1. Furry-Minder
2. Sarah Jorissen, Laura Lopez, Jonathon Hoffman
3. Abstract
   1. Each day of our lives, we are faced with filling up our day with tasks that we don’t want to accomplish. We constantly have to force ourselves to do them or get something out of it to get the job done. Due to natural human nature, we need the incentive to feel accomplished after completing our tasks. To assist in this constant issue, Furry-Minder provides a solution. The pet helps incentivize the user to stay productive and on task by tying the pet's care to a to-do list. If tasks are finished on time, the pet will be fed & cared for and continue to grow. However, neglecting your tasks also means neglecting your pet, eventually leading to its death and starting all over again.
4. Tools & Technologies
   1. **Chrome web browser:** This app's primary browser will be developed and tested.
   2. The team will be developing on **Windows and Ubuntu**, mainly focused on Windows.
   3. **WebStorm IDE (by JetBrains):** A quality JavaScript IDE that works on multiple operating systems and helps simplify and streamline many necessary tasks. Free to use for college students so all team members can access it.
   4. **HTML, CSS, JavaScript:** Required languages for building a web application.
   5. **React:** Create React App helps streamline the creation of a React project and comes with various useful features built-in. It will help ease the process of starting and working on a React app for a team that is mostly new to React.
   6. **Firebase:** For implementing a web server for our project once the front end is complete.
   7. **Discord:** For quick communication with team members.
   8. **Outlook:** For more long-form communication, sending necessary files, etc.
   9. **PivotalTracker:** Keeping everyone on task and seeing what still needs to be done.
   10. **GitHub:** version control.
5. Requirements list
6. Task List System
   1. Add task
      1. Clicking will prompt a pop-up that asks the user to fill in:
         1. Task name
         2. Task description (if applicable)
         3. The date that it is due
         4. Check box asking if the task needs to be repeated
      2. Add a task window with an “OK” button where inputted information will be saved onto the calendar and close the add a task window
         1. Information saved onto the calendar will be stored in the Firebase database
      3. Add a task window with a “Cancel” button that, when clicked by a user, will close the add a task window without saving it onto the calendar
      4. Repeated Tasks
         1. Daily
         2. Weekly
         3. None
   2. View task
      1. Tasks for the current day will be displayed on the application's home screen 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 task
      1. Tasks for the current day that are displayed on the task list window will be able to be “checked-off” by the 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
7. Calendar System
   1. View Calendar
      1. After our “View Calendar” symbol is clicked, a calendar will generate in a modal
         1. To the right of the current month will be an option for selecting a “Weekly view” or “Monthly view”
   2. View Tasks on the calendar
      1. While the Calendar modal is open, the user may be able to view the tasks on any given day within the selected view
      2. When clicked, there will be options on the calendar that will open the “Add Task” modal or “Edit Task” modal
   3. Edit Tasks on a specific date of the calendar
      1. When the user views specific dates tasks, there will be a button in the top-right (location may change) that says “Edit Task”
         1. If the user clicks the “Edit Task” button, the Edit task modal will open where the user can change:
            1. Task name
            2. Task description
            3. Date it's due
            4. Check box asking if the task is to be repeated
   4. Add Tasks on a specific date of the calendar
      1. There will be an “Add Task” button in the top-right (location may change) where information can be added to the calendar and stored in the Firebase database
         1. If the user clicks the “Add Task” button, the Add task modal will open where the user can create a task with the following information:
            1. Task Name
            2. Task description
            3. Date and Time it's due
            4. Check the box asking if the task is to be repeated
8. Pet System
   1. The Pet View will display the pet and its current stats
      1. The Health Bar will display its current health mirroring the database status and will change over time
         1. The pet’s health will update at midnight CST every night
         2. If the user completes tasks by their deadlines, the bar will fill little by little until it is complete, simultaneously updating the pet’s stats in the database
         3. If the user misses a task deadline, the bar will deplete little by little until it is empty, simultaneously updating the Firebase database
         4. The pet’s max health will increase as it ages
            1. Once the bar is empty, it will trigger the death of a pet
      2. The pet’s name will be displayed over it
      3. An option to open the customization menu will be displayed in the upper right-hand corner
   2. The pet will be shown 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 form will be determined by its current health and age
         2. The pet will have 3 Life Stages, mirroring the status stored in the 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

There are three “teen” stages the pet can randomly grow into from a baby stage

* + - * 1. After a week from being created, the pet will grow into an adult

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

Each teen stage can grow into one of two final adult stages

* 1. Pet Customization menu
     1. The user can select from a variety of accessories to display on the pet, which are stored in the 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
     3. The user can change the pet’s species
        1. Different species can be unlocked over time, after colors have been unlocked
     4. The user can change the pet’s name at any time

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 documents | 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 |