COMP0035 Tutorial 7: Application design

Refer to the architecture and design guide as it has details that are not repeated here.

The tutorial continues with the paralympics events example used in earlier tutorials.

- Activity 1: Design REST API
- Activity 2: Design web app with machine learning or other feature
- Activity 3: Design dashboard app
- Activity 4: Review your design against the design principles

Activity 1: Application design for a REST API

A REST API takes a URL request and returns a response. The application does not have a user interface. For this type of application you need to include:

- The resources
- The data format
- A list of URIs and the HTTP method(s) for each

Requirements

Data

The data contains the following resources: Event and Region

Event includes the following fields with an example of the data held:

- Host city: Tokyo
- Year: 1964
- Country: Japan
- Type: Summer
- Start: 1964-11-08
- End: 1964-11-12
- Participants (M): 195.0 (total male participants)
- Participants (F): 71.0 (total male participants)
- Participants: 266 (total participants)
- Duration: 4 (number of days the event lasted)

Region includes the following fields with an example of the data held:

- NOC: JPN (3-letter code representing the region)
- region: Japan, the region's name
- notes: any notes relating to the region, typically blank

Note that the data set .csv files do not have an 'id' column. In the database design activity next week you will take the data from the dataset and add it to a database. When you do this, each entry in the database will have a unique ' id'.

User stories

The REST API is designed for a developer who would use the routes to get data for their application.

The user stories have been documented as follows.

1. As a developer, I want all data to be returned in JSON format so that it is consistent with other APIs I use
2. As a developer I want to get a list of all event ids, host city and the year they were held so that I know which events are available.
3. As a developer I want to get all the details for a given event so that I can use these details in my app
4. As a developer I want to amend any detail for a given event so that I can correct errors
5. As a developer I want to add a new event so that future events can be recorded. Acceptance criteria: an event is only created if all fields are provided.
6. As a developer I want to delete an event so that I can remove potential duplicates or resolve issues.
7. As a developer I want to get a list of all region ids and NOC codes they were held so that I know which are available
8. As a developer I want to get all the details for a given region so that I can use these details in my app
9. As a developer I want to amend any detail for a given region so that I can correct errors
10. As a developer I want to add a new region so that new regions can be identified. Acceptance criteria: a region is only created if the NOC and Region fields are provided, 'notes' is optional.
11. As a developer I want to delete an event so that I can remove potential duplicates or resolve issues.

Document the design

Use steps that are detailed in the architecture and design guide:

1. Choose a way to diagram/model the REST API design: table, class diagram, API design tool e.g. Swagger
2. Decide on the resources and data format (you have been given these in the requirements above)
3. For each user story that implies a REST API route:
    1. Decide on the URI and add to the diagram/table
    2. Decide on the HTTP method (or methods) and add to the diagram/table

For example, using a table structure the route for user story 2 is shown below. The 'http://www.mydomain.com/' part of the URI has been omitted.

URI	***HTTP method***	***Description***
/event/	GET	Returns id, host city, year for all paralympic events

Activity 2: Application design for a web app that deploys a machine learning model and/or other functions

This example does not include the deployment of a machine learning model, however the design approach is the same.

You will use Python Flask to create this type of app.

A web application has a user interface that is displayed in a web browser. A user of the application enters a URL which sends a request to the server to return a response (usually an html web page). The server looks to see if it knows about this URL by checking its route definition. If it finds a route then it will run a Python function associated with that route that does something, e.g.

- gets data and puts it into a web page
- takes data from a submitted form and creates or updates a record in the database
- takes data from a form, passes this to a machine learning model to get a prediction and returns the predicted value in a web page
- etc

For the design of this application you will need to include:

- a list of the available routes (these correspond to the URLs) and the type of HTTP request that is allowed
- functions that carry out the actions for each URL (route)
- classes that model any data in the app

Data

The .csv file data set includes the following fields, an example of the type of data in each field is included:

- Host city: Tokyo
- Year: 1964
- Country: Japan
- Type: Summer
- Start: 1964-11-08
- End: 1964-11-12
- Participants (M): 195.0 (total male participants)
- Participants (F): 71.0 (total male participants)
- Participants: 266 (total participants)
- Duration: 4 (number of days the event lasted)
- Region: JPN 3-letter National Olympic Committee code

User stories

The ‘user’ is any member of the public who wants to find out about past paralympic events.

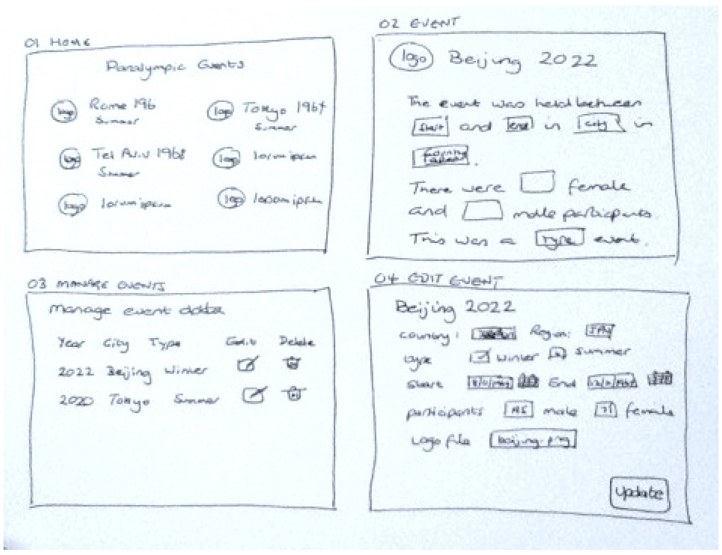
The user stories for this app are as follows. These are not written fully, some are missing the ‘so that...’ and few have acceptance criteria. They should be I’ve just enough for you to use as a basis for the application design:

1. As a user I want to view the web app on any device (laptop, mobile phone, tablet).
2. As a user I want to see a list of all the Paralympic events that are in the database so that I know what is available. Acceptance criteria: Must display the host city, year and type of event (Winter/Summer)
3. As a user I want to see detailed information about each event.
4. As an administrator I want to be able to add a new record to the database when a new Paralympic event occurs. Acceptance criteria: Should only be available if the user is validated.
5. As an administrator I want to be able to amend a record in the database if an error is reported to me. Acceptance criteria: Should only be available if the user is validated.

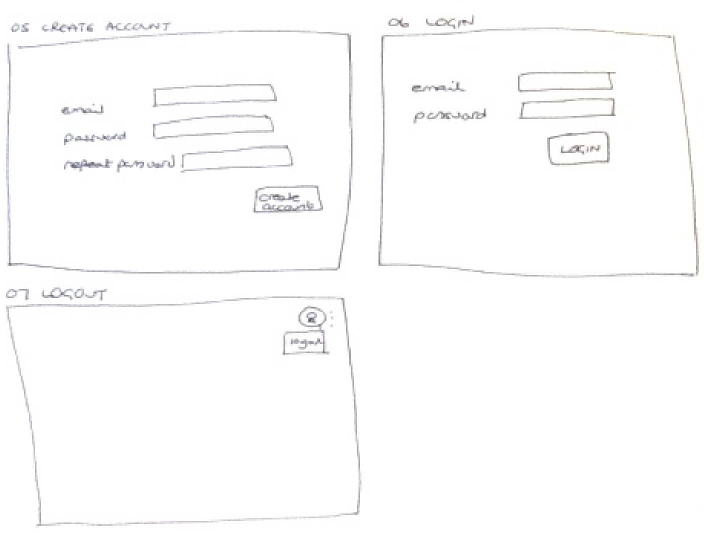
Note: there may be a better way to handle the admin validation, a simple registration and login has been assumed for this example.

Wireframes

The wireframes have been drawn as:



0-1. paralympic app wireframes



0-2. paralympic app wireframes account

1. Design the model classes and their methods

Use the guide which explains how to do this.

Consider the data set data and any other objects in the app. In this case there is a concept of an administrator which could be modelled as follows:

Class: Administrator

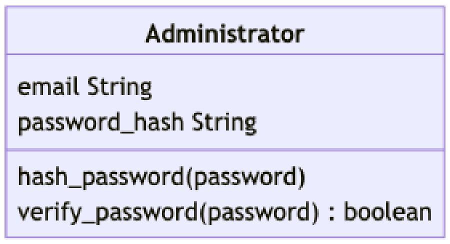
Attributes:

- email String
- password_hash String

Methods:

- hash_password(password) Takes the text password and converts to hash
- verify_password(password) : boolean Takes the text password and checks against the password_hash and returns TRUE if there is a match

Or could be drawn in a class diagram format:



2. Design the routes and controllers

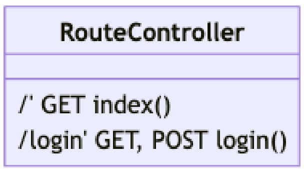
Use the guide which explains how to do this.

Use the user stories and the wireframes to decide which routes you think are needed.

For example, the home page and login could be show either as an entry in a table such as this

Route	***HTTP method***	***Function***	***Wireframe reference***
`/`	GET	index(): Returns the home page	01 Home
`/login`	GET, POST	login(): GET Returns login form page POST Takes the email and password from the submitted form, creates an Admin object and validates the email and password, if valid user is returned to the Manage Events page	06 Login, 03 Manage events

Or as a class diagram entry, this has been modified from a standard class descriptor to represent Flask routes. You won't find this documented anywhere!



Activity 3: Application design for a data visualisation dashboard

A dashboard is accessed via a browser and may be one page, or multiple pages. Each page will display charts or other visualisations (e.g. statistics) and there are typically options to allow the user to interact with the charts, such as zooming in to a particular date range, adding additional variables to be displayed, clicking on map markers to display further details etc. Each of the charts, visualisations and interactive feature implies functionality the app will provide. You will use Plotly Dash open source Python version so you will be writing Python functions and classes.

In this application the design will likely include:

- functions to generate charts/visualisation and interactions on the page, possibly grouped in one or more modules
- classes to represent the data
- relationships between the classes/functions

Requirements

User stories

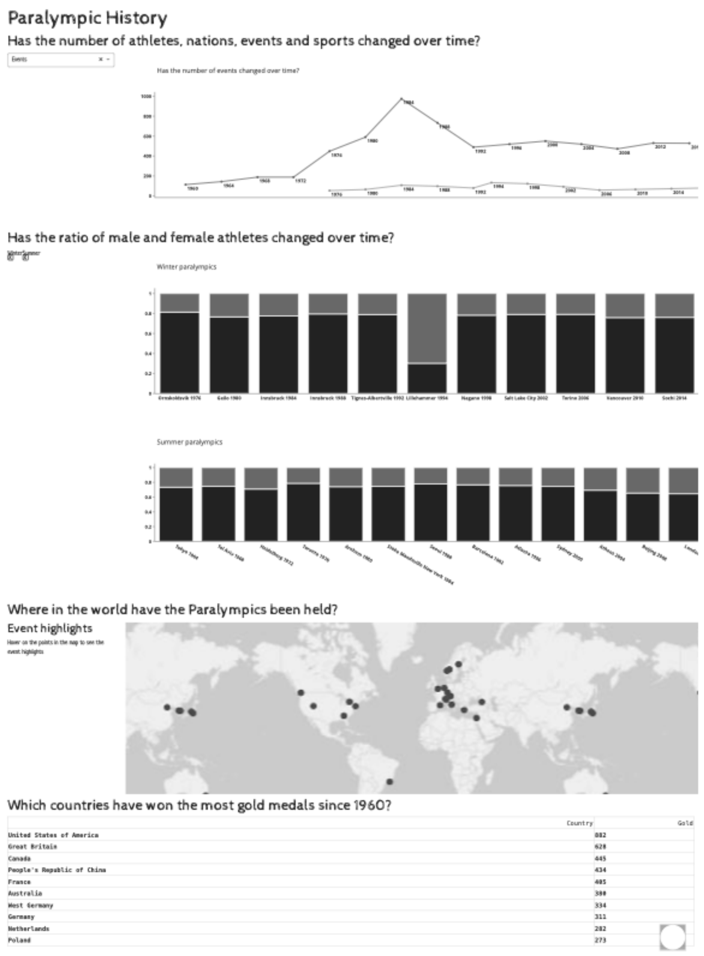
The intended users of the web app are 16-18 year old students researching the paralympics for a school project. This is a niche audience and not really very realistic!.

1. As a student I want to see trends over time to help answer the question "Has the number of athletes, nations, events and sports changed over time?"
2. As a student I want to see for the summer and winter paralympic events the numbers of male and female competitors so that I can answer the question "Has the ratio of male and female athletes changed over time?"
3. As a students I want to see where in the world the paralympic events have been and to be able to see some details such as the year, name and type of event so that I can answer the question "Where in the world have the Paralympics been held?"
4. As a student I want to see the rankings of gold medal winning countries so that I can answer the question "Which countries have won the most gold medals since 1960?"
5. As a student I want an app that helps me to understand which charts will help me to answer each of the homework questions.

Wireframe

In this activity you have a screenshot of an actual app rather than a wireframe, so it doesn't use the standard shapes.

The following shows a layout for a single page Dashboard app.



Paralympic History

Has the number of athletes, nations, events and sports changed over time?

Has the ratio of male and female athletes changed over time?

Where in the world have the Paralympics been held?

Event highlights

Hover on the points in the map to see the event highlights

Which countries have won the most gold medals since 1960?

Country	Gold
United States of America	882
Great Britain	628
Canada	445
People's Republic of China	434
France	405
Australia	380
West Germany	334
Germany	311
Netherlands	282
Poland	273

0-3. Dashboard layout

Identify functions to create the charts

In this example the chart types are already chosen for you.

Consider the image of the page above, identify the functions needed to create charts:

1. create_line_chart(events, chart_type): Returns a line chart based on the selection made in the dropdown and the events data.
2. ...
3. ...
4. ...

Identify the functions for any interactive elements (e.g. dropdowns, sliders, checkboxes, tabs etc):

1. get_line_chart_type_dd(selected_option): Captures the chart type selection from the dropdown and passes it to the create_line_chart() function.
2. ...
3. ...

Having identified the functions, you may wish to consider how related they are and group them into relevant modules.

Identify classes that map to the data

The .csv file data set includes:

- Host city: Tokyo
- Year: 1964
- Country: Japan
- Type: Summer
- Start: 1964-11-08
- End: 1964-11-12
- Participants (M): 195.0 (total male participants)
- Participants (F): 71.0 (total male participants)
- Participants: 266 (total participants)
- Duration: 4 (number of days the event lasted)
- Region: JPN 3-letter National Olympic Committee code

Identify the classes and attributes. One pattern is to match the classes to each table in the database. The database design is covered in the next tutorial, so you may need to review your class design again after doing the database design.

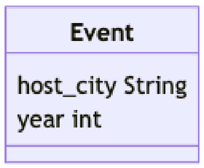
Class: Event

Attributes:

host_city String year int

... etc

If you are drawing a class diagram this could be drawn as follows (this is incomplete, only two attributes are shown):



Consider the relationships between the classes and functions

In this example you have functions to generate and modify charts that you might group into a ‘chart’ module; and a single class representing the data. Drawing a diagram of the relations doesn’t really give much insight but is included to illustrate how you could adapt a class diagram.

The classes are incomplete, you should have completed these in the previous steps! Your ‘chart’ class may be a module rather than a Python class.

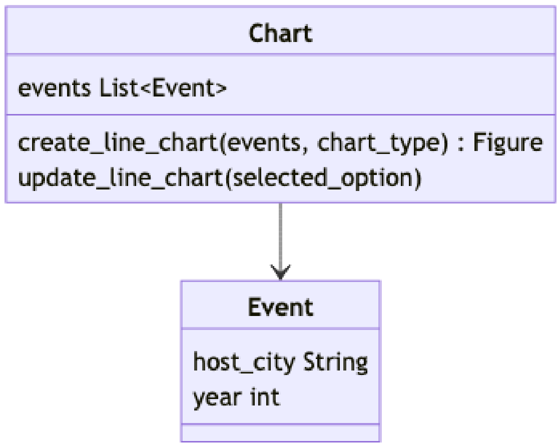


Chart
events List<Event>
create_line_chart(events, chart_type) : Figure
update_line_chart(selected_option)
Event
host_city String
year int

Activity 4: Review your design against the principles

Review your design (or designs if you completed more than one of the activities) against the design principles listed in the application design guide. You may then to iterate the design and review steps.

This is not easy as you are trying to critically evaluate our own work against a general design principle or pattern.

As a starting point, look at the tables or diagrams you have created and consider:

- Are the relationships between classes, modules, functions only those that are needed?
- Are the attributes and methods/functions within a class or module cohesive? If not do you need to split them into more classes, functions, modules.
- Does the structure seem clear (simple as possible)?
- Would the structure allow you to reuse components in another application?
- Is the same business logic represented in more than one place? If so then remove the duplication.

This is not an exhaustive list of questions. This aspect is challenging!