

CS 320 Course Project - Software Design Document

for

Todolist

Prepared by

Group Name: *Rat Squad*

Riley Barnes

11566375

riley.g.barnes@wsu.edu

Kevin Zavadlov

11569849

Kevin.Zavadlov@wsu.edu

Date:

11/20/2020

Contents.....	ii
1 Introduction.....	3
1.1 Project Overview.....	3
1.2 Definitions, Acronyms and Abbreviations.....	3
1.3 References and Acknowledgments.....	3
2 Activity Diagram(s).....	4
2.1 Todolist Complete Task Activity Diagram.....	4
2.2 Todolist Edit Task Activity Diagram.....	5
3 Class Diagram(s).....	6
3.1 Todolist Class Diagram.....	6
4 Behavioral Diagram(s).....	7
4.1 Home Page Diagram.....	7
4.2 Task Page Diagram.....	8
4.3 Tutorial Page Diagram.....	9
4.4 Setting PageDiagram.....	9
Appendix A - Group Log.....	10



1 Introduction

1.1 Project Overview

Our project is a to-do list where a user can add, complete, edit, and delete tasks. There is a point and streak system. This document describes activity diagrams that show the actions a user can take and how the program responds. There is a class diagram that shows the variables, methods, and relationship between other classes of the classes. There are also sequence diagrams that show how the system reacts when a user interacts on certain tabs.

1.2 Definitions, Acronyms and Abbreviations

The menu tabs of our webpage are: Homepage, Tasks, tutorial, and settings.

The available buttons on the Homepage are: complete task.

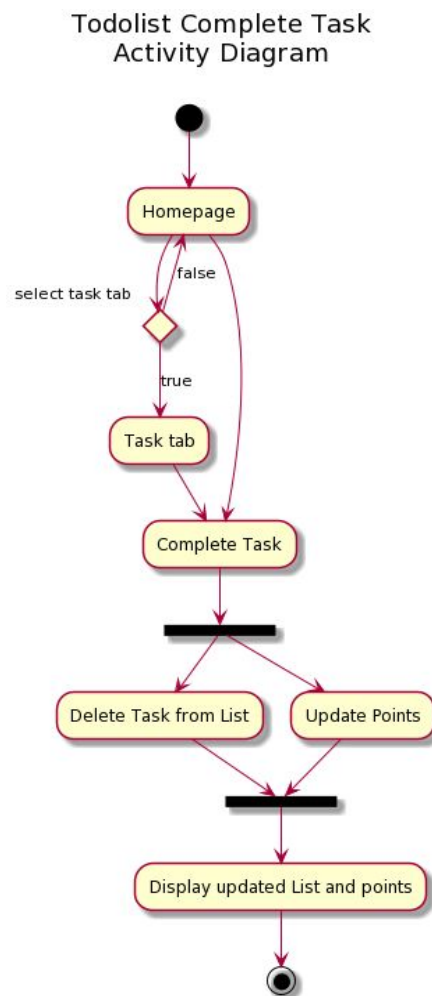
The task editor is only available on the task page.

The task editor has the buttons: add task, delete task, clear all inputs.

The individual tasks on the task page have the buttons: edit task, complete task.

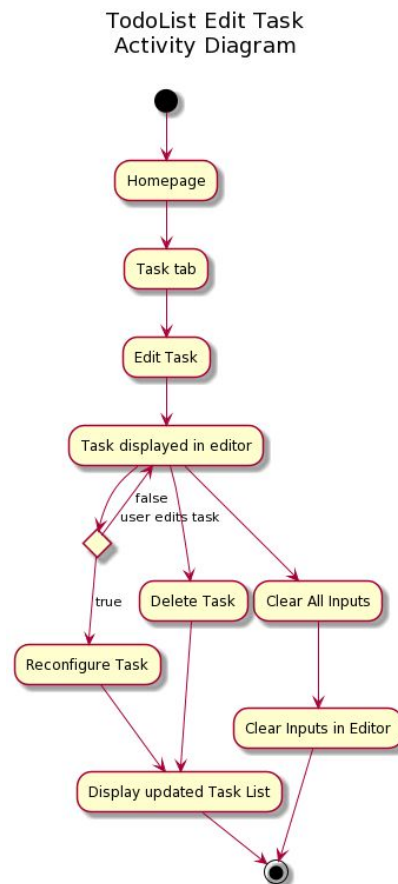
2 Activity Diagram(s)

2.1 Todolist Complete Task Activity Diagram



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.

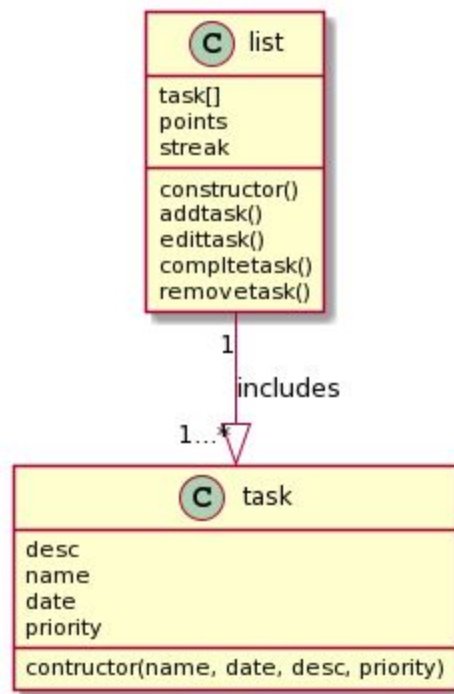
2.2 Todolist Edit Task Activity Diagram



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.

3 Class Diagram(s)

3.1 Class Diagram

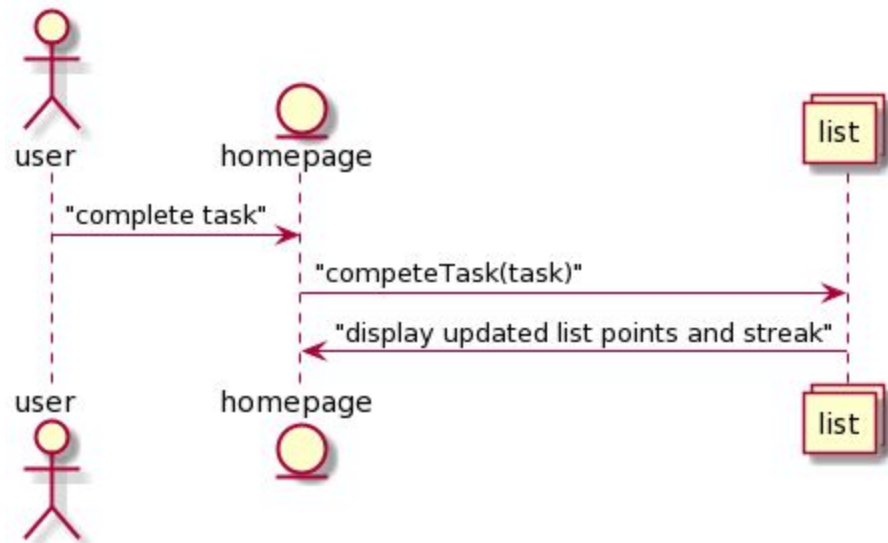


List: The main todolist it holds the current state of the todolist and all of the tasks

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

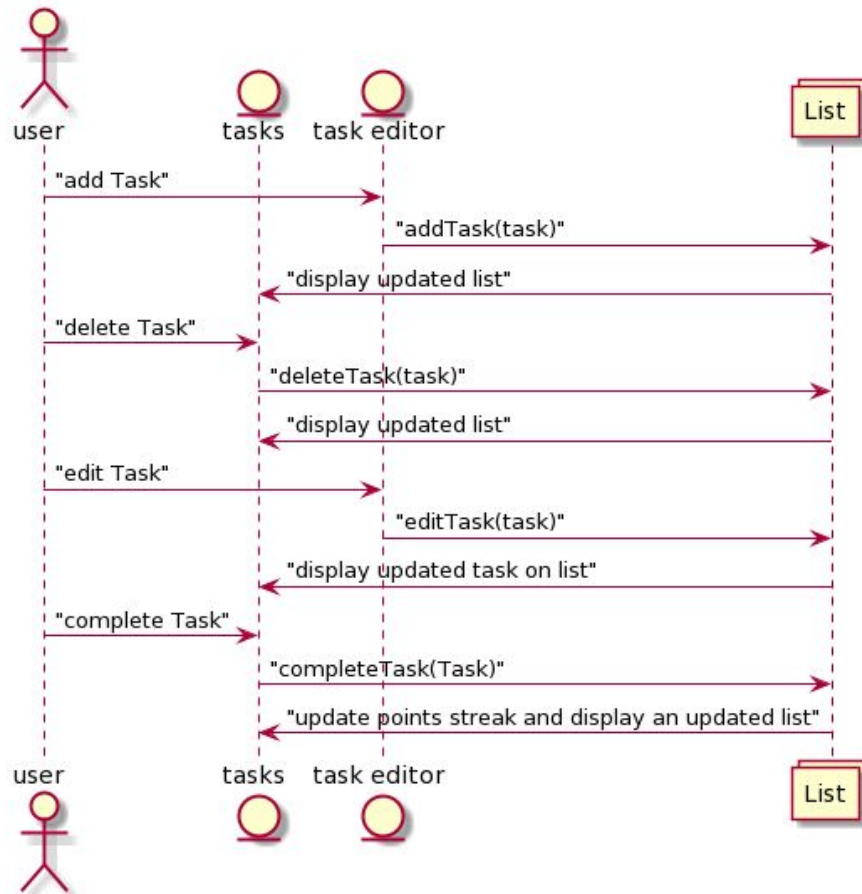
4 Behavioral Diagram(s)

4.1 Homepage Diagram



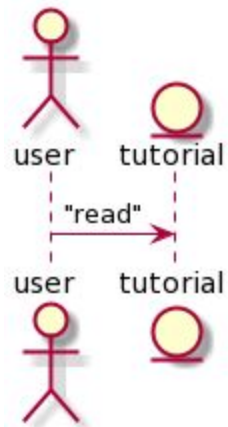
Home page the user completes tasks the complete task button is activated list spits out a new list updates points and streak

4.2 Task Page Diagram



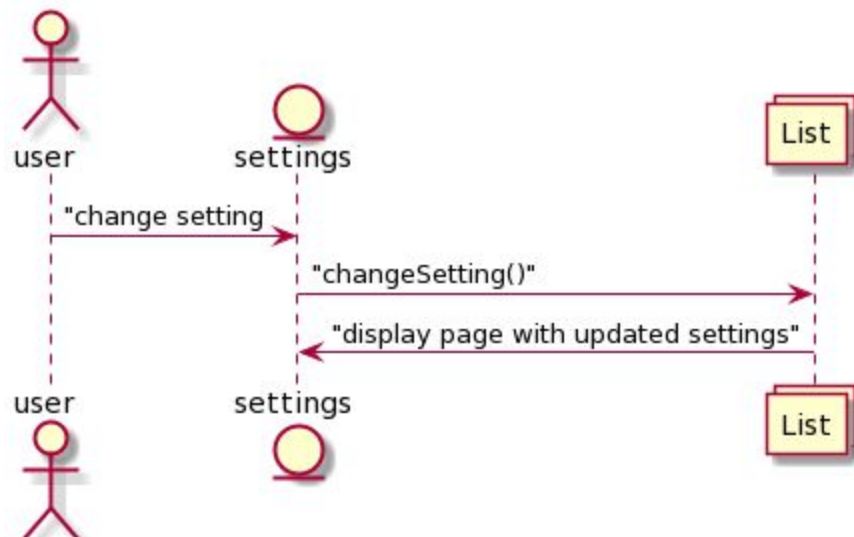
Tasks page user adds a task after pressing the add task button activating the addTask function in the list which then displays an updated list of tasks. User deletes a task deletetask function is activated which list spits out an updated list. User presses the edit button, task editor changes the attributes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

4.3 Tutorial Page Diagram



User reads tutorial page

4.4 Settings Page Diagram



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

Appendix A - Group Log

11/18/2020: Activity Diagrams (Kevin), Class Diagram (Riley)

11/20/2020: Sequence Diagrams (Riley), fill out descriptions/write rest of the document (All).