|  |
| --- |
|  |
| Project Design |
| CSCI 4805-10 Spring 2024 |

|  |
| --- |
| 2-22-2024 |

Table of Contents

[Project Name 3](#_Toc159929854)

[Team Member Names 3](#_Toc159929855)

[Abstract 3](#_Toc159929856)

[Tools & Technologies 3](#_Toc159929857)

[Requirements List 4](#_Toc159929858)

[Onboarding 4](#_Toc159929859)

[1. Sign-in Form 4](#_Toc159929860)

[2. Forgot Password Form 4](#_Toc159929861)

[3. Registration Form 4](#_Toc159929862)

[Dashboard 5](#_Toc159929863)

[1. Dashboard Page 5](#_Toc159929864)

[2. Create a New Project Page 5](#_Toc159929865)

[3. Project’s Name Dashboard Page 6](#_Toc159929866)

[4. Project Page 8](#_Toc159929867)

[5. Library 8](#_Toc159929868)

[6. Completed Projects 9](#_Toc159929869)

[7. User Profile 9](#_Toc159929870)

[General 10](#_Toc159929871)

[1. Dialog Box 10](#_Toc159929872)

[2. Filter Options 10](#_Toc159929873)

[3. Image Upload Options 10](#_Toc159929874)

[4. Project Icon 10](#_Toc159929875)

[5. Mobile Support 10](#_Toc159929876)

[TimeLine 11](#_Toc159929877)

[Design Description 12](#_Toc159929878)

[Major Components of system 12](#_Toc159929879)

[Components in depth 12](#_Toc159929880)

[User Interface Storyboard 14](#_Toc159929881)

[Entity Relation Diagram 15](#_Toc159929882)

[Appendix 16](#_Toc159929883)

[Block Diagram 16](#_Toc159929884)

[Component Diagram 17](#_Toc159929885)

[User Interface Storyboard 18](#_Toc159929886)

[Message Documentation 20](#_Toc159929887)

[Storage Documentation 21](#_Toc159929888)

# Project Name

Hobby Pro.

# Team Member Names

Candelario Aguilar Torres

Yazmin Alvarado

Dustin Bailey

Samuel Fletcher

# Abstract

Hobby Pro is a web-based project management application created for individual hobbyists. Projects can be created and broken up into stages. Users can dedicate tasks to be completed for each stage as well as add any associated notes and images. Hobby Pro aims to streamline tracking and project organization by providing a centralized space for users to efficiently manage all aspects of their hobbies.

# Tools & Technologies

HTML

CSS

JavaScript

Vue.JS

Firebase

# Requirements List

## Onboarding

### Sign-in Form

* 1. The user will be given a screen that welcomes the user and present the following:
     1. A username input textbox
     2. A password input textbox
        1. The username must match any current usernames within the system.
           1. If the username input does not match any current username’s registered in the system, it will be notified that the username is not valid.
           2. Else the username will be accepted.
        2. The password must match with the password under the specified username in the system. If the password does not match what is registered in the system, an invalid credentials notification will be shown to the user.
  2. The user will see under the textboxes two buttons:
     1. Submit button
     2. Forgot password button
        1. Pressing the submit button will check if the inputted info matches any system user info
           1. If the user's info is valid, it will take the user to the home page signed in under that user
           2. Else it will remain on the page and ask the user for valid info.
        2. Pressing the Forgot Password button will take the user to the Forgot Password Form

### Forgot Password Form

* 1. The user will be prompted to input an email address
     1. The user is given the option to submit button
        1. Pressing submit sends an email with a link to enter a new password.
     2. A link is available that takes the user back to the “sign-in form”

### Registration Form

* 1. The user Will be given a screen that welcomes the user and asks for the following:
     1. First Name input textbox
     2. Last Name input textbox
     3. Email input textbox
     4. Username input textbox
     5. Password input textbox
     6. Retype Password input textbox
  2. The first and last name Text boxes must contain alphabetic letters only
  3. Username must be a valid unique name not matching one already in the system.
     1. If the username matches one in the system, the system will ask the user to change it.
     2. Else it will verify that the username is not taken
  4. Password must be unique containing:
     1. Uppercase letter: “A,” “B” “C,” etc...
     2. Lowercase letter: “a,” “b,” “c,” etc...
     3. A Number: “1”, “2”, “3”, etc...
     4. Be 8 to 12 characters long.
        1. If the password does not follow any of the restrictions above, it is considered invalid.
        2. Else it is valid.

## Dashboard

### Dashboard Page

* 1. When a user logs in they will be taken to the Dashboard Page where they will see
     1. Basic Page Layout
        1. A side bar that will be at the left side of all pages and contain links to the following pages.
           1. Dashboard
           2. Projects
           3. Library
           4. Completed Projects
        2. A navbar at the top of all pages and containing:
           1. User Profile
           2. Username
           3. Search Bar
           4. Hobby Pro Logo
        3. Add New Project Button
           1. A floating action button that will redirect the user to create a new project on the New Project Page
     2. List of Starred Projects
        1. Project Icon(s)
           1. See general for basic layout
        2. The area will be empty if no projects are starred
     3. Filter Option
        1. A Sort and Filter option with general sort/filtering options. See General section.

### Create a New Project Page

* 1. A form will prompt users to enter information for the project that include the following
     1. Project Name
     2. Date
        1. Current Date is set as default in the form
        2. User can change the date
           1. Date cannot be set further ahead than the current date
     3. Tags
        1. A search bar that populates similar tag names as the user is typing
           1. User can select at least one or more tag
     4. Deadline
        1. If user selects deadline check box, user can select a date to set
           1. Deadline is added to Main Dashboard Page and Project Page
        2. If not selected, no changes occur
     5. Text Area
        1. A text area where user can add Notes or a Description for the Project
     6. Attachment box
        1. Users can add attachments to their Projects.
        2. Added attachments will appear in the under the Overview in the Project Name’s Page
  2. Bottom of the form will have a Continue and Cancel Button
     1. If the user clicks Continue the project is added to the Project Page and the Main Dashboard Page
        1. The user is then redirected to Project Page of the newly created project
     2. If the user clicks Cancel the user is redirected to the Main Dashboard Page

### Project’s Name Dashboard Page

* 1. Basic Layout users will see for every Project created
     1. Project’s Name Dashboard that consists of “sections” that the user can select that include
        1. Overview section which will have the following
           1. Project Name
           2. Start Date
           3. Deadline (if added) with days left
           4. Last Updated
           5. Tag Name(s)
           6. Total Time
           7. Progress
           8. Notes (if added)
        2. Stage(s)
           1. The project dashboard will list the current stages of the selected project.

Three empty stages are automatically created

* + - * 1. A drop-down menu available next to stage name that includes

Add new stage

A new stage will appear Project’s Name Dashboard

Add an images section

A page is added for user to add images to their project

See general for upload formats

A drop-down menu to give the user the option to rename, view, or delete the image

Rename stage

The selected stage will be renamed.

Duplicate stage

A duplicated stage will appear on the Project’s Name Dashboard

Delete stage

Selected stage will be deleted form the Project’s Name Dashboard

* + - * 1. Users can add Kanban tasks, Notes, and Images within this section

Kanban board with To Do, Doing and Done titles

If there are no tasks, nothing will be displayed

User can add tasks to the kanban board

Pop up block will appear asking user to enter title and description of task

User can move task between To Do, Doing and Done area

A textbox form where user can add notes

Images can be uploaded when the user selects to add a new image section

* + - * 1. Completed Check Box next to each stage name

If checked, progress is updated in the Overview tab and the Main Page

If all stages check boxes are selected

Message Alert appears and prompts the user to add optional items:

Final notes to a textbox

If notes are added, then it will be displayed in the Overview Layout of the Completed Project Page

Nothing will be displayed if left blank

Images of the completed project

See general for upload formats

Nothing will be displayed if left blank

Project removed from the active project list and moved to the Completed Project Page

* + - 1. Time
         1. Users can add time spent on a project in minutes or hours by selecting a button

Time entered is updated on the Project Dashboard

* + - 1. Settings
         1. Users can edit information originally entered from the Create a New Project Page.

Project Name

Project Date

Deadline

Total Time

Notes

Tags

Remove or add more Tags

* + - * 1. The default image that appears in the Project Icon can be changed here

A button the user clicks to open their files and select the image of their choosing.

A button is available to revert to the default image

* + - * 1. Delete the whole Project

A button for the user to select to delete the project from Hobby Pro

A general dialog box will be presented if the user selects to delete the project. See General Section

### Project Page

1. The Project Page will have the same basic page layout as the Dashboard Page
   * 1. Basic Page Layout
        1. A side bar that will be at the left side of all pages and contain links to the following pages.
           1. Dashboard
           2. Projects
           3. Library
           4. Completed Projects
        2. A navbar at the top of all pages and containing:
           1. User Profile
           2. Username
           3. Search Bar
           4. Hobby Pro Logo
        3. Add New Project Button
           1. A floating action button that will redirect the user to create a new project on the New Project Page
     2. List of Projects
        1. Project Icon(s)
           1. See general for the basic layout
        2. Area will be empty if no projects are created
     3. Filter Option
        1. A Sort and Filter option with general sort/filtering options. See General section.

### Library

* 1. The user's personal library on Hobby Pro. The Library Page will have the same basic page layout as the Dashboard and Project Page
     1. The library will have an upload button that will open the computer's file system to allow the following files.
        1. PNG
        2. JPEG
        3. PDF
     2. After the user selects the file under the correct format options, the user’s file will be uploaded.
     3. Once the file is added to the library page the user can view the file by clicking it which then will open the file in another browser tab
  2. The library will have filtering options that allows file to be sorted by the general filtering options
  3. The library will have buttons beside each file that will give the options:
     1. Delete
        1. A general dialog box will be presented if the user selects Delete. See General section.
     2. Rename
        1. A textbox dialogue box opens for the user to rename the given file
           1. The user can enter a new name and then have the option of hitting the confirm or cancel button.
           2. If confirm is hit will close the dialogue box and change the name of the project
           3. If cancel is hit it will close the dialogue box and nothing is changed.
     3. Copy
     4. Paste

### Completed Projects

* 1. The user has access to all their completed projects, if they have not completed a project then nothing will be displayed.
     1. Completed projects will be sorted by the general filtering options:
  2. Users can view completed projects. An icon layout where users can select will include:
     1. Photo of Completed Project
     2. Project Name
     3. Date Range
        1. Start Date to End Date
     4. “Lesson Learned” Notes
  3. When the icon is selected, the project dashboard appears for the user to browse through but not edit.

### User Profile

* 1. The profile will display information pertaining to the user
     1. First and Last Name
     2. Username
     3. Email
  2. Users will have the ability to edit their profile
     1. Change password
        1. A button the user can select to change their password
           1. The new password will follow the same guidelines set in the Registration form
     2. Change user icon
        1. A button the user can select to change the “blank user” image

## General

### Dialog Box

* 1. Will prompt user-associated message and present two options:
     1. OK
     2. Cancel
  2. If OK is clicked, then it will complete the associated process
  3. If Cancel is clicked, then it will cancel the associated process

### Filter Options

* 1. Available filtering options include but are not limited to:
     1. Recent
     2. Date Range
     3. Alphabetical ascending
     4. Alphabetical descending
     5. Numerical ascending
     6. Numerical descending

### Image Upload Options

* 1. A dialog opens and will display the following format options to upload:
     1. JPEG
     2. PNG

### Project Icon

* 1. The basic layout (if available to the specific project) will consist of
     1. Default image
     2. Project Name
     3. Date started
     4. Last Updated Date
     5. Deadline Date with days left
     6. Tag Name(s)
     7. Progress Bar Visualization
     8. Star button
        1. If selected project will appear in Dashboard Page

### Mobile Support

* 1. The web application will be able to be resizable to fit on screens for mobile use.

# TimeLine

|  |  |  |  |
| --- | --- | --- | --- |
| Week | Date | Deadlines | |
| 1 | Jan. 22 | Project Ideas. Start on Project Proposal | |
| 2 | Jan. 29 | Completed Project Proposal; finishing project requirements | |
| 3 | Feb. 5 | All team members- refining project requirements | |
| 4 | Feb. 12 | Technology Familiarization and Project Design components | |
| 5 | Feb. 19 | Technology Familiarization and Project Design components diagram  Yazmin – UI design  Sam – Vue.js/Firebase datastore familiarization  Dustin – Vue.js/Firebase datastore familiarization  Candelario – Vue.js/Firebase datastore familiarization | |
| 6 | Feb. 26 | Project Design Components and Diagrams. Presentation | |
| 7 | Mar. 4 | Yazmin – UI/ Design and Vues.js/Firebase datastore familiarization  Sam – Build project skeleton with routing to appropriate pages | Dustin – database design and implementation  Candelario – database design and implementation |
| 8 | Mar. 11 | Yazmin –New Project Page  Sam – Project Dashboard Page | Dustin – database design and implementation  Candelario – Authorization |
| 9 | Mar. 18 | Yazmin – Main Dashboard Page  Sam – Onboarding Pages | Dustin – Leftover database design and implementation  Candelario – Authorization |
| 10 | Mar. 25 | Yazmin – Main Dashboard Page  Sam – Onboarding Pages | Dustin – Functions Call to database  Candelario – Functions Call to database |
| 11 | Apr. 1 | Yazmin – Library  Sam – User Profile | Dustin – Completed Projects Pages  Candelario – User and General Settings Pages |
| 12 | Apr. 8 | Yazmin – Library  Sam – User Profile | Dustin – Completed Projects Pages  Candelario – User and General Settings Pages |
| 13 | Apr. 15 | All – Preparing and presenting for Innovation Experience | |
| 14 | Apr. 22 | All – Final Testing | |
| 15 | Apr. 29 | All – Final Testing | |

# Design Description

## Major Components of system

Hobby Pro’s architecture takes a Model, View, Controller approach, where the View is managed by our team, and the Model, and Controller are handled by the cloud provider Firebase. Utilizing Vue.js as the framework for user interface, calls to Firebase will be made to create, read, update, delete, and authenticate data, using Firebase API’s.

The user interface (View) will be constructed using Vue.js, a JavaScript web framework used to create dynamic web content. Using Vue, the user interface will be developed using a component-based model, where each component represents a section of the page (i.e., navigation bar, menus, footers.). Each component will be made up of HTML, CSS, and JavaScript, capable of making network requests, and updating the respective component, as necessary.

The Model and Controller will be managed by the cloud provider Firebase. Using the Firebase API’s, requests will be made to firebase to fetch, update, or post data to the Firebase Data Store, a NoSQL database technology used by Firebase.

The security aspect of the project will be implemented using Firebase Authentication. A tested library that follows industry standards and best practices to provide user authentication and authorization support. Hobby Pro will use username and password authentication as a sole means of authentication.

## Components in depth

Our components were thought out using our major components and the way our system would rely on user inputs on what pages and how the flow of data would go from server to database then back to the server on our website. So, starting off on the log in and sign-up pages. We are doing typical login using data values consisting of email and password fields and we are thinking of implementing google sign-in use. How this would interact with the firebase database is that the user's login credentials would be sent to the database and firebase begins a check to confirm that a: a user matching the require credentials is available in our system, and b: if one is not found it will dispense required errors.

Continuing the next component was the signup page. The data values for this page were first name, last name, email address and password. Once the user entered the required information a call to the server using this required information would be sent to the database and firebase would check that the email used to create said account does not already exist, if this is the case an error is thrown to let the user known that an account already exist with that email address. If not, then the user’s information will be saved in the database and a token will be saved on the computer to act like a pass to allow access by that user to enter the site.

The last of onboarding required components we land on forgot password; this one was simple as all we would need of data values from the user is an email address matching a current user that just forgot their password. Once the email had been entered the user entered email would be checked by the firebase database system to see if said user has an account with them. If so, an account recovery email would be sent with a new password the user can use for now to enter the site. If not, the user will be notified that the user has entered an invalid email address.

We move on to dashboard required components. For this we came up with a few matchings out current requirements doc. Well, start with the main Dashboard, the only data value we thought up was the projects list since the main dashboard will present all currently made projects in the middle of the page. We also have some functions for the main dashboard including get\_project() which will get projects upon initial page load. Another method is create\_project() which will allow the user to create a new project which will redirect them to create projects page. And finally, a remove\_project(int project ID) method in which a project\_ID is taken which will make sure to ask the user for confirmation before deleting the project from the list of projects available on the main dashboard page.

The main dashboard along with a few extra pages have Filtering options they include filter Project by name ascending and descending, filter project by number ascending and descending, filter project by recent ascending and descending these methods will filter the projects by their respective restrictions for example filter by name would take all projects and looking at the projects Name and alphabetically sort them in ascending or descending order.

Next is the Create Project Page. On this page, it will act as a form in which the user must fill in information; the following data values were thought up from the requirements. Project Name, Project Date, a list of defining tags, project deadline, and finally a project description. The methods from this that seemed valid were a create\_project() method, an overview method, getProjects by completed, and filtering options for projects. that took all values and created a project to then be put into the list of current projects when it is redirected to the projects page for the new project.

The next component is the Project Page which consists of the data values of Project Name, progress, project Tags, totalTime, projectDescription, and Stages. Methods in this component consist of overview() which shows a page of an overview of the project information, add\_stages() which adds stages to the list of stages available. We also have a method for remove\_stages() which would remove a stage from the list of stages, another is edit\_stages() which allows manipulation of stage names and stage descriptions. A few more are editProjectName() allows you to rename a select project name and get\_stages() gets all stage information upon page reload and initial load. getProgress() which shows the total progress made on the project shown as a progress bar on the page. GetStageTimes() will get the total time spent on a specified stage of a project.

After this is the completed Projects page, which is the list of projects that are labeled in the database as being completed. On this page, it will run through the list on load to sort the list of projects on a Boolean variable that defines each project as completed. It has the get\_projects method, a create projects method, and an overview method the getProject method gets the list of projects while looking for said projects that the complete Boolean of the project is true. The overview method shows a page of an overview of the project's information. Create project runs like the other create project buttons. This page will also have the filter methods as well.

Then, we have the component called library which will take files as a data value and has three methods: get files, add files, rename files and remove files. add\_files method will open the users file explorer on their computer and allow a file to be transferred into the website's library of available files system only allowing predetermined file types described in the Requirements above. Remove\_file just allows you to remove a select file from its library system. And finally get\_file loads in the files in the library on loads and reloads. Rename Files allows the user to rename a file name present in the library. Will also have the Filter methods

Finally, is the user profile. This page has 4 data values that are email total projects total projects completed and total projects not completed. This component page will have one method that is new password, which will allow the user to make a new password.

## User Interface Storyboard

The user interface storyboard starts at the Login Page. A user can log in, register, or reset their password. The design of these pages was kept simple, aligning with what most users see on other platform log-in pages. (See PNG of LoginPage ->LoginErrorPage ->Forgot Password Page ->Registration Form Page for higher quality.)

Once a valid user logs in, they will be directed to the dashboard page. Here users can move through the main pages of Hobby Pro. Initially, all pages will be empty if the user is logging in for the first time. Originally, the first concept design began with a navigation bar at the top of the page. However, when researching similar platform layouts and designs, it was determined that a mixed navigation bar at the left and top would better suit our needs. It was also established that the availability to create a new project from any of the four pages was required. Which further pushed the need for a mixed navigation bar design. (See PNG of Dashboard -> Project-page ->Library page ->Completed Page for higher quality.)

All the main pages have the same basic layout, although the Library Page has an added component next to its heading. A drop-down menu is available for the user to upload the specified files listed on the Requirements Page. The next steps are intuitive, and the design will try to be comparable to the layout of similar cloud file storage platform designs. (See PNG of Library page ->Upload Library page ->Open file in Library ->FileSelection in Library for higher quality.)

From any of the main pages, a user icon and username are visible. A user can view their profile or log out from Hobby Pro when a drop-down menu is selected next to the username. The user profile design was kept simple as well, once again aligning with what most users might see on other platforms. Here we display the information gathered from the registration form. (See PNG of User profile options ->User profile page for higher quality).

Creating a new project layout is straightforward, with the necessary components, project name, start date, deadline, tag names, etc. laid out in an eye-pleasing design. Once a user selects OK, the project is created, and the project icon is also created and populated to the Project Page. A user can toggle the star on the project icon to add it as their "starred"/"favorite". Only project icons that are starred appear on the Dashboard page. (See PNG of Create Project Page ->Set Deadline ->AddTags ->AddAttachements ->Project Icon for higher quality).

Original layout ideas for the Stage page in the Project's Name Dashboard had tasks, notes, and images squeezed into one stage page. This felt cluttered so instead we opted to separate images from tasks and notes. This translated to adding an image page under the selected project that could be added by the user using a drop-down menu.

## Entity Relation Diagram

Our database is a noSQL database. We are building our database using Google’s Cloud Firestore and Google’s cloud storage. Storage will be used to house all documents and images the project needs. Our database will have Five tables. These tables are Projects, Users, Tags, Stages, and Tasks.

All users are assigned a UserID at the time of account creation. This Id will come from the token returned by firebase auth when login is complete. Users will also have an array of file paths pointed to the Storage. The User will also have an array of ProjectIDs.

The second table is the Projects table. Each project will be assigned a unique ID at creation. Each project has a name and a text field. Each project will have a UserID. Each project will have a StartDate: this is a time/date data type. Each project will have an array of tags (more on tags later). Each project has an image associated with it; this will be a file path to the storage. Each project has a Boolean variable to keep track of if that project has been completed. Each project also has an array of StageIDs that will make up the data about the project

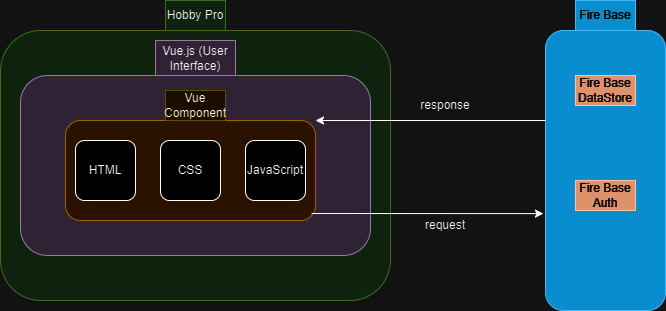
The third table is Stages. Stages are a small table containing information about stages. Each stage has a Stage ID, A completed Boolean, A time variable to summarize the time spent on the stage and an array of TaskID.

The fourth table is Tasks. Tasks are where the user-created information will be mostly. Tasks have an auto-generated TaskID, an array of file paths to the Storage that will contain images and other relevant documents for a project. There will be a notes section on each task that is Text String type. And a Boolean variable called Completed.

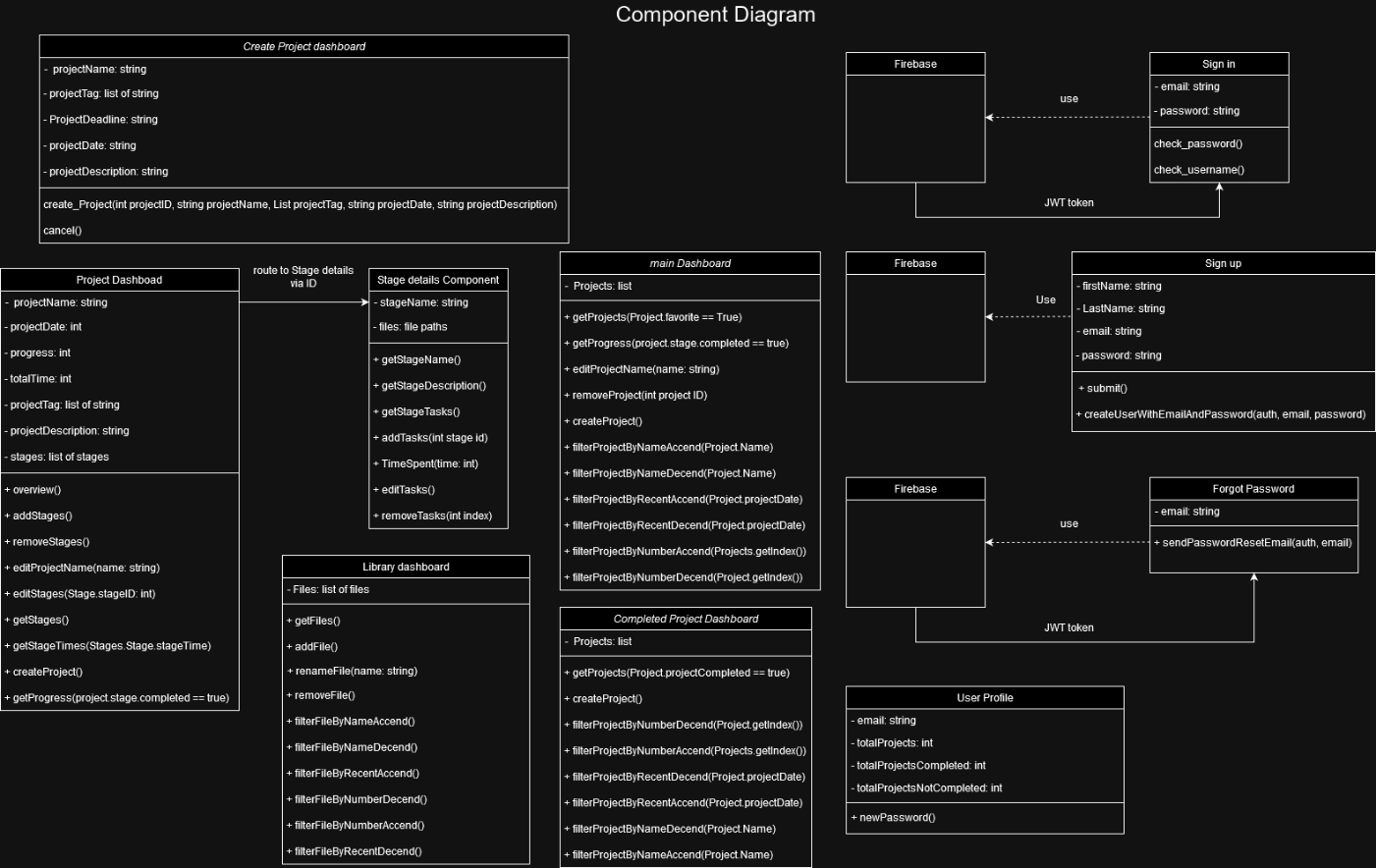
The fifth and final table is called Tags. Tags are our quarriable object. Tags will be a list of Tags.

# Appendix

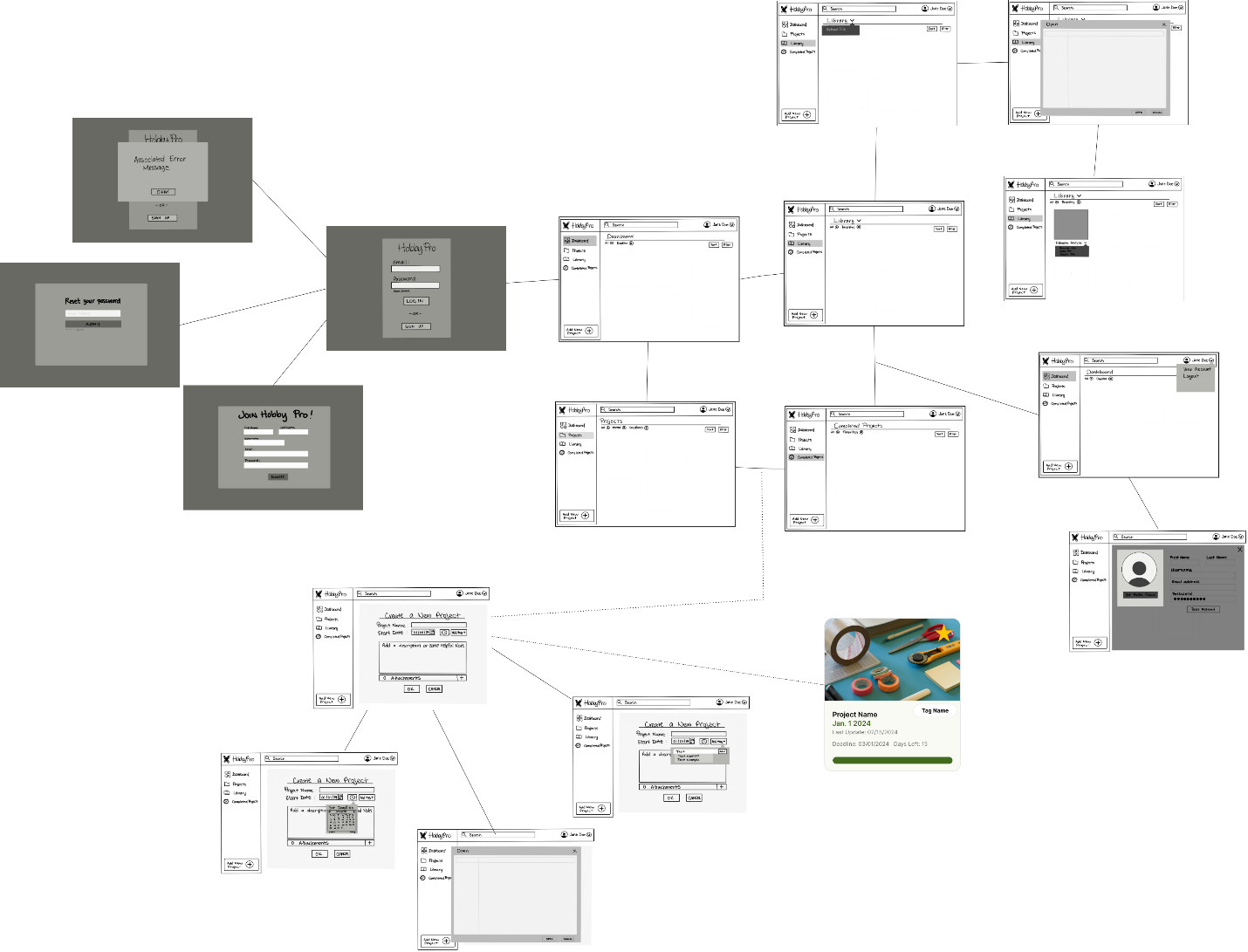
## Block Diagram

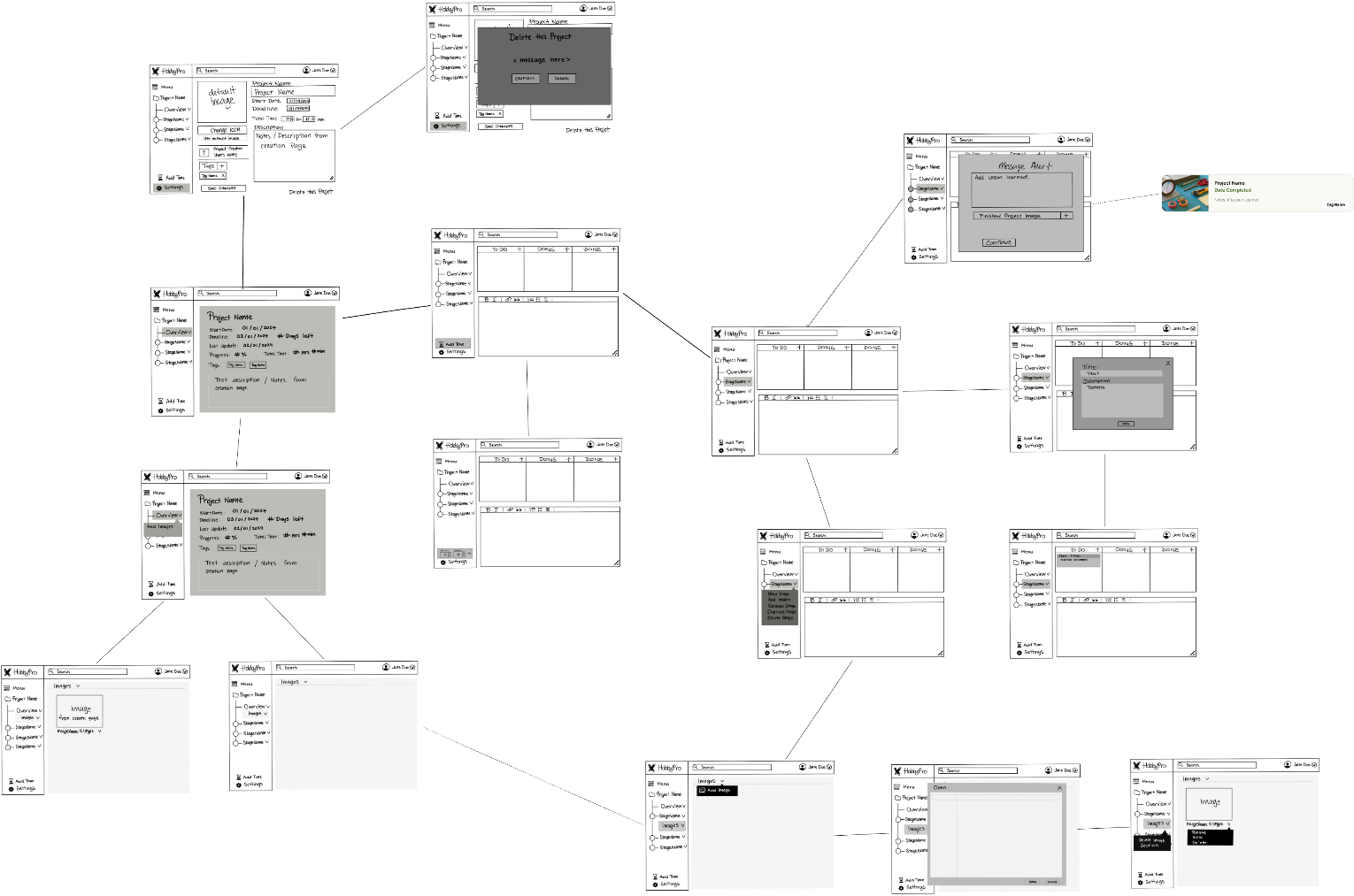


## Component Diagram



## User Interface Storyboard





## Message Documentation

## Storage Documentation