SE302 Design Document of Group 4

Ulaş Doğan, Naci Berkay Caner, Muhammed Furkan Doğan and Furkan Burucu

December 2020

Contents

1 Introduction

The main purpose of the diagrams is to explain the work of the program by modeling within its own rules, which indicates how the program will work. Use case diagram, class diagram and sequence diagrams will be used in the document. These diagrams are basically used for the same purposes, but their modelling is different. With this way of approach, the customer will be able to fully understand how the program works. Apart from these, thanks to the UI mock up, how the program will appear on the user's screen (shown as formatted images) can be seen.

2 Use Case Diagram

Use Case Model indicates the main purpose of the program with a general view. Use cases are identified as important scenarios that are included. Every case has a specification. Editing and Changing syllabus software's purpose is to provide a user to view, edit and change the syllabus after importing from the website. Additionally, it provides user to edit and create the contents of the syllabus, assign teachers and maintain the books according to the course. Therefore, the system can be exported as a HTML format by the user.

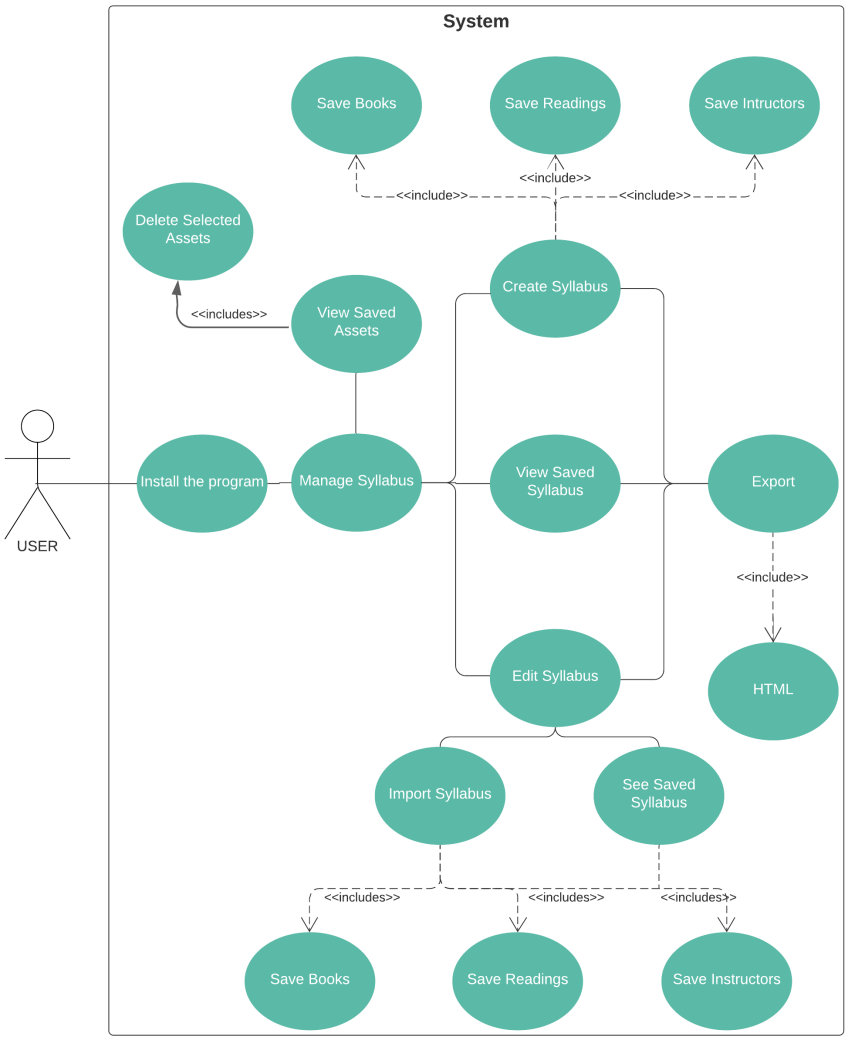

Figure 1: The use case diagram of the software.

3 Class Diagram

The class diagram is used to describe the classes in the program along with their relationships. A class can have more than one relationship with other classes, as well as a single relationship. We used multiple classes to give information on how the newly created syllabus will look like and how the operations and methods will vary upon the class.

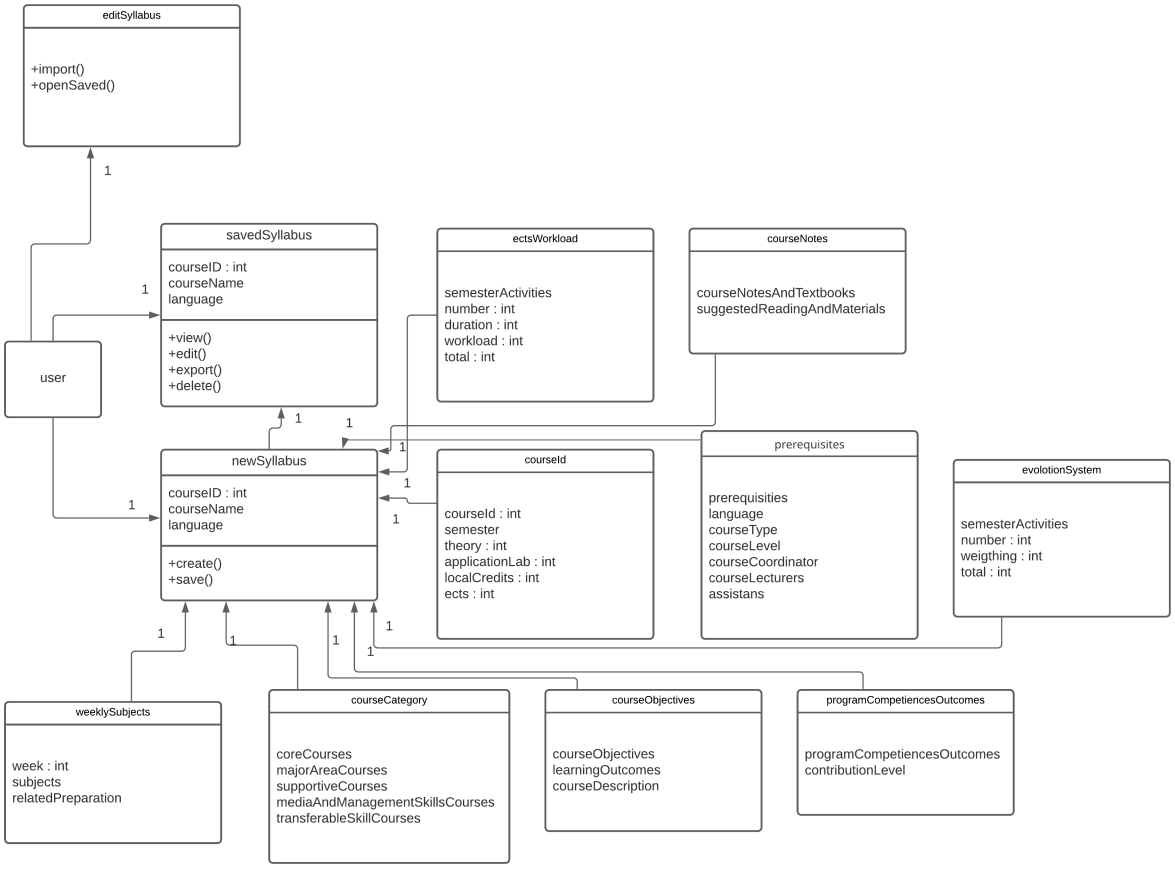

Figure 2: The class diagram of the software.

4 Sequence Diagram

Sequence diagrams are interaction diagrams to show how the operations are carried out. They are usually time focused and show the order of interactions. We used the sequence diagram to show how the commands from user will be carried and how will the system handle the operations.

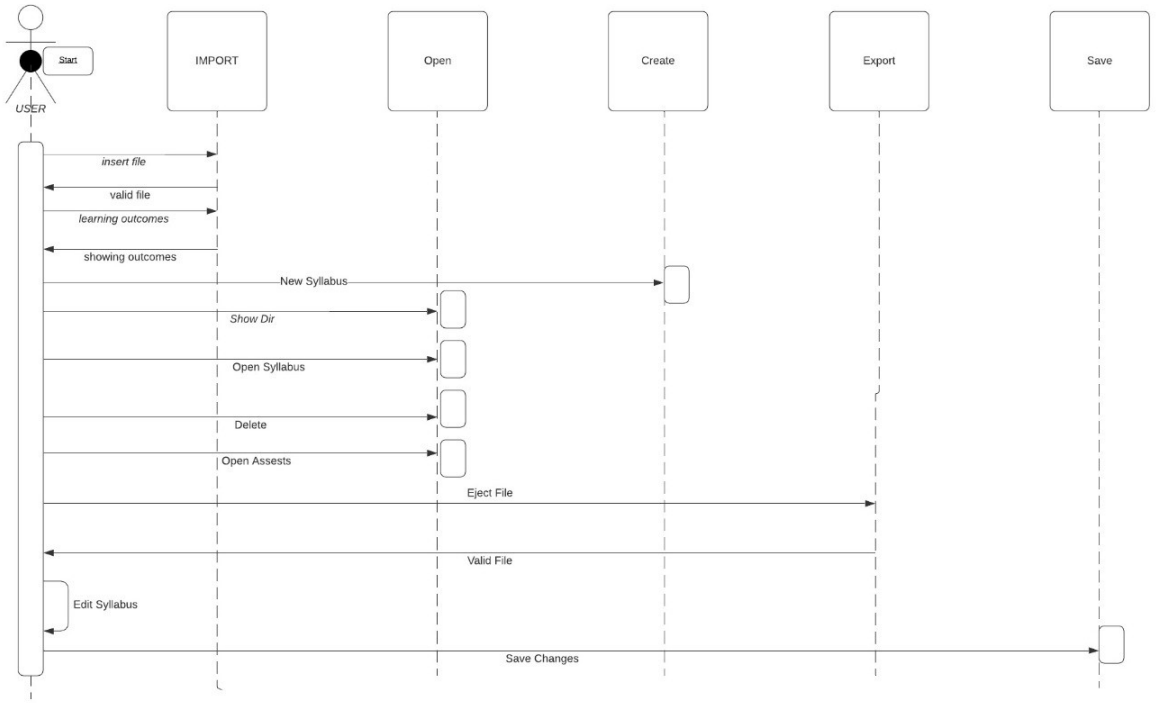

Figure 3: The sequence diagram of the software.

5 State Diagram

The state diagram is used to give an abstract description of our program's behaviour. The behaviour is created and represented by states that can occur more than once. So each operation behaves unique and operates alone. We used the state diagram to show how the system will operate through decision and result based states and to see what will become upon selections and results.

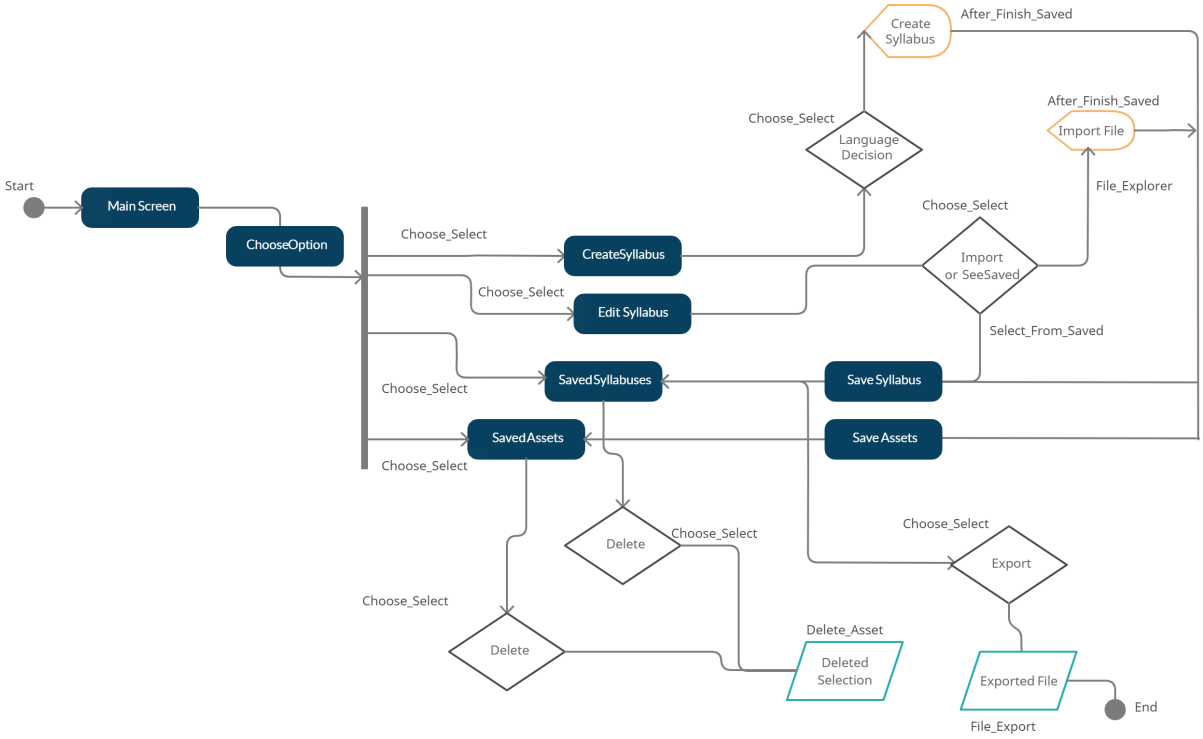

Figure 4: State Diagram of the software.

6 User Interface Mock Up

The User Interface Mock Up is the estimated look of what the software's user interface will look like and a couple of wire-frames created to give a better idea of the available options and window pop ups.

Figure 5: Example of the program's start up screen.

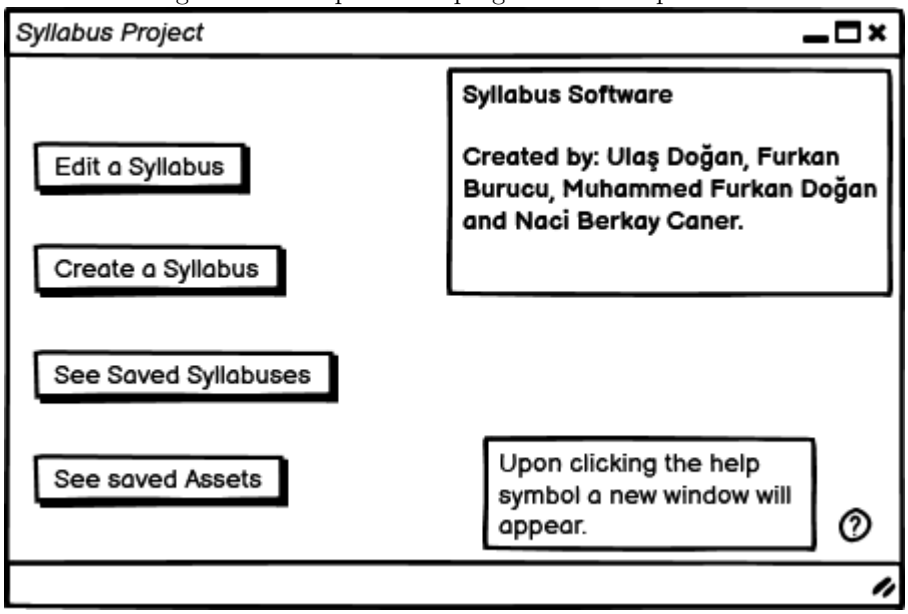

Figure 6: The screen that comes up upon clicking the "edit a syllabus" option.

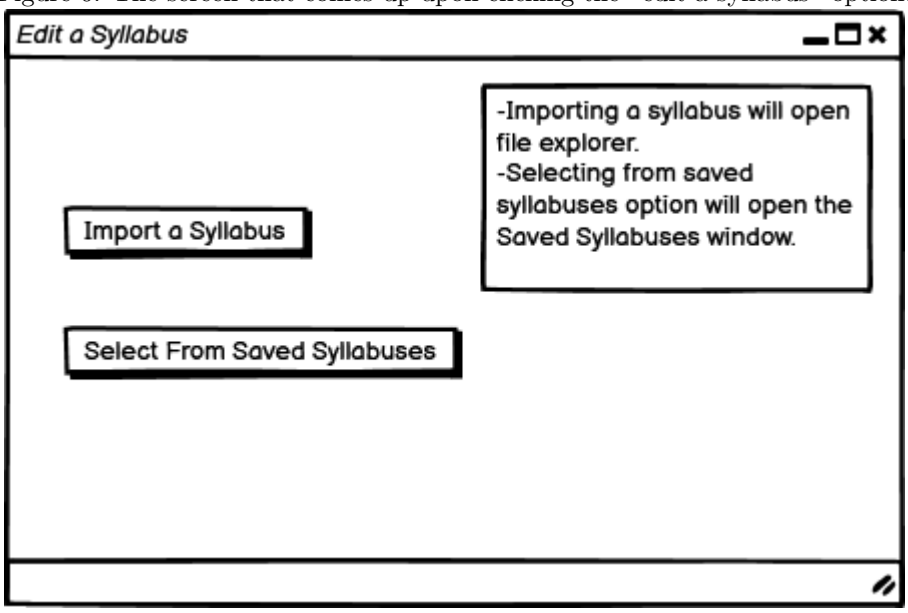

Figure 7: The screen that comes up upon clicking the "create a syllabus" option.

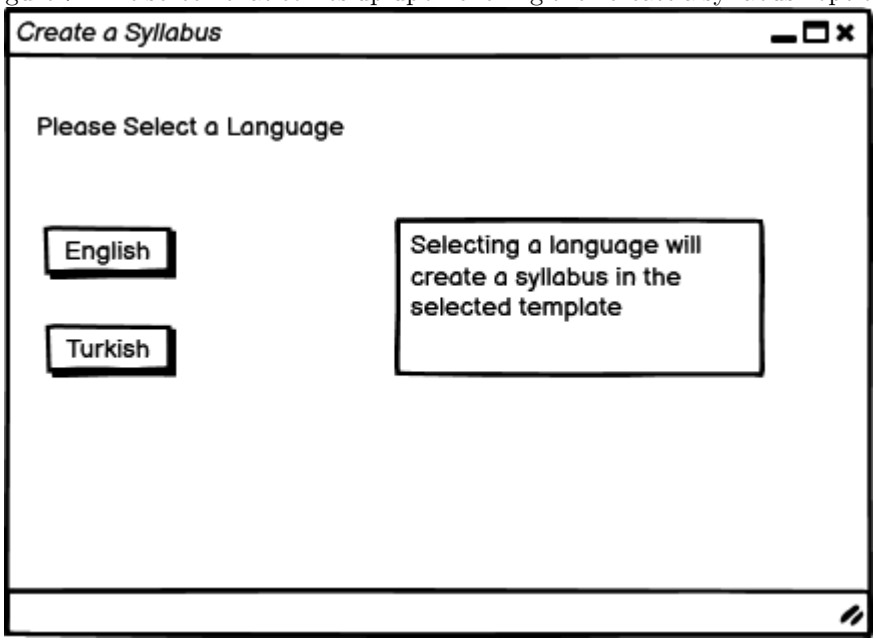

Figure 8: The screen that comes up upon clicking "saved syllabuses" option or the "select from saved syllabuses" option in "edit a syllabus" screen.

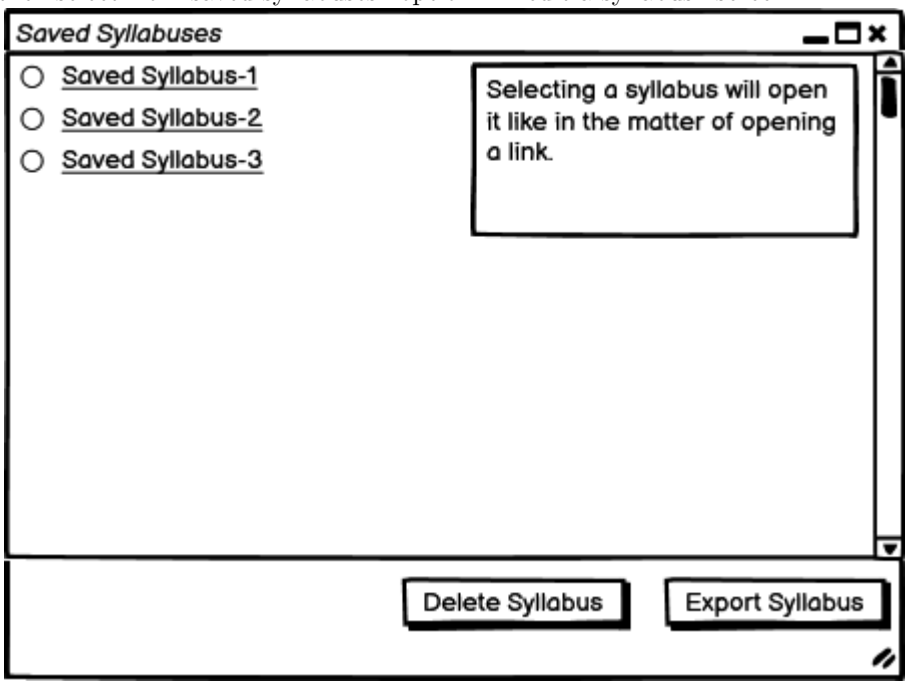

Figure 9: The screen that comes up upon clicking the "saved assets" option.

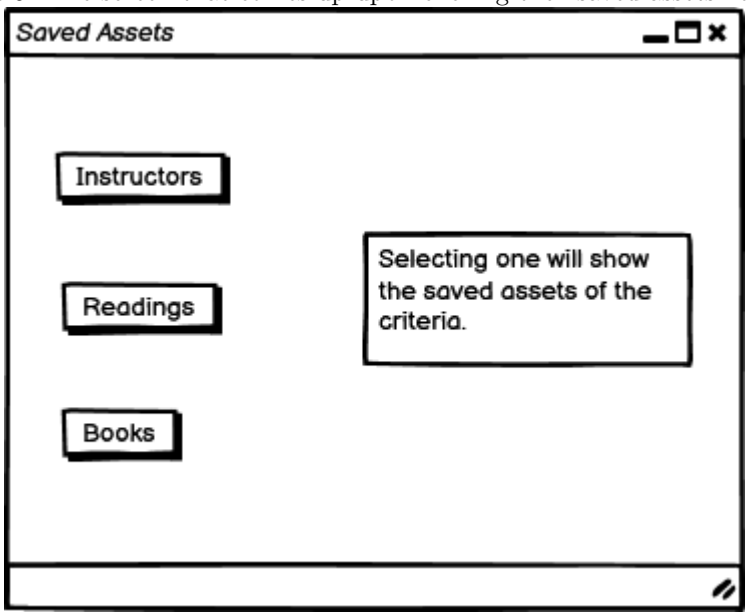

Figure 10: The screen that comes up upon clicking the "instructors" option.

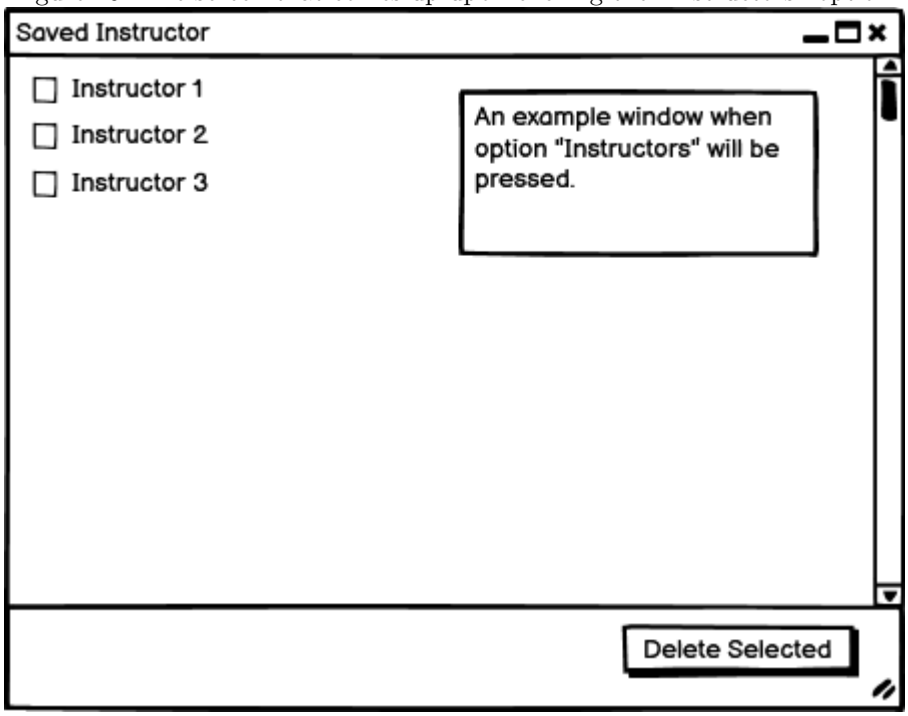

Figure 11: The screen that comes up upon clicking the "help" option in the main menu.