

---

# WIRE POWDERLESS

EE461L Team 3:

Lian Fang Liu, Mona McElroy, Rachel Tan, Sohaib Khan, Yue Cheng

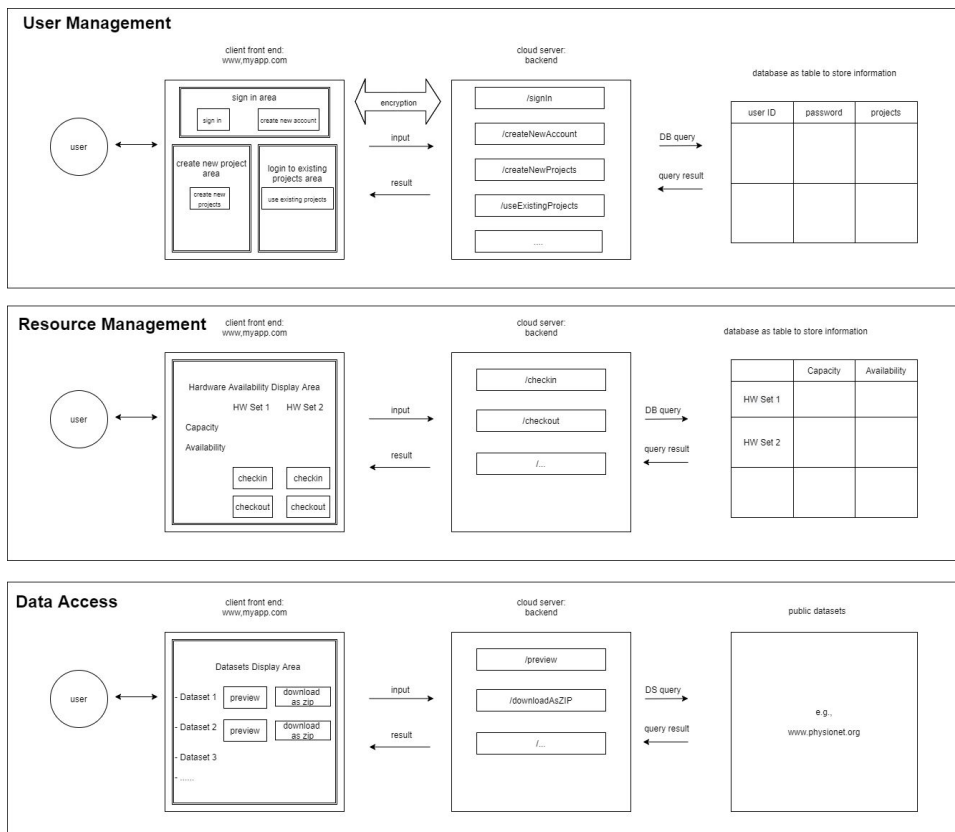
---

# 1: Team Project

## Wire Powderless

Project pdf


# 1: Team Project



## 2: SDLA, Agile, Git

- [Meeting Notes](#)
- [Github](#)



# 3: Python

 main ▾ **EE461L-Project** / data\_service /






Go to file

Add file ▾

...

 **yue-cheng-wind** this file is not used ... 2 hours ago  **History**

..

 <code>__init__.py</code>	flask dynamic table with google chart,	27 days ago
 <code>dataset.py</code>	changed dataset downloads to hrefs with links	10 days ago
 <code>hardware.py</code>	frontback connect done working on dwnload button	17 days ago
 <code>project.py</code>	merge connection-Rachel-Yue to test branch	12 days ago
 <code>user.py</code>	merged with dataset page; all 3 modules fully work	10 days ago

# 4: Database

This screenshot shows the MongoDB Compass interface for the **SemesterProject** database, specifically the **Project** collection. The left sidebar shows the database structure with **Project** selected. The main panel displays the collection's metadata: **COLLECTION SIZE: 5.87KB** and **TOTAL DOCUMENTS: 22**. A filter bar contains the text `{"filter": "example"}`. Below, the query results show a single document with the following fields:

```
{
  "_id": ObjectId("6064f85e0eb2ea5f785136db"),
  "date_created": "2021-03-31 17:31:58",
  "last_edited": "2021-04-06 23:45:14",
  "project_name": "Yue's wednesday",
  "user_id": "60601c82cdd298d4ea3c5a04",
  "comment": "Liu's project",
  "hardware_set_dict": Object,
  "team": Array
}
```

This screenshot shows the MongoDB Compass interface for the **SemesterProject** database, specifically the **HardwareSet** collection. The left sidebar shows the database structure with **HardwareSet** selected. The main panel displays the collection's metadata: **COLLECTION SIZE: 304B** and **TOTAL DOCUMENTS: 4**. A filter bar contains the text `{"filter": "example"}`. Below, the query results show a single document with the following fields:

```
{
  "_id": ObjectId("60620d64d962298b2837d3d7"),
  "hwSet_name": "test1",
  "capacity": 55,
  "availability": 0
}
```

This screenshot shows the MongoDB Compass interface for the **SemesterProject** database, specifically the **User** collection. The left sidebar shows the database structure with **User** selected. The main panel displays the collection's metadata: **COLLECTION SIZE: 3.12KB** and **TOTAL DOCUMENTS: 13**. A filter bar contains the text `{"filter": "example"}`. Below, the query results show a single document with the following fields:

```
{
  "_id": ObjectId("604e7d148aaacd6a12855cbb"),
  "username": "Yue",
  "password": "$pbkdf2-sha256$29000$HqP03lsLwRjJPKe",
  "email": "chengyue@utexas.edu",
  "projects": Array
}
```

# 4: Database

## User: user.py

- user\_id
- username
- password
- Email
- [project\_id]
- Credit information

## Project: project.py

```
▪ project_id
▪ project_name
▪ status: ongoing / completed
▪ Date_created
▪ Last_edited
▪ comment
▪ Hw_set_dict:{HWSets_id_1: 5,
HWSets_id_2 :5 }
▪ user_id
▪ team: [user_id]
=====Phase 3=====
▪ Credit= 300 + amount(credit card)
▪ Hw_set_dict:
{HWSets_id: [(datetime, number_in_rent)]}
e.x.
{"hw_set_id_1": [
(2021-04-22 8:00 , 2),
(2021-04-22 9:30 , 1),
],
"hw_set_id_2": [
(2021-04-20 7:00 , 5),
(2021-04-23 15:30 , 4),
],
}
```



## HardwareSet: hardware.py

- Hardware set id
- HWSets name
- capacity
- availability

# 5: Web Development

Frontend: ReactJS

Backend: Flask

Database: MongoDB Atlas

Deployment: Heroku

UI: CSS, Material UI





# 6: Software Testing

- Frontend
  - React Testing Library
  - Rendering vs functionality
    - Rendering Tests
      - ✓ both check in/out renders without crashing
      - ✓ render CheckIn heading (user-friendly)
      - ✓ render CheckIn heading
      - ✓ render CheckOut heading
      - ✓ render checkbox labels
    - Functionality Tests
      - ✓ test checkbox response
      - ✓ test CheckInTable.js against dummy data
      - ✓ test stepUp/Down in CheckinTable.js

Test Suites: 1 passed, 1 total. Tests: 8 passed, 8 total

# 6: Software Testing

- Backend
  - Flask Testing Library, Pytest
  - Rendering vs functionality

A test client is created to test the hardware module. Four tests are conducted:

✓ `test_hardware_module_hwtable(client)` # test the rendering and length of hardware table

✓ `test_hardware_module_get_projects(client)` # test projects can be retrieved from database for the test client

✓ `test_hardware_module_checkout(client)` # pick up a random project to checkout random amount for random existing hardware set, check if the changed hardware amount in the 'Project' collection and 'HWSet' collection in MongoDB meets expected checkout value.

✓ `test_hardware_module_checkin(client)` # pick up a random project which has hardware in rent to checkin random amount for random existing hardware set, check if the changed hardware amount in the 'Project' collection and 'HWSet' collection in MongoDB meets expected checkout value.

All tests passed, which indicates the rendering and functionality of hardware module in flask backend works normally.

# 7: Information Hiding and Refactoring

1. app.py: improve the `@app.route('/project', methods=['POST', 'GET'])` to separate each request to another another method(Yue)
2. Generate Datasets page dynamically: both structure and download links (Mona)
3. Combine the Login component and Register component into one component,(Sohaib)
4. Figure out how do perform unit testing using React.js and Python (Yue, Lian)
5. Refactor the code in the HWform.js, separate come component out (Rachel)
6. Clarify billing requirements (Mona)

Refactor the current version -> start unit testing for one module for frontend and backend, separately -> refactor again -> add in new features(billing, then the features we are interested in) -> refactor

# 7.1: Refactoring

- Frontend
  - Login Page and Register Page
  - Dataset Page
- Backend
  - Use Flask blueprint to split app.py into 4 modules

## 7.2: Information Hiding

- Frontend
  - Hardware page, separate out different forms
  - Project Page, separate out different components
- Backend
  - In the `project_module.py`, separate the codes into helper functions.

# 8: Scalability

- Add More Hardware Resources

## Database:

In data\_service/hardware.py, we have implemented two functions

```
def create_new_HWSet(HWSet_name: str, capacity: int) -> str:
    """
    create new HWSet
    :param hardware_name: HWSet name
    :param capacity: HWSet capacity
    :returns: HWSet_id in str format if success; -1 if capacity is negative
    """
```

and

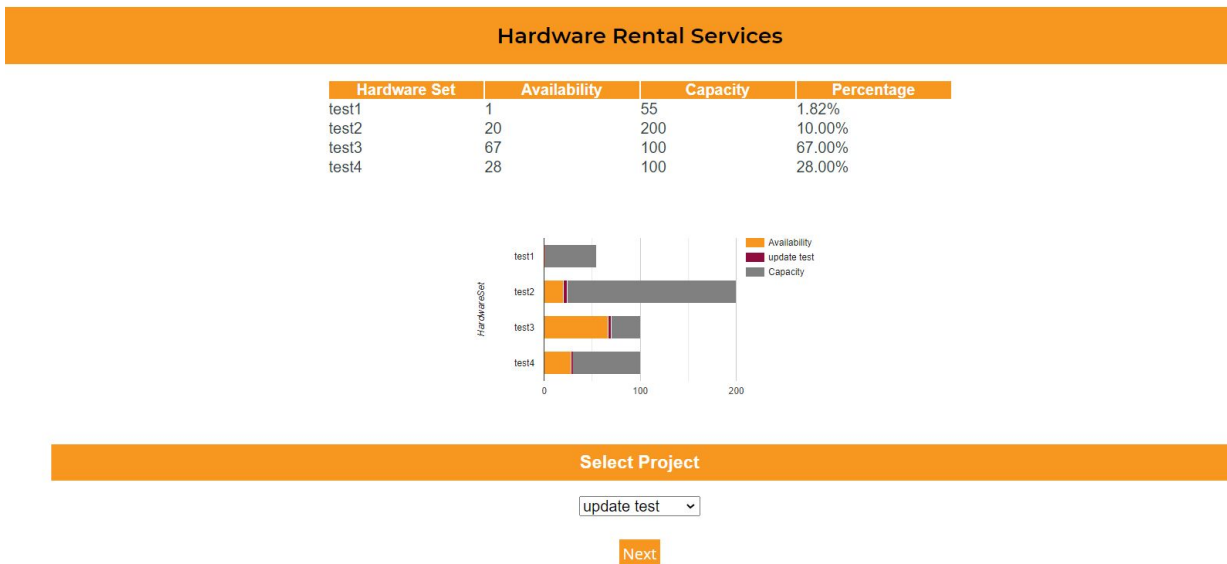
```
def set_HWSet_capacity(HWSet_id: str, amt_of_hardware: int) -> int:
    """
    set hardware_set capacity, amt_of_hardware can be positive or negative
    :param HWSet_id: HWSet id
    :param amt_of_hardware: change of amount of hardware capacity in this hardware set
    :returns: new capacity if success; error code as int. -1 when hardware cannot be found ; -2 when decrease of capacity exceeds cur
    """
```

# 8: Scalability

- Add More Hardware Resources

## Web UI:

This feature can be easily implemented by add a new button 'Add Hardware Resources' on hardware web page(<https://wirepowderless.herokuapp.com/hardware>).



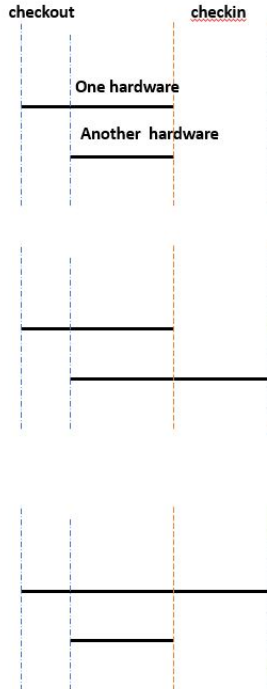
## 8: Scalability

- Tailor the Look and Feel of Web UI by Stakeholders
  - For more general styling like font style, colors, sizes, and background, make changes to App.css.
  - For more specific styling of pages or components, make changes to their individual CSS files.



# 8: Scalability

- View Billing Information by Client
  - Database:



**Total credit used = sum (price per hour \* time span)**

**Change in the MongoDB data structure(no coding needed):**

**(1) In MongoDB, project collection, modify the attribute of HW\_set\_dict**

**From**

```
■HW_set_dict :{HWSet_id_1: 5,  
HWSet_id_2 :5 }
```

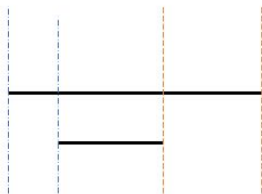
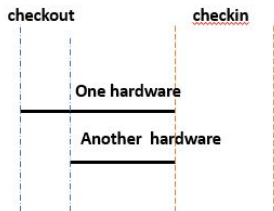
**To**

```
{“hw_set_id_1”: [  
(2021-04-22 8:00 , 2),  
(2021-04-22 9:30 , 1),  
],  
“hw_set_id_2”: [  
(2021-04-20 7:00 , 5),  
(2021-04-23 15:30 , 4),  
],  
}
```

**(2). Add the attribute of ‘credits’ in project collection**

# 8: Scalability

- View Billing Information by Client
  - Database:



Total credit used = sum (price per hour \* time span)

Change in the code:

2. Change the related code associated with HW\_set\_dict and credit in project.py in data\_service folder

Especially the setter

3. Change the related code in flask project\_module when client is checkin/checkout the hardware, update the datetime into HW\_set\_dict

4. Display the credit info for each project on reactJS side

## 8: Scalability

- Web UI
  - Add a column to indicate the 'credit' amount for each project.
  - The history of checkin and checkout log can also be stored and displayed if necessary.



[+ Create Project](#) [+ Join Project](#)

Project ID	Project Name	Owner ID	Date Created	Last Edited	Comment	Actions
607b966fb530d6b79dea9058	update test	604e7d148aaacd6a12855cbb (me)	2021-04-17 21:16:15	2021-05-01 11:04:59	update test	<a href="#">Edit</a> <a href="#">Delete</a>
607dcd1c2e573675664f2ce1	fwefwe131231	604e7d148aaacd6a12855cbb (me)	2021-04-19 13:34:04	2021-04-19 13:34:36	fwefwe321312	<a href="#">Edit</a> <a href="#">Delete</a>
608d798a52ede6a4485f1b01	3132312	604e7d148aaacd6a12855cbb (me)	2021-05-01 10:53:46	2021-05-01 10:53:46	4124124	<a href="#">Edit</a> <a href="#">Delete</a>

Rows per page: 5 1-3 of 3

# 8.1 Future Scalability

- **Implementing custom dataset display:**
  - Use table like with projects page
  - When projects are created, an extra field where user selects tag about topic
  - Sort recommended based on projects in dataset page.
- **Making more datasets available**
  - Change bounds of for loop
- **Allow changing password**
  - Add an additional field to the edit user info form
- **Changing Visual Look**
  - Change color and font in our css file
- **Refactoring CSS**
  - Consolidate the different css files into app.css and change the import statements on relevant .js files.
- **Specific Project pages**
  - Use the template of the main page which has title at the top with comments, timeline by expanding creation date and last updated to an array in MongoDB, show the same graph as in the hardware set for what you have checked out for the project, replace bios with user info.
- **Hardware sets of a type are not interchangeable**
  - Add attribute called “serial number” in MongoDB
  - Hardware page - multiselect

# Module 9 Design Patterns

# Module 10: Machine Learning DP