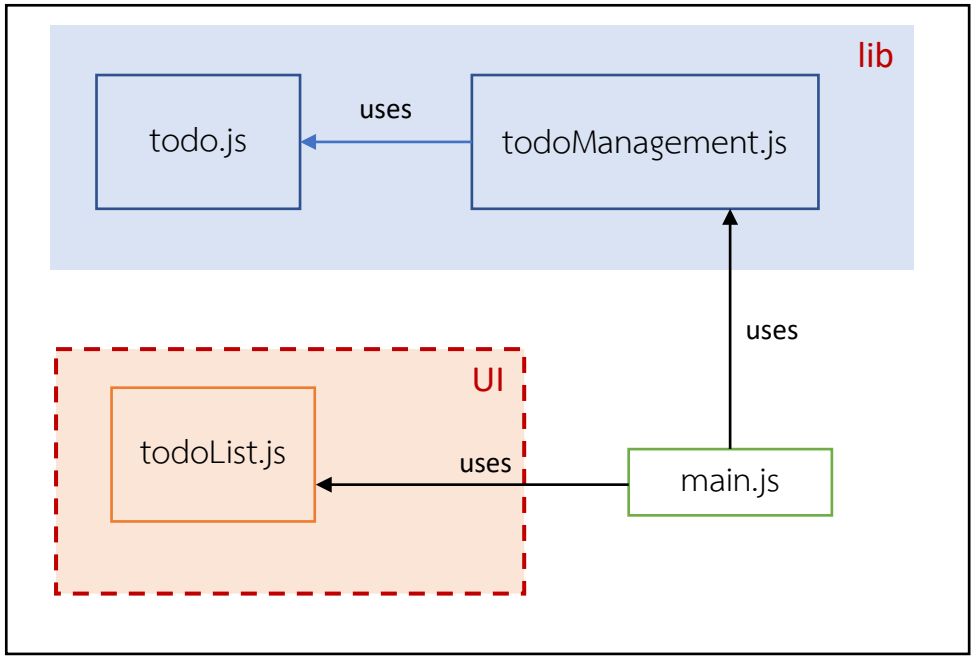
Document Object Modeling

DOM
TodoList Project Diagram



1. ไฟล์ ./lib/todo.js เพื่อจัดการกับ property และฟังก์ชันของแต่ละ todo
2. ไฟล์ ./lib/todoManagement.js เพื่อจัดการโครงสร้างข้อมูลและฟังก์ชันของ todo ทั้งหมด
3. ไฟล์ ./UI/todoListUI.js เพื่อจัดการ dom ของเอกสาร HTML
4. ไฟล์ ./main.js สำหรับเรียกใช้ไฟล์ต่าง ๆ ที่เกี่ยวข้อง

TodoList Project Requirement

ไฟล์ ./lib/todo.js เพื่อจัดการกับ property และฟังก์ชันของแต่ละ todo

1. ให้เพิ่ม property done (Boolean) และให้กำหนดค่าเริ่มต้น done เป็น false ที่ constructor
2. ให้เพิ่ม property done ในการคืนค่า object ที่ฟังก์ชัน getTodo

ไฟล์ ./lib/todoManagement.js เพื่อจัดการโครงสร้างข้อมูลและฟังก์ชันของ todo ทั้งหมด

1. ให้แก้ไขฟังก์ชัน addTodo(desc) ให้ return todo.id แทนการ return length ของ array
2. ให้เพิ่มฟังก์ชัน getNumberOfDone เพื่อ return จำนวนของ Todo ที่อยู่ในสถานะ Done
3. ให้เพิ่มฟังก์ชัน getNumberOfNotDone เพื่อ return จำนวนของ Todo ที่อยู่ในสถานะ Not Done

สร้างไฟล์ ./UI/todoList.js เพื่อจัดการ dom ของเอกสาร HTML

1.ให้เพิ่มฟังก์ชันที่ชื่อ showTodoItem(*newId, newDescription*) โดยทำการรับค่า todo id และ description เป็นพารามิเตอร์ เพื่อทำการแสดงรายการ todo ภายใต้ `<div id="listTodo"></div>` ของเอกสาร HTML โดยแต่ละรายการ todo ให้มีโครงสร้าง tags เรียงลำดับดังนี้

```
<div class="todoItem" id="newId">
    <p>newDescription</p>
    <button>Not Done</button>
    <button>remove</button>
</div>
```

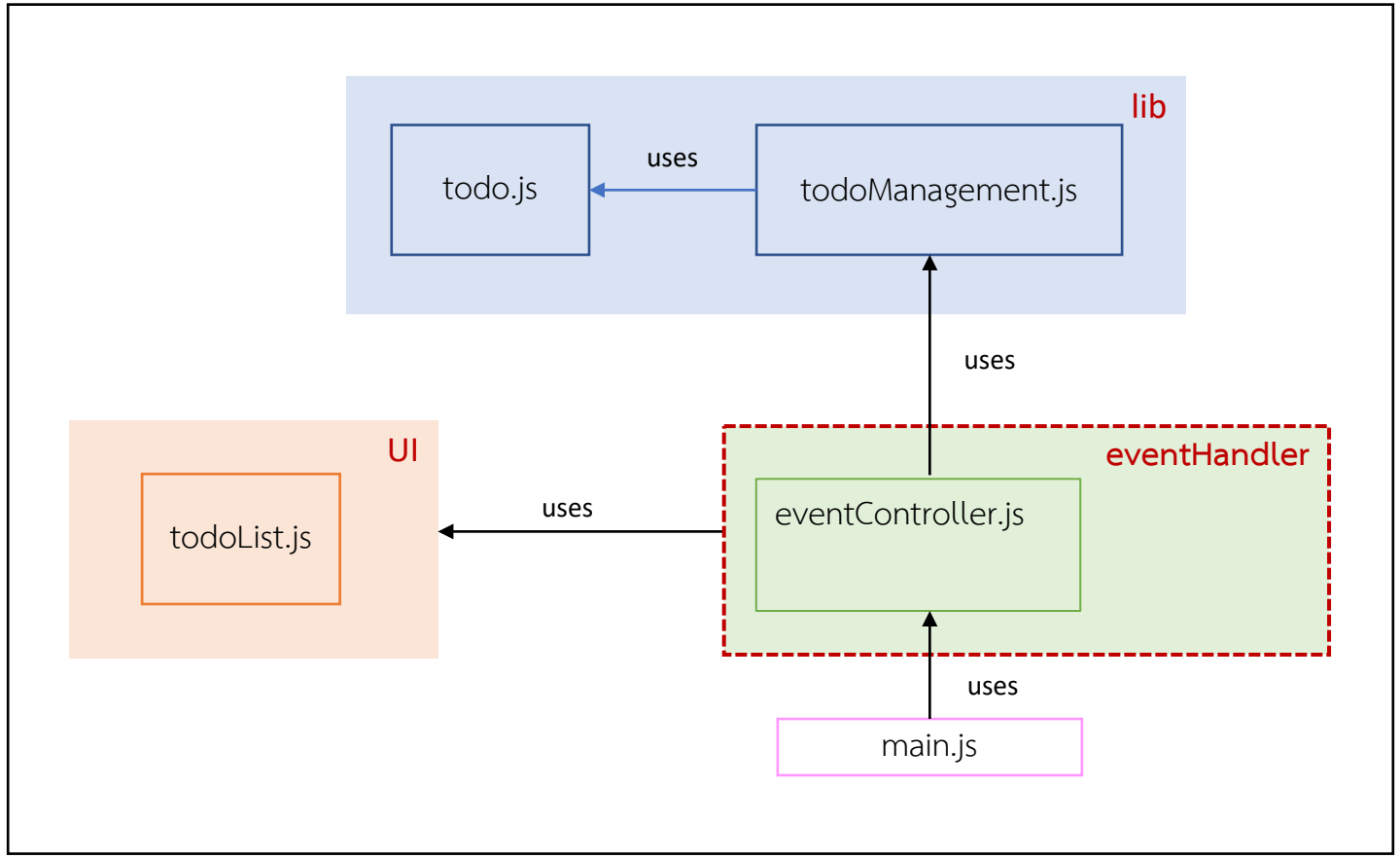
2. ให้เพิ่มฟังก์ชัน showNumberOfDone(numberOfDone) เพื่อแสดงเลขจำนวนของ Todo ที่อยู่ในสถานะ Done ต่อท้ายข้อความ ภายใน `<p id="done">Number of Done:</p>`
3. ให้เพิ่มฟังก์ชัน showNumberOfNotDone(numberOfNotDone) เพื่อแสดงเลขจำนวนของ Todo ที่อยู่ในสถานะ Not Done ต่อท้ายข้อความ ภายใน `<p id="notDone">Number of Not Done:</p>`

สร้างไฟล์ ./main.js

1. ให้ทดสอบเรียกใช้ addTodo() และรับ id ที่ return จากฟังก์ชัน เพื่อส่งต่อให้ฟังก์ชัน showTodoItem() เพื่อทำการสร้าง tags แสดงผลใน HTML File
2. ให้ทดสอบเรียกใช้ getNumberOfDone() และ getNubmerOfNotDone() และรับค่าจำนวนของ Todo ที่อยู่ในสถานะ Done และ NotDone เพื่อส่งต่อฟังก์ชัน showNumberOfDone() และ showNumberOfNotDone() เพื่อแสดงผลใน HTML File

Event Handling

Event Handlers
TodoList Project Diagram



1. ไฟล์ ./lib/todo.js เพื่อจัดการกับ property และฟังก์ชันของแต่ละ todo
2. ไฟล์ ./lib/todoManagement.js เพื่อจัดการโครงสร้างข้อมูลและฟังก์ชันของ todo ทั้งหมด
3. ไฟล์ ./UI/todoListUI.js เพื่อจัดการ DOM ของเอกสาร HTML
4. ไฟล์ ./eventHandler/evenController.js เพื่อจัดการกับ event ที่เกิดขึ้นบน DOM ของเอกสาร HTML
5. ไฟล์ ./main.js เริ่มต้นทำงาน

LAB Exercise

ปรับปรุงไฟล์ todo.js เพื่อจัดการกับ property และฟังก์ชันของแต่ละ todo

1. ให้เพิ่มฟังก์ชัน setDone(value) เพื่อจะกำหนดค่า done ตามค่าพารามิเตอร์ที่รับมา

ปรับปรุงไฟล์ todoManagement.js เพื่อจัดการโครงสร้างข้อมูลและฟังก์ชันของ todo ทั้งหมด

1. ให้เพิ่มฟังก์ชัน setItemToDone(doneId) เพื่อค้นหา todo ที่มีค่าตรงกับ doneId จากนั้นให้เรียกฟังก์ชัน setDone() เพื่อกำหนดค่า done ให้เป็น true

ปรับปรุงไฟล์ todoList.js เพื่อจัดการ DOM ของเอกสาร HTML

1. ให้เพิ่มฟังก์ชัน removeTodoItem(removeId) เพื่อลบ ของรายการ Todo นั้นออกไปจาก HTML

สร้างไฟล์ eventController.js เพื่อสร้าง event handle functions ดังนี้

1. ให้สร้างฟังก์ชัน addTodoHandler() เพื่อจัดการเมื่อผู้ใช้กดปุ่ม add ให้ตรวจสอบว่าไม่เป็นค่า empty string จึงเพิ่มรายการ todo นั้นที่ todos array และให้แสดง todo ภายใต้ ของเอกสาร HTML จากนั้นค้นหาปุ่ม Not Done และ Remove ของ todo ที่ add เพื่อลงทะเบียน event handler function notDoneButtonHandler() และ removeButtonHandler() ให้กับปุ่ม Not Done และ Remove
2. ให้สร้างฟังก์ชัน notDoneButtonHandler() เพื่อจัดการเมื่อผู้ใช้กดปุ่ม Not Done ข้อความจะเปลี่ยนเป็น done โดยแสดง background เป็นสีเขียว และตัวอักษรเป็นสีขาว จากนั้นให้ทำการเรียกใช้ฟังก์ชัน setItemToDone (ไฟล์ todoManagement.js) โดยทำการส่ง todo id ของรายการ todo นั้น เพื่อให้เปลี่ยนค่า property done เป็น true
3. ให้สร้างฟังก์ชัน removeButtonHandler() เพื่อจัดการมื่อผู้ใช้กดปุ่ม remove จะทำการเรียกใช้ฟังก์ชัน removeTodoItem(removeId) (ไฟล์ todoList.js) และฟังก์ชัน removeTodo(removeId) (ไฟล์ todoManagement.js) เพื่อลบรายการ Todo นั้นออกจาก DOM HTML และจาก todos array

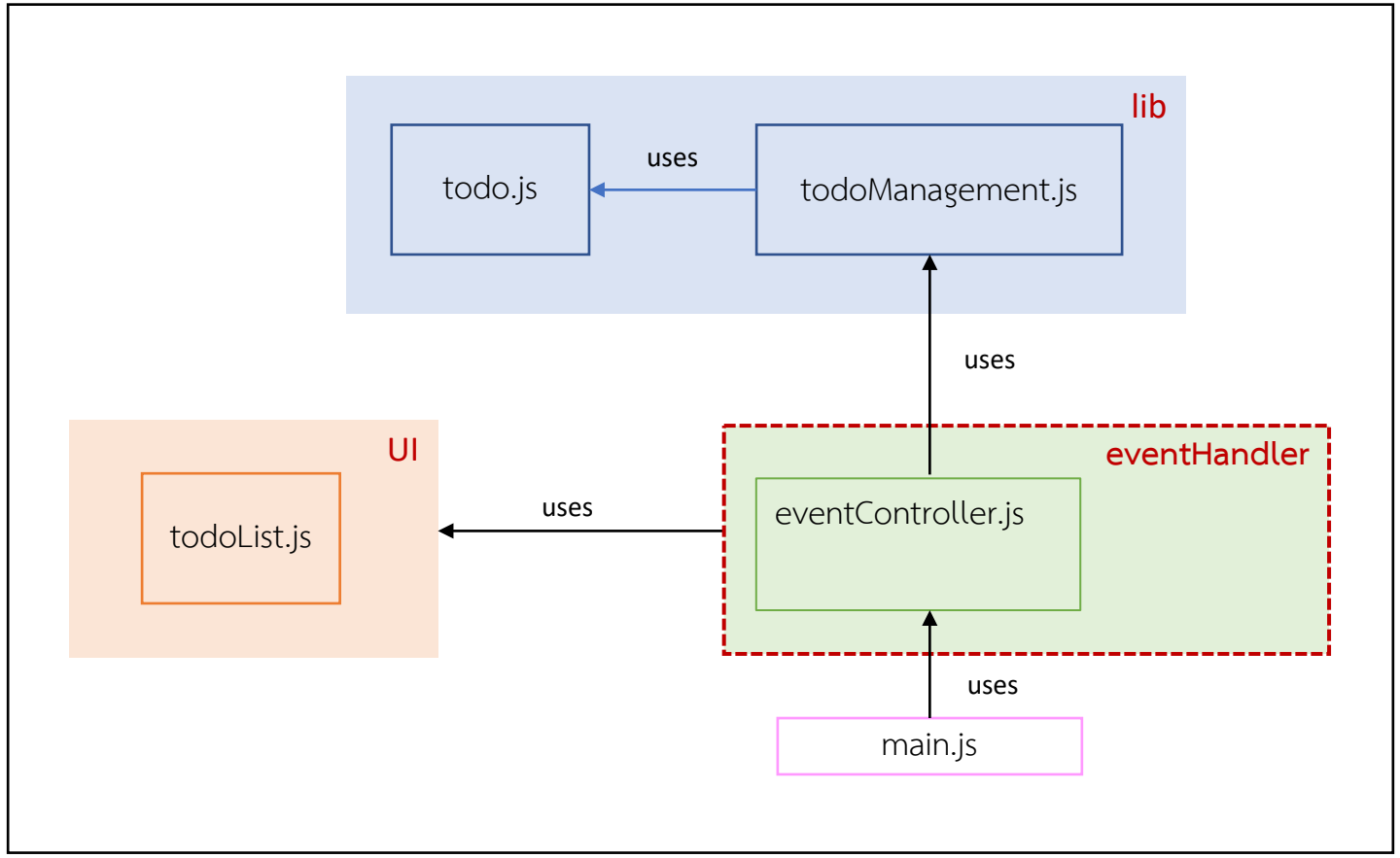
ท้ายฟังก์ชันในข้อ 1 - ข้อ 3 ให้ update จำนวน todo ที่มีสถานะ Done และ Not Done โดยการเรียกใช้ฟังก์ชัน showNumberOfDone() และ showNumberOfNotDone()

แทนที่ code ไฟล์ main.js ด้วยคำสั่ง ดังนี้

1. ให้ทำการค้นหาปุ่ม Add Todo และลงทะเบียน addTodoHandler() ให้กับปุ่ม Add Todo

Client-Side Storages

Client Storages
TodoList Project Diagram



1. ไฟล์ ./lib/todo.js เพื่อจัดการกับ property และฟังก์ชันของแต่ละ todo
2. ไฟล์ ./lib/todoManagement.js เพื่อจัดการโครงสร้างข้อมูลและฟังก์ชันของ todo ทั้งหมด
3. ไฟล์ ./UI/todoListUI.js เพื่อจัดการ DOM ของเอกสาร HTML
4. ไฟล์ ./eventHandler/evenController.js เพื่อจัดการกับ event ที่เกิดขึ้นบน DOM ของเอกสาร HTML และบันทึกข้อมูลของผู้ใช้ไว้ใน client-side storage
5. ไฟล์ ./main.js เริ่มต้นทำงาน

LAB Exercise

ปรับปรุงไฟล์ todoManagement.js เพื่อจัดการโครงสร้างข้อมูลและฟังก์ชันของ todo ทั้งหมด

1. ให้เพิ่มฟังก์ชัน loadTodos (userTodos) เพื่อกำหนดค่า todo ที่เก็บใน local storage ให้กับ todos array

ปรับปรุงไฟล์ eventController.js เพื่อสร้าง event handle functions ดังนี้

1. ให้เพิ่มฟังก์ชัน loadHandler() เพื่อทำการ get ค่า todos จาก local storage โดยถ้าค่าไม่เป็น null ให้ส่งไปเป็นพารามิเตอร์ของฟังก์ชัน loadTodos() จากนั้นให้เรียกฟังก์ชัน getTodos() เพื่อจะ show แต่ละรายการ todos ในเอกสาร HTML โดยการเรียกฟังก์ชัน showTodoItem() และให้ add handler ของปุ่ม Not Done และปุ่ม remove กรณี todo รายการนั้น done แล้ว ให้ใส่ style background เป็นสีเขียว และตัวอักษรเป็นสีขาว จากนั้นupdate จำนวนรายการ done และ not done โดยเรียกฟังก์ชัน showNumberOfDone() และ showNumberOfNotDone() ตามลำดับ
2. ให้เพิ่มฟังก์ชัน beforeUnloadHandler(event) โดยให้เรียกฟังก์ชัน preventDefault() เพื่อจะได้สามารถบันทึกค่า todos ปัจจุบัน โดยตั้งชื่อ todos เพิ่มเข้าไปใน local storage ได้ก่อนจะมีการ unload HTML Page

ปรับปรุงไฟล์ main.js ด้วยคำสั่ง ดังนี้

1. (คงเดิม) ทำการค้นหาปุ่ม Add Todo และลงทะเบียน addTodoHandler() ให้กับปุ่ม Add Todo
2. เพิ่ม event เมื่อ window ถูกโหลด โดยเรียกใช้ loadHandler() เพื่อโหลดค่า todos ของ user ที่บันทึกไว้ใน local storage
3. เพิ่ม event ก่อน window จะ unload โดยเรียกใช้ beforeUnloadHandler(event) เพื่อบันทึกค่า todos ของ user ใน local storage