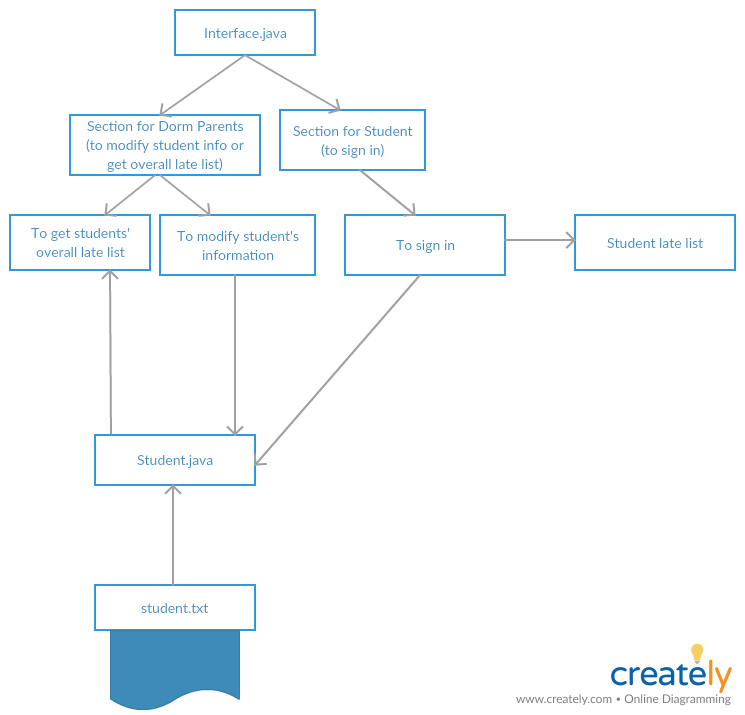
Criterion B: Record of tasks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task number | Planned action | Planned outcome | Time estimated | Target completion date | Criterion |
| 1 | Choosing a topic and discuss idea with future client | Decided the problem and confirm with the willingness of Ms. Xx to collaborate as the client of my project | 2 days |  | A |
| 2 | Discuss idea with computer science teacher | Teacher approved the idea | 3 days |  | A |
| 3 | First interview with Mr. xx | Clear description of the problem | 2 days |  | A |
| 4 | Confirm my solution for the client’s problem with computer science teacher | Java chosen to be used as a solution for client | 2 days |  | A |
| 5 | Define criteria for success and finish with Criterion A | Criteria for success is defined and finished Criterion A | 2 days |  | A |
| 6 | Thinking on design of a product | Start developing basic design and create a flow chart | 1 week |  | B |
| 7 | Create a schedule for developing the software | Develop schedule in discussion with Mr. xx about the software | 1 week |  | B |
| 8 | Work on interface | Finish designing interface and agreed with Mr. xx | 1 week |  | B/C |
| 9 | Draw diagrams and flow charts to clarify the program working process | Finish diagrams and process description flow charts; finish Criterion B | 1 week |  | B |
| 10 | Think about a test to check whether the program works | Finished Detailed test plan | 1 week |  | B |
| 11 | Start writing actual code | Finished source code of Student class and SignIn class | 1 week |  | C |
| 12 | Finding more information about ActionListner and Scanner methods | Researched information from the internet | 5 days |  | C |
| 13 | Develop code and test every time to assure that everything works fine | Different classes and methods are added in whole program with that makes the code more readable; function testing should be made in whenever a change is made | 4 weeks |  | C |
| 14 | Finish writing criterion C | Finished extended writing together with the source code | 1 week |  | D |
| 15 | Test the final product solution | Show Mr. xx the final product | 2 days |  | D |
| 16 | Interview with Ms. xx | Get fedback from Ms. xx: testing results and suggestions about further developments | 1 day |  | A/ E |
| 17 | Think about how to further improve the program | Written ideas on how to improve program in future | 1 day |  | A/D/E |
| 18 | Record video to show programs functionality | Video is recorded successfully | 1 day |  | D |

Criterion B: Design

**Program basic (initial) structure:**



**Input data (Dorm parents)**

* Dorm parents can either choose to modify student information or get students’ overall late list
* When choose to modify student information, dorm parents have to first input the student’s ID number

|  |  |
| --- | --- |
| Ex: Normal data  Student ID number: 1111111 (must be seven-digit) | Ex: Abnormal data  Student ID number: 13ueydghbkasdkv |

Limitation: Dorm parents can only modify one student information at a time. Also, dorm parents cannot separate overall late list of a student, but every student’s overall late list.

* Student.txt must be typed in this specific way

|  |  |
| --- | --- |
| Ex: Normal data  Red(tie color) 1111111(ID number) Nerissa(name) 0(total late time) | Ex: Abnormal data  1111111 Red 1 Nerissa |

**Input data (Students action)**

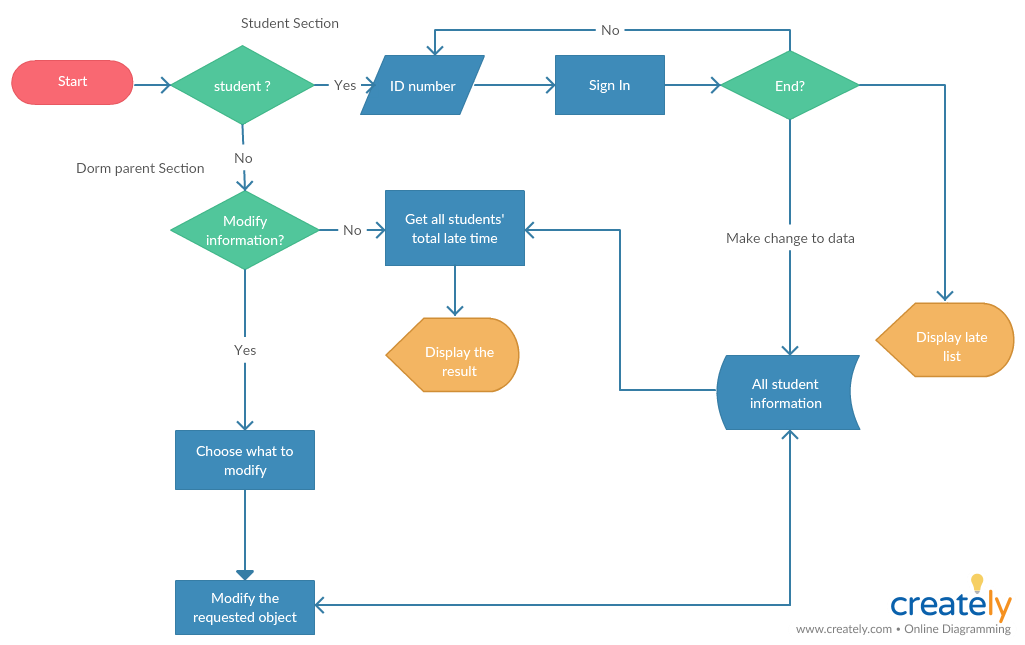
* Data- students will input their ID number in order to sign in

|  |  |
| --- | --- |
| Ex: Normal data  Your student ID number: 1111111 | Ex: Abnormal data  Your student ID number: 21ssdkvosdljv |

When student input their ID number, their late status variable in their Student object will change to false, so that later they will not appear in the returning late list.

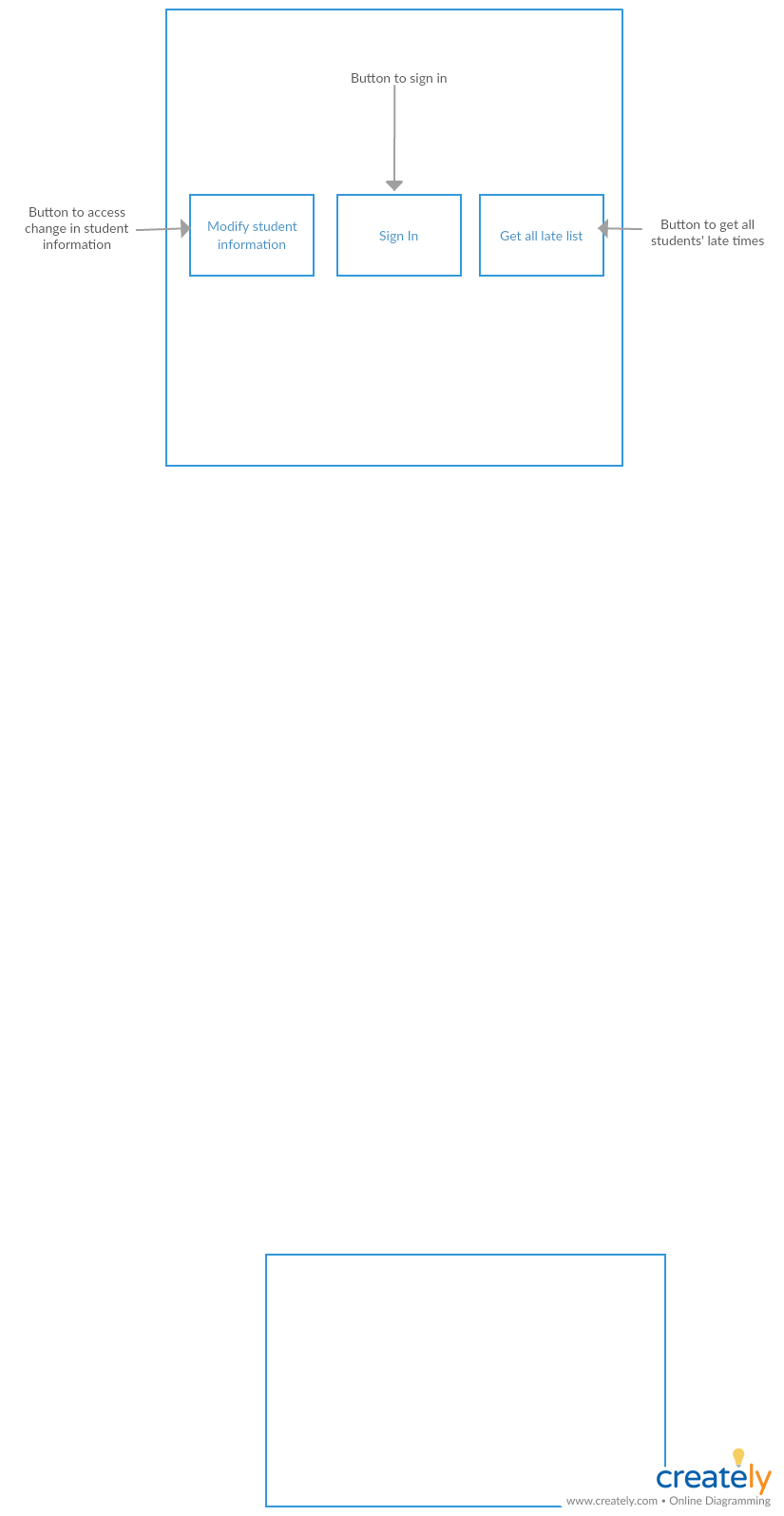
**Output data**

|  |  |
| --- | --- |
| Dorm parent’s section | Student Section |
| 1. a student late list in JTextArea will appear in a new window (can be printed) 2. student information can be modified as requested | 1. When the student sign in, they change their status in their information to false, so late they will not later identify as late |

**Flow Chart**

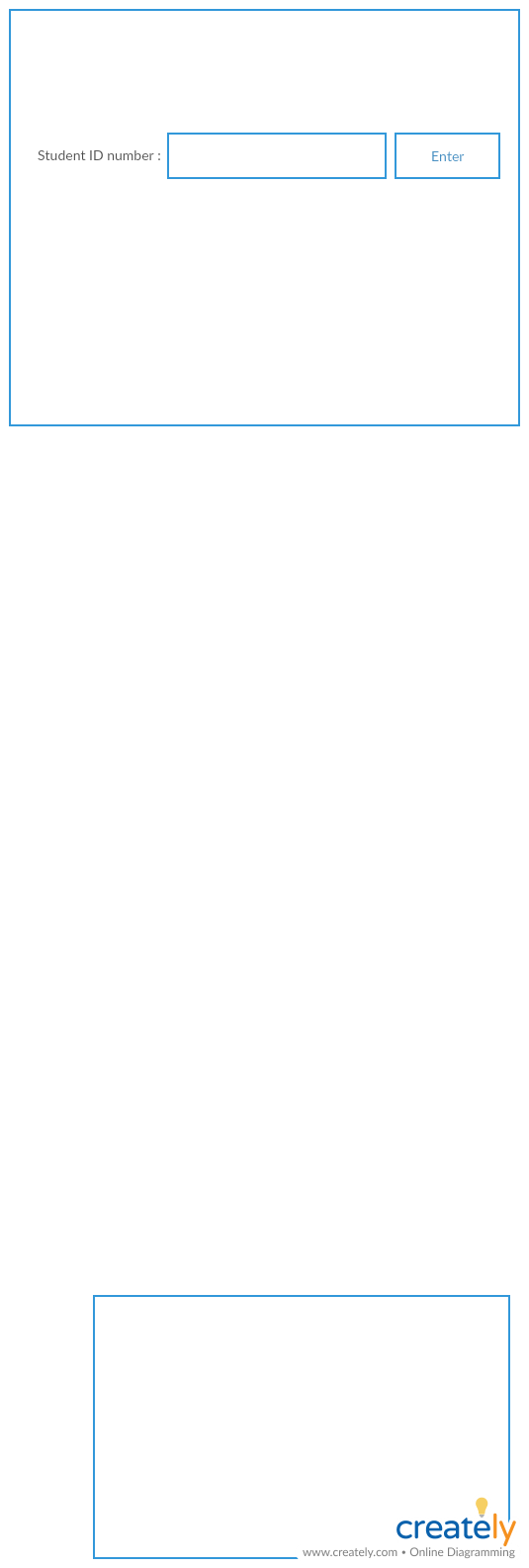
**Software Interface**

* Home window

****

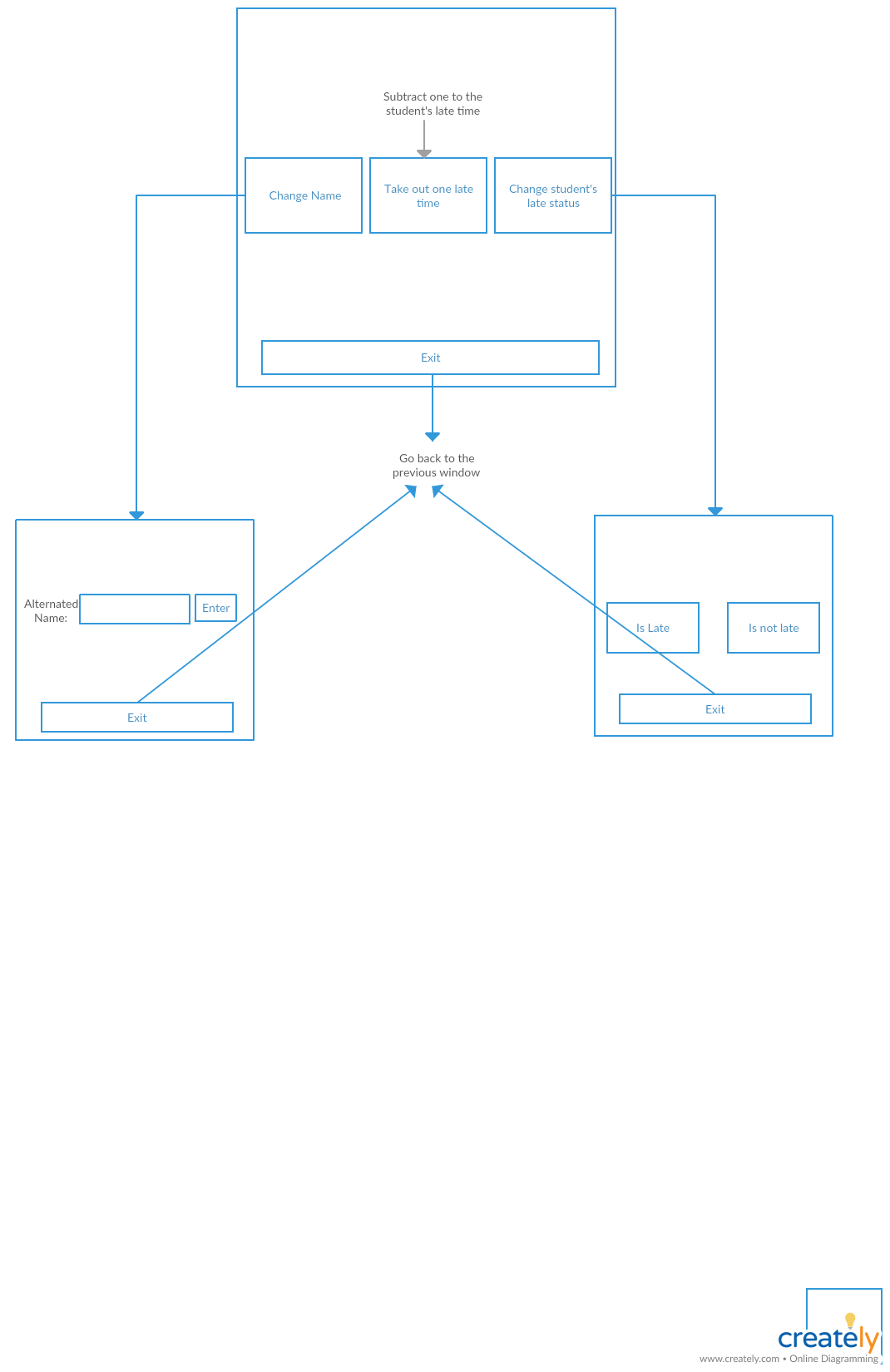
* Modify student information (for dorm parent’s section)

Dorm parents have to first clarify the student whose information they want to make change of.

****

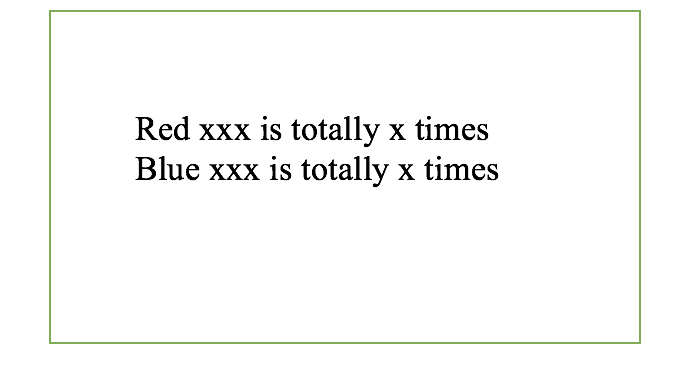
Exit

Then, the dorm parents can pick what they want to change among three options (requested by the client)

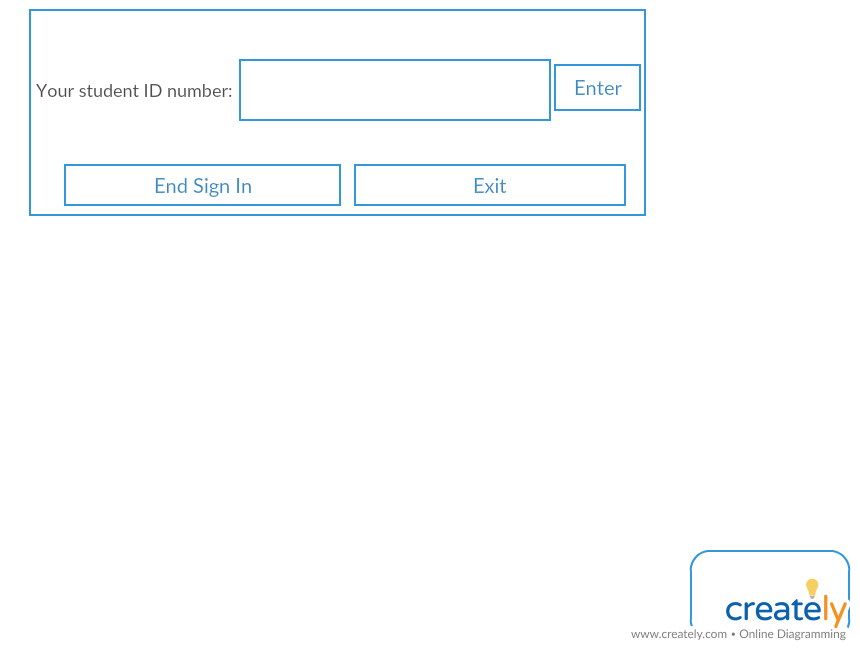


Get all late list (for dorm parent’s section)

* Result:

****

Sign in (for student’s section)



**Schedule for developing the program**

Program will be divided in three sections; Modify Student Information section for dorm parents to make change in student’s information; Get All Late List section for dorm parents to change the student’s late status; Sign In section for students to sign in for dinner

2 + 3 weeks will be enough to complete all sections:

|  |  |  |
| --- | --- | --- |
| Modify Student Information  (for dorm parent’s section) | Get All Late List  (for dorm parent’s section) | Sign In  (for student’s section) |
| * Create interface * Write a method that can find the student file by inputting the student’s ID number * Write code for exit button that allows user to go back to the previous window * Write code to prevent dorm parents from data entry errors | * Create interface * Write code that gets every student late time from the student file stored in Student object * Write code for exit button that allows user to go back to the previous window * Write a code that create a new window to display the data | * Create interface * Write code that will hold every student’s information from student.txt using Scanner and Student class * Write code for changing student late status and record the students who are late into an ArrayList * Write code for end sign button that create a new window that displays student who are absent list * Write code for exit button that allows user to go back to the previous window |
| Action Test | Action Test | Action Test |
| Test if the program runs correctly and appears on the main window | Test if the program runs correctly and appears on the main window | Test if the program runs correctly and appears on the main window |
| Check if the button works and leads to the right window | Check if the button works and leads to the right window | Check if the button works and leads to the right window |
| Check if the requested change is done and changed from the student.txt | Check if all the data is correct and compare it with the data in student.txt | Check if the students who sign in do not appear on the late list and appear those who supposed to do |
| Check if the program catches invalid student ID number | Check if the program has a proper size | Check if the program catches invalid student ID number |