**s o f i a**

**Functional Specification**

*TeachEasy and LearnEasy Functional Specifications*

**Document Control**

|  |  |  |
| --- | --- | --- |
| **Editor** | **Date** | **Update** |
| **Alex Cash** | **28/01/15** | **1.0** |
| **Alex Cash** | **30/01/15** | **1.1** |

# Table of Contents

[**Introduction** 4](#_Toc410300318)

#### Product Introductions and Target Markets 4

#### User Requirements 5

#### TeachEasy User Stories 5

#### LearnEasy User Stories 6

#### Proposed GUI Wireframes 7

## Introduction

This document shall define the functional requirements and usage cases for both the sofia TeachEasy and the sofia LearnEasy applications. This document shall also define the target market for the applications which shall be used to explain the initial design choices. This document shall aim to provide an indication of how the final product shall behave from a user’s perspective through the use of user stories and user requirements.

## Product Introductions and Target Markets

sofia TeachEasy is a teaching product that is aimed at providing a platform for teachers and tutors to create interactive lessons for students to complete in their own time. This is a way for tutors and teachers to get the best out of their students, and for the students to get the most out of their lessons, as they can support their learning independently at their own pace. TeachEasy will allow a teacher to create a lesson for their student(s) that can then be transferred to the student(s) for completion. These lessons also have the benefit of being very interactive, taking advantage of the fact that they are to be completed on a computer, the teacher will be able to set lessons which require direct input from the student, be that text or mouse clicking inputs.

The target market for TeachEasy is primarily for tutors and teachers who only see their students for a limited amount of time; a prime example being a maths tutor who has been hired to support a child’s learning in school, yet only gets to work face to face with the child for one hour a week. In this situation, the tutor could create lessons to their own specifications which the student could then complete; effectively increasing the amount of time they are working with the child. Whilst this product is to be aimed at tutors and teachers with limited levels of interaction with the student, the intended features of the product also enable a regular school teacher to utilise the product – assigning homework to a whole class easily.

Additionally, it is worth noting that whilst there is no age range associated with the product, the lesson plans will be best suited to students who are beginners. By this it is meant that the lessons themselves can be full of information, but will be restricted to a simple format, thus not enabling overly complex topics or learning methods to be taught effectively.

sofia LearnEasy is the sister product of TeachEasy which functions as a “player” that can display the previously created lessons to the student. This player will open the lessons and enable interaction from the student, without allowing the student to edit the lessons, thus removing the fear of any foul play by a student.

LearnEasy is primarily designed as an accompanying product to the TeachEasy product, but may be marketed at a later date to students keen to use the product for lessons provided online, by either ourselves or a community of users. This is not within the scope of this project, and as such LearnEasy will be treated as an accompanying application to TeachEasy.

## User Requirements

The following are a set of user requirements that are necessary to define before beginning the project, and apply to any user using either TeachEasy or LearnEasy:

* The user must be able to input information to the product(s) via keyboard and mouse
* The product(s) should output information to the user via computer monitor (visual) and speakers/headphones (aural)
* User must be able to use the product(s) on a computer in a Windows environment

## TeachEasy User Stories

The following are a series of user stories from the perspective of the teacher, from which the product shall be interpreted, designed, implemented and tested. The overall TeachEasy project can be summarised by the following user story:

“As a teacher, I can create an interactive lesson for a student of mine to complete independently”

From which we can derive the following user stories, on which we can base our iterative design, implementation, and testing routines:

“As a teacher, I can provide any number of my students with access to a lesson I created previously”

“As a teacher, I can create a lesson comprised of a number of discrete pages, each of which I can customise”

“As a teacher, I can pick a page category from a number of pre-defined templates (e.g. video, quiz, etc.)”

“As a teacher, I can include multimedia objects in the lessons I create”

“As a teacher, I can save lessons I am working on then access them again and edit them at a later date”

“As a teacher, I can assign marks to exercises on each page, where necessary”

These are the user stories we shall use to create our product, but may be subject to alterations throughout the project.

## LearnEasy User Stories

The following are a series of user stories from the perspective of the student, from which the product shall be interpreted, designed, implemented and tested. The overall LearnEasy project can be summarised by the following user story:

“As a student, I can complete an interactive lesson created for me by my teacher, in my own time”

From which we can derive the following user stories, on which we can base our iterative design, implementation, and testing routines:

“As a student, I can view a lesson created by my teacher”

“As a student, I cannot edit a lesson”

“As a student, I can choose the lesson I want to work on from a selection of lessons provided to me by my teacher(s)”

“As a student, I can pause and resume lessons”

“As a student, I can view all forms of multimedia that my teacher has included in the lesson”

“As a student, I can interact with all suitable forms of multimedia (e.g. pause a video)”

“As a student, I am provided with a record detailing my lesson completion and level of achievement, which I can choose to print out”

As before, these are the user stories that will be used throughout the project, however may be subject to alterations throughout the project.

## Proposed GUI Wireframes

The following is a suggestion of how the GUI of the TeachEasy application may look:

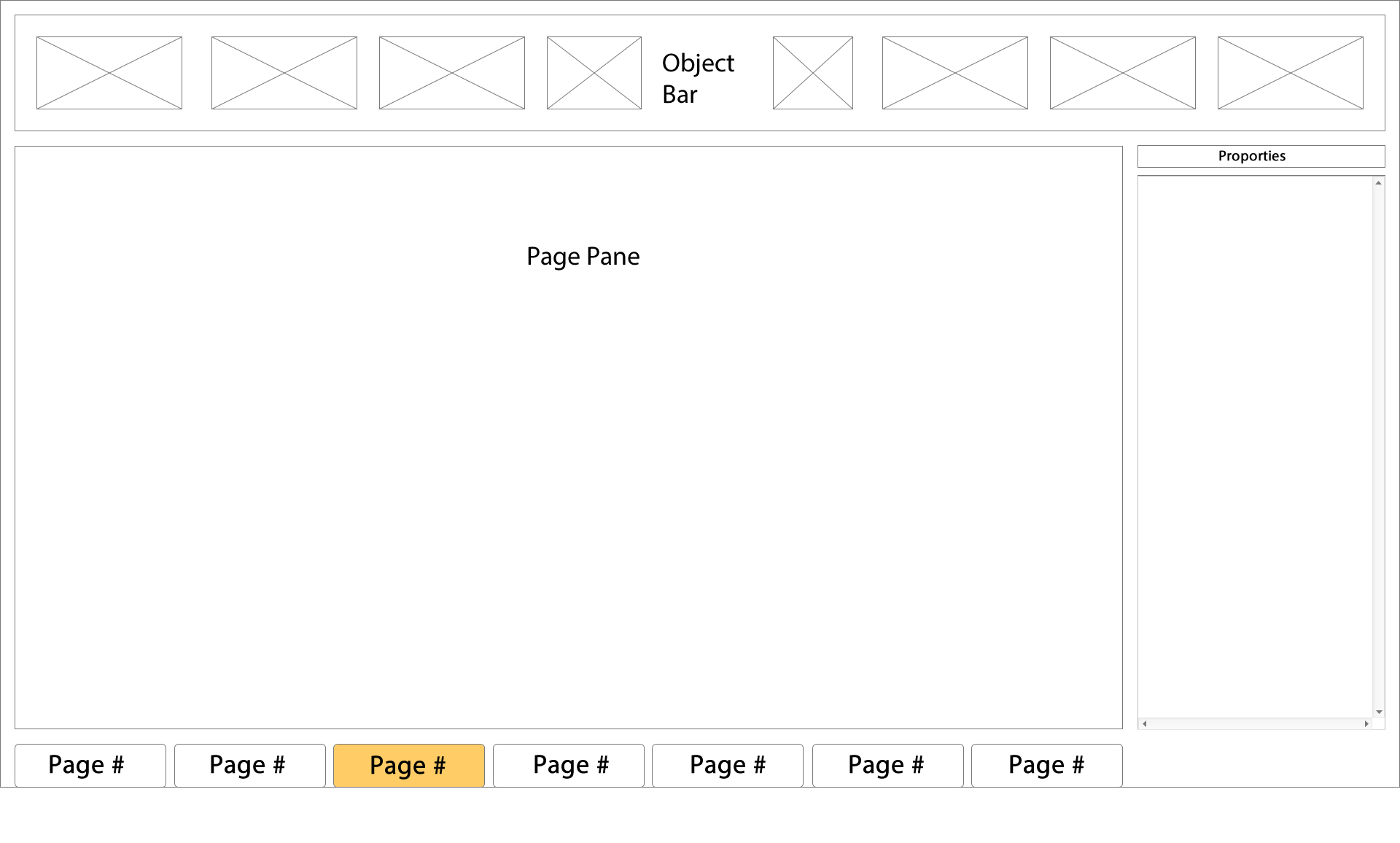
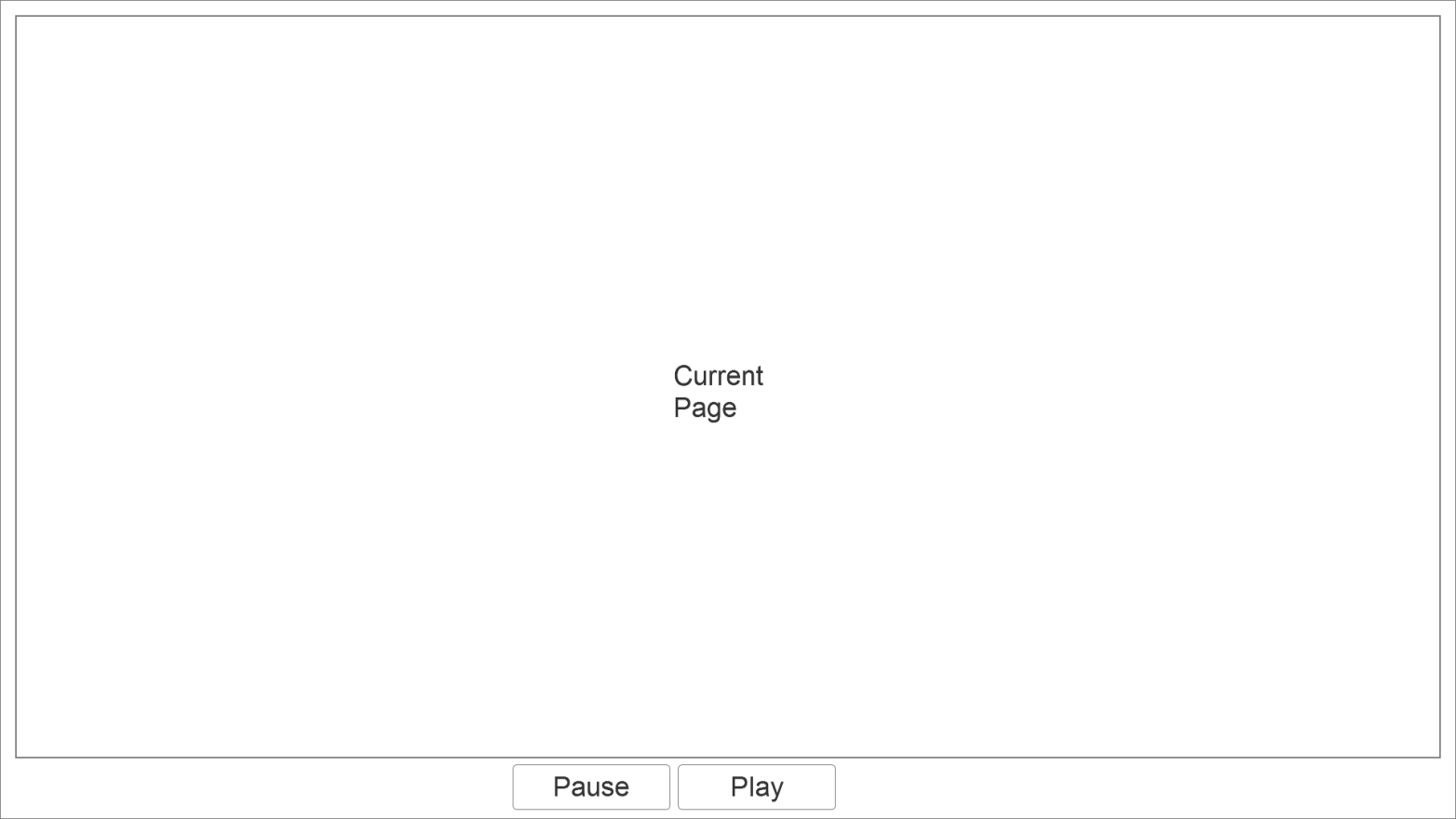


Figure 1: Wireframe depicting an initial TeachEasy GUI proposal

As can be seen from the diagram, the user-created pages are selectable using buttons along the bottom of the screen which will change in appearance to make it instantly obvious to the user which page is selected. Along the top of the screen will be a series of objects that can be used to customise the page, including things such as drag-and-drop shapes. To the right of the page is the main properties pane, which will be used to edit a page’s properties and thus the properties of the lesson. The properties pane is intended to be context sensitive, so will be used to edit the properties of a selected element of a lesson, such as a table or diagram. This pane will also include properties such as template formats and user-defined options. Not displayed in the diagram are the traditional menus such as “File” etc. These will be included and used to handle file saving and opening and other typical functions required by a user in a Windows environment.

The following is a suggestion of how the GUI of the LearnEasy application may look:

Figure 2: Wireframe depicting an initial LearnEasy GUI proposal

As can be seen from the diagram, the LearnEasy player will be a very much simplified interface, removing the user’s ability to edit the lesson plans. The two main buttons available to the user will be used to pause and resume lessons, offering the user the chance to stop working for a while and return to the lesson at a later time. Not shown in the diagram are the traditional menus, as before, such as “File” etc. These will once again be used to handle file operations.