CS 320 Course Project Final Report

for

TodoList

Prepared by

Group Name: Rat Squad

Riley Barnes	11566375	riley.g.barnes@wsu.edu
Kevin Zavadlov	11569849	Kevin.zavadlov@wsu.edu

Date: 12/12/2020

1 Introduction

The Todolist program was created by Kevin Zavadlov and Riley Barnes.

1.1 Project Overview

The Todolist program to allows users add tasks to their todolist with a name, date, priority, and description. Users will gain points and a streak for all tasks completed on time. Users can set their own username and save their todolist to their local pc storage based on their username.

1.2 Definitions, Acronyms and Abbreviations

1. Chai: a JS assertion framework.
2. Github: version control program.
3. Html: Hyper Text markup Language.
4. Ide: Integrated Development Environment
5. Javascript: A programming language that easily interfaces with HTML
6. Mocha: A JS testing framework.
7. UI: User Interface.
8. Uml: a software design tool.

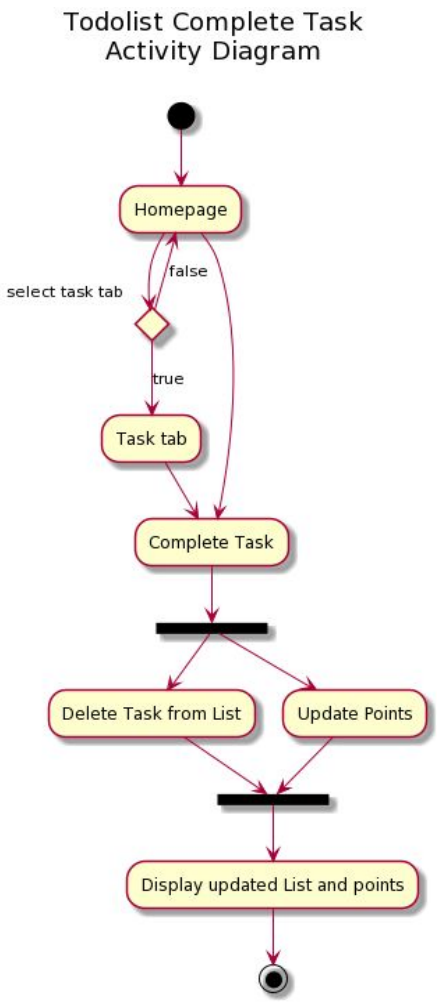
1.3 References and Acknowledgments

No reference or acknowledgments

2 Design

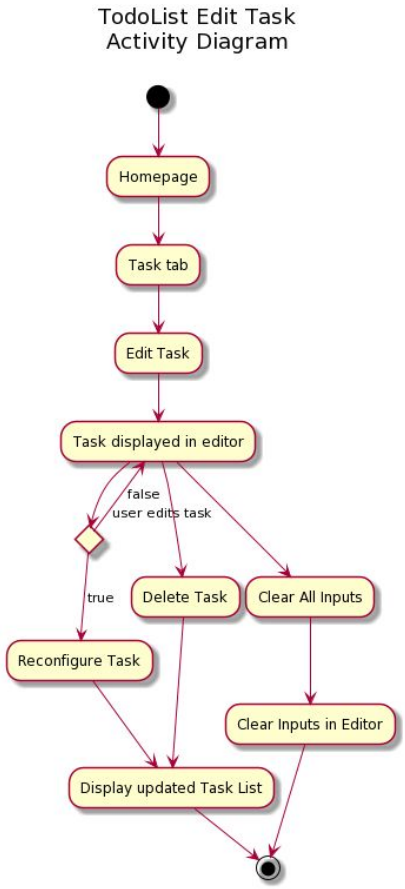
2.1 System Modeling

2.1.1



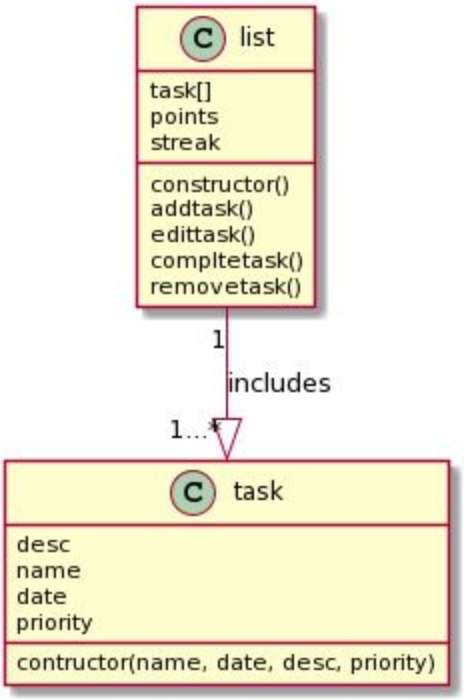
The start state is when the user opens the webpage. The end state is after the user completes a task and the webpage updates the necessary things. The diagram shows the paths the user can take to complete a task and how the system reacts when they complete a task.

2.1.2 Todolist Edit Task Activity Diagram



The start state is when the user opens the webpage. The end state is after the user finishes editing a task and the task editor is cleared. The diagram shows the paths the user can take to edit a task and how the system changes when they edit a task such as changing what buttons are useful to the user.

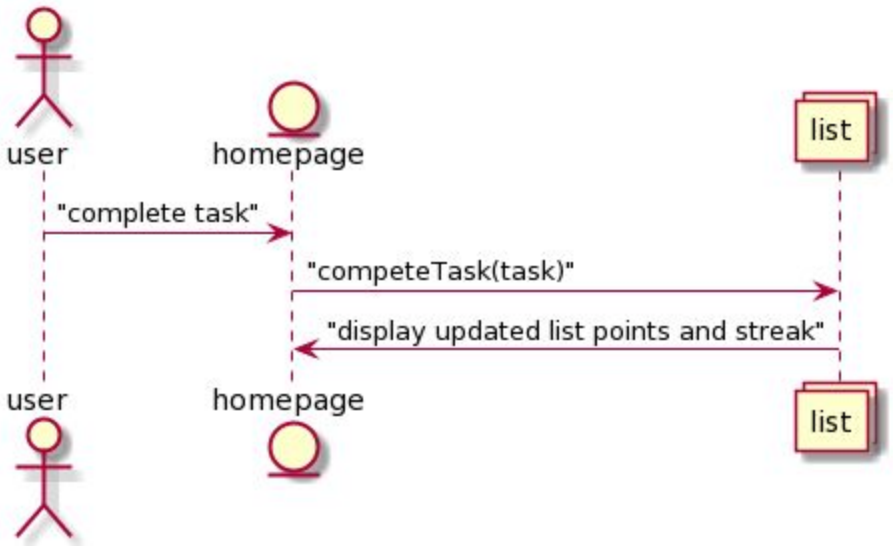
2.1.3 Class Diagram



List: The main todolist it holds the current state of the todolist and all of the tasks

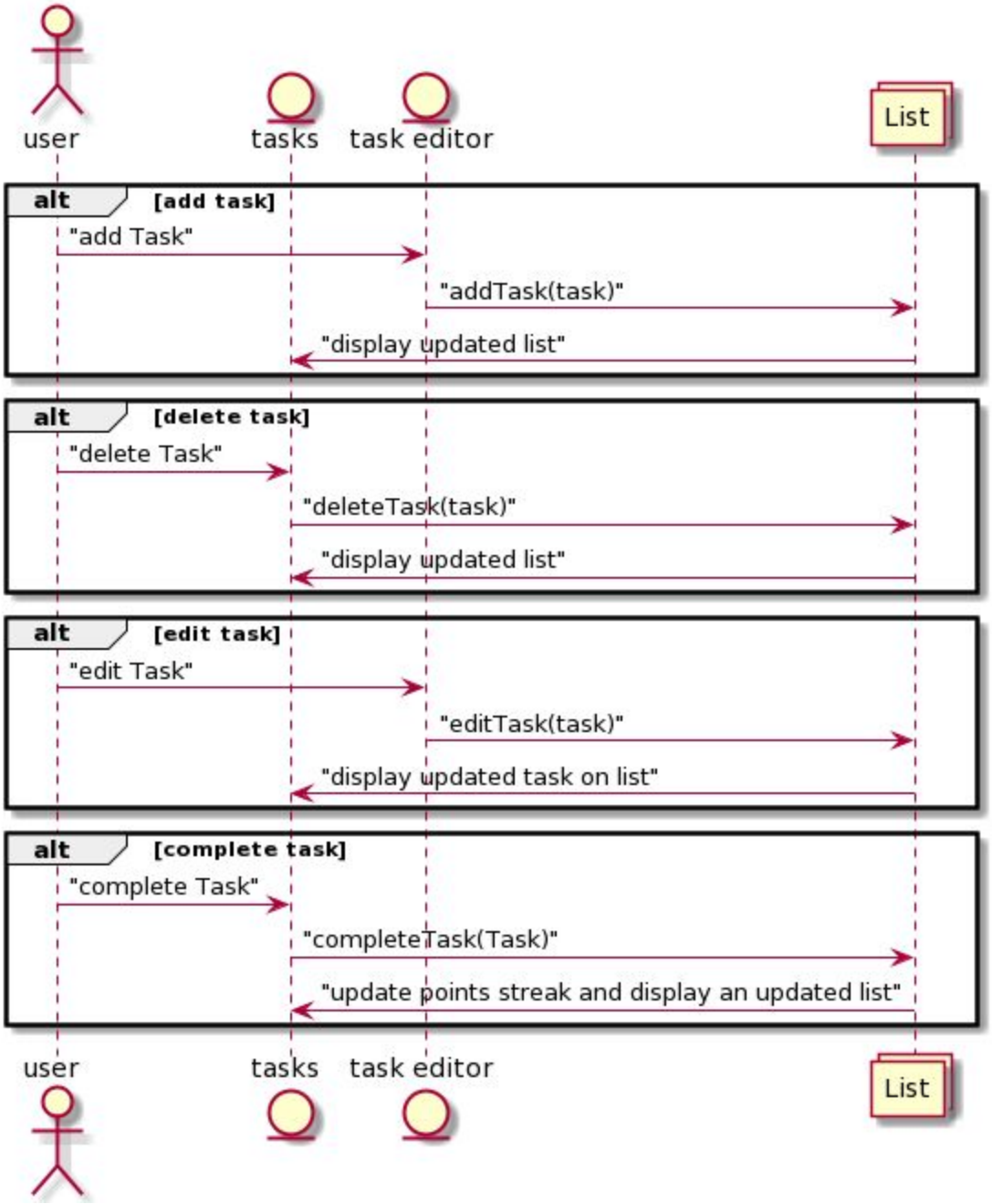
Task: A single task only holds the name, description, date, and priority of that task.

2.1.4 Homepage Diagram



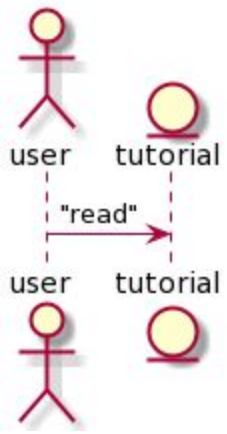
Home page the user completes tasks the complete task button is activated list spits out a new list updates points and streak

2.1.5 Task Page Diagram



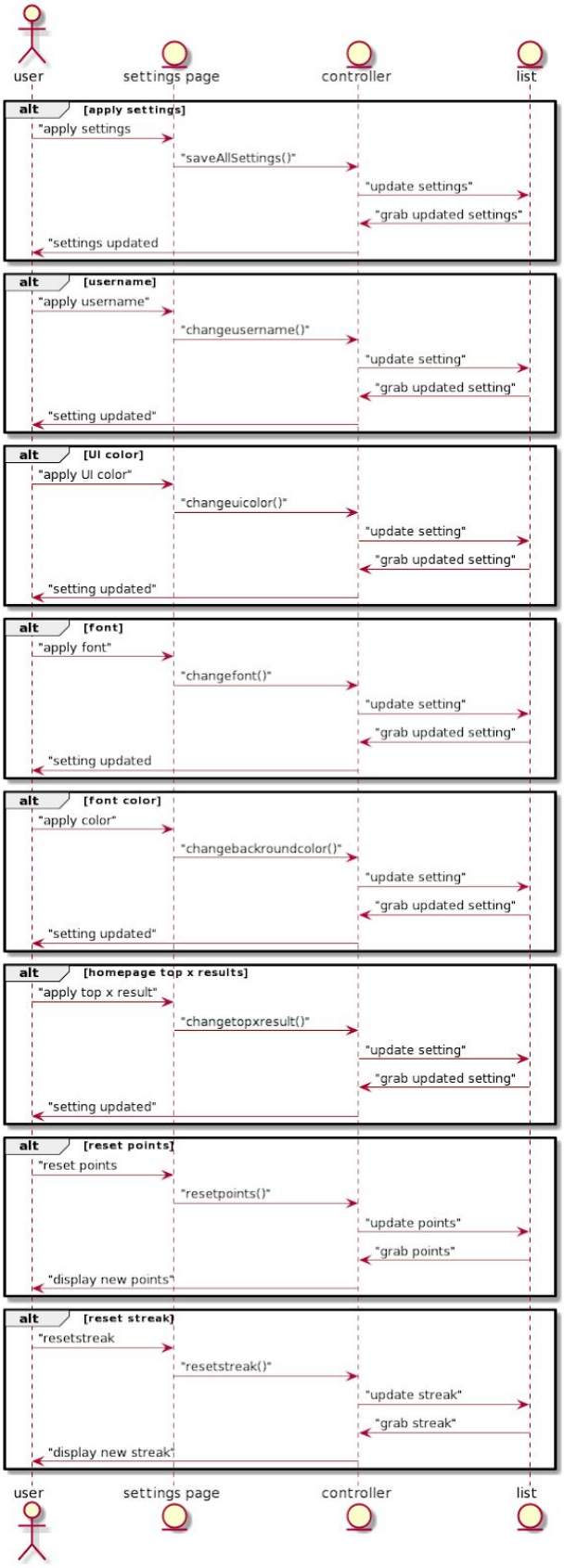
Tasks page user adds a task after pressing the add task button activating the addTask function in the list which then displays an updated list of tasks. User deletes a task deletetask function is activated which list spits out an updated list. User presses the edit button, task editor changes the attributes of the task list spits out an updated list. User completes task complete task is activated list spits out an updated list and updates points an streaks

2.1.6 Tutorial Page Diagram



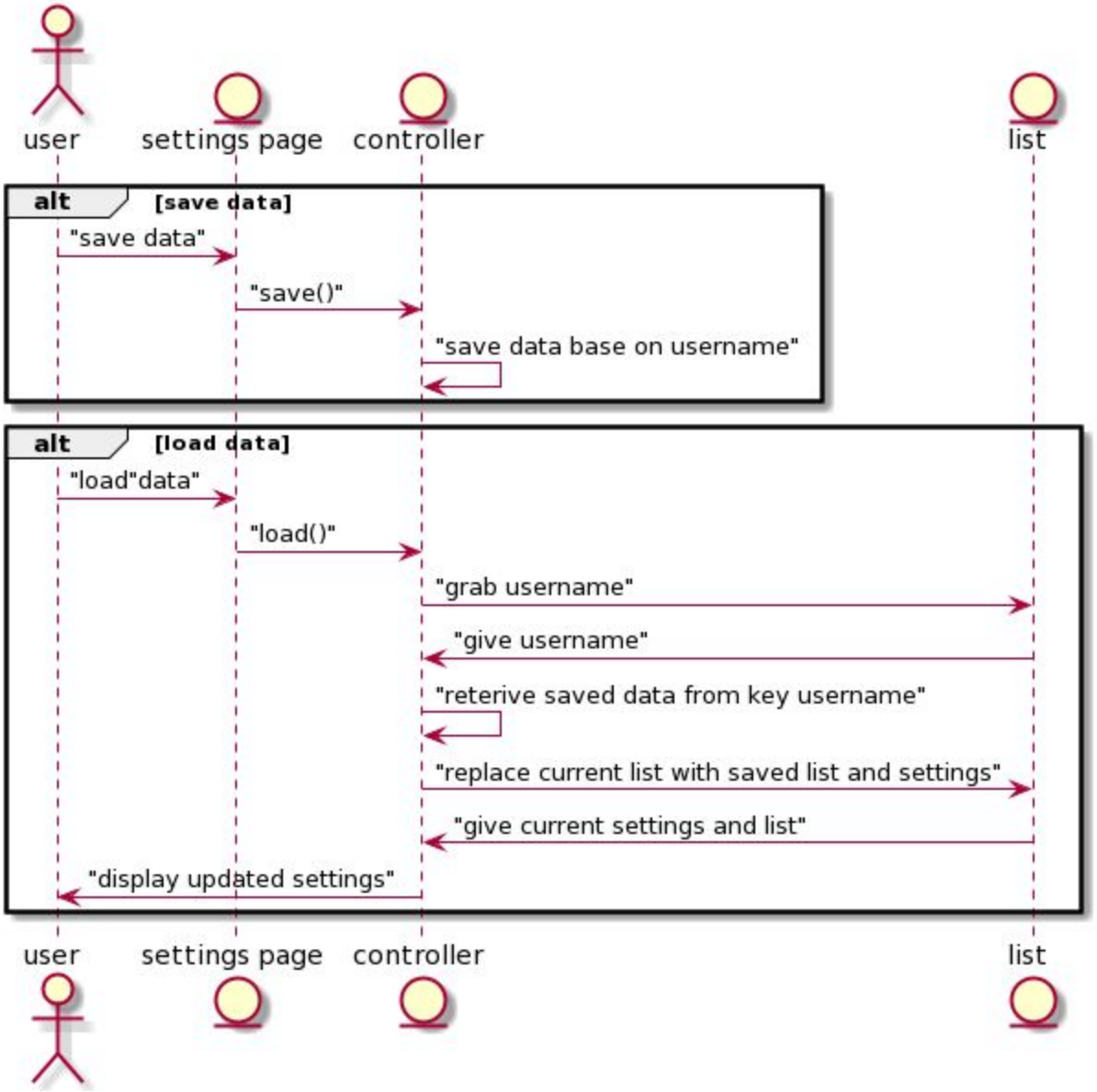
User reads tutorial page

2.1.7 Settings Page Diagram


user
settings page
controller
list
alt [apply settings]
"apply settings"
"saveAllSettings()"
"update settings"
"grab updated settings"
"settings updated
alt [username]
"apply username"
"changeusername()"
"update setting"
"grab updated setting"
"setting updated"
alt [UI color]
"apply UI color"
"changeuicolor()"
"update setting"
"grab updated setting"
"setting updated"
alt [font]
"apply font"
"changefont()"
"update setting"
"grab updated setting"
"setting updated
alt [font color]
"apply color"
"changebackroundcolor()"
"update setting"
"grab updated setting"
"setting updated"
alt [homepage top x results]
"apply top x result"
"changetopxresult()"
"update setting"
"grab updated setting"
"setting updated"
alt [reset points]
"reset points
"resetpoints()"
"update points"
"grab points"
"display new points"
alt [reset streak]
"resetstreak
"resetstreak()"
"update streak"
"grab streak"
"display new streak"
user
settings page
controller
list

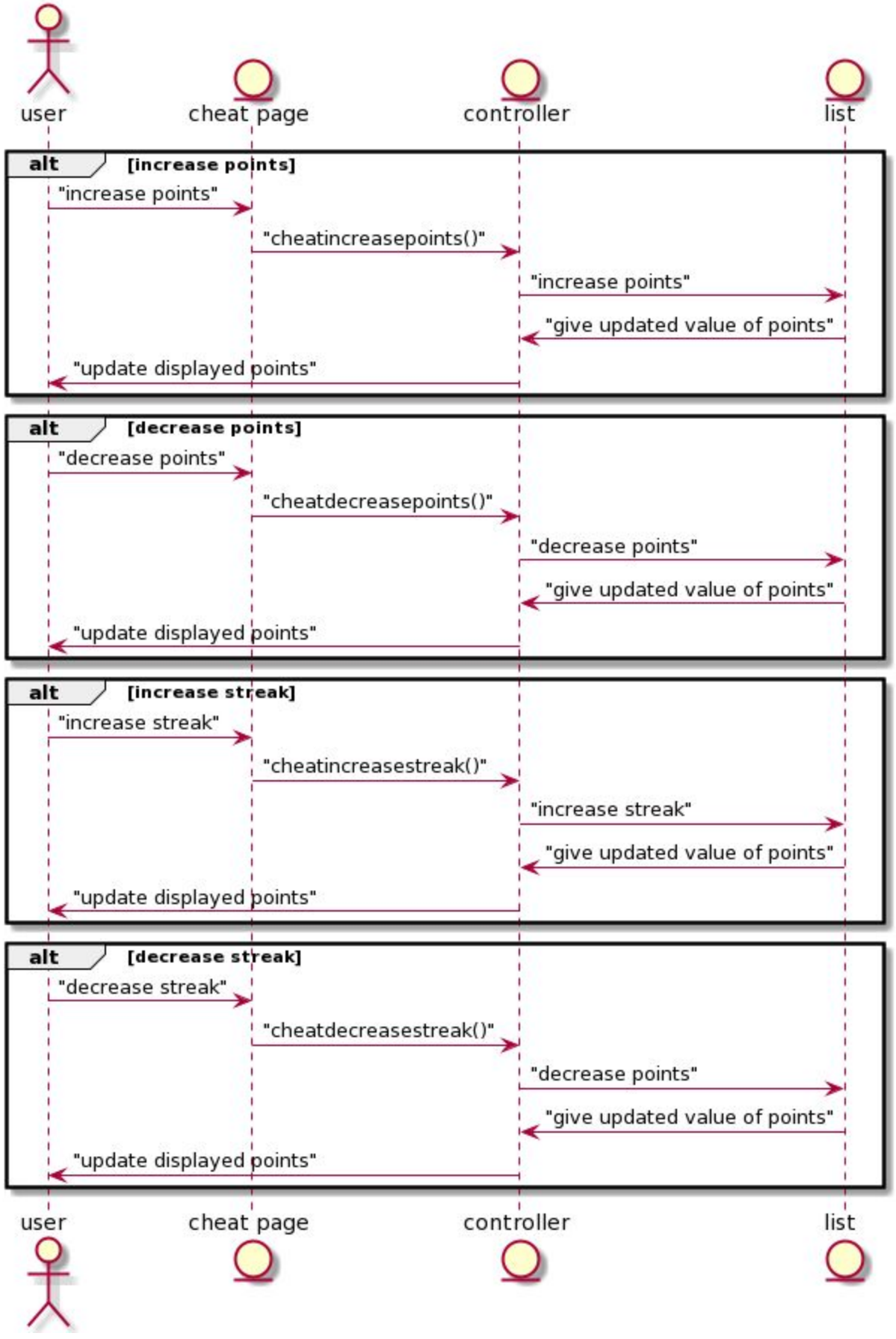
User changes a setting changeSetting is activated list displays an updated page of settings.

2.1.8 Save and Load buttons Diagram



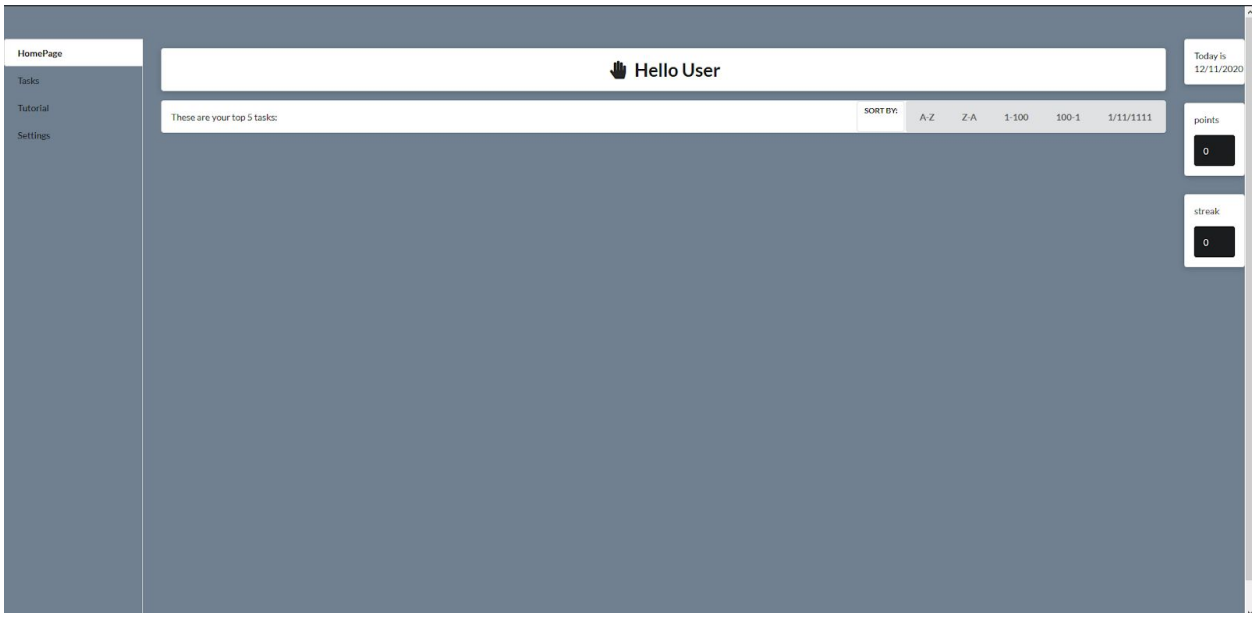
User loads and saves data

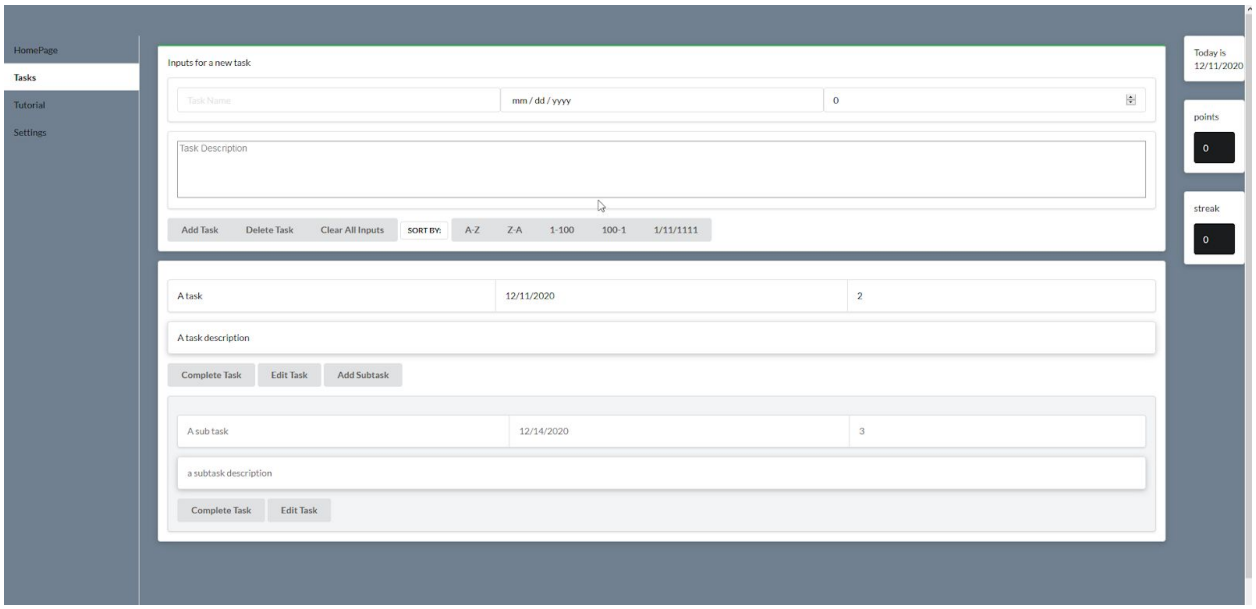
2.1.9 Cheat Page Diagram

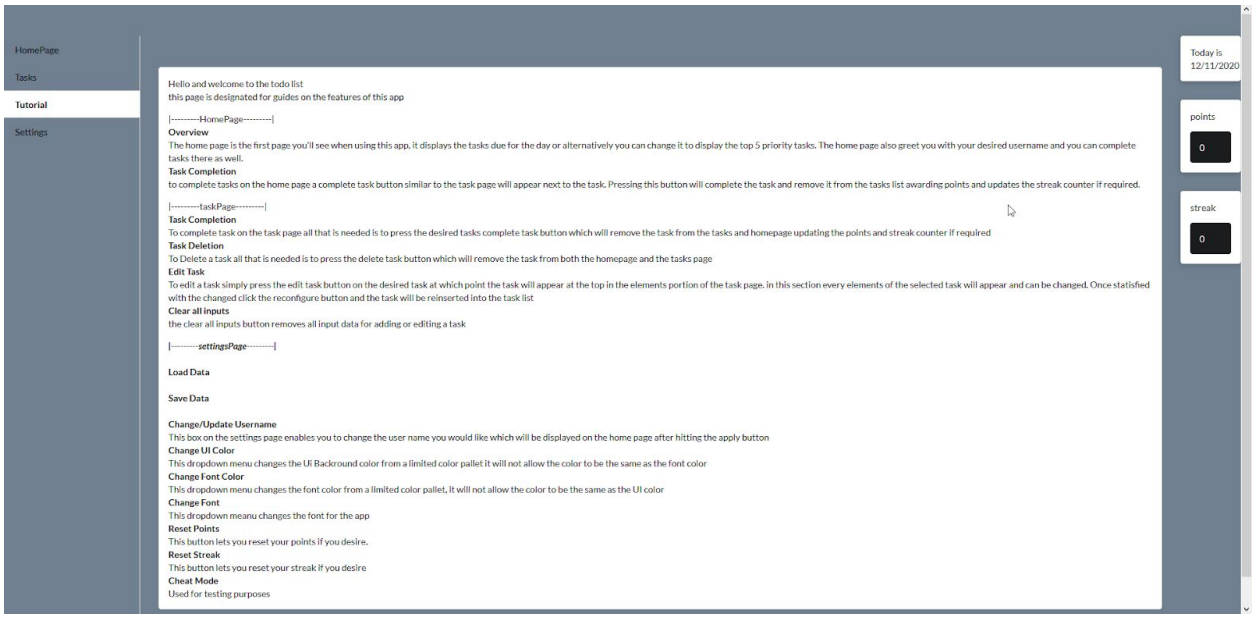

user
cheat page
controller
list
alt
[increase points]
"increase points"
"cheatincreasepoints()"
"increase points"
"give updated value of points"
"update displayed points"
alt
[decrease points]
"decrease points"
"cheatdecreasepoints()"
"decrease points"
"give updated value of points"
"update displayed points"
alt
[increase streak]
"increase streak"
"cheatincreasestreak()"
"increase streak"
"give updated value of points"
"update displayed points"
alt
[decrease streak]
"decrease streak"
"cheatdecreasestreak()"
"decrease points"
"give updated value of points"
"update displayed points"
user
cheat page
controller
list

User changes the current points and streak for test purposes

2.2 Interface Design






HomePage
Tasks
Tutorial
Settings
Hello and welcome to the todo list
this page is designated for guides on the features of this app
|---------HomePage---------|
Overview
The home page is the first page you'll see when using this app, it displays the tasks due for the day or alternatively you can change it to display the top 5 priority tasks. The home page also greet you with your desired username and you can complete tasks there as well.
Task Completion
to complete tasks on the home page a complete task button similar to the task page will appear next to the task. Pressing this button will complete the task and remove it from the tasks list awarding points and updates the streak counter if required.
|---------taskPage---------|
Task Completion
To complete task on the task page all that is needed is to press the desired tasks complete task button which will remove the task from the tasks and homepage updating the points and streak counter if required
Task Deletion
To Delete a task all that is needed is to press the delete task button which will remove the task from both the homepage and the tasks page
Edit Task
To edit a task simply press the edit task button on the desired task at which point the task will appear at the top in the elements portion of the task page, in this section every elements of the selected task will appear and can be changed. Once statisfied with the changed click the reconfigure button and the task will be reinserted into the task list
Clear all inputs
the clear all inputs button removes all input data for adding or editing a task
|---------settingsPage---------|
Load Data
Save Data
Change/Update Username
This box on the settings page enables you to change the user name you would like which will be displayed on the home page after hitting the apply button
Change UI Color
This dropdown menu changes the Ui Backround color from a limited color pallet it will not allow the color to be the same as the font color
Change Font Color
This dropdown menu changes the font color from a limited color pallet, it will not allow the color to be the same as the UI color
Change Font
This dropdown meanu changes the font for the app
Reset Points
This button lets you reset your points if you desire.
Reset Streak
This button lets you reset your streak if you desire
Cheat Mode
Used for testing purposes
Today is 12/11/2020
points
0
streak
0


HomePage
Tasks
Tutorial
Settings
save data
desciption
load data
desciption
apply Settings
applys all setting at the same time
username
apply
UI Color
apply
font
apply
Font Color
apply
reset points
resets points
reset streak
resets streak
cheat mode
admin password
enter
Today is 12/11/2020
points
0
streak
0

3 Implementation

3.1 Development Environment

All developers used the Intellij IDE. The display is HTML and uses the semantic UI framework. The program itself is coded in javascript. The test harness uses mocha+chai. Used github for version control and github desktop for ease of use interacting with the github cloud.

3.2 Task Distribution

Kevin: Homepage, Tasks page and task functionality, save/load functionality.

Riley: Settings page and functionality, Tutorial page and functionality, Tests, cheat page and functionality.

3.3 Challenges

Kevin: Getting the controller to know which task it is modifying and communicating it to the model was tough. The subtask implementation wasn't as simple as I would have liked adding many more checks and extra references (a subtasks[] in the main list). The subtasks also only go 1 lvl deep as the implementation wasn't modular enough to allow multiple levels.

4 Testing

4.1 Testing Plan

Create a test harness for the model (the List class) that tests all of its functions. Do the demonstration which will test the functionality of the display. Manually verify that each button has intended effect when clicked and that inputs are used appropriately.

4.2 Tests for Functional Requirements

Test the model of our program through a mocha+chai test harness.

todolist tests

List class functions

addtask()

- ✓ should add a task object to the task list that has the correct properties and values 57ms
- ✓ should be able to add many tasks and are in the order they were put in 183ms
- ✓ should be able to add subtasks to tasks
- ✓ should be able to add many subtasks to tasks

edittask()

- ✓ should be able to edit a given task given an index for the task
- ✓ should be able to edit subtask

compltetask()

- ✓ should be able to complete tasks as well as update streak and points
- ✓ should be able to reset streak if a task is overdue and not add points
- ✓ should be able to add more points if a streak is going
- ✓ should be able to reset streak if a task is overdue and not modify the current points

removeTask()

- ✓ should be able to remove tasks
- ✓ should be able to remove a subtask

changeUsername()

- ✓ should be able to change the username

resetStreak()

- ✓ should be able to reset the current streak

resetPoints()

- ✓ should be able to reset the current points to zero

sortAZ()

- ✓ should be able to sort the current list of tasks by lexographic order

sortZA()

- ✓ should be able to sort the current list of tasks by reverse lexographic order

sort01()

- ✓ should be able to sort the current list of tasks by priority lowest first

sort10()

- ✓ should be able to sort the current list of tasks by priority highest first

sortDate()

- ✓ should be able to sort the current list of tasks by date earliest first

4.3 Tests for Non-functional Requirements

Manually verify that each button has intended effect when clicked and that inputs are used appropriately.

1. Homepage
    a. Sort functions
        i. A-z verified
        ii. Z-a verified
        iii. 1-100 verified
        iv. 100-1verified
        v. 1/11/1111 verified
    b. Complete task verified
2. Tasks functions
    a. Sort functions
        i. A-z verified
        ii. Z-a verified
        iii. 1-100 verified
        iv. 100-1verified
        v. 1/11/1111 verified
    b. Complete task verified
    c. Edit task verified
        i. Delete task verified
        ii. Reconfigure verified
    d. Add subtask verified
    e. Clear all inputs verified
3. Settings functions
    a. Save data verified
    b. Load data verified
    c. Apply settings verified
    d. Username apply verified
    e. Ui color apply and dropdown verified
    f. Font apply and dropdown verified
    g. Top x homepage dropdown and apply verified
    h. Reset points verified
    i. Reset streak verified
    j. Cheat mode enter button verified
4. Cheat functions
    a. Increase points verified
    b. Decrease points verified
    c. Increase streak verified
    d. Decrease streak verified

4.4 Hardware and Software Requirements

A pc or laptop to run the program and its test.

5 Analysis

Kevin: spent approximately 30-40 hours total. Milestone 1 was the hardest because it was the first time I had to completely design a program before writing any code.

Riley: spent approximately 30-40 hours. Again milestone 1 was the hardest due to the required knowledge of the product you wanted to create.

6 Conclusion

We learned how to use plant UML to help design a program's structure. We learned valuable skills in software design such as: functional vs non functional requirements, software quality attributes, and Design and Implementation Constraints. We also learned how to use mocha and chai to create a robust testing framework with the coverage testing criteria in mind.

Appendix A - Group Log

We met at least once a week since we started working on the project usually more than once when a milestone was due. We also communicated through discord messaging.