Project Specification

Xinyi Li Juntao Li Yaxi Li

Project Proposal:

We decided to implement “beat the time”, the description of original approved proposal is as below:

*We plan to build a time tracking web app for user to record and reflect his/her time/effort spent on different projects, and major milestones/achievements along the way. Users might look at their friends' statistic page.*

*1. Predefined projects and sub-projects for user to track time. User can input or use a timer to add a log to these projects. User may also add/edit new projects.*

*2. Achievement mechanism. Users can earn various metals based on their accumulated time on different projects.*

*3. User skills statistics. Besides displaying user's time accumulated on different projects, we also provide a RPG-like real-time spider diagram showing users' relative strength on different areas. For example, a user might be very experienced in Computer Science related fields but weak in Arts.*

What we want to implement is based on this proposal with some modifications. The detailed description of our project is:

Our plan is to build a time tracking web application for users to record and manage his/her time spend on different projects. Instead of only focusing on users’ individual project, we further introduce a team project time tracking function which will could the manager to coordinate the team’s effort.

1. A user can set his/her own goal by adding the projects to be done previously. Use can also input the time spending on different projects afterwards. For the users’ goal, we will notify the user before their planned starting time for this project begins. There are individual projects and group projects.
2. Individual Project: Predefined projects and sub-projects for user to track time. Use can input or user a timer to add a log to these projects. User may also add/edit new projects.
3. Group Project: A user can form a group with others. Group must have a manager who is responsible for defining precise goal, assigning tasks for team members, setting a time period and deadline for every task. The tasks assigned by the manager will automatically appear on the team member’s goal list without manually adding by the team member.
4. Notification: For every project user added for a goal, our application will notify the user before the time when this project starts via inbox or email.
5. Timer: When user see the notification, he/she could accept the notification then we will start tracking time for this project. When user finish this task, he/she could close the timer.
6. Achievement mechanism. Users can earn various metals based on their accumulated time on different projects.
7. Reports: We will generate the reports for user time accumulated on different projects and also whether goals have been achieved or not. For team projects, we will generate reports for the progress of this projects.
8. Calendar: User can see their detailed time tracking for the previous and incoming projects in a calendar page.

To implement those functions, we decided to use Django, MariaDB for backend development, React.js, HTML, Chart.js for frontend. For APIs, we want to use Calendar API from Google. Potentially, maybe we will use Google Map API to add some geo-location services.

Project functionality:

1. Login/Signup module: User can use this module to sign up then log into our application. User can also retrieve their password with this module. This is the basic function that almost all the web applications should implement.
2. Home module: User can add new individual project and tasks for this project. User can also select existing individual projects then start the timer. User can also set the goals for the next week and the expected finishing time for all the goals.
3. Dashboard module: Users can see the reports for themselves in this module. We will provide information like the effectiveness, the tasks finished, the tasks unfinished, etc.
4. Projects module: In this module, a user can see the details for a specific project. We will provide information like the progress, the progress of each member, the goals, etc.
5. Notes module: A manager can write notes for a specific group project to set goals and assign tasks. A team member can see the tasks assigned by the manager. We also introduce a function to import the tasks assigned by the manager into the goal list by clicking one button. Also, a manager can manually send a notification to a member if he/she is late for a deadline.
6. Groups module: A user can see the group information in this module. Which groups have the user joined, the members of this group will all be included in this module.
7. Calendar module: This module will visualize the tasks the user need to do in the future or the tasks the user has done previously in a calendar. For this part, we will use Google Calendar API, so it will look like the Google Calendar.
8. Notification module: When the expected start time of a task starts, the notification module will notify the user by 3 different ways. The first way is to notify the server via Inbox of our application 15 mins before it starts, via Email 5 mins before it starts, and via Phone Text 1 mins before it starts. User can accept notification then the timer will on.
9. Profile module: User can modify password, user name, notification on/off, etc. A user can also see other users’ profile. For others profile, we can see others effectiveness curve.
10. Navigation module: This module will help the user navigate through our pages. There will be 2 navigation bars. One is on the left of this page, one is on the top of this page. For the navigation bar on the left, a user could navigate through Home module, Dashboard module, Projects module, Notes module, Groups module and Calendar module. For the navigation bar on the top, user can click the notifications available and go to the Profile module.

For the 10 modules, Xinyi is responsible for module 1, 2, and 3. Yaxi is responsible for module 4, 5, 6 and 7. Then Juntao is responsible for module 8, 9 and 10.

Draft implementation of the data models:

There will be 4 models (may change afterwards):