

## Criterion B → Design

### Input & Outputs:

Central UI Input:

Input	Data type	Example
Name	String	Haruto
Day	Int	18
Month	Int	01
Year	Int	2019
Task	String	Computer Science IA
Nickname	String	Hal
Age	Int	18
Task type	String	Study

Central UI Output:

Output	Data type	Example
Name	String	Haruto Tanaka (Hal)
Date	String (in integer form)	18 / 01 / 2019
Task	String	Computer Science IA (Study)
Sort	String	Result in table
Filter	String	Result in table
Help	String	***Instructions***

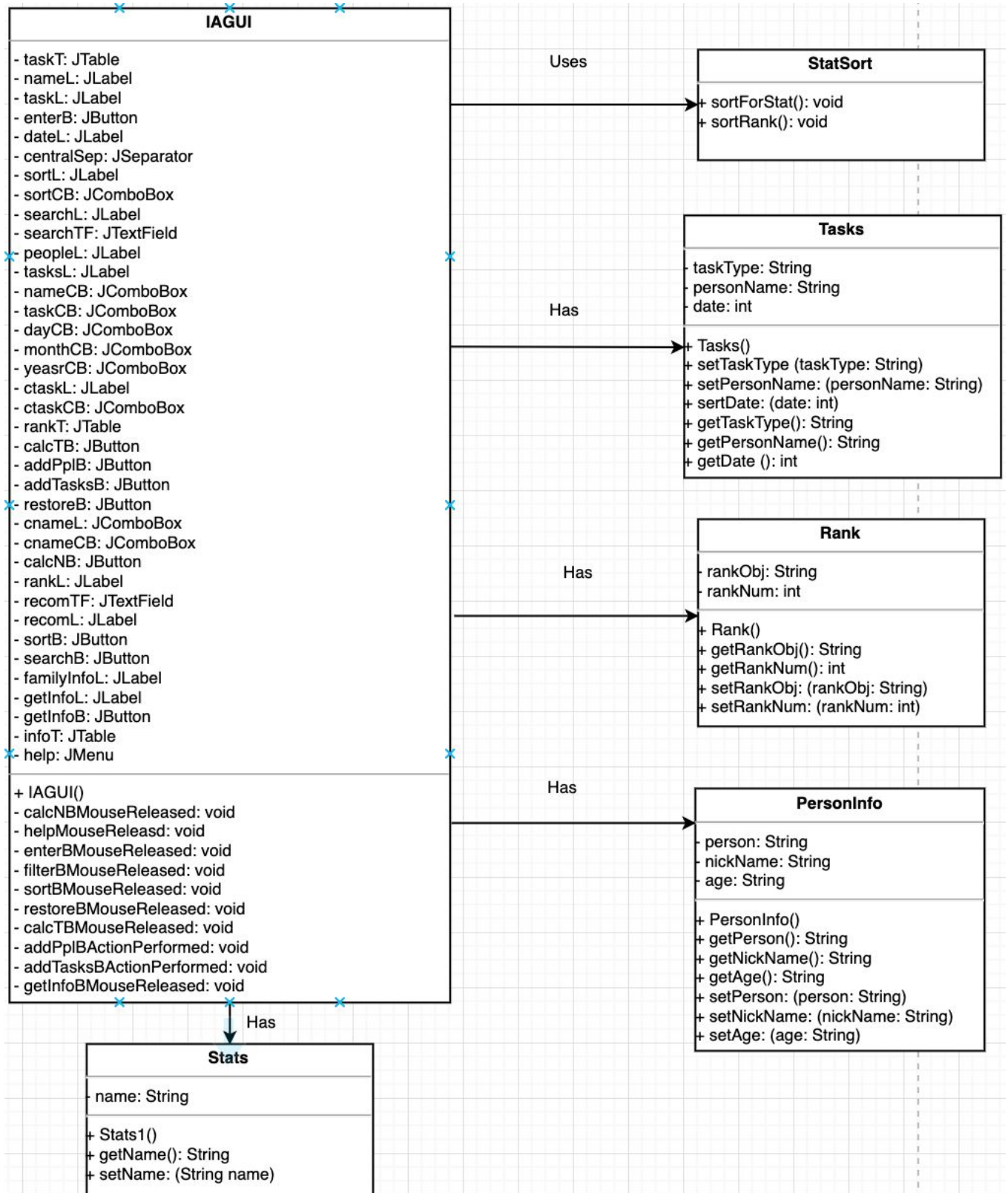
Statistics Input:

Input	Data type	Example
Task	String	Computer Science IA
Name	String	Study

Statistics output:

Output	Data type	Example
Name rank	String	Shows the name of people who should do the input task in Table
Task rank	String	Shows the ranked tasks in Table which shall be done by person who inputted
Recommendation (person ver)	String	“The task: Computer Science IA should be done by Haruto”
Recommendation (task ver)	String	“Haruto should do the Computer Science IA”
List of family member	String	Shows the basic information related to family member (name, nickname, and age)

## UML Diagram:

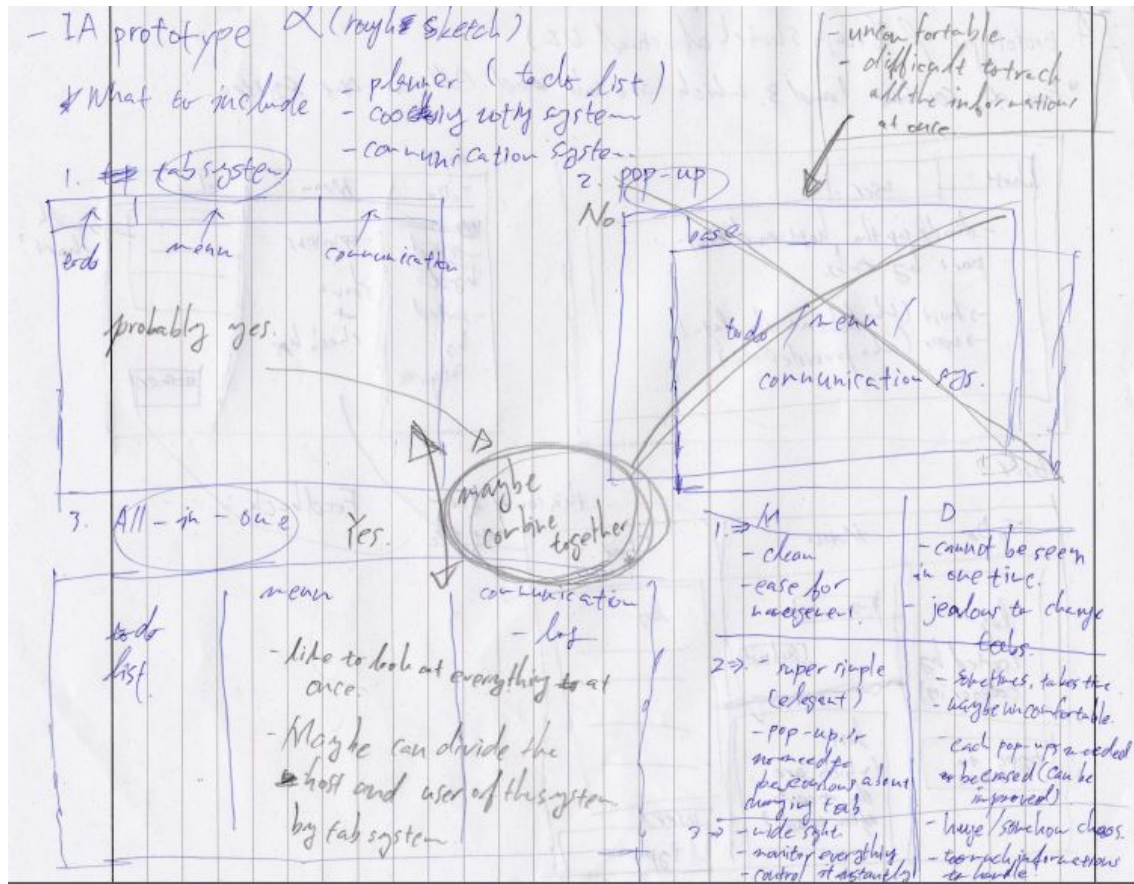


**Prototype Process:**

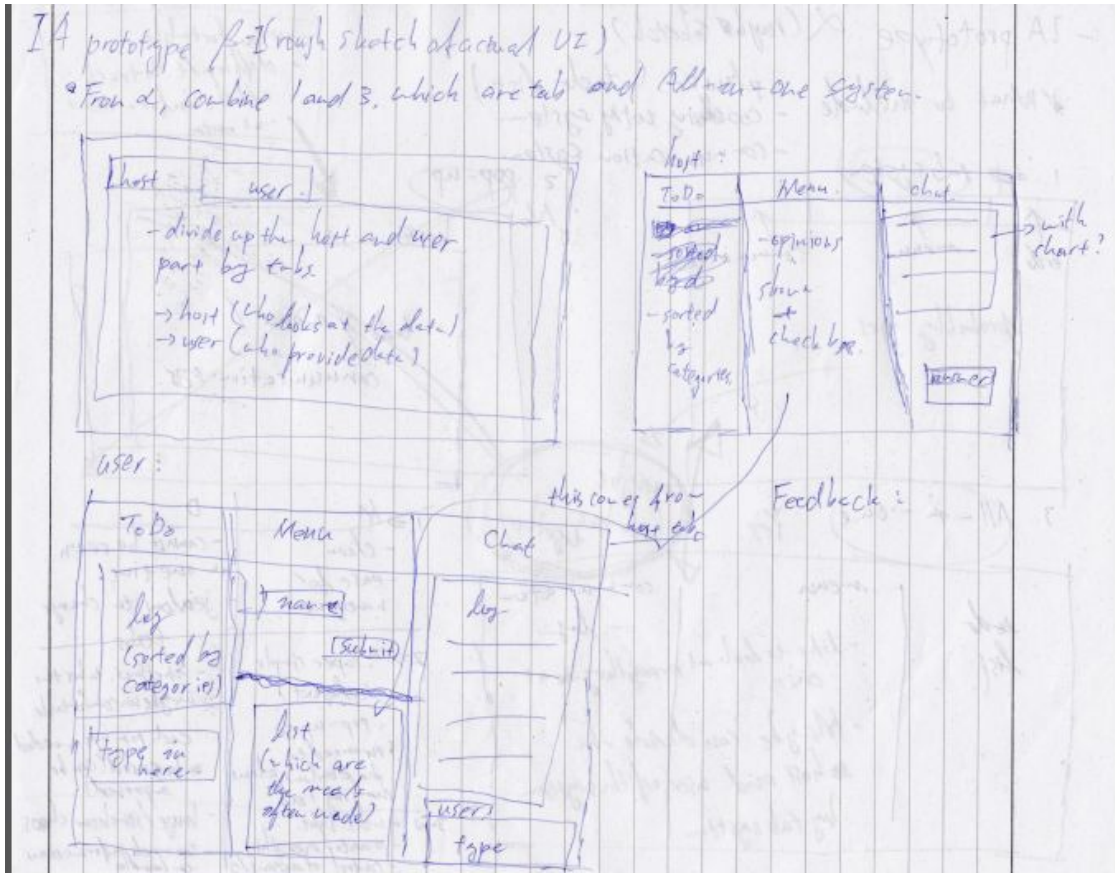
1. The original prototype without any comments or advices
2. The prototype with the comments

Three prototypes made prior to each interview sessions:

1: Revised version (Ver.)  $\alpha$  - I (for first interview)

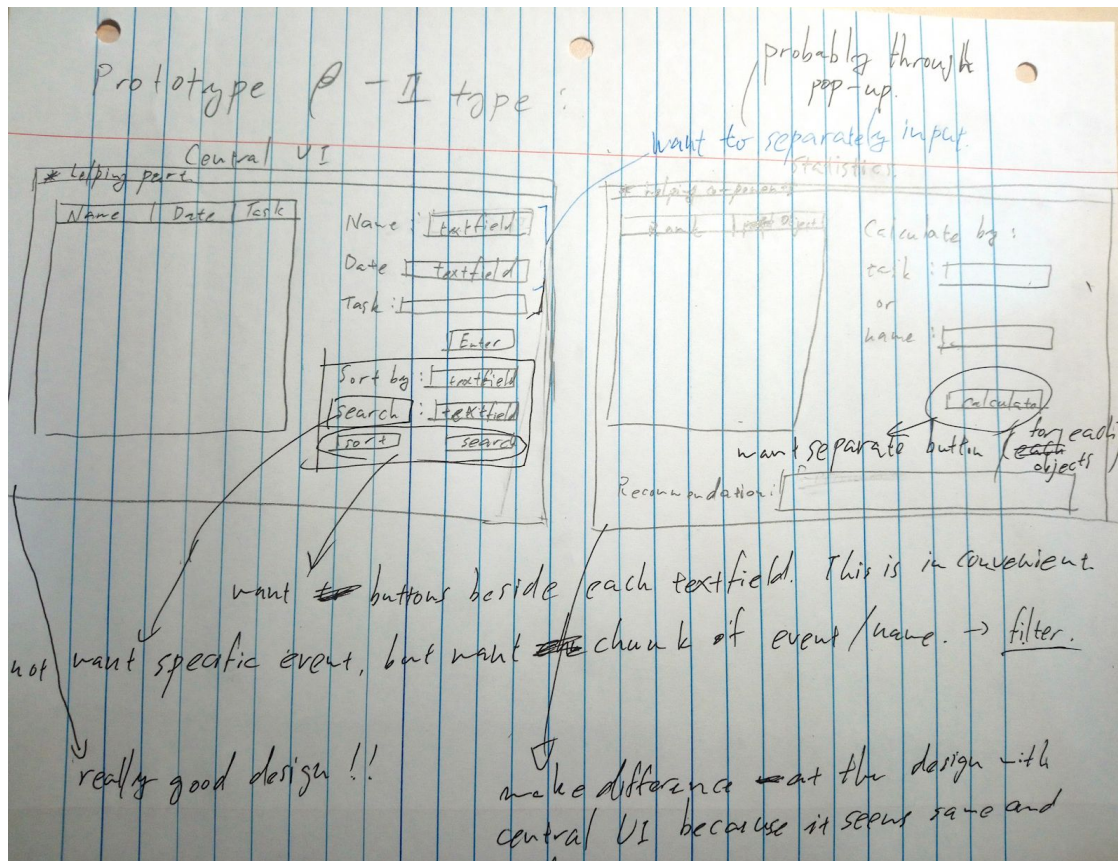


## 2: Revised Ver. $\beta$ - I (Initial idea / for second interview)





### 3: Revised Ver. $\beta$ - II



### 3. Change made in the prototype

- $\alpha$  - I  $\rightarrow$   $\beta$  - I
  - First prototype  $\alpha$  - I was specifically made for first interview. In order to get what client wants, three styles of Graphical User Interface (GUI) were made; tab, pop-up, and all-in-one. Through the interview process, client found that combination of tab and all-in-one might have high accessibility and much comfortable. Thus, in prototype  $\beta$  - I, both attributes combined and abstract features with sentences added.
- $\beta$  - I  $\rightarrow$   $\beta$  - II
  - At  $\beta$  - I, there were only words showing what the features are, so there were none of the GUI visualized. As the second interview progressed, client did like ToDo function, but due to the high frequency that every family member directly communicate, client did not feel any demand for chatting and menu function. In  $\beta$  - II, unnecessary functions removed and focused more on ToDo, statistics functions.
- $\beta$  - II  $\rightarrow$  type  $\phi$ 
  - The difference between  $\beta$  - II and type  $\phi$  are significant at the system for adding people and tasks (adding pop-up system for it). Otherwise, some gimmicks are added or deleted based on client's request.

4: Final Prototype (type φ) \*\*\*This can be varied with finalized one.\*\*\*

File Edit Help

Central UI Statistics

Name	Date	Tasks
------	------	-------

Name:

Date:

Task:

Enter

Sort by:

Search:

---

Add people:

Add Tasks:

File Edit Help

Central UI Statistics

Choose tasks:

Calculate tasks

Choose name:

Calculate Name

Get family member information:

↓Recommendation↓:

Recommendation ranking:

Rank	Object
------	--------

Family member information:

Name	Nickname	Age
------	----------	-----

---

**\*\*\*This part is not included in the word count\*\*\***

Second interview with client:

1. What features do you want to add or delete?
    - a. I would like to delete the menu and chatting function because we communicate those things mainly in direct way.
  2. Do you feel comfortable for this system?
    - a. Mostly yes, but partially no.
  3. If you feel comfortable, which part did you feel most comfortable?
    - a. The part where the Central UI and Statistics separated by tabs.
  4. If you feel any uncomfortable part, which part did you feel most uncomfortable? Why?
    - a. As said, because family member frequently communicate with each other directly, there is no demand for menu and chatting functions.
  5. How do you feel about that Central UI and Statistics windows are separated?
    - a. I think it works well.
  6. Any comments on the prototype?
    - a. I would like to have a better system to collect the family member name and tasks.
-



**Record of Task:**

<b>Criterion B: Record of tasks</b>					
Task number	Planned action	Planned outcome	Time estimated	Target completion date	Criterion
1	Find a client	Have a client	A week	Dec. 7th	A
2	Initial Interview	Know the problem better	Half an hour	Dec. 9th	A
3	Make prototype	Prototype to take to client	An hour	Dec. 10th	A
4	Problem statement	Specify problem in narrative text	15 minutes	Dec. 12th	A
5	Describe the scenario	Know the context better	15 minutes	Dec. 12th	A
6	Criteria for success	List of features to be implemented	30 minutes	Dec. 12th	A
7	Additional interview against the client	Coloring function added	30 minutes	Dec. 29	A
8	Input & Output list	Input & Output	30 minutes	Jan 11	B
9	Class Diagrams	Get some class diagrams	1.5 hour	Jan 11	B
10	Prototyping Process	Show off how prototype released	20 minutes	Jan 11	B
11	Chronological Development Plan	Development plan	1.5 hour	Jan 11	B
12	Testing Plan	Testing plan	45 minutes	Jan 11	B
13	Programming I	Template class	1 hour	Jan 23	P
14	Programming II	Fundamental systems (tables, combo box, etc)	2 hours	Jan 25	P
15	Programming III	Connecting systems	2 hours	Jan 27	P
16	Programming IV	Statistics	1 week	Feb 5	P
17	Programming V	Error handling	2 week	Feb 23	P

18	Development I	Structure of program & Structure of data	1 hour	Mar 3		C
19	Development II	Unique algorithm & GUI work	1.5 hours	Mar 19		C
20	Video	Video of product	1 hour	Mar 25		D
21	Evaluation	Section E	1 hour	Mar 27		E
22	Final check	Finalized products	1 hour	Mar 28		All

**Chronological Development:**

<p>_____</p> <p>Tabs (panes)</p> <p>_____</p> <p>Tasks</p>	<p>Central UI</p>	<p>Statistics</p>
#1 - Construction of fundamental GUIs (1 day)	All the GUIs are completed and settled as the final prototype (refer to final prototype above)	All the GUIs are completed and settled as the final prototype (refer to final prototype above)
#2 - Completion of fundamental features (2 weeks)	<p>(All codes are to be coded into the GUI at this level)</p> <ul style="list-style-type: none"> <li>● Name, Date, Task properly shown on the table immediately when the “Enter” button is hitted</li> <li>● Sort when the “Sort” button is hitted</li> <li>● Filter the object when the “Filter” button is hitted</li> <li>● Pop-up shown when “Add people” button is hitted               <ul style="list-style-type: none"> <li>○ When the “Add” button in that pop-up is hitted, the Name should be added into combo box of “Name” section</li> </ul> </li> <li>● Pop-up shown when “Add Tasks” button is hitted               <ul style="list-style-type: none"> <li>○ When the “Add” button in that pop-up is hitted, the name of Tasks should be added into combo box of “Task” section</li> </ul> </li> </ul>	<p>(All codes are to be coded into the GUI at this level)</p> <ul style="list-style-type: none"> <li>● Whenever the “Calculate tasks” button is hitted, the Date and name of people who are recommended to do is ranked in that system</li> <li>● Whenever the “Calculate Name” button is hitted, the Date and tasks who should do is ranked in that system</li> <li>● Recommendation should appear in certain form when the refresh button is hitted</li> </ul>
#3 - Error checking & handling (2 days)	<ul style="list-style-type: none"> <li>● GoTo → Testing Plan</li> </ul>	
#4 - Add/delete small features (1 week)	<ul style="list-style-type: none"> <li>● Add feature of “help” button, which guides how to use GUI by pop-up instruction</li> <li>● (For each tabs) Add the little features which requested by the client</li> </ul>	
#5 - Overall checking (2 - 3 days)	<ul style="list-style-type: none"> <li>● Check all potential errors, codes flaws, and GUI accessibility.</li> </ul>	

**Testing Plan:**

Input	Normal	Border	Abnormal	Extreme
Name	Haruto → Normally inputted.	No inputs → No inputs Ex. (blank)  <i>#Name will be accepted if characters exceeds or even equal to 2</i>	Character out of ASCII → No inputs Ex. ηαρθτο  <i>#Only English ASCII code is available for Name. If other characters typed in and submitted, error message comes out and request to retype it.</i>	Very long name → Input Ex. Pablo Diego José Francisco de Paula Juan Nepomuceno María de los Remedios Cipriano de la Santísima Trinidad Ruiz y Picasso  <i>#No limits on the length of name</i>
Tasks	Complete essay draft → Normally inputted	No inputs → No inputs Ex. (blank)  <i>#ToDo should be 2 characters or more. If no inputs, warning shown within the text field.</i>	Characters other than ASCII → Normally inputted Ex. Ψομπλετε εσσαυ δραφτ  <i>#ToDo do not have character limitation. Thus any languages can be written.</i>	Long ToDo object → Conditionally reject input Ex. Completing essay for Computer Science. Finish this first, then ...  <i>#ToDo should be under 25 characters. If exceeded, input rejected and request to retype.</i>
Nickname	Hal → Normally inputted	No inputs → Normally inputted Ex. (blank)  <i>#Nickname is only a conditional object, so it should be inputted if there is no inputs</i>	Characters other than ASCII → Normally inputted Ex. Ψθρρυ ριψε  <i>#Nickname can be other characters out of ASCII code.</i>	Long nickname → Conditional Ex. Chirashizushi, the steamed rice with ingredients of ...  <i>#Nickname is for shortening their original names, so it should not be</i>

				<i>longer than 15 characters</i>
Age	18 → Normally inputted	No inputs → No inputs Ex. (blank)  <i>#The warning shown in text field. Age should be a number and properly typed in</i>	Characters other than ASCII → No inputs Ex. +  <i>#Because the age should be typed in as integer form, characters other than numbers are rejected.</i>	Impossible numbers → No inputs Ex. 1200  <i>#Age should be under 120, and if the number exceeded, it shows the error text inside the text field and request for retype</i>

Word Count: 186\*<sup>1</sup>

---

<sup>1</sup> Considering that Prototype Process, Change made in the prototype part as sentences and not as bullet point