Team Information
- 14' Kim Seongmin
- 15’ Kim Sebin
- 13’ Lee Jaehong
- 15’ Song Joonho
- 15’ Choi Jihyeok

Final Release

Name of the product
- Software Development Project Management Tool

Final User Guide (Written in Markdown)

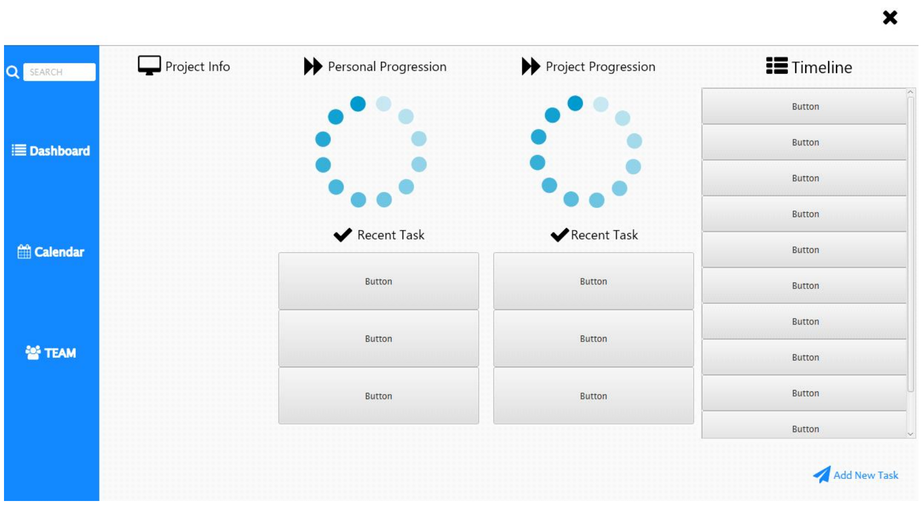
# User Guide

## Project Description

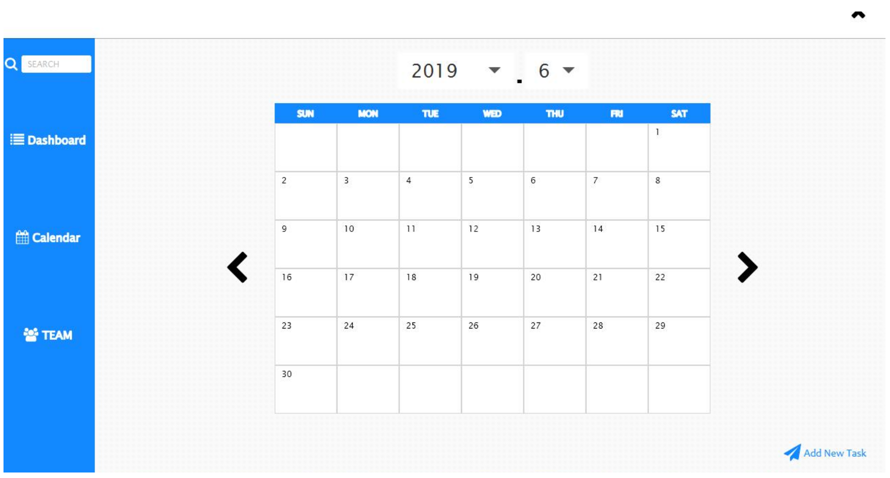
* This tools is used to managing/scheduling some tiny team project effectively.

## How to use?

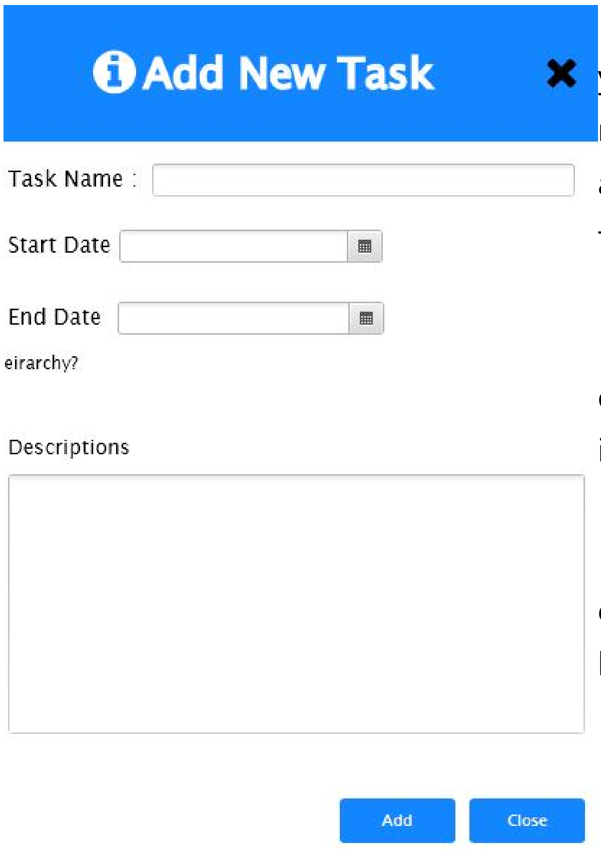
* First, When you execute our program, you will see this "dashborad"



* You Can See Calendar If you click "Calendar" Button on the left side bar.



* The first view of calendar is like this. if you click the left button(<), you can move your calendar to before one month. and also if you click right button(>), you can move the date one month after.



* If click date button on the calendar, you can meet this windows for adding new task. also, you can meet this window after you click "Add New Task" button on the right below of calendar window.

* You can assign task name and start date, end date. you can assgin hierachy(the importance of task)

* If you finished to write simple description of task, just push add button or close button to finish this job.

* You can see time line of team tasks on the Dashboard page.
* If you add/change/delete some tasks on your program, firebase databse will be automatically updated by firebase API(REST request).

Final Developer Guide (Written in Markdown)

# Developer Guide

## Project Description

* This tools is used to managing/scheduling some tiny team project effectively.

## Main Feature

* This program is consisted of `Schedule Manager, Calendar, TaskList, Search, Timeline diagram`.
* This diagram shows structure of our program.
* Our program has real-time synchronization of database between all team

members with Firebase Database.

## How to use

 * First, Login to our program with account registered in database.(We will not serve login in MVP version)

## How to obtain and install the software

 * Clone our repository and just build.
 * The command to clone our repository is
 ```git clone https://github.com/pm-tool-se2019/Source```
 * You need to use `Maven` to install all dependency to execute our program.
 * If you use `intellij` IDE, you can get the dependency library easily.

## How to run our software

 * Just build and execute.

## How to report a bug

 * We are open to receive contribution or issues for our program.
 * Please don't hesitate to report a bug or to give an opinion to our project :).
 * Just write `issue` of gitHub repository. but, I recommend to write specified issues to get exact reply.

## Dependency

 * Our program uses `JavaFX`, `firebase4j`, `log4j`, `gson(google json parser)`, `jackson`.

## Class Method/Member Variable Usage

### Schedule Manager

*** Member Variables**

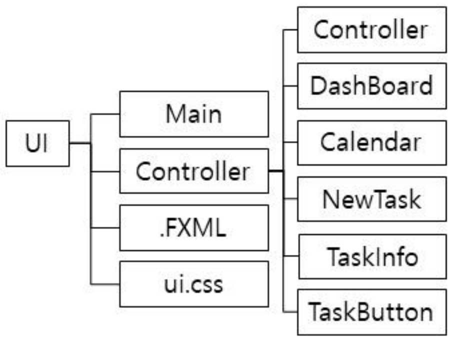
 * Public Class Schedulemanage : The class about dealing with functions about Task.
 * firebase_baseurl : Our Firebase URL. For connect with firebase remote DB.

* current_user_task_list : User's task list. This Arraylist will be shared with all of other modules.
* **METHODS**
* **searchFromTaskList(int id)** : Search the given task from current_user_task_list. If not found, returns null.
* **addTask(Task t) :** Add task to current_user_task_list and Firebase DB. Task is you want to add. It should pre-defined from other interface. This method does not give a function of setting property of task object.
* **deleteTask(Task t)**: Delete the Task from current_user_task_list and Firebase DB.
* **updateTask(Task t)** : updating the Task information.
* **deleteTask(Task t)** : Delete the Task from current_user_task_list and Firebase DB.
* **fetchFromDB(Task t) :** Fetch specific Task from DB using task id. Use when cannot find task from current_user_task_list. Return Task or null(failed to find).
* **fetchAllFromDB() :** Get all Tasks from firebase DB. It should be used when the program starts. All tasks are saved at current_user_task_list.
* **Class firebase4j.firebasesrc** : It is a libarary to use Firebase API with Java. For dependency, we put this on the ScheduleMan Folder.

### Calendar
* **MyCalendar singleCalendar** : singleton reference variable of Calendar instance.
* **LocalDate local_date_time** : Saving date of today. Java.time (JDK 1.8)
* **MyCalednar(..**) : Constructor of Calendar Instance.
* **getNowMonthValue()** : return today's month
* **getNowYear()** : return today's year
* **getCalLastDateOfMonth() :** Return array of calendar end day.
* **getCalLeafLastDateOfMonth**() : return leaf year array of calendar end day.
* **MyCalednar getSingleCalendar()** : return reference of singleton calendar.
* **loadTaskList()** : Set Tasklist to calendar.

### UI
* .FXML : .fxml files contains actual UI components such as HBox, Button, Anchor etc., Written in xml, some of controller listeners are included in this file. Each FXML files has their own controller.
* Ui.css : css style sheet define most of fxml components' style. Such as background **color, text color, hover action etc,.**

#### Java Classes :



 * **UI.Main** : Application Main, Calling JavaFx Main stage.
Method : main()
 * **UI.Controller** : Application Main stage Controller. Defines how main stage components work which contains such as SideButton Listeners.
 * **UI.DashboardController** : From Main stage, user can add dashboard scene in Main stage by clicking sidebar button, and this is Controller of Dashboard scene. Defines dashboard scene components' work.
 * **UI.CalendarController** : Same as DashboardController, user and add Calendar scene in Main stage by clicking sidebar button, this is Controller of Calendar scene. Defines calendar scene components' work.
 * **UI.NewTaskController** : From Main stage, by clicking Add New Task button, user can enter to new Scene named newTask Stage. If this stage activated, Main stage will be disabled. Defines NewTask Stage's components' work.
 * **UI.TaskInfoController** : From Calendar Scene or Dashboard Scene, User can enter TaskInfo Stage by clicking task buttons. If this stage activated, Main stage will be disabled Defines TaskInfo Stage's components' work.
 * **UI.taskButton** : This button is custom components which extends VBox that elements of javaFx. This button contains VBox-Button-HBox-Texts. By Clicking this button, user can enter to the TaskInfo stage
 * **UI.taskButtonController** : Controller of taskButton. If taskButton Clicked, enter to TaskInfo Stage.

### TaskList

 * **ShowTodoList** : With the input (ArrayList full of 'Task' variables), output the task names of which their states are 'TODO'.
 * **ShowDoingList** : With the input (ArrayList full of 'Task' variables), output the task names of which their states are 'DOING'.
 * **ShowDoneList** : With the input (ArrayList full of 'Task' variables), output the task names of which their states are 'DONE'.
 * **AlarmTodayTask** : With the input (ArrayList full of 'Task' variables), output the task names of which their states are 'TODO' and the start date of the tasks and

today's date is same.

 * **UI :** There are three split scenes, each of them will show the tasks 'TODO', 'DOING', and 'DONE'.

### TimeLine

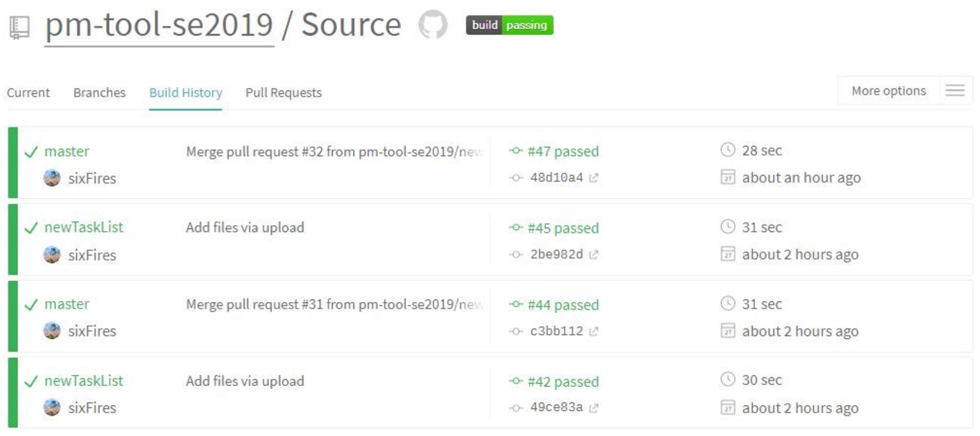
 * **TimelineManage.java :** ByStartDate () using comparator sort the task order of date -ascending | descending. Search() find match with input

 * **TimelineController :** show task or all of task. With time order which is done by TimelineManage

Test Reports

1) Build Test

- We are using gitHub as collaboration tool. So, We can use 'Travis CI' for build testing every commit/pull-request. If someone`s code break our complete build, we can reject or revery his/her commit.



pm-tool-se2019 / Source build passing

Current Branches Build History Pull Requests More options

✓ master sixFires	Merge pull request #32 from pm-tool-se2019/new	#47 passed 48d10a4	28 sec about an hour ago
✓ newTaskList sixFires	Add files via upload	#45 passed 2be982d	31 sec about 2 hours ago
✓ master sixFires	Merge pull request #31 from pm-tool-se2019/new	#44 passed c3bb112	31 sec about 2 hours ago
✓ newTaskList sixFires	Add files via upload	#42 passed 49ce83a	30 sec about 2 hours ago

2) Unit Test

- We used JUnit for testing our components.
- For specified input, the method or constructor have to return some special return value or output. If someone`s code can't pass this JUnit component test, someone has to re-develope his/her program.

3) UI Test - testFX

- testFX is automatically testing JavaFX UI.
- testFX will serve click macro of JavaFX, So we don't need to click each button manually for testing fuctionality of our program.

Project Review

- Calendar Developer, Seongmin Kim (김성민)

> Very interesting project ever I had. All member did perfect jobs assigned to each other. and perfectly did cooperation, program testing, issuing many problem with github. Thanks for all team members. I want to see you all this team members again another class by a new perfect team. For this project, we challenged to use many support pre-developed service to make good to our project development and service. For example, Firebase(real-time database for synchronization), JavaFX(Efficient Interface design library). and developed all member class and database with MVC, Observer, Singleton pattern(design patterns). and we did it all. thanks for everyone.

- ScheduleManager Developer Junho Song(송준호)

> It was very interesting job. Every member did the given jobs, and with assembling them. I was so happy with watching the process that each part was gathered, and the result is complete program. I think this is the attraction of the team programming.
By doing this job, I met many challenges about unknown libararies, techiniques, and so many architects. But, with Google, and help from other teammates are the solution of the problems. I learned many things from this project, for example JavaFX, Firebase and etc.
My teammates also do so many things. No one got lazy, Each person do the given jobs so perfectly. So I don't have to do thing out of my range. I really appreciate to my all teammates.

- UI(JavaFX Developer) Jaehong Lee(이제홍)

> This is first time I participate in this kind of software project implementation. Communication with team member was good, progressing with github was fun. It was very fresh experience.

- Time Line Module Developer, Sebin Kim(김세빈)

> The project was straight. Beginning of the project, – the design part - it was locked up. During the development, some of features might be added for the convenience but still main frame was not changed. The goal was always fixed. The existence of the leader of the project means a lot to team members and the project itself.
Manage the schedule of members and aim of rightful project direction. Leader and vice (I would like to call vice – leader) took lots of their time on this project for managing and modifying faults of codes. Also, they encouraged other members to focus on current task and solved encountered problems. I believed that with their hard effort, other members including myself could focus on our own tasks (at least I believe).

Personally, if I pick the problem of our project, I will pick myself. First, leaders did huge role in the project. I think my own skill or technique, and effort were insufficient compare to them. Lack of understanding and unfamiliarity of language. In fact, I did not recognize the big picture of the program, I just wrote code only based on current task.

The important of design level, The role of leader and conversation.

In conclusion, I give thanks to all members of our team. Especially to leaders.

- **Task List(Topologically sorted task list module), Choi-ji Hyeok(최지혁)**

>This was the first time to me making a whole completed software with full UI. Frankly, it made me worried whether I can make this well. However, the more I developed, the more confident I could make it. Plus, I didn't know there are many support-software such as 'Maven', 'Firebase' and so on. These reminded me of the professor's saying. "Software developers must not make their software from the ground up. They must use supporting tools well." It helps a lot while doing this project. I really appreciate on attending this lecture and developing a software with the team members. From this project, I could get to know how to handle the project schedule, distribute the tasks, adjusting on new software.

Team Website

- gitHub Organization
- Website : https://github.com/pm-tool-se2019