TimerController
- timer_todolist_btn : Button
- timer_timer_combobox : ComboBox<?>
- timer_timer : Label
- timer_stats_btn : Button
- timer_reset_timer : Button
- timer_progress_bar : ProgressBar
- timer_music_choice : ChoiceBox<?>
- timer_exit_btn : Button
- timer_avatar_picture : Rectangle
- timer_avart_name : Label
~ Exit(event : ActionEvent) : void
~ open_stats(event : ActionEvent) : void
~ open_todo_list(event : ActionEvent) : void
~ timer_activate_timer(event : ActionEvent) : void
~ timer_resettimer(event : ActionEvent) : void


StatsController
- stats_totalwork : Label
- stats_return_btn : Button
- stats_graph : BarChart<?, ?>
~ stats_return_to_timer(event : ActionEvent) : void
BarChart<?,?>
Button
Label


TodoController
- add : boolean
- tx3_5 : TextField
- tx3_4 : TextField
- tx3_3 : TextField
- tx3_2 : TextField
- tx3_1 : TextField
- tx2_5 : TextField
- tx2_4 : TextField
- tx2_3 : TextField
- tx2_2 : TextField
- tx2_1 : TextField
- tx1_5 : TextField
- tx1_4 : TextField
- tx1_3 : TextField
- tx1_2 : TextField
- tx1_1 : TextField
- todo_return_btn : Button
- save_todo : Button
- reload_todo : Button
- checkbox_3_5 : CheckBox
- checkbox_3_4 : CheckBox
- checkbox_3_3 : CheckBox
- checkbox_3_2 : CheckBox
- checkbox_3_1 : CheckBox
- checkbox_2_5 : CheckBox
- checkbox_2_4 : CheckBox
- checkbox_2_3 : CheckBox
- checkbox_2_2 : CheckBox
- checkbox_2_1 : CheckBox
- checkbox_1_5 : CheckBox
- checkbox_1_4 : CheckBox
- checkbox_1_3 : CheckBox
- checkbox_1_2 : CheckBox
- checkbox_1_1 : CheckBox
~ reloading_todo(event : ActionEvent) : void
~ saving_todo(event : ActionEvent) : void
~ todo_return_to_timer(event : ActionEvent) : void
~ list_retrival() : ArrayList<String>
~ take_todo_from_GUI() : ArrayList<String>
~ adding(t : TextField) : String
TextField
CheckBox
boolean
Button

Flowchart of our Project:



Thank you