My Task Planner App -

My Task Planner App

Iteration 3- Apr, 15,2018

Elham Hamid
Group 1

Contents:

Requirements

MTP is a mobile android application that gives you a calendar to manage your task (homework, test, assignments, projects, and important events) in which you can track your upcoming task, as well where you can sync your dues for (assignments and exams) from blackboard and receive reminder for tasks when they're due.

The purpose of MTP is to provide a single platform where students can manage all their task from all classes and important events and to remind their upcoming task.

R1. MTP shall allow a student to create an account and login from different devices.

R1.1 MTP shall allow student to type ID (email address), password, name, and the university name when creating an account.

R2. MTP shall allow a student to manage a task.

R2.1 MTP shall allow a student to create, edit, and delete a task based on variety type (class, Homework, tests, projects, events, meeting, personal work type, and others).

R2.2 MTP shall allow a student to set the due date and time of a task.

R2.3 MTP shall allow a student to set the priority of a task. (high, medium, low).

R2.4 MTP shall allow a student to set a time for how long it will take to accomplish a task and to decrement the given time when a student spends the time for the specific task.

R2.5 MTP shall allow a student to manage a daily list task.

R2.6 MTP shall allow a student to set a reminder/notification of a task based on his/her preferred date and time.

R2.7 MTP shall allow a student to add their percentage progress on the available task.

R3. MTP shall allow a student to view his/her task in a variety of display format.

R3.1 MTP shall allow a student to view his/her task by list or calendar.

R4. MTP shall allow a student to sync tasks from blackboard announcement and syllabus.

R5. MTP shall allow a student to log out when a student is completing the interaction with MTP.

Constraints

- Students should be able to provide their blackboard account to sync with MTP. sometimes blackboard may not allow to be accessed by a third system since it has confidential data. A student may need to call or change setting to sync blackboard with MTP.

Description

- Task: includes homework,project, test, quiz and important events.
- Weight: how long it takes. Heavy weight consumes more than 4 hours). Light is up to 4 hours.
- Manage includes add, edit/update and delete.

Use Case Modeling

<Abstract Use Cases>

1. Create an account - UC1: Create an account
2. Log in - UC2: Log in
3. Manage tasks - UC3: Manage tasks
4. Change the order of tasks - UC4: Change the order of tasks
5. Sync tasks - UC5: Sync tasks
6. Log out - UC6: Log out

<High Level Use Cases>

UC1: Create an account
TUCBW a student hits ‘n’ button.
TUCEW a student sees a confirmation.

UC2: Log in
TUCBW a student hits ‘l’ and type ID (email address) and password.
TUCEW a student sees the blackboard.

UC3: Manage a task
TUCBW a student hits the ‘n’, ‘e’, or ‘d’ button.
TUCEW a student comes back to the list.

UC4: Change the order of tasks
TUCBW a student sees the list in the blackboard, and hits the ‘s’ button.
TUCEW a student views the tasks by the order the student wanted.

UC5: Sync tasks
TUCBW a student hits the ‘b’ button.
TUCEW a student sees success message.

UC6: log out
TUCBW a student hits the ‘x’ button
TUCEW a students sees a logout message.

<Requirements-Use Cases Matrix>

	Priority Weight	UC1	UC2	UC3	UC4	UC5	UC6
R1	2	X					
R2	4		X	X			
R3	3		X		X		
R4	1		X			X	
R5	2						x
Score		2	8	4	3	1	2

<Expanded Use Cases>

UC1: Create an account

Actor	System
	0. MTP shows a login page.
1. The student sees the login page	2. MTP moves to the creating

and hit the ‘n’ button.	account page.
3. The student type ID and password	4. The system creates the account shows a confirmation message.
5. The student sees the confirmation	

UC2: Log in

Actor	MTP
	0. MTP shows a login page.
1. The student hits ‘l’ and type ID (email address) and password.	2. The MTP verifies password and pull out tasks.
3. The student sees tasks	

UC 6

UC3: Manage a task

Actor	MTP
	0. The system shows the MTP tasks.
1. The student hits the ‘n’, ‘e’, or ‘d’ button.	2. If creating or editing a task, the system shows attributes to fill out or edit such as type, due date and time, priority, the time for how long it will take to accomplish a task, and a reminder/notification. If deleting a task, system asked for the index number.
3. The student fills out or edit attributes of a task, or types index number.	4. The MTP saves updated information. The MTP shows the list.

5. The student comes back to the list.	

UC4: Change the order of tasks

Actor	MTP
	0. The MTP shows the MTP tasks.
1. The student sees the list in the blackboard, and hits the ‘s’ button.	2. The MTP shows the options such as weekly, monthly, daily, class, due dates, priorities, weight, and others.
3. The student chooses an option.	4. The MTP shows the tasks by an option that the student chose.
5. The student views the tasks by the order the student wanted.	

UC5: Sync tasks from blackboard

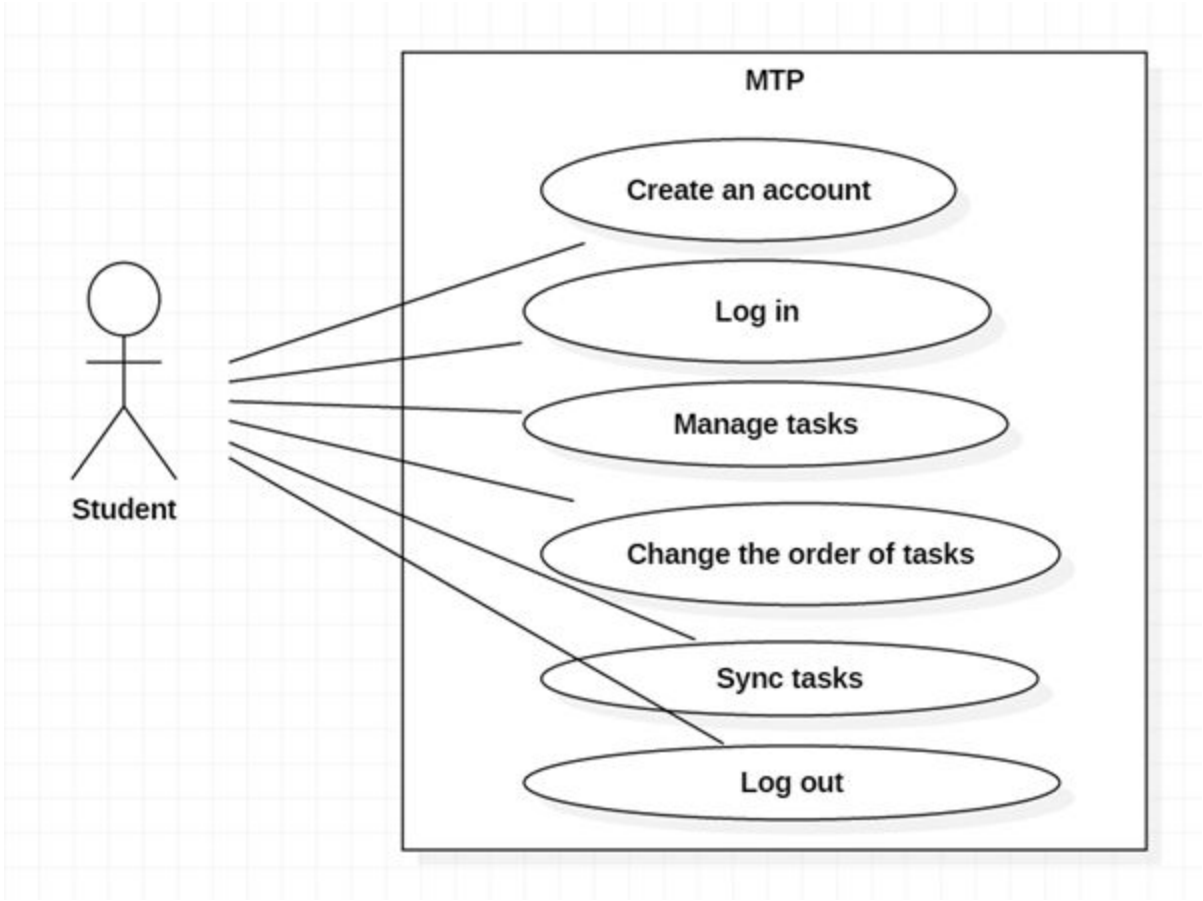
Actor	MTP
	0. the MTP shows that blackboard is logged in.
1. The student hits the ‘b’ button.	2. For each item to be added, MTP creates a task object with the same name and due date, then displays success message
3. The student sees success message.	

UC6

Log out:

	0. MTP shows a displays page for logout
1. The students hit ‘x’ button	2. MTP confirm all the updates for task is saved and request confirmation message.
3. The student enter confirm and see the logout page. The program closed.	

<Use Case Diagram>



Domain Modeling

<Brainstorming>

R1:

- Nouns/noun phrases: student, account
- Transitive verbs: create, log in

R1.1:

- Nouns/noun phrases: student, ID (email address), password, name, the university name, account
- Transitive verbs: type, create

R2:

- Nouns/noun phrases: student, task
- Transitive verbs: manage

R2.1:

- Nouns/noun phrases: student, task, variety type, class, Homework, tests, projects, events, meeting, personal work type, others
- Transitive verbs: create, edit, delete

R2.2:

- Nouns/noun phrases: student, due date and time
- Transitive verbs: set

R2.3:

- Nouns/noun phrases: student, priority, high, medium, low
- Transitive verbs: set

R2.4:

- Nouns/noun phrases: student, time for how long it will take to accomplish a task, given time, time for the specific task
- Transitive verbs: set, decrement, spend

R2.5:

- Nouns/noun phrases: student, daily list task
- Transitive verbs: manage

R2.6:

- Nouns/noun phrases: student, reminder/notification, preferred date and time

· Transitive verbs: set

R2.7:

· Nouns/noun phrases: student, percentage progress, available task

· Transitive verbs: add

R3:

· Nouns/noun phrases: student, task, a variety of display format

· Transitive verbs: view

R3.1:

· Nouns/noun phrases: student, task, weekly, monthly, daily, class, due dates, priorities, weight, others

· Transitive verbs: view

R4:

· Nouns/noun phrases: student, tasks, blackboard announcement, syllabus

· Transitive verbs: sync

R5:

· Nouns/noun phrases: student

Abbreviations

(A): attribute (of a class)

(AC): association class (of an association)

(AG): aggregation

(AS): association

(C): class, may be a subclass of another class

(I): inheritance relationship

(m,n): multiplicity of each class in a binary association

(rl,r2): role name of each class in a binary association

(V): attribute value (of an attribute of a class)

<Classification>

(C) Student

(C) Account

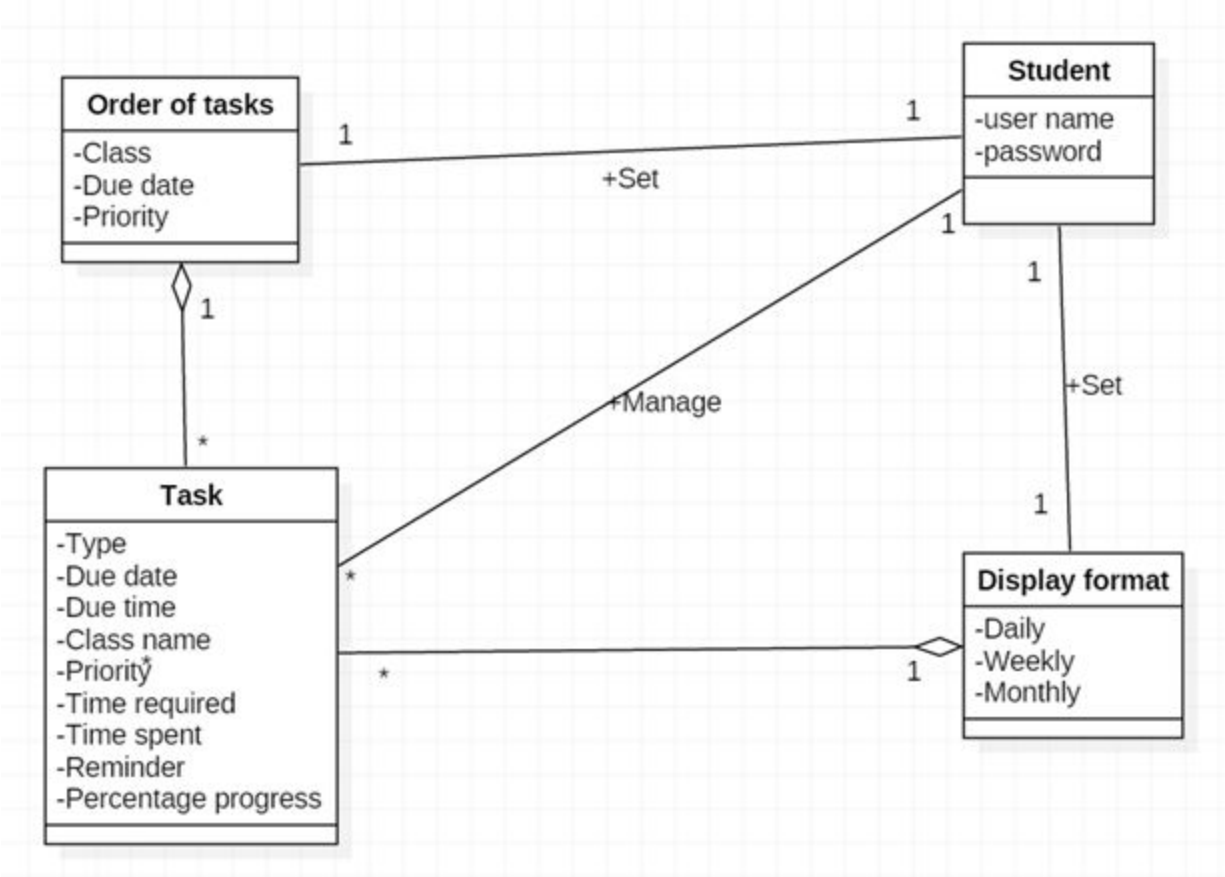
(A) ID (email address)

(A) Password
(A) Name
(A) University name

(C) Task
(A) task id
(A) Type (homework/test/project/event/meeting/personal work/other)
(A) Due date
(A) Due time
(A) Class name
(A) Priority (high/medium/low)
(A) Time required
(A) Time spent
(A) Reminder/Notification
(A) Percentage progress
(A) Order of tasks (class/due date/priority)

(C) Display format
(A)daily
(A)weekly
(A)monthly

<DM class diagram>



Object Interaction Modeling

<Scenarios & tables>

Create an account:

3) The student Enter ID and password.

4.1) The interface see if the ID already exists.

4.2) The controller checks for ID through the DB manager.

4.3) The DB manager returns the result of the request to the controller.

4.4) The Controller creates an account through the DB manager.

4.5) The DB manager pulls up new account.

4.6) The DB manager returns a confirmation message to the controller.

4.7) The controller returns a confirmation message to the interface.

4.8) The interface displays a confirmation message to the student.

	Subject	Action of Subject	Other Data/Objects	Object Acted Upon
3)	Student	Enter	ID, password	Interface
4.1)	Interface	Check	ID	Controller
4.2)	Controller	Check	ID	DB manager
4.3)	DB manager	Return	Result	Controller
4.4)	Controller	Create	Account	DB manager
4.5)	DB manager	Pull up	Account information	Account
4.6)	DB manager	Return	Confirmation message	Controller
4.7)	Controller	Return	Confirmation message	Interface
5.8)	Interface	Display	Confirmation message	Student

Log in:
1) The student hits ‘l’ and type ID (email address) and password.
2.1) The interface compares ID and password.
2.2) The controller get an account through the DB manager.
2.3) DB manager pulls up the account.
2.4) DB manager object returns tasks to the controller.
2.5) The controller returns tasks to the interface.
2.6) The interface display tasks to the student.

	Subject	Action of Subject	Other Data/Objects	Object Acted Upon
1)	Student	Enter	ID, password	Interface
2.1)	Interface	Compares	ID, password	Controller
2.2)	Controller	Get	Account	DB manager
2.3)	DB manager	Pull up	Account	Account
2.4)	DB manager	Return	Tasks	Controller
2.5)	Controller	Return	Tasks	Interface
2.6)	Interface	Display	Tasks	Student

Create/Edit a task:
3) The student fills out or edit attributes of a task.
4.1) The interface sends the new task information.
4.2) The controller create or edit the task.
4.3) The DB manager pulls up the task.
4.4) The controller get the task list.
4.5) The DB manager pulls up the task list.
4.6) The controller adds the task to the task list.
4.7) The controller returns task list to the interface.
4.8) The interface displays task list to the student.

	Subject	Action of Subject	Other Data/Objects	Object Acted Upon
3)	Student	Type	New task information, index	Interface
4.1)	Interface	Send	New task information, index	Controller
4.2)	Controller	Create/Edit	Task	DB manager
4.3)	DB manager	Pull up	Task	Task
4.4)	Controller	Get	Task list	DB manager
4.5)	DB manager	Pull up	Task list	Task list
4.6)	Controller	Add	Task	Task list
4.7)	Controller	Return	Task list	Interface
4.8)	Interface	Display	Task list	Student

Delete a task:
3) The student types index number.
4.1) The interface sends the index number.
4.2) The controller gets the task with the number.
4.3) The DB manager pulls up the task list.
4.4) The controller deletes the task with the index.
4.5) The controller returns task list to the interface.
4.6) The interface displays task list to the student.

	Subject	Action of Subject	Other Data/Objects	Object Acted Upon
3)	Student	Type	Index	Interface

			number	
4.1)	Interface	Send	Index	Controller
4.2)	Controller	Get	Task list	DB manager
4.3)	DB manager	Pull up	Task list	Task list
4.4)	Controller	Delete	Index	Task list
4.5)	Controller	Return	Task list	Interface
4.6)	Interface	Display	Task list	Student

Change the order of tasks:
3) The student chooses an option.
4.1) The interface gets the option from the student.
4.2) The controller gets the task list.
4.3) The DB manager pulls up the task list.
4.4) The controller sorts the task list by the order.
4.5) The controller returns the task list.
4.6) The interface displays the task list.

	Subject	Action of Subject	Other Data/Objects	Object Acted Upon
3)	Student	Enter	Order type	Interface
4.1)	Interface	Send	Order type	Controller
4.2)	Controller	Get	Task list	DB manager
4.3)	DB manager	Pull up	Task list	Account
4.4)	Controller	Sort	Task list, Order type	Task list
4.5)	Controller	Return	Task list	Interface

4.6)	Interface	Display	Task list	Student

Sync tasks from blackboard:

1) The student hits the “b” button.
2.1) The system sync the tasks.
2.2) The controller collects the tasks from the blackboard.
2.3) The controller adds the tasks to the task list.
2.4) The DB manager pulls up the task list.
2.5) The controller adds the tasks to the task list.
2.6) The controller returns the task list.
2.7) The system displays the task list.

	Subject	Action of Subject	Other Data/Objects	Object Acted Upon
1)	Student	Hit	Sync	MTP
2.1)	System	Sync		Controller
2.2)	Controller	Collect	Task	Blackboard
2.3)	Controller	Get	Task list	DB manager
2.4)	DB manager	Pull up	Task list	Task list
2.5)	Controller	Add	Task, task list list	Task list
2.6)	Controller	Return	Task list	MTP
2.7)	System	Display	Task list	Student

Log out:

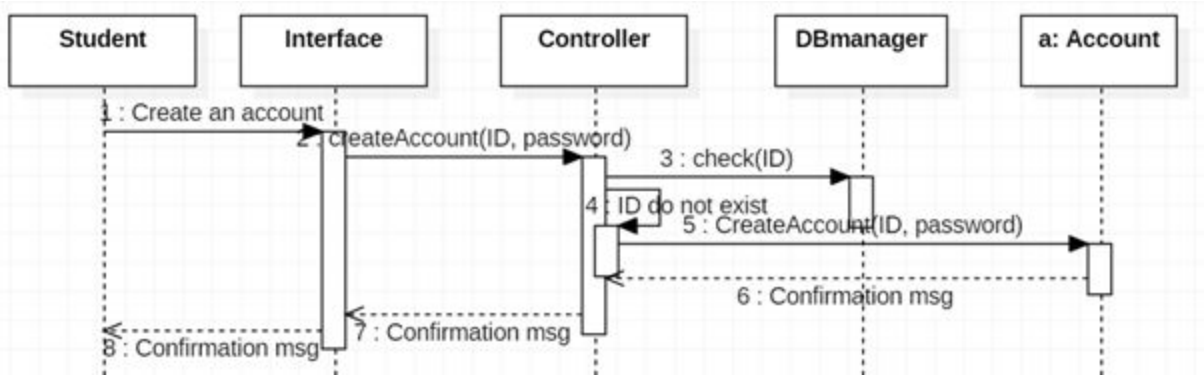
1) The student hits ‘x’ to logout .
2.1) The interface sends the message to controller.

2.2) The controller sends the message to account. The account closes the student task and send a close message to the DB manager.
2.3) DB manager close the student account.
2.4) The controller returns close object to the interface.
2.5) The interface display logout to the student.

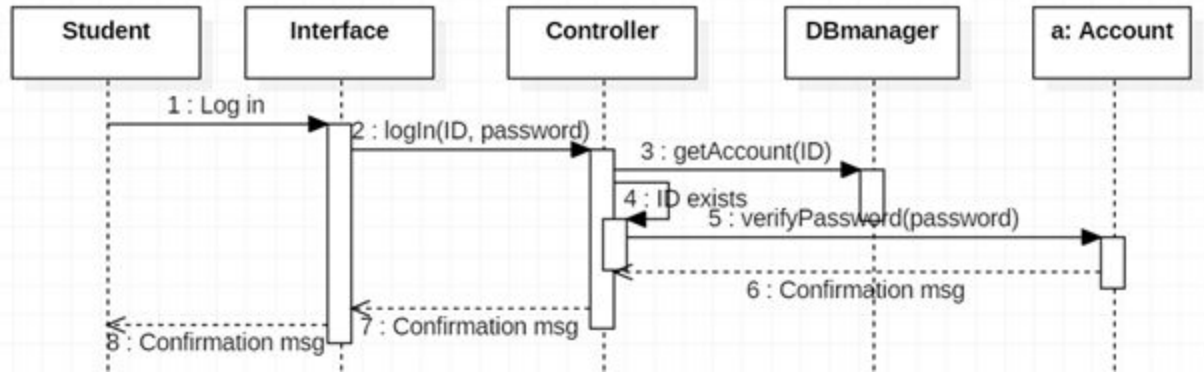
	Subject	Action of Subject	Other Data/Objects	Object Acted Upon
1)	Student	Hit	Close option x	Interface
2.1)	Interface	sends	Close message	Controller
2.2)	controller	Sends	Close message	account
2.3)	account	close	DBmanager	Student account
2.4)	controller	close	object	interface
2.5)	interface	display	Logout message	student

<Sequence Diagram>

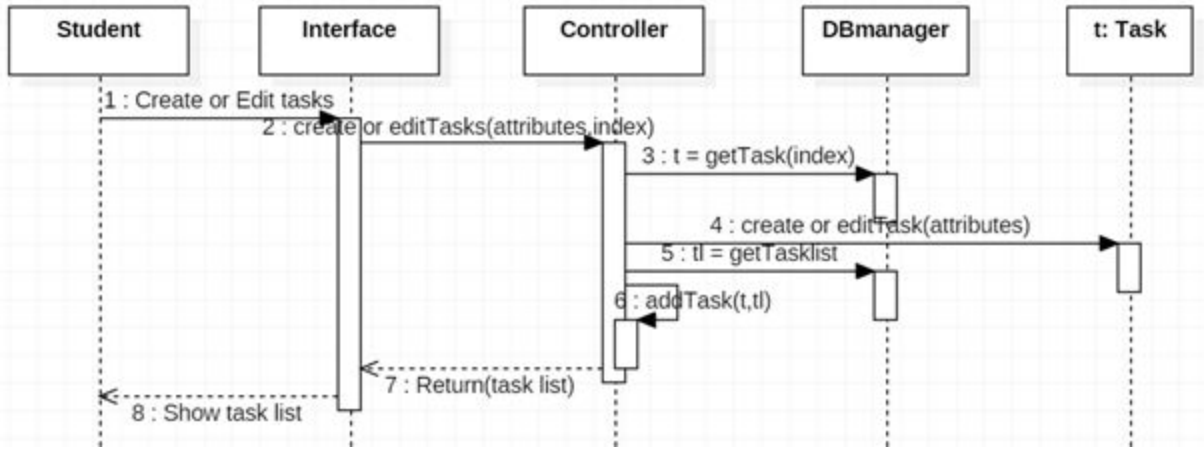
Create an account:



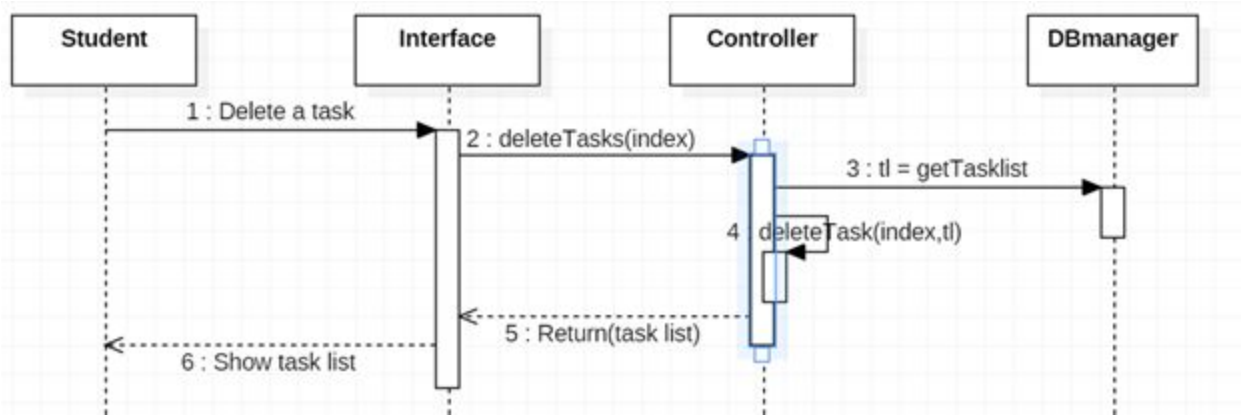
Log in:



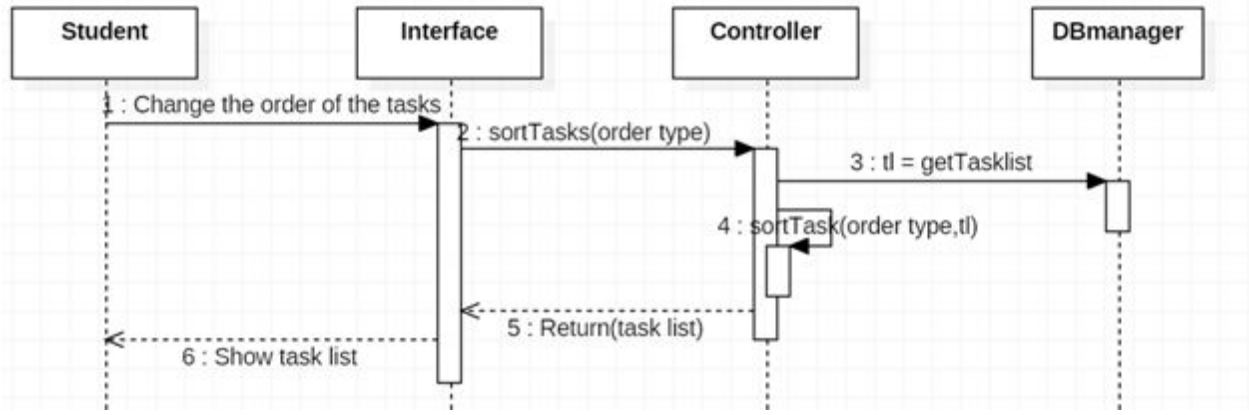
Create/Edit a task:



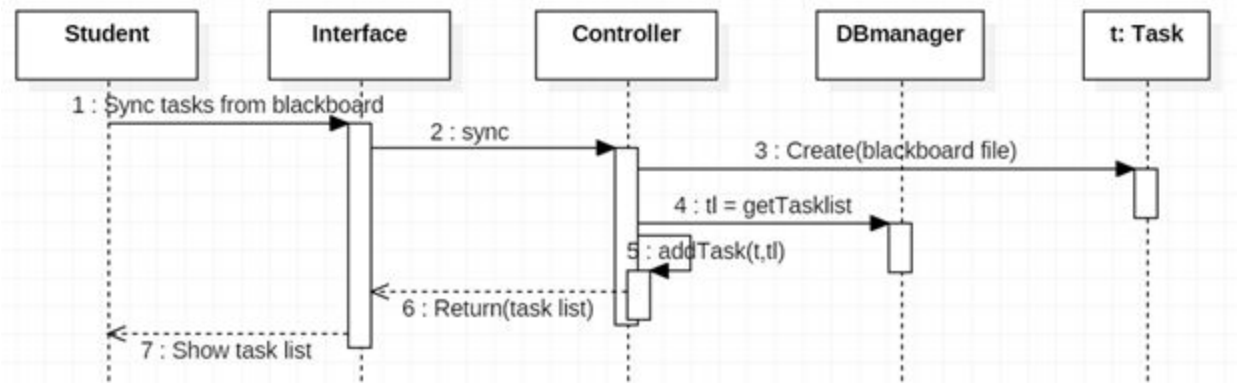
Delete a task:



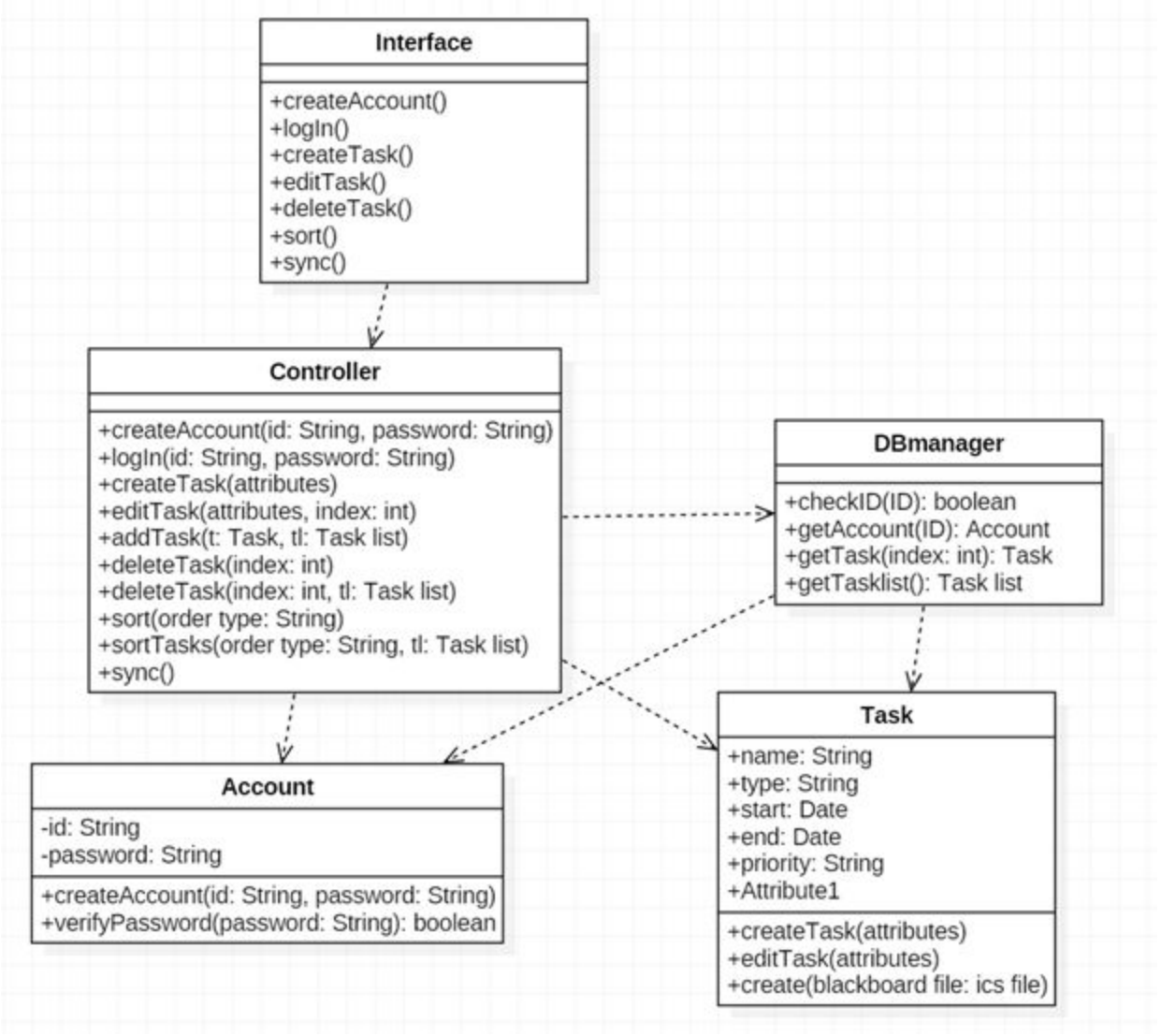
Change the order of tasks:



Sync tasks from blackboard:



Design Class Diagram



Iteration Presentation

The document is a methodology to create a student task planner app. The document is consist of all the methodology needed to implement student task planner software life cycle. It consists from requirement to implementation.

The requirement specify that the software must include major components like, login- logout, calendar display, sync schedule from blackboard,

reminder of tasks, manage different tasks like home work, class schedule, exams etc..,
The second part is the use-case. The use-case is a business process. It begins and end with the user. In our program the user is mostly a student. We have six use-cases for this program. Create account, login, manage task, sync task from blackboard, change order of the task based on student priority, and log out. The document include use case matrix, high level use case which include the beginning and the end, expanded use case, and use-case diagram.
The third part is Domain model. It is conceptual plan of the software. It consist of major classes that needs to be included in MTP to make the software completed. Domain model is the part where brainstorming take place. Class names, attributes and relations can be found from this steps to implement to class diagram.
The fourth part is the interaction diagram and sequence diagram. This part is very vital for code. It clarify the steps in the background during business process. In other word, it shows the steps in the non-trivial step of each use case (if they have). If the system has to pull, change a request, or save a requests, these steps need to be clarified in the interaction diagram.
The fifth part is the class diagram. It is the blueprint for the code. It consists of the class name, operations, attributes and the relation between the classes in the code.
The last part is the code and testing for the code. The code is written using Java language. The testing is done by block boxing method.

Software Demo/Testing

The zip file of java files are attached with this document.

Listed are Test Cases, sorted by Type of Operation, then followed by the results of the testing conducted during iteration 3.
V = Valid I = Invalid
Create an account

Test Case ID	Scenario	Username	Password	Exp. Result
TC1	Successful Registration	V	V	task prompt
TC2	Duplicate Registration	I	N/A	Error msg

These test cases work correctly

Log in

Test Case ID	Scenario	Username	Password	Exp. Result
TC3	Successful Login	V	V	Task prompt
TC4	Invalid username	I	N/A	Error msg
TC5	Invalid password	V	I	Error msg

These test cases work correctly

Manage a task

Test Case ID	Scenario	Task index	Task info	Exp. Result
TC6	Success	V	V	Task prompt
TC7	Task does not exist	I	N/A	Error msg
TC8	Invalid info	V	I	Error msg

TC7 and TC8 did not have validation

Change the order of tasks

Test Case ID	Scenario	Task type	Exp. Result
TC9	Success	V	Task prompt
TC10	Invalid type	I	Error msg

Sync tasks from blackboard

Test Case ID	Scenario	file	Exp. Result
TC11	Success	V	Task prompt
TC12	File does not exist	I	Error msg