

# **Software Requirements Specification**

**for**

**DailyLife**

**Version 1.0 approved**

**Prepared by JATH**

**Hoyeon Moon(Team Leader), Audris Gaerlan, Tyler VanLerberghe,  
Jaspinder Singh**

**03/09/2022**

## **1. Introduction**

### **1.1. Purpose**

The software is an application that allows users to plan activities and track their progress. The program will allow the user to visually assess their progress with a detailed and friendly user interface.

### **1.2. Document Conventions**

Each requirement has its own priority.

### **1.3. Intended Audience and Reading Suggestions**

This document is intended for all parties involved in the design, production, and maintenance of the software.

Design and Production teams should read the entire document in order. Testing teams may skip section 3.

### **1.4. Product Scope**

The software is an application that allows users to plan activities and track their progress. The program will allow the user to visually assess their progress with a detailed and friendly user interface.

### **1.5. References**

[Request For Proposal - University - 02/09/2022](#)

[Project Use Cases - Team JATH - 02/23/2022 -v1.0](#)

## **2. Overall Description**

### **2.1. Product Perspective**

The software is a standalone application. It will be written originally and entirely by Team JATH.

### **2.2. Product Functions**

- Add Checklist
- Change Checklist Date
- Delete Checklist
- Mark Checklist Status
- Add Tags to Checklist

- Search Tags
- Add Checklist Entry
- Delete Checklist Entry
- Edit Checklist Entry
- Disable/Enable Notifications
- Add Journal
- Edit Journal
- Lock Information
- View Locked Information
- Create PIN
- Edit PIN
- Go Back to Previous Page

### **2.3. Operating Environment**

This application will be a web application made to work on computers, more specifically Mac computers.

### **2.4. Design and Implementation Constraints**

The software must be able to run on MacOS machines.

### **2.5. User Documentation**

A short manual explaining the functions of the buttons and definitions of terms used within the software will be provided.

### **2.6. Assumptions and Dependencies**

The software is not dependent on any external factors, besides the assumption that the chosen programming language remains stable and secure.

## **3. External Interface Requirements**

### **3.1. User Interfaces**

Daily Life



search Tag



## March 2022

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Daily Life 3/9/2022

CHECKLIST







JOURNAL







Daily Life 3/9/2022



**CHECKLIST**   

In Progress 

(todo)   



**JOURNAL**   

(your journal content)

Daily Life

123



tag name



## March 2022

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Daily Life

123



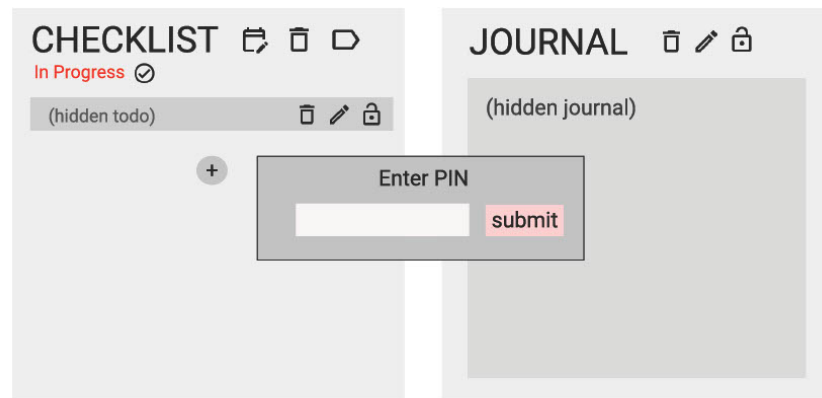
search Tag



## March 2022

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
		1	2	3	4	5
6	7	Enter PIN (pin you want to create) submit			11	12
13	14			17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Daily Life 3/9/2022



### 3.2. Hardware Interfaces

Computers that are able to run macOS High Sierra or above are able to make use of the desktop application.

### 3.3. Software Interfaces

The application does not require any software interfaces.

### 3.4. Communications Interfaces

The application does not require any communications interfaces since the application is fully local.

#### 4. System Features

- Checklist
  - Change due date
  - Delete checklist
  - Change status
  - Add tag
  - Add entry
    - Delete entry
    - Edit entry
- Search tag
- Notification
- Journal
- Lock information
- Go back to previous page

##### 4.1. Use Case 1

Use Case #1	Add Checklist	
Actor	User	
Pre-Conditions	Calendar must be displayed.	
Flow of Control	User clicks on a date on the calendar.	
		System switches screens and displays a “+” button to add checklist
	User selects the ‘+’ button to create a checklist.	
		System creates the new checklist, with a status of “In Progress”
Post Conditions	A new checklist is created, with a status of “In Progress”, on the date previously selected by the user.	

Error Condition	System is unable to create the new checklist.	
Non-Func Requirements	Checklist is limited to only one per date.	

#### 4.2. Use Case 2

Use Case #2	Change checklist date	
Actor	User	
Pre-Conditions	<ul style="list-style-type: none"> <li>• A checklist must exist</li> <li>• Date View must be displayed</li> </ul>	
Flow of Control	User selects "Change Date" on desired checklist	
		System displays a mini calendar and a text box to enter the date to change the checklist to.
	User select the date they want the checklist to move to.	
		System changes the checklist to correspond with the new date.
Post Conditions	The checklist can now be found under the new date.	
Error Condition	The system cannot move the checklist to the correct date.	
Non-Func Reqts	Application is updated in under 5 seconds.	

#### 4.3. Use Case 3

Use Case #3	Delete checklist	
Actor	User	
Pre-Conditions	<ul style="list-style-type: none"> <li>• Checklist must exist.</li> <li>• Date view must be displayed</li> </ul>	
Flow of Control	User clicks delete checklist button on the checklist's panel	



		System asks for confirmation to delete that checklist
	User selects "Confirm"	
		System deletes the checklist.
Post Conditions	The checklist is deleted, and no longer exists.	
Error Condition	Selected checklist cannot be deleted.	
Non-Func Reqts	Application is updated in under 5 seconds.	

#### 4.4. Use Case 4

Use Case #4	Mark Checklist Status	
Actor	User	
Pre-Conditions	<ul style="list-style-type: none"> <li>• Checklist must exist.</li> <li>• Date View must be displayed</li> </ul>	
Flow of Control	User selects "Change Checklist Status"	
		System displays statuses available for this checklist
	User selects which status to apply to the checklist	
		System applies the status to the checklist.
Post Conditions	The selected checklist is marked as the selected status.	
Error Condition	<ul style="list-style-type: none"> <li>• Checklist Status cannot be changed.</li> <li>• Desired Status is not displayed</li> </ul>	
Non-Func Reqts	Application updates in 5 seconds	

#### 4.5. Use Case 5

Use Case #5	Add Tags to Checklist	
Actor	User	
Pre-Conditions	<ul style="list-style-type: none"> <li>• Checklist must exist.</li> <li>• Date View must be displayed</li> </ul>	
Flow of Control	User selects "Add Tags"	
		System displays a text box for the user to name the tag.
	User typing in the tag they wish to add, then clicks submit.	
		System adds the tag/tags on the checklist
Post Conditions	The checklist now has updated tags.	
Error Condition	An invalid tag is added.	
Non-Func Reqts	Application updates in 5 seconds	

#### 4.6. Use Case 6

Use Case #6	Add Checklist Entry	
Actor	User	
Pre-Conditions	<ul style="list-style-type: none"> <li>• Date must be selected to get to the correct checklist.</li> <li>• Checklist must exist on the given date.</li> </ul>	
Flow of Control	User will click to add an entry	
		System will take in the input and add it to the entry
Post Conditions	A new entry is added to the correct checklist	
Error Condition	The entry does not appear where specified or the entry is not saved	
Non-Func	Application will update in 5 seconds	

Reqs		
------	--	--

#### 4.7. Use Case 7

Use Case #7	Delete Checklist Entry	
Actor	User	
Pre-Conditions	<ul style="list-style-type: none"> <li>There must be an entry to be deleted</li> <li>Date view must be displayed</li> </ul>	
Flow of Control	User will select the entry they wish to delete	
		System will delete chosen entry
Post Conditions	The entry will now be deleted	
Error Condition	The entry is not deleted	
Non-Func Reqs	Application is updated in 5 seconds	

#### 4.8. Use Case 8

Use Case #8	Edit Checklist Entry	
Actor	User	
Pre-Conditions	<ul style="list-style-type: none"> <li>An entry must be created to edit</li> <li>User must be editing a checklist</li> </ul>	
Flow of Control	User selects an entry on the checklist that they wish to edit.	
		System displays text fields for the user to edit data in the entry.
	User clicks submit.	
		System updates entry with the new information.
Post	Entry's data reflects the changes the	

Conditions	user made.	
Error Condition	System is unable to update the entry.	
Non-Func Reqts	Application updates in 5 seconds	

#### 4.9. Use Case 9

Use Case #9	Search tags	
Actor	User	
Pre-Conditions	<ul style="list-style-type: none"> <li>Tags must be present on the checklist they wish to search</li> <li>Calendar View must be displayed</li> </ul>	
Flow of Control	User clicks search bar from the main page and types in tag name	
		System will search for the tags and highlight dates with items corresponding with the tags
Post Conditions	User will be able to find the correct item	
Error Condition	Wrong tags are found	
Non-Func Reqts	System will find the corresponding tags within 30 seconds.	

#### 4.10. Use Case 10

Use Case #10	Disable/Enable notifications	
Actor	User	
Pre-Conditions	Calendar must be displayed	
Flow of Control	User clicks on notifications toggle	
		System either enables or disables notifications, based on the state of the button.

Post Conditions	<ul style="list-style-type: none"> <li>• If DISABLED: The system will no longer send notifications.</li> <li>• If ENABLED: The system will send notifications.</li> </ul>	
Error Condition	Notification status is unable to be changed	
Non-Func Reqts	Application updates in 5 seconds	

#### 4.11. Use Case 11

Use Case #11	Add Journal	
Actor	User	
Pre-Conditions	The Calendar must be displayed	
Flow of Control	User clicks on a date on the calendar.	
		System shows “+” button on the journal option
	User clicks on “+” button to add journal	
		System shows text box for user to write journal
	User clicks submit after done writing	
		System displays saved journal
Post Conditions	Journal will be saved to the application and displayed	
Error Condition	User cannot write(add) journal User cannot submit journal	
Non-Func Reqts	Journal is limited to only one per date.	

#### 4.12. Use Case 12

Use Case #12	Edit Journal	
--------------	--------------	--

Actor	User	
Pre-Conditions	<ul style="list-style-type: none"> <li>• Date view must be displayed</li> <li>• A journal must exist</li> </ul>	
Flow of Control	User selects "Edit Journal"	
		System allows the user to edit the journal
	User can click the "done" button when finished editing the journal	
		System saves edited journal
Post Conditions	The journal is edited according to the new data input	
Error Condition	The changes are not saved in the journal because of a system error	
Non-Func Reqts	The system must update the journal in under 5 seconds.	

#### 4.13. Use Case 13

Use Case #13	Lock information	
Actor	User	
Pre-Conditions	<ul style="list-style-type: none"> <li>• User must open the program</li> <li>• User must have created PIN</li> <li>• User must have created at least one checklist/entry, or journal</li> </ul>	
Flow of Control	User clicks lock button on the entry that user want to lock	
		Lock symbol changes and information is hidden
Post Conditions	Information that user chose is locked and cannot be seen without PIN entered	
Error Condition	<ul style="list-style-type: none"> <li>• User hasn't created PIN</li> <li>• Information is unable to be locked.</li> </ul>	

Non-Func Reqts	System should update the entry's status and save it within 3 seconds	
----------------	--	--

#### 4.14. Use Case 14

Use Case #14	View information that is locked	
Actor	User	
Pre-Conditions	<ul style="list-style-type: none"> <li>User must open the program</li> <li>User must have previously locked information</li> </ul>	
Flow of Control	User clicks on the unlock button	
		System displays pop up prompts for user to enter the PIN
	User enters PIN and click unlock(ok) button	
		System closes the pop up prompt and shows the hidden information
Post Conditions	The information is now unlocked	
Error Condition	<ul style="list-style-type: none"> <li>User entered wrong PIN</li> <li>Unable to unlock information</li> </ul>	
Non-Func Reqts	System should update the entry's status and save it within 3 seconds	

#### 4.15. Use Case 15

Use Case #15	Create PIN	
Actor	User	
Pre-Conditions	User must open the application	
Flow of Control	User clicks PIN button on the main page	
		System displays prompts with the message "Enter 4 digit PIN you want to create"

	User enter PIN then clicks the submit button	
		System displays confirmation message
Post Conditions	User created a PIN	
Error Condition	<ul style="list-style-type: none"> <li>User enters a PIN that is more than 4 digits.</li> <li>User enters a PIN that uses characters other than standard digits.</li> </ul>	
Non-Func Reqts	System must update user's password in 3 seconds	

#### 4.16. Use Case 16

Use Case #16	Edit PIN	
Actor	User	
Pre-Conditions	User must open the application User must have created PIN	
Flow of Control	User clicks PIN button on the main page	
		System displays prompts with the message "Enter 4 digit PIN to change the PIN"
	User enter the old PIN then clicks the submit button	
		System displays prompts with the message "Enter 4 digit pin that you want to change"
	User enter the new PIN and click submit button	
		System displays confirmation message
Post Conditions	User changed PIN	



Error Condition	<ul style="list-style-type: none"> <li>• User enters wrong old PIN</li> <li>• User enters a new PIN that uses characters other than standard digits</li> <li>• System displays wrong prompts</li> </ul>	
Non-Func Reqs	System must update user's password in 3 seconds	

#### 4.17. Use Case 17

Use Case #17	Go back to previous page	
Actor	User	
Pre-Conditions	User must have a page in view, having left a previous page.	
Flow of Control	User clicks the "Previous Page" button.	
		System displays the previous page.
Post Conditions	System displays the previous page.	
Error Condition	System is unable to display the previous page.	
Non-Func Reqs	Users should be able to go back to the very first page they visited.	

### 5. Nonfunctional Requirements

- Checklist is limited to only one per date.
  - Used in use case #1
- Application is updated in under 5 seconds
  - Used in use case #2, #3, #4, #5, #6, #7, #8, #10, #12
- System will find the corresponding tags within 30 seconds
  - Used in use case #9
- Journal is limited to only one per date
  - Used in use case #11
- System will update and save it under 3 seconds

- Used in use case #13, #14, #15, #16
- User's will be able to go back to a previous page
  - Used in use case #17

**6. Other Requirements**

Some simple documentation about the application's use must be provided.