TimeSprite

Status Reports and Project Submissions

Date: 08/10/2015 to 15/10/2015

University of the Pacific

COMP 188, Senior Project, 2015 Fall ---Yunpeng Zhang

Contents

[2 System Description and Project Overview 2](#_Toc437454969)

[3 Non-Functional Requirements 3](#_Toc437454970)

[3.1 Open Source 3](#_Toc437454971)

[3.2 Security and privacy 3](#_Toc437454972)

[3.3 backup 3](#_Toc437454973)

[3.4 internet 3](#_Toc437454974)

[4 Functional Requirements 4](#_Toc437454975)

[4.1 Stakeholders 4](#_Toc437454976)

[4.2 Actors and Goals 4](#_Toc437454977)

[4.3 Casual Use Case Descriptions 4](#_Toc437454978)

[4.4 Fully-Dressed Use Case Descriptions 4](#_Toc437454979)

[4.5 System Sequence Diagrams 7](#_Toc437454980)

[5 User Interface Design Sketches 12](#_Toc437454981)

[5.1 Interface Sketches 12](#_Toc437454982)

[5.2 Interface interaction 18](#_Toc437454983)

[6 Glossary of Terms 19](#_Toc437454984)

[7 References 20](#_Toc437454985)

# System Description and Project Overview

Workplace time management is a real challenge. Emails, texts, assignments, and even snack breaks prevent us from focusing on – and effectively executing – a single task at a time. For decades, countless people have used the Pomodoro technique to improve work and project productivity.

TimeSprite is a professional time manager App for iPhone and iPad. This App will help user manager their tasks and calendar more efficiency. It will split task to small tasks, and fill into free time before due date. Also, users can easy edit tasks to adjust calendar to their desire.

This App will make users calendar more efficiency and compact, that is why I create this awesome App.

# Non-Functional Requirements

## Open Source

This project is an open source project, which means any individual developer or group can access source code via free license. Any Individual developer or group can redesign, re-publish, and modify this project. This project will apply Apache License 2.0 (The Apache Software Foundation, 2004). The developer and group should follow this license to redistribution.

## Security and privacy

This project will use users’ data, but not collect data include:

Calendar Data.

After user accept TimeSprite access the calendar data on their device, However, TimeSprite will not collect data, and upload user data to any internet server.

## backup

TimeSprite will not back up any data. Because calendar application on iPhone or iPad can connect to iCloud drive, the system will automatic back up calendar to iCloud via Apple Account. If user want to back up any events data, TimeSprite recommend user use Apple account to back up to iCloud.

## internet

TimeSprite will not use internet. User can use it offline. However, for back up (see 3.3), system will use internet communicate with Apple iCloud server.

# Functional Requirements

## Stakeholders

* 1. Everyone who want to have an efficiency calendar to manager their time, and find best way to finish their tasks

## Actors and Goals

* 1. People who want to easy manager their time
  2. People who want finish their tasks efficiently

## Casual Use Case Descriptions

* 1. TimeSprite will response the user’s action from touch screen. The users will tap buttons on main screen, and application will go into screen which user’s choice. The main screen will be the first stage, and it will include but no limited those stages: Setup Profiles, Add Tasks, and Generate Calendar.
  2. For edit and delete task, those two use case will only be touched off on task details stage.

## Fully-Dressed Use Case Descriptions

**Pre-Condition for all use cases: iPhone or iPad is on→ TimeSprite is open to the Home Stage**

* Use Case: Setup profiles
  1. This use case will be touched off only at the first time when TimeSprite running!

1. User enters unavailable time for each day
2. User enters relax time for each day
3. User taps finished button → System: check input is legal → pull events and tasks from system calendar → set busy time → save unavailable time, relax time, and busy time to file

* Use Case: Add Tasks

1. User tap Add Tasks button
   1. Add Tasks stage show up
2. User enters task information
3. User tap save button
   1. System check input is legal
      1. Illegal input → display error message
         1. User tap OK, and change information, go back to 3. User tap save button case
      2. Input acceptable
         1. Save task information to file
         2. Pop up message ask user if want to add more tasks
            1. If user want to add more tasks, go back to 3. User tap save button case
            2. If user doesn’t want to add more tasks, go back to main stage

* Use Case: Generate Calendar

1. User tap generate calendar button
   1. “Generating….” Message show up
   2. System read user profile
      1. System read unavailable time, relax time, and busy time from profile
   3. system read tasks files
      1. system read task information from tasks files
   4. check task can be divided
      1. If task can be divided, then divided task to small parts in order can fill up free time. → assign small tasks to each free time space.
      2. If task cannot be divided, find enough free time to assign task
   5. Save each tasks to system calendar
   6. Read events from system calendar, and show calendar.

* Use Case: Edit Profile

1. User tap Edit profile button
   1. System read user profile
   2. show edit profile stage
   3. user edit profile
   4. User taps finished button → System: check input is legal → pull events and tasks from system calendar → set busy time → save unavailable time, relax time, and busy time to file

* Use Case: Task details(This use case will use system calendar app)

1. User tap task in calendar stage
2. Show task details.

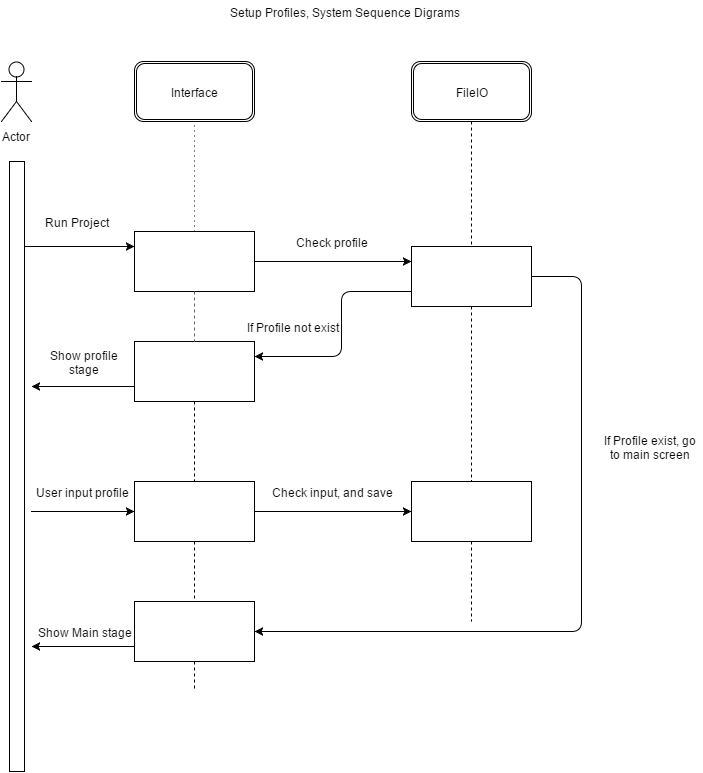
* Use Case in Task details stage(This use case will use system calendar app)
* Use Case: Edit Task(This use case will use system calendar app)

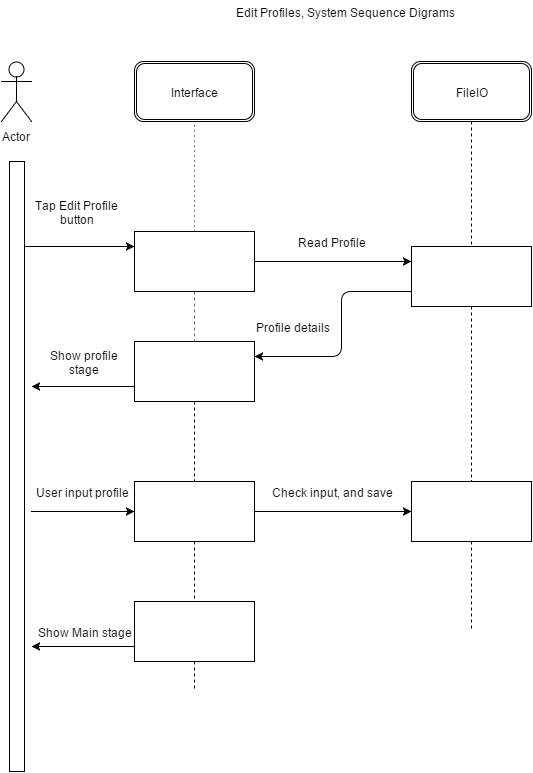
1. User tap Edit Task button →System read events from system calendar → System show calendar to stage
   1. User tap event to edit →system show edit stage
   2. User change event information
   3. User click save button → System check input is legal
      1. Illegal input → display error message
         1. User tap OK, and change information, go back to 3. User tap save button case
      2. Input acceptable
         1. Save task information to file
   4. Back to main stage

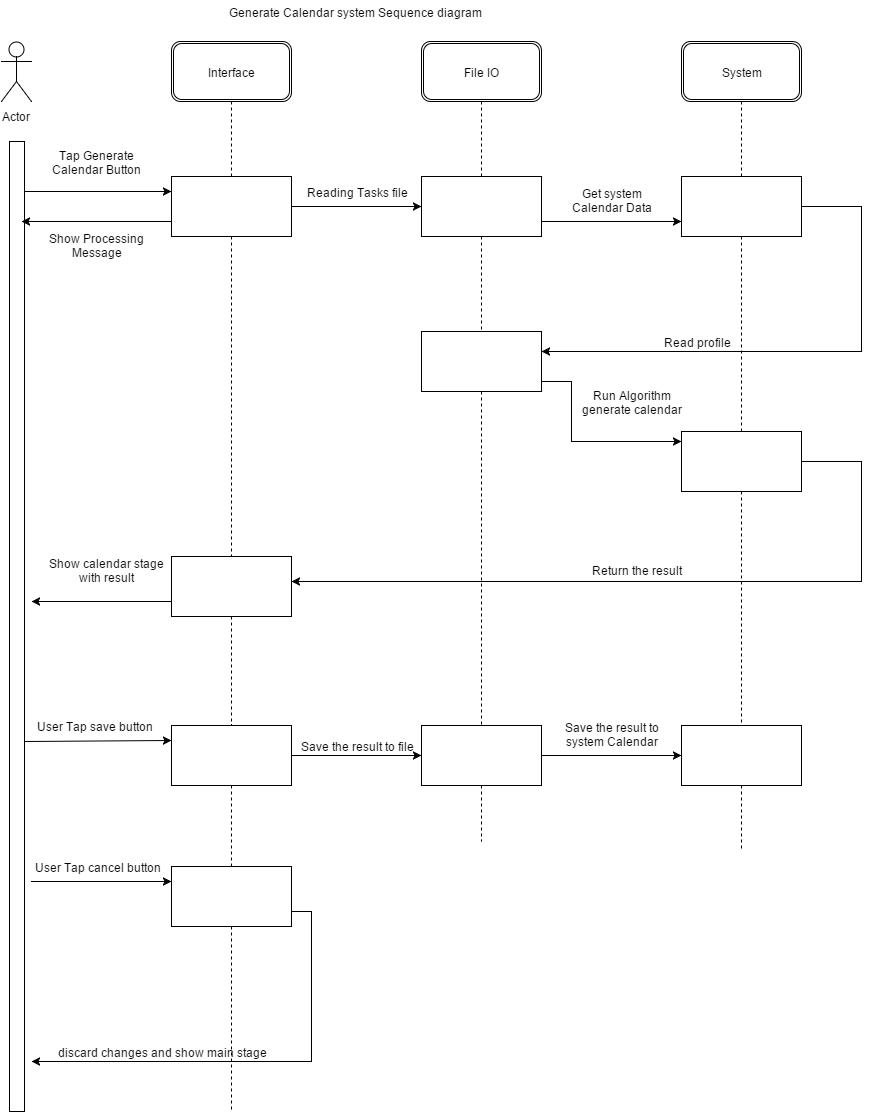
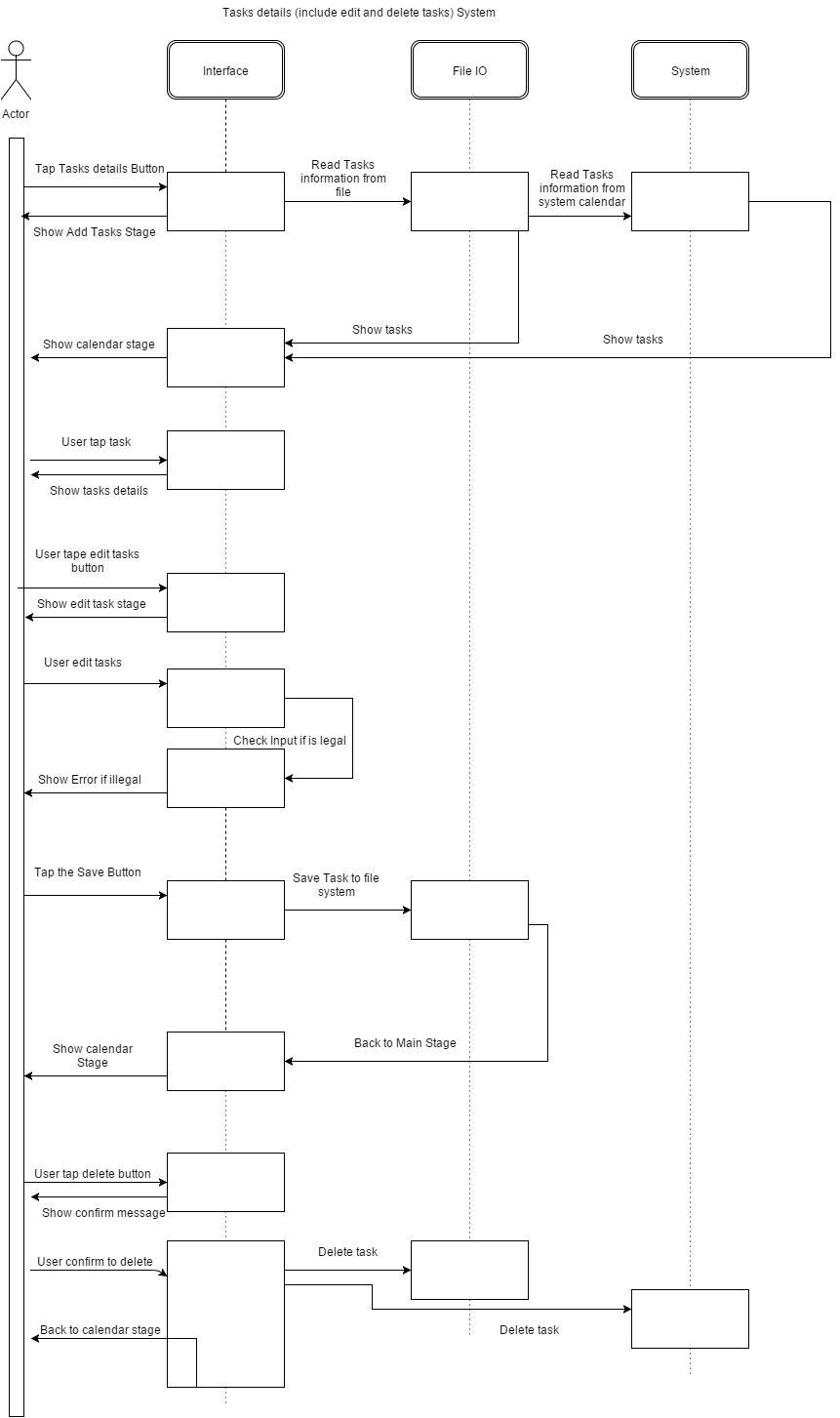
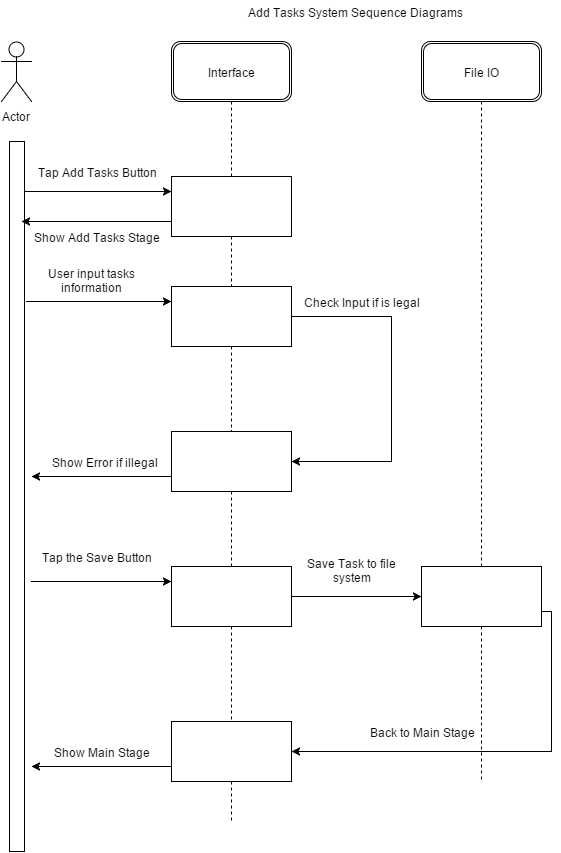
* Use Case: Delete Task(This use case will use system calendar app)

1. User click delete button → System pop up confirm message
   1. Yes → Delete task → back to show calendar stage
   2. No →keep this task → back to task details stage

## **System Sequence Diagrams**

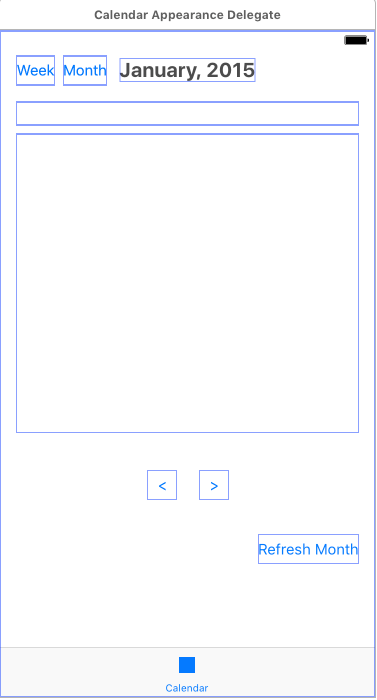
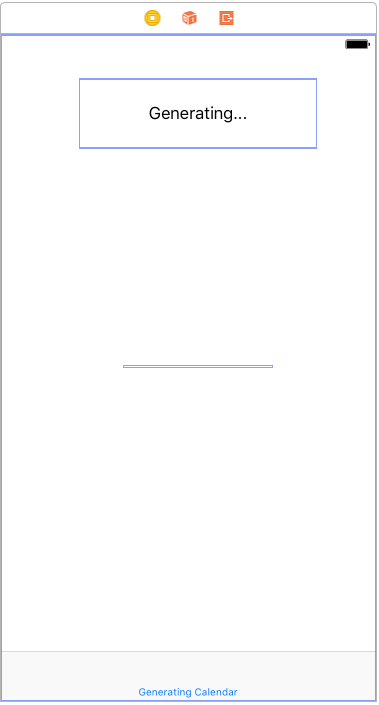
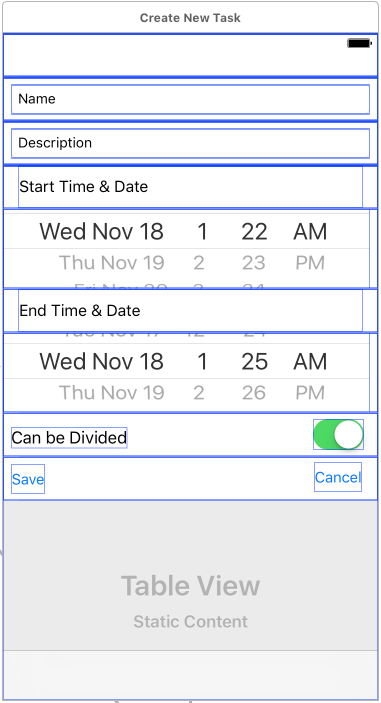
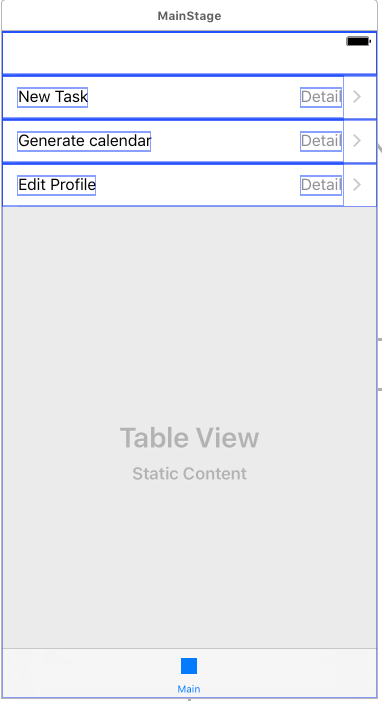
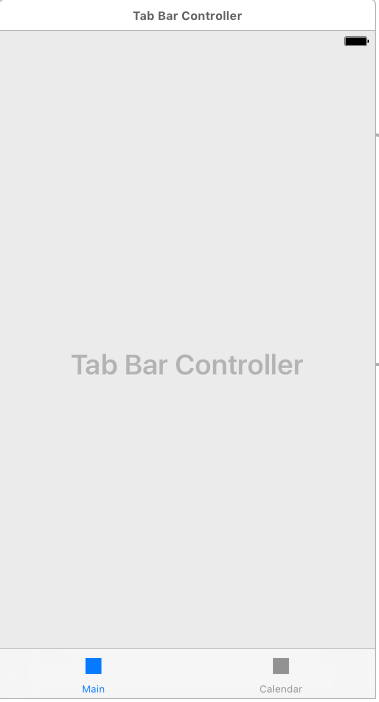
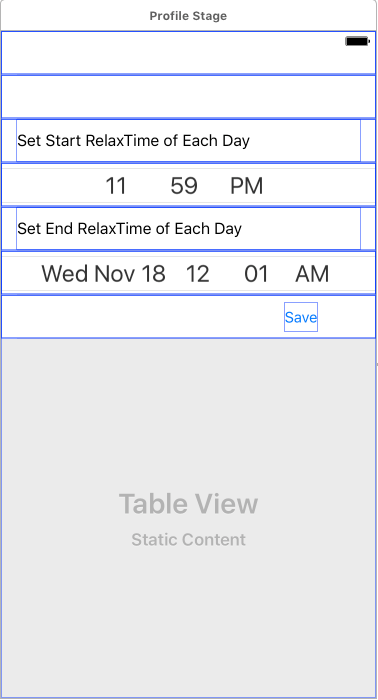






# User Interface Design Sketches

## Interface Sketches



## Interface interaction



# Glossary of Terms

1. Pomodoro: The Pomodoro Technique is a time management method developed by Francesco Cirillo in the late 1980s. The technique uses a timer to break down work into intervals traditionally 25 minutes in length, separated by short breaks. (Wikipedia, 2015)
2. Stage: Stage in this project means each view controller in Apple App. Each user interface screen defines each stage.
3. Unavailable time: Set by user. Unavailable time means in this period TimeSprite will not sign any tasks.
4. Busy Time: Busy time means in this period, there are events already sign by system calendar.

# References

The Apache Software Foundation. (2004, January). *The Apache Software Foundation*. Retrieved from Apache License, Version 2.0: http://www.apache.org/licenses/LICENSE-2.0.html

Wikipedia. (2015, October 3). *Pomodoro Technique*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Pomodoro\_Technique