

# THE OPPENHEIMER PROJECT

---

## BACKGROUND

In the year 1969, the City of Carson is a growing city that housed close to millions of working class heroes<sup>^</sup>. To support the growing city population, a bill was passed that:

*‘each working class hero is to contribute a fraction of their yearly income towards city building’*

This year, as part of the governor’s initiative to pony up surplus cash in the vault, each working class hero is gifted with taxation relief as recognition for their voluntary contribution to city building efforts.

To facilitate this, the governor has drafted out the Oppenheimer Project. This is a software system that has to support 3 features:

- Enable Clerks to populate a list of working class heroes to the system
- Enable Bookkeepers to retrieve the payable taxation relief for each working class

<sup>^</sup>: also known as employees

[tech.gov.sg](https://tech.gov.sg)

GOVERNMENT TECHNOLOGY AGENCY  
10 Pasir Panjang Road #10-01  
Mapletree Business City, Singapore 117438  
T +65 6211 0888  
E [info@tech.gov.sg](mailto:info@tech.gov.sg)

## How to setup the application and database

To start the application you can run the following commands in your terminal

1. `docker-compose -f {path-to-this-docker-file}/local-docker-compose.yml up -d`
2. Verify that your database is up and running. The credentials are a. Url:  
`jdbc:mysql://localhost:3306/testdb`  
b. Username: user  
c. Password: userpassword
3. Run `java -Dspring.profiles.active=prd -jar {path-to-this-jar}/OppenheimerProjectDev.jar`
4. Verify that you can access the login page via `http://localhost:9997/login`
5. Login credentials for various roles:

Role	Username / Password
Clerk	clerk / clerk
Book Keeper	bk / bk

## Expectations:

1. Automate your tests, preferably in Java.
2. Upload the test suite into a repository and provide us with a link to it
3. In your submission, feel free to document what further tests, if any, that you may wish to implement as part of the application's automated testing pipeline.

[tech.gov.sg](https://tech.gov.sg)

GOVERNMENT TECHNOLOGY AGENCY  
10 Pasir Panjang Road #10-01  
Mapletree Business City, Singapore 117438  
T +65 6211 0888  
E [info@tech.gov.sg](mailto:info@tech.gov.sg)

## USER STORIES

**(User Story 1) Application allows the creation of a single working class hero via API call.**

**AC1:** POST request `/api/v1/hero` with the following payload:

Payload	Example
<pre>{   "natid": &lt;natid&gt;,   "name": &lt;name&gt;,   "gender": &lt;gender&gt;,   "birthDate": &lt;birthDate&gt;,   "deathDate": &lt;deathDate&gt;,   "salary": &lt;salary&gt;,   "taxPaid": &lt;taxPaid&gt;,   "browniePoints": &lt;browniePoints&gt; }</pre>	<pre>{   "natid": "natid-12",   "name": "hello",   "gender": "MALE",   "birthDate": "2020-01-01T23:59.59",   "deathDate": null,   "salary": 10.00,   "taxPaid": 1,   "browniePoints": 9 }</pre>

**AC2:** Field validations are as follow

1. natid must be in the format "natid-*number*", where *number* is between **0** to **9999999** (inclusive)
2. name must contain letters and spaces with a length between **1** and **100** (inclusive)
3. gender must be either **MALE** or **FEMALE**
4. birthDate format is in **yyyy-mm-dd** and cannot be a future date. 'T' is a literal string and the time is in HH:mm:ss
5. deathDate format is in **yyyy-mm-dd'T'HH:mm:ss**
6. salary is a decimal and cannot be negative
7. taxPaid is a decimal and cannot be negative
8. browniePoints and deathDate are nullable

**AC3:** If the natid already exists in the database, return error status code **400**. No changes should be made to the existing record.

**AC4:** Verify record is created in database table **WORKING\_CLASS\_HEROES**.

[tech.gov.sg](https://tech.gov.sg)

**(User Story 2) As the Clerk, I should be able to upload a csv file through the portal so that I can populate the database from a UI.**

**AC1:** When I login as a Clerk, I should be able to upload a csv file to create multiple heroes when I click on "Upload a CSV file" button.



Clerk Dashboard Log out

Welcome Home Clerk

What would you like to do?

Add a hero +

Add

Upload a csv file



Clerk Dashboard Log out

Upload CSV file

Choose file No file chosen

Back Create

**AC2:** The CSV file format is as follows which contains the data to be inserted (without headers):

`<natid>, <name>, <gender>, <birthDate>, <deathDate>, <salary>, <taxPaid>, <browniePoints>`

**AC3:** When I click on Create after I upload the csv file, I should see a success notification. All records are persisted successfully into the database table `WORKING_CLASS_HEROES`.



Clerk Dashboard Log out

Created Successfully!

Upload CSV file

Choose file hero.csv

Back Create

[tech.gov.sg](https://tech.gov.sg)

GOVERNMENT TECHNOLOGY AGENCY  
10 Pasir Panjang Road #10-01  
Mapletree Business City, Singapore 117438  
T +65 6211 0888  
E [info@tech.gov.sg](mailto:info@tech.gov.sg)

**AC4:** There is an erroneous record in the csv file. When I click on **Create** after I upload the csv file, I should see a failure notification stating that the SUT failed to persist that record. All records which passed validation will still be persisted in **WORKING\_CLASS\_HEROES**.

## Clerk Dashboard

Log out

### Unable to create hero!

There are 1 records which were not persisted! Please contact tech support for help!

Upload CSV file

Choose file hero.csv

Back

Create

**(User Story 3) As the Book Keeper, I should be able to generate a Tax Relief Egress File from the UI**

**AC1:** When I login as a Book Keeper, I should be able to generate a tax relief egress file **taxrelief.txt** by clicking the **Generate Tax Relief File** button

## Book Keeper Dashboard

Log out

Welcome Home Moh Pehl

### What would you like to do?

Generate Tax Relief File

Nothing running at the moment

### Search for a hero

Show 10 entries

Search By Name: Enter name here...

Name	Gender	Birth Date	Death Date	Salary	Tax Paid	Tax Relief
No records found!						
Name	Gender	Birth Date	Death Date	Salary	Tax Paid	Tax Relief

Showing 0 to 0 of 0 entries

Previous Next

**AC2:** File contain a body where each line is in the format: <natid>, <tax relief amount>, followed by a footer which is a number that indicates the total number of records written to the file

**AC3:** If there are no records to be generated, a file is still being generated containing only the footer

[tech.gov.sg](http://tech.gov.sg)

GOVERNMENT TECHNOLOGY AGENCY  
10 Pasir Panjang Road #10-01  
Mapletree Business City, Singapore 117438  
T +65 6211 0888  
E [info@tech.gov.sg](mailto:info@tech.gov.sg)

**AC4:** Each time a file process is being triggered, a file of **FILE\_TYPE: TAX\_RELIEF** record is being persisted into a database table **FILE** containing the file status, total count of records being written into file. File statuses at the end of a job are **COMPLETED** (on successful generation) or **ERROR** otherwise

**(User Story 4) As the system owner, I want to provide an API that creates a working class hero with vouchers.**

**AC1:** POST request **/api/v1/hero/vouchers** with the following payload:

Payload	Example
<pre>{   "natid": &lt;natid&gt;,   "name": &lt;name&gt;,   "gender": &lt;gender&gt;,   "birthDate": &lt;birthDate&gt;,   "deathDate": &lt;deathDate&gt;,   "salary": &lt;salary&gt;,   "taxPaid": &lt;taxPaid&gt;,   "browniePoints": &lt;browniePoints&gt;,   "vouchers": [     {       "voucherName": &lt;voucherName&gt;,       "voucherType": &lt;voucherType&gt;     }   ] }</pre>	<pre>{   "natid": "natid-12",   "name": "hello",   "gender": "MALE",   "birthDate": "2020-01-01T23:59.59",   "deathDate": null,   "salary": 10.00,   "taxPaid": 1,   "browniePoints": 9,   "vouchers": [     {       "voucherName": "VOUCHER 1",       "voucherType": "TRAVEL"     }   ] }</pre>

**AC2:** Field validation are the same as **User Story 1**

**AC3:** If vouchers is null or empty, the working class hero cannot be created

**AC4:** Vouchers are created in table **VOUCHERS**

**AC5:** If any validation fails, nothing is persisted.

**(User Story 5) As the system owner, I want to have an API that hits an external service's API to find out if a working class hero owes money**

**AC1:** GET request `/api/v1/hero/owe-money` with query parameter `natid=<number>`

```
/api/v1/hero/owe-money?natid=1
```

**AC2:** The `natid` value should only accept numeric values.

**AC3:** The system should receive a response with the payload format like:

```
{
  "data": "natid-<number>",
  "status": "OWE"
}
```

where ``natid-<number>`` is the `natid` provided as part of the query and ``status`` should be either ``OWE`` or ``NIL``

**AC4:** The system should respond in the following format

```
{
  "message": {
    "data": "natid-<number>",
    "status": "OWE"
  },
  "timestamp": "<timestamp>"
}
```

where ``natid-<number>`` is the `natid` provided as part of the query and ``OWE`` is the value returned by the external system.

**(User Story 6) As the system owner, I want to provide an API that gives insight into the number of vouchers each customer has for each voucher category**

**AC1:** `GET` request to `/api/v1/voucher/by-person-and-type`. System will provide a response in the following format:

```
{
  "data": [
    {
      "name": "<name of working class hero>",
      "voucherType": "<voucher category>",
      "count": <count>
    }
  ]
}
```

where `<voucherType>` is the voucher category and the `<count>` reflects the number of vouchers in that category