# Digital Pet Task App

# Team members

## Sarah Jorissen, Laura Lopez, Jonathon Hoffman

# Revised Requirements

1. Requirements list
2. Task List System
   1. Add a task
      1. When Clicked, will prompt a pop-up that asks the user to fill in:
         1. Task name
         2. Task description (if-applicable)
         3. Date and time that it is due
         4. Check box asking if task needs to be repeated
      2. Add a task window will have an “OK” button where inputted information will be saved onto the calendar and will close the add a task window
         1. Information saved onto the calendar will be stored onto the Firebase database
      3. Add a task window will have a “Cancel” button that when clicked by a user, will close the add a task window without saving onto the calendar
      4. Repeated Tasks
         1. Daily
         2. Weekly
         3. Custom
   2. View a task
      1. Tasks for current day will be displayed on home screen of application upon log-in on task list manager window
   3. Scroll through tasks
      1. Two arrows will be at the top of the task list system allowing users to scroll through dates, one day at a time, and see the tasks of that day
   4. Check off a task
      1. Tasks for current day that are displayed on the task list window will be able to be “checked-off” by user via checkbox, indicating completion of task
   5. View Calendar
      1. A symbol that’s on the top-right of the task list window, when clicked a calendar system will appear to user
3. Calendar System
   1. View Calendar
      1. After our “View Calendar” symbol was clicked a calendar will generate in a pop up window
         1. To the right of the current month will be an option for selecting a “Weekly” or “Monthly view”
      2. There will be an “X” button in the top-right hand corner of the window that will close the window when clicked.
   2. View Tasks on the calendar
      1. While the Calendar window is open, the user may be able to click on any date to view the tasks for that current date
      2. After clicking on a specific date, the task list corresponding to that date will generate
      3. There will be options to Add/Edit the tasks on the tasklist for the specific date the user clicked
   3. Edit Tasks on a specific date of the calendar
      1. When the user views a specific dates tasks, there will be an button in the top-right (location may change) that says “Edit Task”
         1. If the user clicks the “Edit Task” button, the Edit task window will open where the user can change:
            1. Task name
            2. Task description
            3. Date and Time its due
            4. Check box asking if the task is to be repeated
   4. Add Tasks on a specific date of the calendar
      1. When the user views a specific dates tasks, there will be an “Add Task” button in the top-right (location may change) where information can be added to the calendar and stored onto the Firebase database
         1. If the user clicks the “Add Task” button, the Add task window will open where the use can create a task with the following information:
            1. Task Name
            2. Task description
            3. Date and Time its due
            4. Check box asking if the task is to be repeated
4. Pet System
   1. The Pet View will display the pet and all of its current stats
      1. The Health Bar will display its current health mirroring the Firebase database status, and will change over time
         1. If the user completes tasks by their deadlines, the bar will fill little by little until it is full, simultaneously updating the Firebase database
         2. If the user misses a task deadline, the bar will deplete little by little until it is empty, simultaneously updating the Firebase database
            1. Once the bar is empty, it will trigger the death of a pet
      2. The pet’s name will be displayed over it
         1. An option to allow a Name Change will be displayed next to the name
      3. An option to open the customization menu will be displayed in the upper right-hand corner
   2. The pet will be displayed in the middle of the view, and will move around its area with basic animations
      1. The pet’s appearance will update each day based on its current state
         1. Its state will be determined by its current health and age
         2. The pet will have 3 Life Stages, mirroring the status stored in the Firebase database
            1. The pet will start as a baby after being created and named by the user
            2. After 2 days without death, the pet will grow into an adolescent
            3. After a week from being created, the pet will grow into an adult

There will be multiple different forms that the pet can randomly grow into once it reached adulthood

* 1. Pet Customization menu
     1. The user can select from a variety of accessories to display on the pet, which are stored in the Firebase database
        1. A baby pet can have at most 1 accessory on at a time
        2. An adolescent pet can have at most 2 accessories on at a time
        3. An adult pet can have at most 3 accessories on at a time
     2. The user can change the pet’s color
        1. Different colors can be unlocked over time upon consistently good care of the pet

1. Settings System
   1. Alert Options
      1. Users will be able to pick (via checkbox) while simultaneously updating the Firebase database when their pet alerts or reminds them them of:
         1. An upcoming task
         2. An upcoming expiration date
         3. Pet happiness/deterioration
      2. User will be able to confirm changes about their pet alerts or reminders when they click the “Save” button on the settings window
      3. User will be able to close out the settings window by clicking the “X” at the top of the settings window
2. Authentication System
   1. Logging in
      1. Users will be able to select the “Log in” button displayed on the homepage of the digital pet desk app
         1. Create an account
            1. Username

If a user does not already have an existing account, they will be able to create a username of their choosing, as long as it hasn’t been used for another account

* + - * 1. Passwords

Will be used in conjunction with username of their choice and will need to abide by these guidelines:

At least 8 characters long

At least 1 uppercase and 1 lowercase character

At least 1 number

* + - * 1. “Create Account” button

When hit after user enters credentials of their choice and follows password requirements, credentials will be stored on a database for future log-ins

If password requirements are not met, user will be prompted with an error message stating “Password does not meet requirements”

* + - 1. Logging into an existing account
         1. Users will be able to enter their existing username and password combination, which is stored onto the Firebase database to log into their pet desk application

Username and password combination will be validated by the database to confirm that user already exists

1. Updated Timeline
   1. Provide a weekly timeline that outlines the work that will be accomplished each week on the project from now until final presentation.

| **Week** | **Task** | **Member** |
| --- | --- | --- |
| Week 1  8/29 - 9/5 | 1. Set up tools, make sure everyone is on the same page 2. Sketch out basic UI for app, decide how to tie features together 3. Decide on design for pet & its different statuses | 1. ALL 2. ALL 3. ALL |
| Week 2  9/5 - 9/12 | 1. Requirements List 2. 15 Minute Presentation | 1. ALL 2. ALL |
| Week 3  9/12 - 9/19 | 1. Project design diagrams 2. Project design document | 1. ALL 2. ALL |
| Week 4  9/19 - 9/26 | 1. Begin authentication system 2. Begin task list development. 3. Decide on basic UI design for all parts to follow (colors, etc.) 4. Create basic art for pet to use for testing features 5. Implementation of adding and removing items from task list. 6. Calendar mock up | 1. Laura 2. Laura 3. ALL 4. Sarah 5. Laura 6. Jonathon |
| Week 5  9/26 - 10/3 | 1. Create UI of task list 2. Complete authentication system - Confirm password requirements, log into existing user, create new user 3. Begin settings UI window 4. View calendar pop up 5. Add tasks to calendar 6. Set different “conditions” that determine how the pet will look/respond & test these | 1. Laura 2. Laura 3. TBD 4. Jonathon 5. Jonathon 6. Sarah |
| Week 6  10/3 - 10/10 | 1. Create UI of task list 2. View task list implementation. Checking items on and off task list. 3. Remove tasks from calendar 4. Create UI for viewing individual tasks on calendar 5. Implement pet aging & changing over time intervals, depending on status | 1. Laura 2. Laura 3. TBD 4. Jonathon 5. Jonathon 6. Sarah |
| Week 7  10/10 - 10/17 | 1. Setting tasks at certain time intervals for calendar 2. Implement alert systems for tasks 3. Create UI for editing individual tasks on calendar 4. Make announcements to user when they are coming close to a due date for their task | 1. Laura 2. TBD 3. Jonathon 4. Sarah |
| Week 8  10/17 - 10/24 | 1. Adjustable calendar UI 2. Implement alert systems for tasks 3. Allow regeneration of pet upon the death of one 4. Create a settings window for application 5. Prepare for midterms | 1. Jonathon 2. TBD 3. Sarah 4. ALL 5. ALL |
| Week 9  10/24 - 10/31 | 1. Implement alert systems for tasks 2. Mid-semester meeting/Progress management (Longer than Scrum) | 1. ALL 2. TBD 3. ALL |
| Week 10  10/31 - 11/7 | 1. Confirm calendar and task list interaction is successful 2. Confirm task list and calendar are unaffected by pet death | 1. Jonathon 2. Laura |
| Week 11  11/7 - 11/14 | 1. Confirm pet is programmed to follow task list and date/time provided by the calendar 2. Program alerts to occur following settings checklist 3. Create poster for project (due the 14th) | 1. Sarah 2. TBD 3. ALL |
| Week 12  11/14 - 11/21 | 1. Implement notification system for upcoming tasks/dates in calendar 2. Possibly implement pet to overlay other windows other than desktop | 1. Jonathon 2. Sarah |
| Week 13  11/21 - 11/28 | 1. Complete any unfinished items 2. Thanksgiving Break | 1. ALL 2. ALL |
| Week 14  11/28 - 12/5 | 1. Complete any unfinished items | 1. ALL |
| Deadline  12/6, 1:30 - 3:30 PM | 1. Complete any unfinished items / Prepare for finals 2. Submit final report 3. Live demonstration and presentation 4. Submit all code to D2L and GitHub | 1. ALL 2. ALL 3. ALL 4. ALL |

# Design Description

This design description will cover the entire functionality of the application. This will be broken into sections to make it easier to follow and understand.

**The home page**

The home page will consist of several features which include: the “Login” button, the “Create” button, and finally a tasklist that contains a “+” symbol, the current date, and a calendar symbol.

**Login Window**

When the user clicks the “Login” button a small pop-up window will generate that prompts the user for their Username and Password. In this window, an additional “Login” button will be under the area where they are prompted for their username and password, taking the user to their account’s main page. Underneath the “Login” button will be a “Sign up!” button that will take the user to the signup page. There will also be an “X” symbol at the top right that will allow the user to close out of this window. After the user logs in, the main page will be the exact same but where the login button once was, will be their username. If the user fails to login with existing credentials, an error will generate on the login page stating to try again.

**Sign-up Window**

This window will generate when the user clicks the “Sign up!” button in the Login window. This window prompts the user for their email, username, and password, and to verify their password. At the bottom of these prompts will be a “Sign up” button that when clicked will take the user back to the Login window, allowing the user to log in with their newly signed up account. At the top, right-hand corner of the window will be an “X” symbol that allows the user to close the window. If the user tries to create an account with a faulty email, a username that’s already been taken, or attempts to create a password that does not follow the minimum requirements, they an error message will appear stating which field made the creation of their account fail.

**Pet Creation Window**

Back on the user’s home page, when they click the “create” button a small popup window will generate a pet egg and prompt the user to give the pet a name. Below this prompt will be a “Confirm” button that when clicked will take the user back to their home page that will look exactly the same except now their pet will be displayed on the right side of the screen with its name above it and its health bar below it. To view any pet customization available to the user they would click their pet and it will take them to the Pet Customization window.

**Pet Customization Window**

This window will contain the user’s pet’s name, a display of the user’s pet below the name, and below the display, two arrows pointing left and right with the pet's customization item’s name in between them, and finally a confirm button beneath that. When the user clicks on either arrow it will cycle through available customization options for the pet, and when the user likes their newly customized pet they can click the “Confirm” button to save all changes.

**Tasklist Window**

Back on the user's home page, we will be breaking down the task list and each of its buttons. The task list will be on the left side of the screen, and as previously stated, contain a “+” symbol for adding tasks, a calendar symbol for viewing the user's calendar, and the current date. When the user clicks the “+” button it will take the user to the add task window. There will also be, if there are any tasks in the current task list, an “edit” button next to each task that will take the user to the Edit Task Window. When the user clicks on the calendar button it will generate the user's monthly calendar.

**Add Task Window**

In this window, the user will be prompted with several text fields such as task name, description, date, and a dropdown list of different options for whether the task repeats daily, weekly, or monthly. There will also be a “Confirm” button that when clicked will take the user back to their homepage and adds that task to whatever selected dates tasklist. There will be an “X” symbol at the top right-hand corner of the window that will allow the user to close the window.

**Calendar Window**

Back on the user's homepage task list, there will be a calendar button that, when clicked, will generate the user's monthly calendar with the current day highlighted. The user can then select any day from the calendar and it will generate the selected day’s task list. This task list will contain the “+” symbol that allows the user to add tasks, an “X” symbol to close out of that task list, and finally, an “Edit” button located next to each of the tasks that will take the user to the Edit task Window. There will also be an “X” symbol to close out of the calendar when the user is done viewing/adding/editing their daily task lists.

**Edit Task Window**

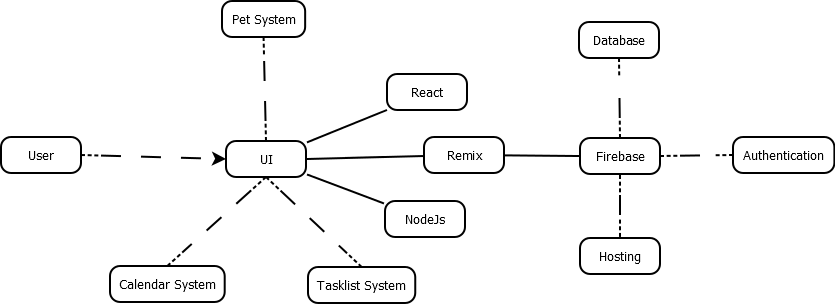
The edit task window will look exactly the same as our Add Task Window, however, each of the fields will fill with the task's original information. The user will be able to change any information in each of the fields and when they finish they can press the “Confirm” button to confirm all the changes. There will be an “X” symbol located in the top-right of the window that will allow the user to close out of the window.

**Setting Window**

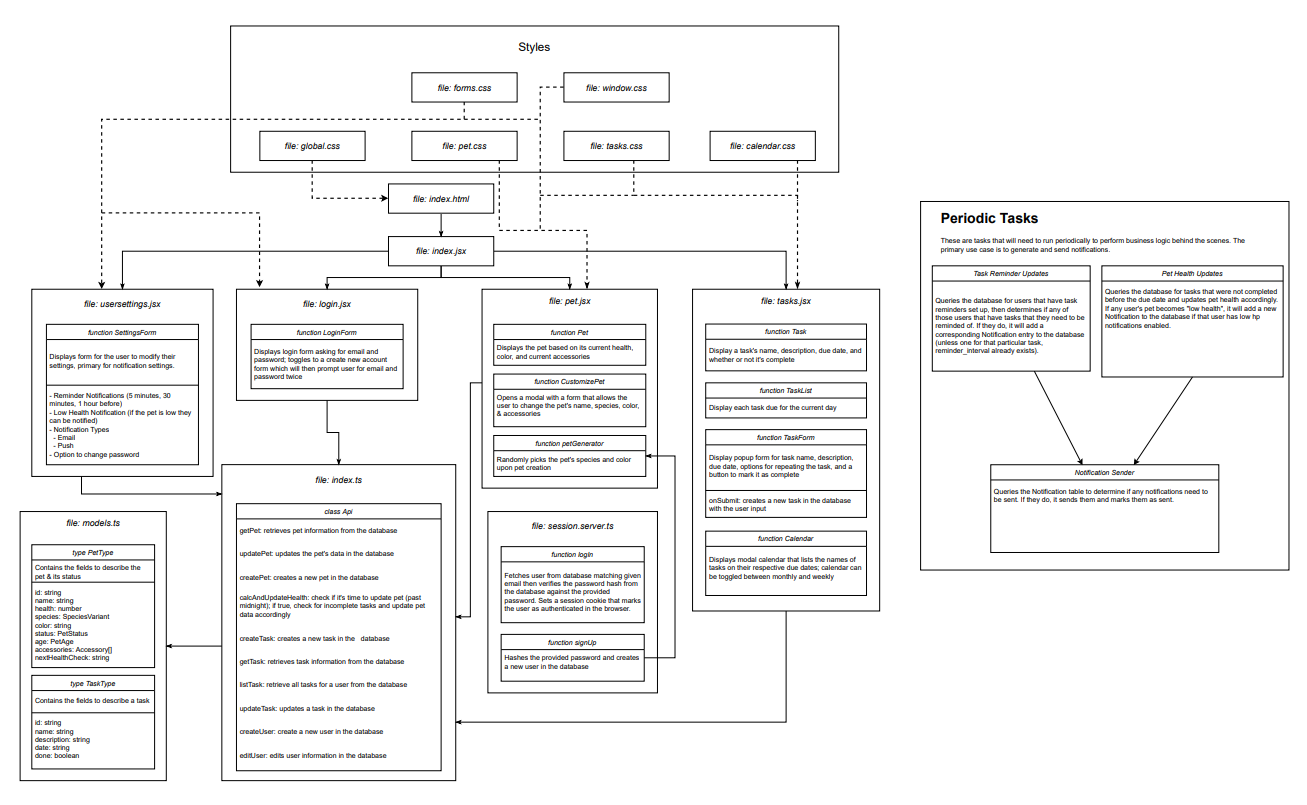
Back on the user's homepage, when the user wishes to set up what they get notified about regarding the application, they will need to go to their settings window. This is achieved by clicking on the user’s username. The settings window will contain several options for notifications and toggles for each of them. The user will be able to toggle on whichever notification they want to receive and then they will be able to select the “Confirm” button located at the bottom of the page.

# APPENDIX

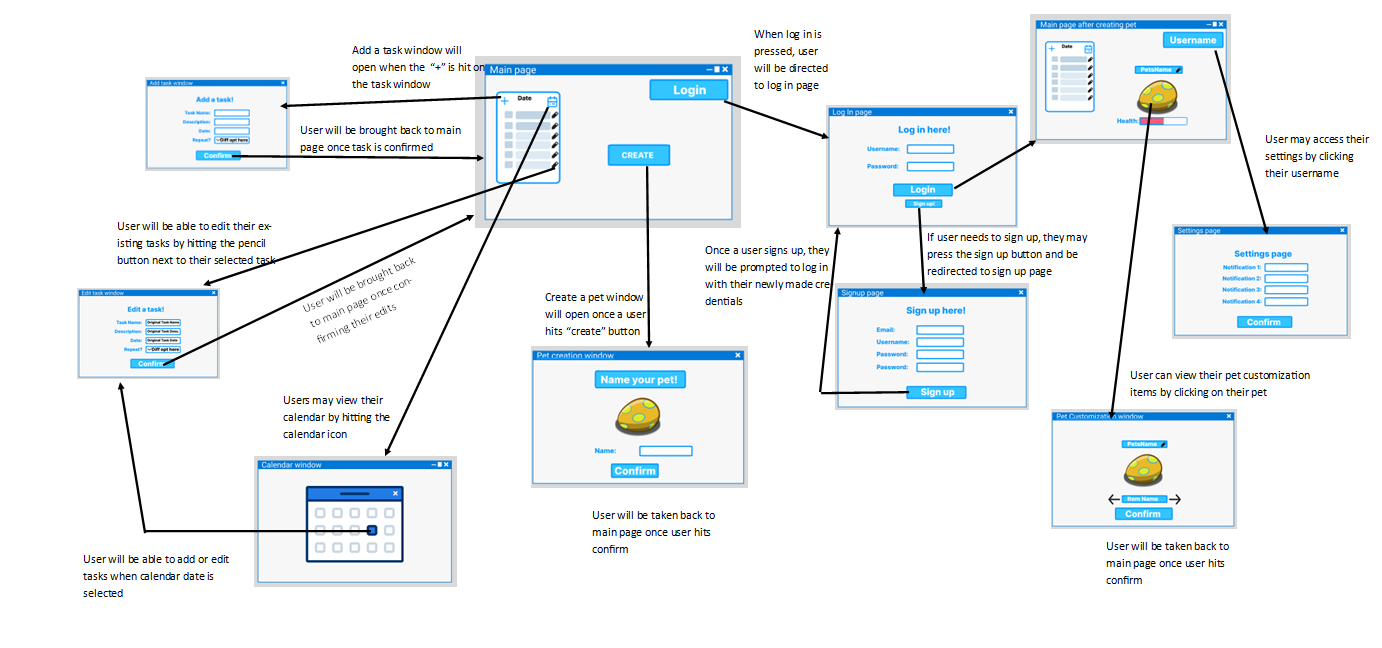
## BLOCK DIAGRAM



## COMPONENT DIAGRAM



## USER INTERFACE STORYBOARD



## STORAGE DOCUMENTATION

