CLASS-2
Sprint 1
Technical Requirements
and Guidelines

Olarn Rojanapornpun

❍ Recap on Requirements

❍ Sprint 1 PBIs Descriptions, Test Data, and Test Cases

❍ Sprint 1 Guidelines

❍ Nominal Sprint 2 Goal

Week 1 Tasks

○ Discuss with advisor on how to work together, communication channel, meeting schedule, ...

○ Design infrastructure architecture

○ Design communications between client and server :
    protocol, message type, ...

○ Design API specifications

○ Design software architecture

○ Design DB

○ Setup dev env

   ○ git repo, task board, communication channels

   ○ IDE, localhost servers

○ Setup dev server

○ (Create Sprint 1 wireframes and discuss with advisor)

○ (Prepare data/test cases/test scripts for Sprint 1)

○ Spike/PoC for Sprint 1

○ plan with advisor on what will be committed in Sprint 1 (Sprint goals)

Week 1 Technical Spike

○ date/time/time-zone (client: display, DB: type)

○ createdOn, updatedOn (DB)

○ UTF-8 (DB, Infra)

(Nonfunctional) Business Requirements

○ **IT-Bangmod Kradan Kanban (ITB-Kradan Kanban, ITB-KK)
— Release 1**

- ○ There is only one Kradan (task board).

- ○ Development team is the owner of the Kradan.

- ○ There is no user authentication/authorization support yet, anyone with access to the system can view and modify the board.

- ○ The board must be accessible from within KMUTT network (but not limited to).

- ○ The system uses UTF-8 character-encoding.

- ○ The system supports ICT (Asia/Bangkok) time-zone.

1	VIEW-TASK-TABLE VIEW	1
2	VIEW-TASK-DETAIL	
3	ADD-TASK-BASIC	2
4	DELETE-TASK	
5	EDIT-TASK-BASIC	
6	ADD-STATUS	3
7	EDIT-STATUS	
8	DELETE-STATUS	
9	NESTED-PUBLIC-PATH	
10	SORT-TASK-BY-STATUS-TABLE-VIEW	4
11	FILTER-TASK-BY-STATUS-TABLE-VIEW	
12	LIMIT-TASKS-IN-STATUS	
13	VALIDATE-INPUT-FE-BASIC	
14	VALIDATE-INPUT-BE-BASIC	

o Release-Level Priority

o The priority is modified to accommodate for the level of difficulties of each PBI

● Must-have
● Should-have
● Could-have

task Attributes : v1 [PBI1–PBI5]

Attribute Name	Description / Constraints	Example
taskTitle	Task name : UTF-8 TEXT; Limit to 100 characters; NOT NULL; NOT EMPTY; Leading and trailing whitespaces are allowed;	design API specs for PBI1–2
taskDescription	Task description : UTF-8 TEXT; Limit to 500 characters; OPTIONAL; NOT EMPTY; Leading and trailing whitespaces are allowed;	need to specify endpoints, request body, response body, status
taskAssignees	Task assignees : UTF-8 TEXT; Limit to 30 characters; OPTIONAL; NOT EMPTY; Leading and trailing whitespaces are trimmed.	Pang; Pong;
taskStatus	Task status : ENUM of {No Status, To Do, Doing, Done}; NOT NULL; The default value is 'No Status';	To Do
createdOn	Task creation date/time (on DBMS) : DATETIME; NOT NULL;	2024-04-19 01:00:00
updatedOn	Task last update date/time (on DBMS) : DATETIME; When a task is created, updatedOn is the same as createdOn; NOT NULL;	2024-04-19 01:00:00

Nominal Sprint 1 Goals

PBI #	PBI NAME	PBI DESCRIPTION / CONDITIONS
1	VIEW-TASK-TABLE VIEW	1. List all task in table view, with no pagination. 2. The table shows task: Title, Assignees, Status. 3. The task is sorted ascending order of created time. 4. If there is no task in the board, a table with no record is shown.
2	VIEW-TASK-DETAIL	1. When the user click on task's title, the task's detail is shown in a modal window. 2. Task's detail includes Title, Description, Assignees, Status, Created date, and Updated date. 3. If task's description is null or empty, the statement '*No description provided*' is shown in italic and grey. 4. If task Status is null or empty, the statement '*Unassigned*' is shown in italic and grey. 5. The dates are shown in local timezone (with timezone identifier such as 'Asia/Bangkok') 6. If the requested task (id) does not exist, the backend returns status code 404 and the frontend shows message "The requested task does not exist" and redirects to previous page.

ID	taskTitle	taskDescription	TaskAssignees	taskStatus	createdOn	updatedOn
1	TaskTitle1TaskTitle2TaskTitle3TaskTitle4TaskTitle5TaskTitle6TaskTitle7TaskTitle8TaskTitle9TaskTitle0	Descripti1Descripti2Descripti3Descripti4Descripti5Descripti6Descripti7Descripti8Descripti9Descripti1Descripti1Descripti2Descripti3Descripti4Descripti5Descripti6Descripti7Descripti8Descripti9Descripti2Descripti1Descripti2Descripti3Descripti4Descripti5Descripti6Descripti7Descripti8Descripti9Descripti3Descripti1Descripti2Descripti3Descripti4Descripti5Descripti6Descripti7Descripti8Descripti9Descripti4Descripti1Descripti2Descripti3Descripti4Descripti5Descripti6Descripti7Descripti8Descripti9Descripti5	Assignees1Assignees2Assignees3	No Status	22/4/2024 09:00:00 +00:00	22/4/2024 09:00:00 +00:00
2	Repository	<empty>	<empty>	To Do	22/4/2024 09:05:00 +00:00	22/4/2024 14:00:00 +00:00
3	ดาต้าเบส	ສ້າງຖານຂໍ້ມູນ	あなた、彼、彼女 (私ではありません)	Doing	22/4/2024 09:10:00 +00:00	25/4/2024 00:00:00 +00:00
4	_Infrastructure_	_Setup containers_	ไก่งวง กับ เพนกวิน	Done	22/4/2024 09:15:00 +00:00	22/4/2024 10:00:00 +00:00

○ Test Scenario : normal - task table is empty

○ Test Data : empty task table

○ Test Steps :

1. Empty task table

2. Open task view page at /task

○ Expected Results :

1. FE sends request to BE

2. BE returns status code 200 with response body = '[]'
(view in browser Inspector)

3. FE shows task table with column headers and no task list

○ Test Scenario : normal - task table has entries

○ Test Data : TASK_BASE

○ Test Steps :

1. Load TASK_BASE test data

2. Open task view page at /task

○ Expected Results :

1. FE sends request to BE
2. BE returns status code 200 and all task entries
3. FE shows task table with column headers and 4 task entries
4. Task Title shows : TaskTitle... , Repository, ดาต้าเบส,   _Infrastructure_   , respectively

❍ Test Scenario : normal - all fields available, max fields, "No Status"

❍ Test Data : TASK_BASE

❍ Test Steps :

1. Open task details for task TaskTitle1...

❍ Expected Results :

1. Task details are shown with max character limits for Title, Description, Assignees

2. datetime for BOTH createdOn AND updatedOn are shown as 22/04/2024 16:00:00 (Asia/Bangkok)

3. taskStatus is shown as "No Status"

4. BE returns 200 and task details

Acceptance Test : VIEW-TASK-DETAIL-2

❍ Test Scenario : empty Description and Assignees, "To Do"

❍ Test Data : TASK_BASE

❍ Test Steps :

1. Open task details for task Repository

❍ Expected Results :

1. Task details are shown with "*No Description Provided*" and "*Unassigned*" in grey italic
2. Status is "To Do"
3. BE returns 200 and task details

Acceptance Test : VIEW-TASK-DETAIL-3

❍ Test Scenario : non-English characters, "Doing"

❍ Test Data : TASK_BASE

❍ Test Steps :

   1. Open task details for task ดาต้าเบส

❍ Expected Results :

   1. Title, Details, and Assignees are shown in non-English characters correctly
   2. Status is "Doing"
   3. BE returns 200 and task details

Acceptance Test : VIEW-TASK-DETAIL-4

❍ **Test Scenario** : leading and trailing whitespace, "Done"

❍ **Test Data** : TASK_BASE

❍ **Test Steps** :

1. Open task details for task _Infrasturcture_

❍ **Expected Results** :

1. FE shows Title and Details with 5 leading and 5 trailing whitespaces (highlight the text to confirm)
2. response from BE shows leading and trailing whitespaces
3. Status is "Done"
4. BE returns 200 and task details

Acceptance Test : VIEW-TASK-DETAIL-5

❍ Test Scenario : normal - view different time zone

❍ Test Data : TASK_BASE

❍ Test Steps :

0. In task detail view for task   _Infrastructure_

1. change the browser's timezone to Australia/Sydney (+10:00)

1.1 verify the time shown

❍ Expected Results :

1. datetime is shown as 22/4/2024 19:15:00 (Australia/Sydney) and 22/4/2024 20:00:00 (Australia/Sydney)

16

Acceptance Test : VIEW-TASK-DETAIL-6

○ Test Scenario : failed - view task with invalid ID

○ Test Data : TASK_BASE

○ Test Steps :

    1. Open task view page at /task/100

○ Expected Results :

    1. FE shows User Appropriate message "The requested task does not exist" and when the message is closed redirects to the TaskList view (/task)

    2. BE returns 404

Server:

http://localhost:8080/itb-kk

URI:

GET **/v1/tasks**   **Get all task with no pagination**

GET **/v1/tasks/{id}**   **Get a task detail by id**

GET /v1/tasks ^

Parameters Cancel

No parameters

Request URL

```
http://localhost:8080/itb-kk/v1/tasks
```

Server response

Code	Details
200	**Response body**

```
[
  {
    "title": "Create Entity",
    "assignees": "Somchai",
    "status": "DOING"
  },
  {
    "title": "Create Rest Service for Task",
    "assignees": "Pichet",
    "status": "TO_DO"
  }
]
```

Download

GET /v1/tasks/{id} ^

Parameters [Cancel]

Name	Description

id * required
integer($int32)
(path)

```
17
```

Request URL

```
http://localhost:8080/itb-kk/v1/tasks/17
```

Server response

Code	Details

200

Response body

```json
{
    "title": "Create Rest Service for Task",
    "description": "Create CRUD for Task",
    "assignees": "Pichet",
    "status": "TO_DO",
    "createdOn": "20-04-2024 23:46:58 ICT",
    "updatedOn": "20-04-2024 23:47:32 ICT"
}
```

[Download]

GET /v1/tasks/{id}

Parameters

Cancel

Name	Description
id * required	
integer($int32) (path)	12

Request URL

```
http://localhost:8080/itb-kk/v1/tasks/12
```

Server response

Code	Details
404 Undocumented	Error: response status is 404

Response body

```json
{
    "timestamp": "2024-04-22T12:33:58.896+00:00",
    "status": 404,
    "error": "Not Found",
    "message": "Task 12 does not exist !!!",
    "path": "/itb-kk/v1/tasks/12"
}
```

Download

IT-Bangmod Kradan Kanban (ITB-KK)
Sample UI: No task

path: /task

IT-Bangmod Kradan Kanban

Title	Assignees	Status
No task		

IT-Bangmod Kradan Kanban (ITB-KK)
Sample UI: Having tasks

path: /task

IT-Bangmod Kradan Kanban

itbkk-item

	Title	Assignees	Status
1	Design Backend API for PBI 1,2	Pang; Pong;	No status
2	Design UI for PBI 1,2	Ping;	To do

itbkk-title   itbkk-assignees   itbkk-status

class name : itbkk-*

IT-Bangmod Kradan Kanban (ITB-KK)
Sample UI: Task Detail

Nominal Sprint 1 Tasks

- In-complete tasks from Week 1 may be carried forward to Sprint 1 backlog

- (At the beginning of Sprint 1) commit PBI (Sprint goal) to advisor

- development of committed PBI in Sprint backlog

- review completed PBI with advisor

- Sprint 2 Spike

Nominal Sprint 1 Tasks

VIEW-TASK-TABLE VIEW

- ❑ [Design] review design: architecture, DB, endpoint, methods, JSON message, response code
- ❑ [DB] create task table
- ❑ [DB] import TASK_BASE test data
- ❑ [Front] create task listing page
- ❑ [Back] create /v1/tasks GET method API
- ❑ [Infra] create dockerfile for DB container, back-end container, front-end container
- ❑ [Infra] deploy codes to dev servers
- ❑ [Test] unit test DB
- ❑ [Test] unit test /tasks GET method using postman (empty table)
- ❑ [Test] unit test task listing page
- ❑ [Test] integration test
- ❑ [Test] acceptance test on dev servers with advisor

Nominal Sprint 1 Tasks

Front-end

- ❑ config environments (.env) and routes
- ❑ create task listing component
  - ❑ html template
  - ❑ java script
  - ❑ css style

Back-end

- ❑ create repository/DAO
- ❑ create entity class
- ❑ create REST controller
  - ❑ route pattern (request mapping)
  - ❑ exception handling

Nominal Sprint 2 Goals

PBI #	PBI NAME	PBI DESCRIPTION / CONDITIONS
3	ADD-TASK-BASIC	1. When the user click on "Add" button, the task detail modal window is shown. 2. The user can add the Task's details: Title [REQUIRED], Description [OPTIONAL], Assignees [OPTIONAL], Status [default: No Status] 3. DB: auto inserts Created date and Updated date (current date, time), these are NOT provided by BE 4. BE returns status code; FE shows user appropriate message 5. FE Shows task table with added task
4	DELETE-TASK	1. When the user click on "Delete" for a task, a confirmation modal window is shown showing taskTitle to be deleted. "Confirm" and "Cancel" options are provided. 2. When "Confirm", BE deletes task and provides response 3. BE provides response code; FE shows user appropriate message 4. FE shows updated task table

PBI #	PBI NAME	PBI DESCRIPTION / CONDITIONS
5	EDIT-TASK-BASIC	1. When the user clicks on "Edit" button for a task, the Task-Detail is shown and all fields are editable, with options to "Save" or "Cancel" 2. BE returns status code, and reason 3. FE shows response from BE with user appropriate message 4. If successful FE, shows updated task details. 5. User can click "Close" or "Back" to return to Task Table view. 6. If the task (id) does not exists, BE returns 404 and FE behaves appropriately

○ **Hostname:** ip23<TEAM-CODE> e.g. ip23at1

○ **Domain:** sit.kmutt.ac.th

○ **OS:** Ubuntu 22.04LTS

○ **Account:** sysadmin

○ **Password:** <<check your email>>

○ **Recommendation:**

  ○ Create account for each team member, each in the same group

  ○ Store source codes in neutral, agreed on directory