Project Specification with modifications (originally myPhDTime)

How many of you feel your grad program is like a black hole for time? You spend hours and days and weeks working on something, but still feel like your time is going to waste. How many times you are so tired at the end of the day but still feel like you did not spend enough time doing work? How many times have you felt guilty of not "spending enough time on task X" or "spending too much time on that task Y". To help save your time (and your degree!!) in this project we propose to develop the Kairos webapp.

Kairos is a webapp that will allow users to keep track of the tasks they are doing as they are doing it. With a very simple interface, in one simple click, you can add the task you are currently working on and start recording the time you spend on it. Kairos will not only keep track of the all the tasks and times you spend on them, but will also provide a cutting edge dashboard to help you visualize the tasks you are spending most (or least!) time on (1) per day, (2) per week and (3) per month. This dashboard is aimed to cognitively support your self-regulation and monitoring activities that will allow you to identify the tasks that you need to spend more or less time on, thus help you improve your productivity, happiness and help , you save your degree. Kairos will provide you with a private account to keep track of your time and data.

On completion of the above features, we will provide the option to "mentor" and "get mentored" by sharing with your data with other students. This way you will receive the necessary support and encouragement (and social responsibility) to do well.

Kairos will be developed fully in Django. To develop the cutting edge dashboard, we will learn and make use of the D3.js data visualization library. We will initially focus on learning how to develop and use charts such as bar charts and pie charts and time permitting, we can move into more complicated types of data representations.

Project Name: Kairos - Your personal time manager

Product Backlog

Modules:

1.User Registration

The user will be presented with a registration form which allows him to register with his email address and select a password for the application. The user will be able to confirm his registration and affiliation with the application if and only if he confirms the email address. Other fields in the registration form would be to take in the user's first name and contact number.

2. Dashboard

Once a new user logs in, he will be presented with a dashboard that allows him to choose from within a wide range of options that we call as a 'high level task / category', presented below. Under these categories the user can create tasks within them (for example the user can create the 'Web Apps" task within the high level 'Course Work' category). After creating the "Web Apps" task, 'subtasks' can be added to it like Assignment 1 or Project Sprint 1 etc.

For a user who has already added tasks, the dashboard will provide a visualization (using d3.js) of the progress of ongoing subtasks. These will show the time elapsed since the task was started, time left depending on the expected finish data entered by the user, due date of the task, the percentage of completion of the task (this can be entered by the user for an ongoing subtask, since the time elapsed is not reflective of the amount of work put in by the user). These can be sorted by due date, percentage completion or time elapsed. The list of ongoing subtasks will have links to the its edit page.

The dashboard will also show the amount of time spent during the week in each high level task. This will give the user a high level idea of how long they spend per week on course-work, research, basic needs or other categories the user creates. This data can be aggregated over more than a week (2 weeks or a month) and displayed if the user wishes to do so.

The navbar on the dashboard will have links to the user's account, a logout button and a search bar, so the user can search for a particular subtask within the ongoing tasks. There will also be link to a page where the user can add new tasks. The dashboard will also have a side menu which will have navigation links to the tasks' individual page. For example there will be a link to the "Web Apps" task which will direct to a page that contains all subtasks (assignments, projects etc) related to Web Apps. Similarly the side menu will have links to tasks such as research or basic needs.

3. Add task in 'High Level Tasks or Categories'

Users can add tasks from a set of predefined categories that they perform on a regular basis. The pre-defined categories will be Course Work, Research and Basic Needs, each of which will have different forms. Users can also add a category if they want but these will have simple forms with just the essential fields. The forms will have some essential fields like task name, start time for task, expected end time. There can be optional fields for a due date (which can be different from expected end time), a fields for any comments associated with the task.

4. Adding Subtasks

Each task in the dashboard page would be associated with subtasks. For example, a student might take several courses and he might want to track down the progress of each course. The subtask page in our application offers exactly that. The subtask page offers the user the ability to edit and decide when he wants to stop, pause, continue tracking the progress of each of his subtasks. The user is also presented with important information, that he could use to track his progress and improve his productivity. We provide the user with details such as how much time the user spent on a particular subtask for that week as well as the amount allocated for sub subtasks within a particular subtask eg. a student taking a course might spend 2 hours on the readings and 10 hours working on an assignment for a particular week. We also provide him with alerts showing him the progress he has made so far, as well as how much time he took to complete (if he has completed a task), compared to the amount of time he predicted for completing the task.

Technologies Used : Django, D3.js, AJAX, JQuery, Bootstrap, HTML, CSS, Materialize

Optional Features

If the above features of the applications are implemented on time, we will consider providing the user a more enhanced version of the application. The enhanced version may include the ability to present other users performance on a subtask as compared to the user and the ability to be mentored by another user. This in turn will allow the user to either send messages to other users in the form of emails, calls or chatting with him. This requires a sophisticated design model, along with the integration of technologies such as WebRTC, Twilio and email integration. In addition, we will give the user a chance to register for application through their Linkedin or Facebook accounts.