# 1.2 Update Plan

First I need to reinstate the goal of this software.

1. List and reminds activities user need to do.
2. Provide well-organized documents of what has been done by the user. That is the key advantage over other GTD software. In order to provide it the activities need to be well modeled in database, so it can be extracted in a well-organized way.
3. Minimize typing while using this application

## Activity model

In order to make activities well documented, we need title, schedule, class. But each of them is redundant in some scenario.

We need to do a brainstorm of common scenarios in usage, then reach an abstract model of activities

* Appointment: It is an activity last some time without interruption.
* Project: large project.
* Scheduled work: some people may schedule a work in some days, but it is not as strict as meeting type in schedule.
* Small task: It can be done in several minutes just like sending an email at some time
* Note: Note
* Waiting: We need to wait for the work of someone else to accomplish this one. For example, receive replied message.
* Some day: It is usually an initial state of a project, but we did not think it through.

As for title, it is sometimes useful, sometimes no. The original plain version will be changed as a plain text with the first line as title.

As for time, it is usually more convenient to use date instead of time. ~~But considering time-zone difference, store time zone offset as well~~. There are three functions of time. The first one is for documenting, and our primary interest would be when this thing is done, in other word, finishTime. The second goal is for sort when it is still active. We are interested in today’s appointment>old scheduled work/small task>today’s scheduled work/small task. Some day/waiting does not have an active time. Third one is default remind time. This time of appointment is based on startTime, but it is deadline for scheduled work or project.

As for class, loose connected activities should be supported using tags instead of projects. Our primitive types are

* Appointment: they are distinct because of sort order and finish feature.
* Task: include project, scheduled work, small task. It has a state: active, pending, inactive. Tasks can be created as subtask when parent is active.

## API

1. GET:/appname/activity/aid?from&to&page: JSON of this activity {aid,type(“appt”/”task”), title,parent,description,state,finishTime,startTime,deadline:,subactivity[]{aid,type,title}}
2. POST:/appname/activity/aid

## Search

## Logging