Wireframer ™
Software Requirements Specification

Author: Richard McKenna
Debugging Enterprises™

Based on IEEE Std 830™-1998 (R2009) document format

1 Introduction

Requirements Engineering is the step in the Software Development Lifecycle where the team determines what to make. This means figuring out all the ways the user will interact with the system, then figuring out all the necessary controls to serve those interactions and determining how to layout those controls.

We can think of user interface design as laying out boxes, as all UI controls have a width and height as well as x, y position. We can also contain such boxes inside of other boxes. Because of this, UI layout and design is sometimes called wireframing as the outlines of our controls approximate to lines that help us visualize how to effectively arrange all needed components.

There are many tools for creating wireframe diagrams, what will set ***Wireframer*** apart is it ease of use, the ability to save our work online to a database, and the simplicity of using the app.

1.1 **Purpose**

The purpose of this document is to specify how our ***Wireframer*** program should look and operate. The intended audience for this document is all the members of the development team, from the requirements engineers to the software engineers and designers. This document serves as an agreement among all parties and as a reference for how the wireframing tool should ultimately be constructed. Upon completing the reading of this document, one should clearly visualize how the application will look and operate.

1.2 Scope

For this project the goal is for users to easily make and update wireframe diagrams and then be able to export those diagrams easily into an image for use in a report.

1.3 **Definitions, acronyms, and abbreviations**

Document Object Model (DOM) – a tree data structure maintained by the browser that contains all content for the currently loaded Web page.

Framework – In an object-oriented language, a collection of classes and interfaces that collectively provide a service for building applications or additional frameworks all with a common need.

GUI – Graphical User Interface, visual controls like buttons inside a window in a software application that collectively allow the user to operate the program.

HyperText Markup Language – a markup language used to describe Web pages. Web pages are text files encoded in HTML that can employ JavaScript and Stylesheets to build and style content.

IEEE – Institute of Electrical and Electronics Engineers, the “world’s largest professional association for the advancement of technology”.

JavaScript – the default scripting language of the Web, JavaScript is provided to pages in the form of text files with code that can be loaded and executed when a page loads so as to dynamically generate page content in the DOM.

Stylesheet – a static text file employed by HTML pages that can control the colors, fonts, layout and other style components in a Web page.

UML – Unified Modeling Language, a standard set of document formats for designing software graphically.

Use Case Diagram – A UML document format that specifies how a user will interact with a system.

1.4 References

IEEE Std 830^{TM}-1998 (R2009) – IEEE Recommended Practice for Software Requirements Specification

1.5 Overview

This SRS will clearly define how the ***Wireframer*** application should look and operate. Note that this is not a software design description (SDD), which would design how to construct the software using UML. This document does not specify how to build the appropriate technologies, it is simply an agreement concerning what to build. Section 2 of this document will provide the context for the project and specify all the conceptual design. Section 3 will present how the user interface should be laid out. Section 4 provides a Table of Contents, an Index, and References.

2 Overall description

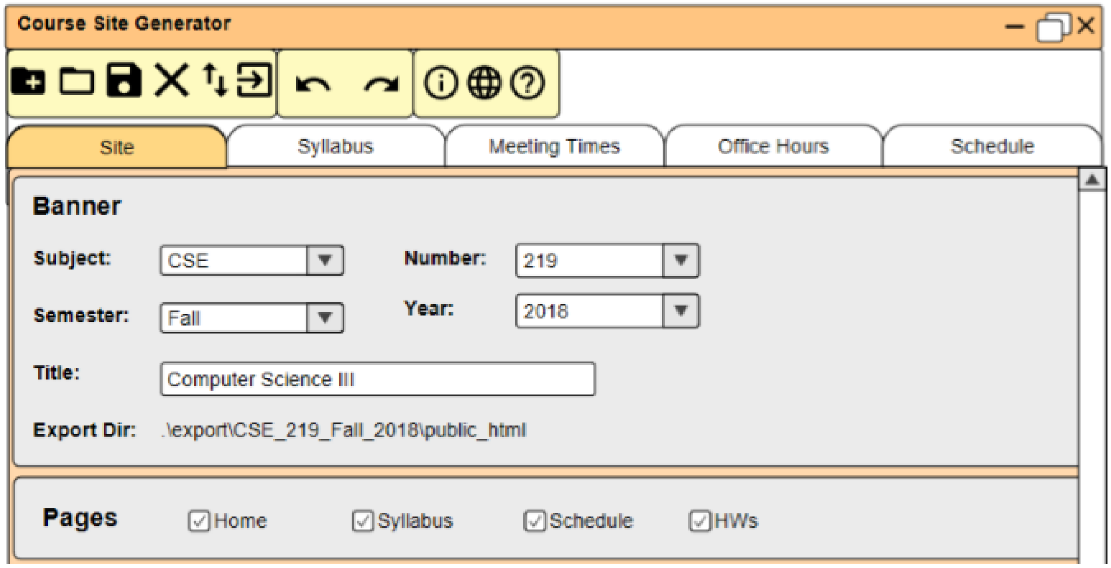
Here you are a software requirements engineer preparing a Software Requirements Specification and it's time to create User Interface Mockups, i.e. rendered approximations of the User Interface to be built. There are many tools to do so, but most are behind a pay wall, have limitations for free use, and tend to be rather complicated to use. All of these present problems, particularly for student users who wish to easily get in and get out and keep their work saved in an online database to make for easy retrieval. With ***Wireframer*** we hope to create a minimalist tool that is easy to get in, get simple work done, and get out with work conveniently saved online for future retrieval.

2.1 Product perspective

Our product will not provide the full range of services that many other wireframing applications provide but instead will provide a core set of important, easy to use services, to allow one to make high quality wireframes without much time or training. It will provide a limited number of controls to use in wireframing.

2.1.1 System Interfaces

Wireframer will be a Web-based application that should be usable via any browser and will let the user create a new wireframe diagrams and edit existing ones they've created as needed. As such each diagram will require a name as well as a unique id. In addition, diagrams will belong to the user who created them, so only the creator can access to view or update a given diagram. Note that there are a number of wireframe-building type applications already on the market like Moqups.com, which was used to create the example wireframe diagram shown below in Figure 2.1.



Subject: CSE

Number: 219

Semester: Fall

Year: 2018

Title: Computer Science III

Export Dir: .\export\CSE_219_Fall_2018\public_html

☑ Home ☑ Syllabus ☑ Schedule ☑ HWs

Figure 2.1 Example Wireframe User Interface Mock-Up made using Moqups.com

2.1.2 User Interfaces

Our site editing program will be a desktop application and so will make use of a mouse and keyboard for user input. Figure 2.2 below summarizes the ways with which the user will interact with our ***Course Site Generator*** application, which will be further detailed using UML Use Case diagrams. These Use-Case diagrams should be fed as input directly into Section 3.1, external interfaces, which is where the design of the user interface is specified. Here is the full list of UML Use-Case Diagrams:

Use Case	UI Context	Use Case
2.1	Splash Screen	Login to Account
2.2	Register Screen	Create new Account
2.3	Multiple Screens	Logout of Account
2.4	Home Screen	Create New Wireframe
2.5	Home Screen	Select Wireframe to Edit
2.6	Edit Screen	Change Wireframe Dimensions
2.7	Edit Screen	Zoom In/Out
2.8	Edit Screen	Add Control
2.9	Edit Screen	Reposition Control
2.10	Edit Screen	Resize Control
2.11	Edit Screen	Duplicate Control
2.12	Edit Screen	Delete Control
2.13	Edit Screen	Unselect Control
2.14	Edit Screen	Change Control Text
2.15	Edit Screen	Change Control Text Font Size
2.16	Edit Screen	Change Control Colors
2.17	Edit Screen	Change Control Border
2.18	Edit Screen	Save Work
2.19	Edit Screen	Close Work/Go Home
2.20	Home Screen	Delete Diagram
2.21	Test Screen	Test Database

Figure 2.2: Overview of Use-Case Diagrams

Note that there are two different actors who will make use of our application:

UI Designer	This user can create, edit, and save wireframe diagrams. Note that this is the only type of actor/account that can be created via the browser registration process.
Administrator	This user can do all the things that a UI Designer can do and in addition can access and make use of the databaseTester screen, which allows them to test all required updates of the database that need to be developed. Users should be made administrators sparingly via the online database console.

Use Case 2.1: Login to Account

Use-Case:	Login to Account
Primary Actor:	UI Designer
Goal in Context:	The user wishes to login to an existing site account
Preconditions:	The user has already created an account for the site
Trigger:	The user can be routed to the Login Page either by going to the site home page (i.e. no routing path) or by trying to visit another route, which will forward them to the login page. Note that one may end up on the Login page by logging out from another screen as well.
Key Shortcut:	N/A
Scenario:	1. User is viewing the Login page 2. User enters a valid account name in the first textfield 3. User enters the corresponding account password in the second textfield 4. User presses the Login button, which will submit the user name/password entries to the server for confirmation. 5. For valid credentials, the user will be forwarded to their Home Screen
Exceptions:	Should the user enter an invalid user name or password the user interface must provide sufficient feedback to the user.
Priority:	Essential, must be implemented
Frequency of use:	Used every time the user wishes to do work
Open Issues:	Size, location, and style of screens and controls should be finalized by UI designer

Use Case 2.2: Create New Account

Use-Case:	Create New Account
Primary Actor:	UI Designer
Goal in Context:	The user wishes to create a new site account, which is required for creating, saving, and editing wireframes
Preconditions:	The application has been started and the user is not logged in.
Trigger:	From the Login Screen, the user clicks the Register button in the site Navbar.
Key Shortcut:	N/A
Scenario:	1. User is viewing the Register Screen 2. User enters their email address in the first textfield 3. User enters their password in the second textfield 4. User enters their First Name in the third textfield 5. User enters their Last Name in the fourth textfield 6. User clicks the Register button 7. If successful, user account is created and user is forwarded to their Home Screen, which would not yet have any Wireframes to edit.
Exceptions:	Should the user enter an email address in the incorrect format or should the user leave out one of the required fields or should the user enter an email address that has already been reserved, registration will fail, and appropriate feedback must be provided to the user. This option is only available in the Navbar when the user is not logged in.
Priority:	Essential, must be implemented
Frequency of use:	Used once per account
Open Issues:	Size, location, and style of screens and controls should be finalized by UI designer

Use Case 2.3: Logout of Account

Use-Case:	Logout of Account
Primary Actor:	UI Designer
Goal in Context:	The user wishes to leave the account they are currently logged in with.
Preconditions:	The application has been started and the user is logged in.
Trigger:	N/A
Key Shortcut:	N/A
Scenario:	1. User is viewing either the Home Screen or Edit Screen, which would mean they are already logged in 2. User clicks the Logout button in the Navigation bar. This causes the user account to be logged out and the user is forwarded to the Login Screen 3. User views Login Screen
Exceptions:	This option is only available in the Navbar when the user is logged in
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	Used every time the user wishes to logout of their account, for example, when they wish to switch to a different account
Open Issues:	Size, location, and style of screens and controls should be finalized by UI designer

Use Case 2.4: Create New Wireframe

Use-Case:	Create New Wireframe
Primary Actor:	UI Designer
Goal in Context:	The user wishes to save the current course site under another file
Preconditions:	The user is logged in
Trigger:	The user navigates to the Home Screen, where all of the wireframes that they have created are listed by order of last viewed, with the most recently viewed/edited at the top.
Key Shortcut:	N/A
Scenario:	1. User is viewing the Home Screen 2. User clicks on the Create New Wireframe button, which routes the user to the Edit Screen. This new wireframe would now be first in the list of the user's wireframes should the user return to the Home Screen. 3. User views Edit Screen and begins editing Wireframe
Exceptions:	The Create New Wireframe button is ***always*** enabled
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	Used every time the user wishes to make a new wireframe, which might be daily, weekly, monthly, or otherwise depending on the user's use of the site.
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

Use Case 2.5: Select Wireframe to Edit

Use-Case:	Select Wireframe to Edit
Primary Actor:	UI Designer
Goal in Context:	User wishes to view or edit an existing wireframe that they own
Preconditions:	The user has logged in and has at least one existing wireframe
Trigger:	The user navigates to the Home Screen, where all of the wireframes that they have created are listed by order of last viewed, with the most recently viewed/edited at the top.
Key Shortcut:	N/A
Scenario:	1. User is viewing the Home Screen, which has at least one named wireframe listed that the user has already created 2. User clicks on one of their wireframes. This routes the user to the Edit Screen and loads all the data for the selected wireframe into the screen controls. This selected wireframe would now be first in the list of the user's wireframes should the user return to the Home Screen. 3. User views Edit Screen and begins editing the selected Wireframe
Exceptions:	N/A
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	Used every time the user wishes to edit existing work
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

Use Case 2.6: Change Wireframe Dimensions

Use-Case:	Change Wireframe Dimensions
Primary Actor:	UI Designer
Goal in Context:	The user would like to make their wireframe diagram larger or smaller
Preconditions:	The user is logged in and has at least one wireframe diagram that they own
Trigger:	The user either creates a new wireframe (which has a default size that can be changed) or selects an existing wireframe to edit
Key Shortcut:	N/A
Scenario:	1. The user is viewing a wireframe they own on the Edit Screen 2. The user enters a new pixel width in the Wireframe width text field. This enables the Update dimensions button. 3. The user enters a new pixel height in the Wireframe height text field. This would also enable the Update dimensions button. 4. The user presses the Update dimensions button. The size of the diagram, as depicted by the background color area, is changed, reflecting the updated dimensions.
Exceptions:	Non-integer dimension or integers smaller than 1 or larger than 5000 should be disregarded and should not update the diagram.
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	Used every time the user wishes to resize the diagram
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer. Note that the Update dimensions button should simply say "Update" and should be disabled after being pressed.

Use Case 2.7: Zoom In/Out

Use-Case:	Zoom In/Out
Primary Actor:	UI Designer
Goal in Context:	Lets the user view the diagram being edited at different zoom levels.
Preconditions:	The user is logged in and has at least one wireframe diagram that they own
Trigger:	The user either creates a new wireframe (which has a default size that can be changed) or selects an existing wireframe to edit
Key Shortcut:	N/A
Scenario:	1. The user is viewing a wireframe they own on the Edit Screen 2. The user presses the Zoom In button, which updates the scale indicator and makes the diagram 2x bigger. 3. The user presses the Zoom Out button, which updates the scale indicator and makes the diagram ½ size.
Exceptions:	Note that when the diagram is too large to fit in the UI window the diagram should be scrollable via a scrollbar, either vertically, horizontally, or both
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	Likely used every time the user wishes to do work
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

Use Case 2.8: Add Control

Use-Case:	Add Control
Primary Actor:	UI Designer
Goal in Context:	Lets the user add a container, label, text button, or textfield control to the diagram
Preconditions:	The user is logged in and has at least one wireframe diagram that they own
Trigger:	The user either creates a new wireframe (which has a default size that can be changed) or selects an existing wireframe to edit. Note that once the user is editing a diagram they may add as many controls as they wish.
Key Shortcut:	N/A
Scenario:	1. The user is viewing a wireframe they own on the Edit Screen 2. The user presses one of the four add control buttons: add container, add label, add text button, add textfield 3. The user views the added control as it appears in the top left hand corner (the default location) of the diagram in its default size.
Exceptions:	N/A
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	Likely used every time the user wishes to do work
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

Use Case 2.9: Reposition Control

Use-Case:	Reposition Control
Primary Actor:	UI Designer
Goal in Context:	User wishes to move a control to a new location in the diagram
Preconditions:	The user is logged in and has at least one wireframe diagram that they own with at least one control in the diagram
Trigger:	The user either creates a new wireframe (which has a default size that can be changed) or selects an existing wireframe to edit. Note that once the user is editing a diagram they may reposition as many controls as they wish.
Key Shortcut:	N/A
Scenario:	1. The user is viewing a wireframe they own on the Edit Screen 2. The user selects one of the controls by pointing the mouse on it an pressing the left mouse button. This will denote that the control is selected by drawing small rectangles in the four corners of the control 3. The user may hold the mouse button down and drag the control about the diagram and then release the mouse button when it is positioned where they wish to leave it
Exceptions:	N/A
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	Used every time the user wishes to do work
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

Use Case 2.10: Resize Control

Use-Case:	Resize Control
Primary Actor:	UI Designer
Goal in Context:	User wishes to resize a control in the diagram
Preconditions:	The user is logged in and has at least one wireframe diagram that they own with at least one control in the diagram
Trigger:	The user either creates a new wireframe (which has a default size that can be changed) or selects an existing wireframe to edit. Note that once the user is editing a diagram they may resize as many controls as they wish.
Key Shortcut:	N/A
Scenario:	1. The user is viewing a wireframe they own on the Edit Screen 2. The user selects one of the controls by pointing the mouse on it an pressing the left mouse button. This will denote that the control is selected by drawing small rectangles in the four corners of the control 3. The user selects and drags (via left mouse button) one of the four corner rectangles, which will resize the control accordingly 4. The user releases the mouse button
Exceptions:	N/A
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	Used every time the user wishes to do work
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

Use Case 2.11: Duplicate Control

Use-Case:	Duplicate Control
Primary Actor:	UI Designer
Goal in Context:	User wishes to duplicate one of the controls in the diagram such that a new control is added to the diagram with all the same values as the one being duplicated
Preconditions:	The user is logged in and has at least one wireframe diagram that they own with at least one control in the diagram
Trigger:	The user either creates a new wireframe (which has a default size that can be changed) or selects an existing wireframe to edit. Note that once the user is editing a diagram they may duplicate as many controls as they wish.
Key Shortcut:	Control-D
Scenario:	1. The user is viewing a wireframe they own on the Edit Screen 2. The user selects one of the controls by pointing the mouse on it an pressing the left mouse button. This will denote that the control is selected by drawing small rectangles in the four corners of the control 3. The user presses Control-D on the keyboard. This duplicates the control, making a new one with all the same settings as the original, and positions it offset by 100 pixels both vertically and horizontally with the new duplicate control selected 4. The user continues to edit the diagram
Exceptions:	N/A
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	Can be used every time the user wishes to do work
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

Use Case 2.12: Delete Control

Use-Case:	Delete Control
Primary Actor:	UI Designer
Goal in Context:	User wishes to delete one of the controls in the diagram
Preconditions:	The user is logged in and has at least one wireframe diagram that they own with at least one control in the diagram
Trigger:	The user either creates a new wireframe (which has a default size that can be changed) or selects an existing wireframe to edit. Note that once the user is editing a diagram they may delete as many controls as they wish.
Key Shortcut:	Delete
Scenario:	1. The user is viewing a wireframe they own on the Edit Screen 2. The user selects one of the controls by pointing the mouse on it an pressing the left mouse button. This will denote that the control is selected by drawing small rectangles in the four corners of the control 3. The user presses Delete on the keyboard. This deletes the selected control, unselecting the control such that none is selected. 4. The user continues to edit the diagram
Exceptions:	N/A
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	Can be used every time the user wishes to do work
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

Use Case 2.13: Unselect Control

Use-Case:	Unselect Control
Primary Actor:	UI Designer
Goal in Context:	User wants to make sure no control is selected
Preconditions:	The user is logged in and has at least one wireframe diagram that they own with at least one control in the diagram.
Trigger:	One of the controls has been selected.
Key Shortcut:	N/A
Scenario:	1. The user is viewing a wireframe they own on the Edit Screen where a control is already selected 2. The user clicks on the diagram in a location that is not on any control. This deselects the selected control, making it so no control is selected. 3. The user continues to edit the diagram
Exceptions:	N/A
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	Can be used every time the user wishes to do work
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

Use Case 2.14: Change Control Text

Use-Case:	Change Control Text
Primary Actor:	UI Designer
Goal in Context:	User wants to change the text that appears on one of the textual controls (button, textfield or label)
Preconditions:	The user is logged in and has at least one wireframe diagram that they own with at least one textual control in the diagram.
Trigger:	One of the controls has been selected.
Key Shortcut:	N/A
Scenario:	1. The user is viewing a wireframe they own on the Edit Screen where a control is already selected. As the textual control is selected, a textfield with the control's textual content is loaded with the control's text in the properties portion of the UI at right 2. The user clicks on the textfield and edits the text inside the control, which will update the text that appears inside the textual control. 3. The user views the updated diagram with the textual changes
Exceptions:	N/A
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	May be used many times per editing session
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

Use Case 2.15: Change Control Text Font Size

Use-Case:	Change Control Text Font Size
Primary Actor:	UI Designer
Goal in Context:	User wants to change the font size of the text that appears on one of the textual controls (button, textfield or label)
Preconditions:	The user is logged in and has at least one wireframe diagram that they own with at least one textual control in the diagram.
Trigger:	One of the controls has been selected.
Key Shortcut:	N/A
Scenario:	1. The user is viewing a wireframe they own on the Edit Screen where a control is already selected. As the textual control is selected, the control text's font size is loaded with the control's current font size (an integer) into a text field in the properties portion of the UI at right 2. The user clicks on the font size textfield and edits the number inside the control, which will update the size of the text inside the textual control. 3. The user views the updated diagram with the text size changes
Exceptions:	N/A
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	May be used many times per editing session
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

Use Case 2.16: Change Control Colors

Use-Case:	Change Control Colors
Primary Actor:	UI Designer
Goal in Context:	User wants to change the colors used for a particular diagram control
Preconditions:	The user is logged in and has at least one wireframe diagram that they own with at least one control in the diagram.
Trigger:	One of the controls has been selected.
Key Shortcut:	N/A
Scenario:	1. The user is viewing a wireframe they own on the Edit Screen where a control is already selected. As the control is selected, the control's background color, border color, and text color (for text controls) is loaded into color pickers in the properties portion of the UI at right 2. The user selects the desired background, border, text color as desired in the appropriate color picker. 3. The user views the updated diagram with the control color changes
Exceptions:	N/A
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	May be used many times per editing session
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

Use Case 2.17: Change Control Border

Use-Case:	Change Control Border
Primary Actor:	UI Designer
Goal in Context:	User wants to change the border thickness and/or radius used for a particular diagram control
Preconditions:	The user is logged in and has at least one wireframe diagram that they own with at least one control in the diagram.
Trigger:	One of the controls has been selected.
Key Shortcut:	N/A
Scenario:	1. The user is viewing a wireframe they own on the Edit Screen where a control is already selected. As the control is selected, the control's border thickness and radius are loaded into text fields in the properties portion of the UI at right 2. The user selects the desired border thickness and border radius as desired in the appropriate slider. 3. The user views the updated diagram with the control borders changes
Exceptions:	N/A
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	May be used many times per editing session
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

Use Case 2.18: Save Work

Use-Case:	Save Work
Primary Actor:	UI Designer
Goal in Context:	User wishes to save the diagram they are working on to the server
Preconditions:	The user is logged in and has at least one wireframe diagram that they own with at least one control in the diagram.
Trigger:	User has made an edit since last saving
Scenario:	1. The user is viewing a wireframe they own on the Edit Screen where at least one edit has been made since the last time saved 2. The user presses the Save button, which saves the diagram to the online database 3. The user continues working on the diagram.
Exceptions:	N/A
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	Used every time the user edits a diagram they may save their diagram
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

Use Case 2.19: Close Work/Go Home

Use-Case:	Close Work/Go Home
Primary Actor:	UI Designer
Goal in Context:	User wishes to stop editing the current diagram and go back to the Home Screen
Preconditions:	The user is logged in and is editing a diagram
Trigger:	The user either creates a new wireframe (which has a default size that can be changed) or selects an existing wireframe to edit. Note that once the user is editing a diagram they may delete as many controls as they wish.
Scenario:	1. User is editing a diagram 2. User saves the diagram after an edit 3. User presses close diagram button, which closes the current diagram and forwards the user to the Home Screen 4. User views Home Screen, which would have most recent diagram on top
Exceptions:	Should the user request to close without saving the user should be prompted to see if they wish to save their current diagram first
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	Used every time the user wishes to work on a diagram they will eventually close it
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

Use Case 2.20: Delete Diagram

Use-Case:	Delete Diagram
Primary Actor:	UI Designer
Goal in Context:	User wishes to delete a diagram from the system
Preconditions:	The user is logged in and has at least one existing diagram
Trigger:	The user has navigated to the Home Screen and can see the list of their diagrams with an “X” next to each one
Scenario:	1. User is viewing the list of their diagrams on the Home Screen 2. User presses the “X” button next to a diagram, this should open a dialog to confirm that they actually wish to delete the diagram 3. User presses Yes to delete diagram, 4. User views remaining diagrams on Home Screen
Exceptions:	Should the user press “No” when the dialog prompts them the diagram should not be deleted and the user should be returned to the Home Screen without any changes to the diagram list.
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	Deleting a diagram would likely be uncommon but could only happen at a maximum of once per diagram.
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

Use Case 2.21: Test Database

Use-Case:	Test Database
Primary Actor:	Administrator
Goal in Context:	Administrator wishes to test CRUD operations for data in the datastore
Preconditions:	Cloud database project has been created and all suitable credentials have been setup
Trigger:	User has logged in as an administrator and gone to the /databaseTester routing
Scenario:	1. User is viewing the databaseTester screen 2. User presses suitable button to perform appropriate test 3. User views updates to database in datastore console
Exceptions:	N/A
Priority:	Essential, must be implemented
When available:	Completed Project
Frequency of use:	Used every time the user wishes to do work
Open Issues:	Size, location, and style of UI screens and controls should be finalized by UI designer

i. Hardware Interfaces

The application should be runnable in any browser, though is best used with a PC as a mouse and keyboard are necessary for some editing.

ii. Software Interfaces

Wireframer will be developed using a React/Redux/Firebase stack. Note that one should use all the appropriate approaches to design and implement high quality software such that additional diagram components can be added over time.

iii. Communications Interfaces

There will be no networking requirements, just reading and writing to the Firestore.

iv. Memory Constraints

This application uses a manageable amount of user provided data so this should not be a concern.

v. Operations

Note that the user will want to take a Screen Shot of diagrams in order to use them in their own report. We will leave this to browser and Operating System functionality, like Print Screen in Windows computers. Our app will not itself provide this feature.

vi. Site Adaptation Requirements

N/A

b. Product functions

N/A.

c. User characteristics

The editor should aim to be as user friendly as possible, using the principles of foolproof design as well as sound UI design principles.

d. Constraints

N/A

e. Assumptions and dependencies

N/A

f. Apportioning of the Requirements

N/A

3 Specific requirements

The ***Wireframer*** application will require a number of user interface screens including:

- Login Screen
- Register Screen
- Home Screen
- Edit Screen

Note that the navbar will provide navigation features between these screen as needed.

3.1 **External interfaces**

The following wireframe mockups provide a look at the types of controls and layout to be used for the User Interface. Note that the User Interface designer should select the appropriate fonts for all text and should carefully choose color and font combinations that provide good contrast and attract the eye. There are four UI diagrams, one for each Screen. Note that the User Interface designer should also consider additional dialogs for providing adequate feedback to the user.

Figure 3.1 Login Screen

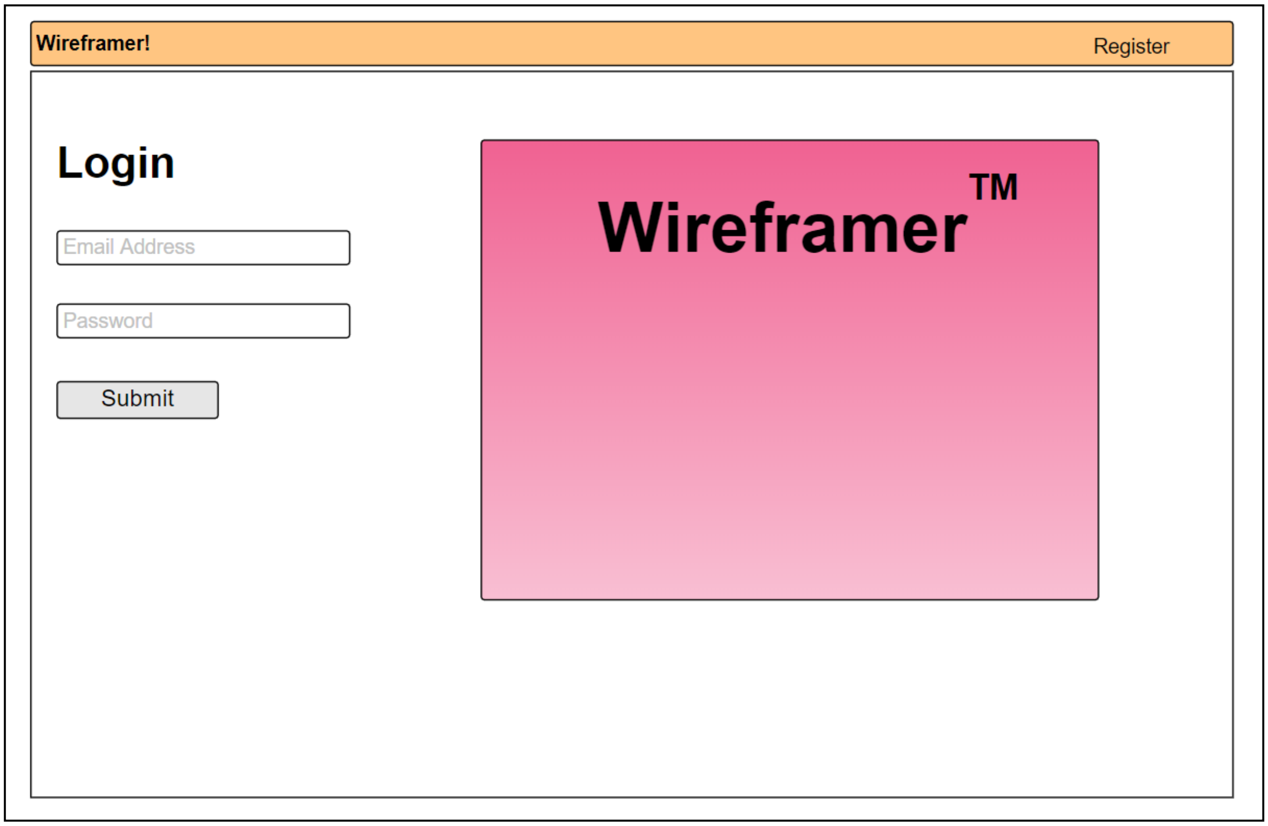


Figure 3.2 Home Screen

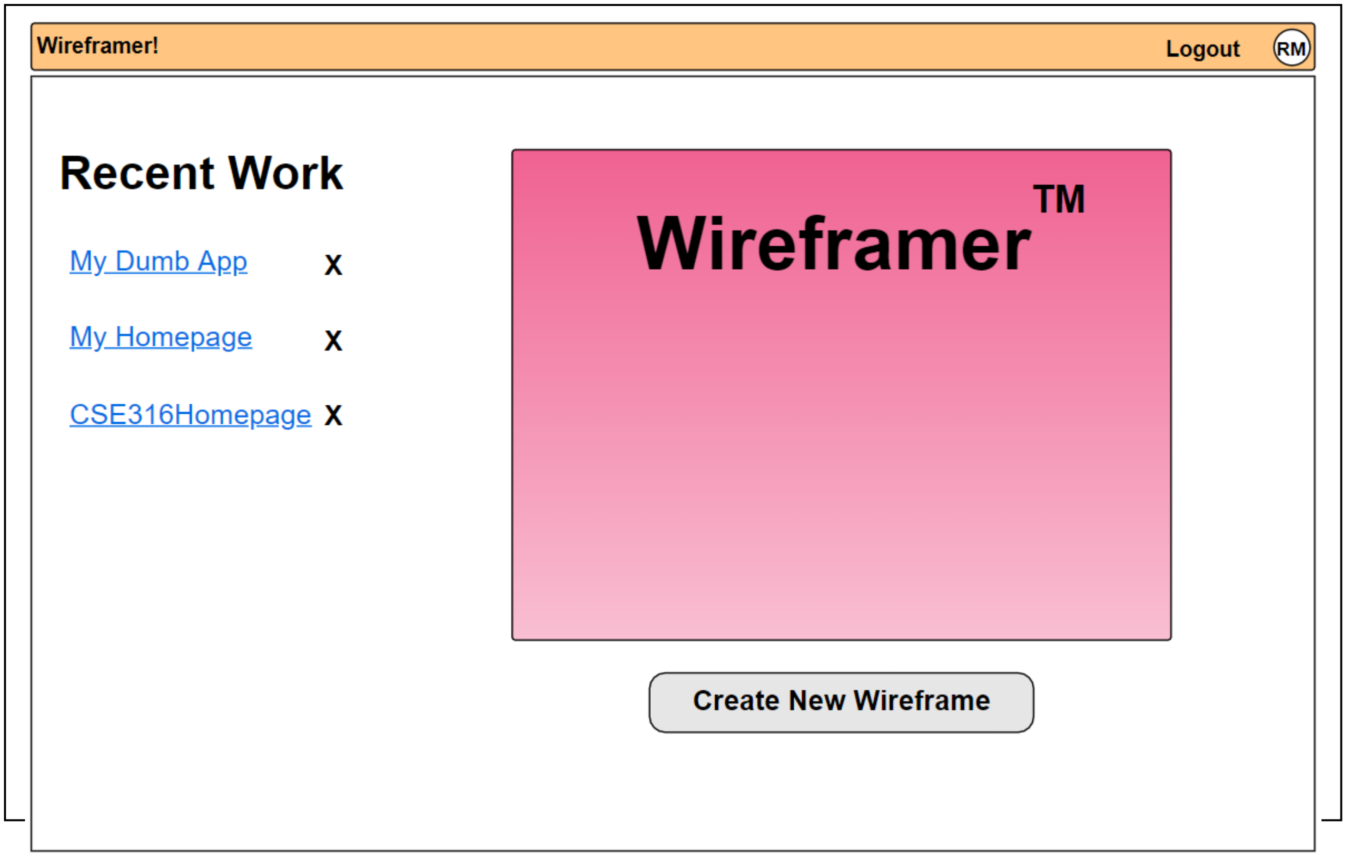


Figure 3.3 Register Screen

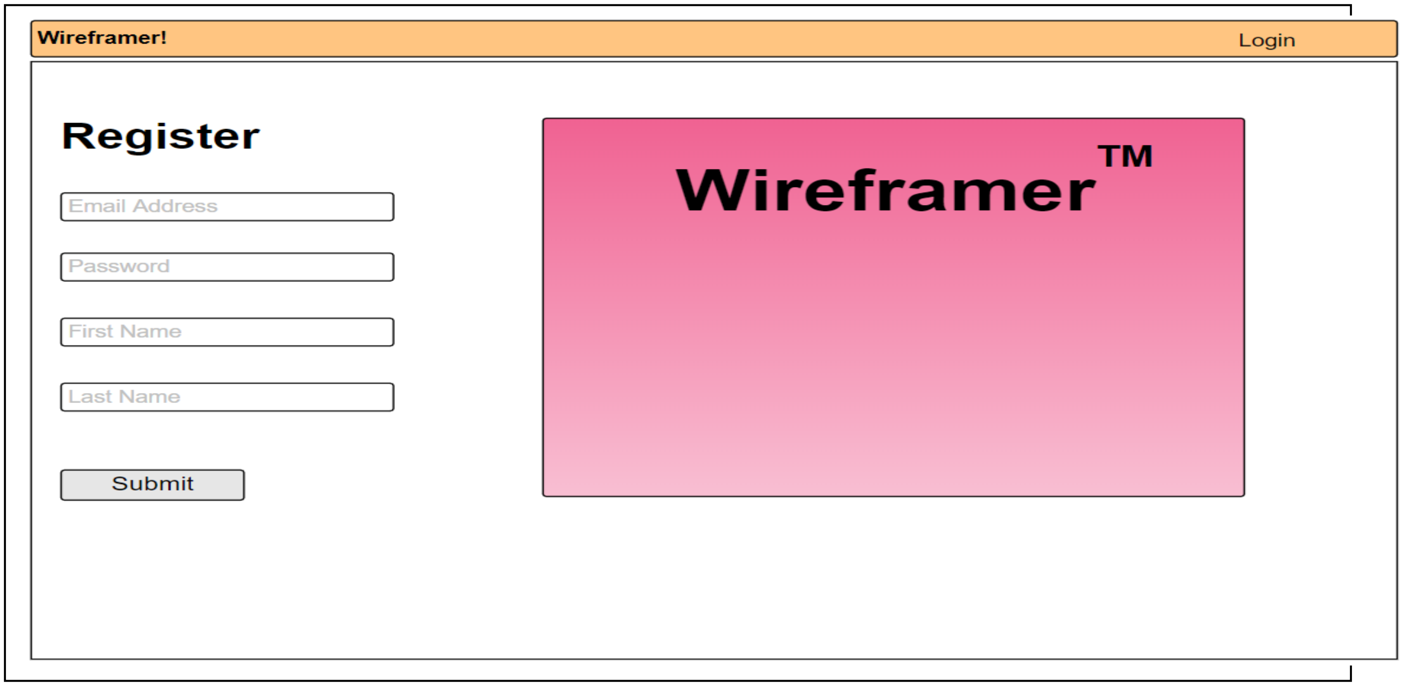
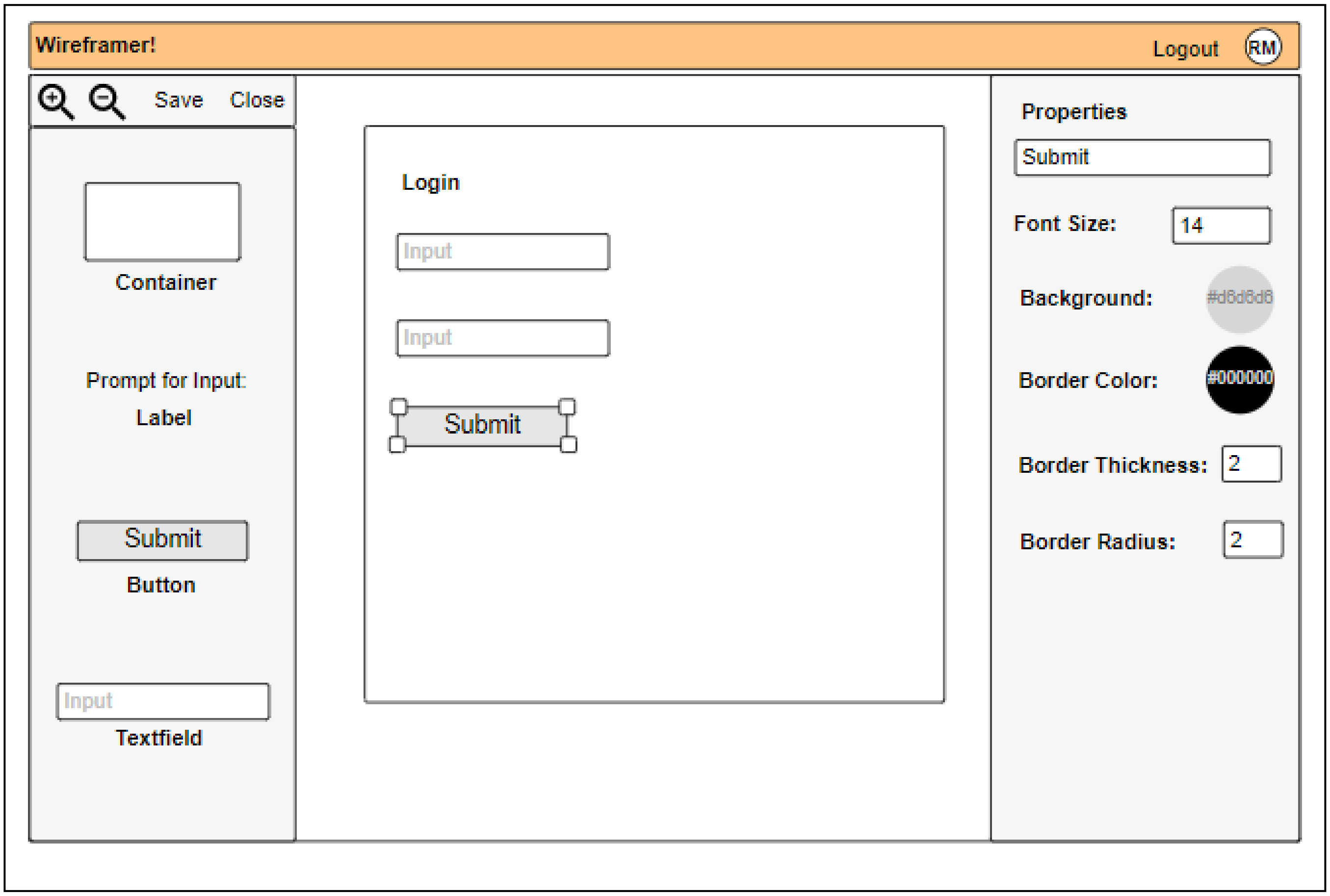


Figure 3.4 Edit Screen



Submit

Font Size: 14

Background: #d6d6d6

Border Color: #000000

Border Thickness: 2

Border Radius: 2

3.2 **Functions**

One of the important things to consider in our application is providing the appropriate feedback to the user. Users need feedback to enjoy their experience. This is typically done with visual cues like dialog boxes.

3.3 **Performance requirements**

N/A

3.4 **Logical database requirements**

N/A

3.5 **Design constraints**

JavaScript is the default language of the Web and so will provide all front-end functionality. This will be done using the React front-end framework.

3.6 **Software system attributes**

As professionals, all members of this project must take this project seriously. We are dedicated to producing robust software that exceeds the expectations of our customers. In order to achieve this level of quality, we should build a product with the following properties in mind:

3.6.1 Reliability – The program should be carefully planned, constructed and tested such that it behaves flawlessly for the end user. Bugs, including rendering problems, are unacceptable. In order to minimize these problems, all software will be carefully designed using UML diagrams and a Design to Test approach should be used for the Implementation Stage.

3.6.2 Availability –Customers may download and install the application for free.

3.6.3 Security – All security mechanisms will be addressed by future revisions

3.6.4 Extensibility – It is possible that more JavaScript/JSON widgets might be added to course sites in the future, so by providing an additional tab with suitable data and making changes to exporting methods, this should be considered during this deaign.

1.6.5 Portability – To start with, the app will target desktop Java applications.

3.6.6 Maintainability – Update mechanisms will be addressed by future revisions.

3.7 **Organizing the specific requirements**

Note that the application is simple enough that we need not worry about using an alternative arrangement of the content of this document. The specific requirements for this application already fit neatly into the sections listed in the IEEE’s recommended SRS format.

3.8 Additional comments

It is important to keep in mind that the UI designers and instructors should make updates to the themes and content as need to make something that looks great. It will be to their discretion to design all the interface controls in an effective, interactive style.

4 Supporting Information

Note that this document should serve as a reference for the designers and coders in the future stages of the development process, so we’ll provide a table of contents to help quickly find important sections.

4.1 Table of contents

4.2 **Appendixes**

N/A

Debugging
Enterprises