Activities 6: Requirements analysis and wireframes

Table of contents

COMP0035 Activities 6: Requirements and wireframes

Theme: **Designing applications**

The next two sets of activities do not involve coding.

This week's activities introduce you to:

- techniques for documenting requirements
- wireframes

The activities are **not** exemplars for the coursework; since many of the examples are partial solutions only.

There are other techniques that can be used for this stage beyond the few covered here; students aiming for higher marks are encouraged to read more widely for this topic and to choose an appropriate blend of methods to use. There are some activities on Moodle that introduce other techniques.

Complete the activities

Tutorial activities can be found in the docs/6_requirements folder. These are:

1. Introduction (6-01-introduction.md)
2. Identify requirements (6-02-identify-requirements.md)
3. Document requirements (6-03-document-requirements.md)
4. Prioritise requirements (6-04-prioritise-requirements.md)
5. Draw wireframes (6-05-wireframes.md)

Next activity (6-01-introduction.md)

1. Starting the requirements analysis phase

Starting the requirements analysis phase

By the time you are ready to start gathering, documenting and analysing the requirements for a system or app, you would have some understanding of what the project, or product, is about. You may have one or more of:

- Persona
- Problem statement
- Goals/objectives
- Questions to answer from the data

From these:

- You may be able to identify groups of functionality that the system should achieve from the product vision or goals.
- The questions to be answered using the dataset may help to guide the requirements for visualisations in a dashboard.
- The persona might inform the requirements if there is information on current frustrations or goals. Its main use in requirements is likely to be in trying to consider from the persona's perspective what the web app needs to do. The persona represents the main user base of your application, also consider if there are others who will use the app as they may have different needs.

Paralympics project

The theme of paralympics is continued based on the paralympics dataset from earlier tutorials.

Two project concepts are used in this week's activities.

Project 1	Paralympics quiz app
Target audience	UK school pupils aged 11 to 16. These students are used to using mobile phones and web based apps.
App overview	A web app with a quiz feature for students, information pages, charts and an area for teachers to enter questions. It uses the existing paralympics data set and will also collect new data (teacher questions).
Scenario	Students have been asked to research facts about the paralympic games. They will use the data they find to generate fact files. Their tutor has asked them as part of their research to find out about:
Project 2	Paralympics prediction app
Target audience	UK school pupils aged 16 to 18. These students are used to using mobile phones and web based apps.
App overview	A web app that includes a machine learning prediction algorithm to predict country medal results in future paralympic games. Historical medal rankings will be included in the app. The app will use the paralympics dataset medal rankings; and a machine learning model trained on the dataset.
Scenario	Students have been asked to learn about the medal rankings in previous paralympic games, and to make predictions for the next paralympic games including:

Optional: gen-AI activity to create a persona

Personas can help to guide the design of an app to ensure it meets the needs and interests of the intended target audience.

Try using a gen AI tool to generate a persona based on the target audience of one of the scenarios. For example, try this prompt for the quiz app project:

```
Generate a persona for a UK school student aged 11-16. This is for a
project to design and develop a web app with statistics and information
about paralympic games.
```

Next activity (6-02-identify-requirements.md)

2. Identify requirements

What is a requirement?

Requirements set out what a software system should do along with any constraints imposed on its operation and implementation.

A requirement is some capability needed by a stakeholder to solve a particular problem or achieve an objective. The requirements define a capability that the solution (web app) you will design and deliver has to have for it to be considered by the stakeholders as meeting their needs.

In the context of the coursework project, the stakeholders are the imagined people who will be using the web app.

Requirements are often discussed in two broad categories:

1. **Functional requirements** are those that specify *what* the system should do; that is a behaviour, feature or function of a system. For example:

   ○ "Create a new customer account"

2. **Non-functional requirements** are those that specify *how* the system should. These often have implications for the system architecture and sometimes sub categorised by performance, availability, reliability, usability, flexibility, configurability, integration, maintainability, portability, and testability. For example:

   ○ "The application must support devices running OS verions 3.4, 3.5 and 3.6"
   ○ "The error rate of users submitting their payment details at the checkout page must not exceed 10%."
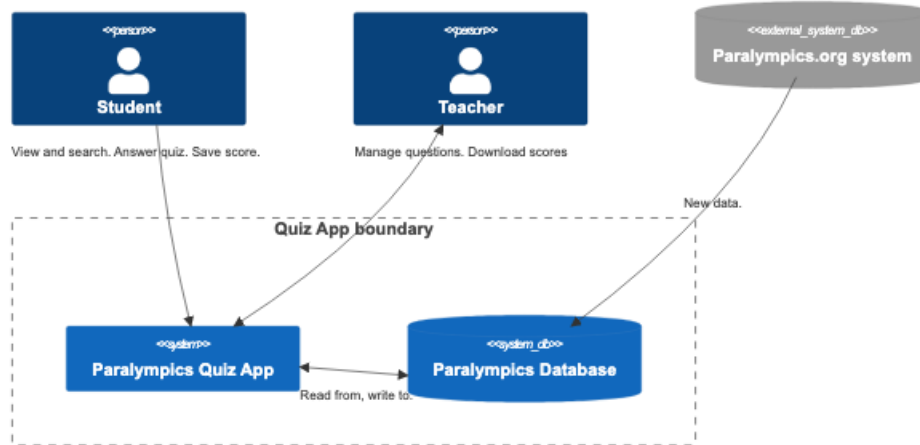
Context diagram

While not essential, a context diagram may be useful as a next step as can help you to consider:

- what is the 'system' boundary, i.e., what is in the web app (the system), what is definitely not in it
- who or what (e.g., another system) interacts with the 'system' (external entities)
- what information needs to flow between the 'system' and the 'external entities'

Consider the following context diagram for the quiz app:

System Context diagram for Paralympics Quiz Web App

Identify the requirements

The Business Analysis Body of Knowledge (BABOK) (https://babokpage.wordpress.com/elicitation/) lists techniques that can be used in a real world project to elicit requirements. You would gather requirements using one or more of these techniques:

- Interview
- Workshop
- Survey/ Questionnaire
- Interface analysis
- Focus group
- Observation or ethnography
- Brainstorming
- Prototyping
- Analysing documentation

In this course, since there are no users for you to talk to, and you are not allowed to involve others outside the course as requires ethics approval that you do not have, then your options are limited.

Useful techniques for the coursework include brainstorming, low fidelity prototyping (e.g. wireframes), looking at similar existing web apps.

For this example consider the context diagram and brainstorm what the users of the app may need to be able to do.

The following mindmap represents the potential results of a brainstorming activity (apologies for the awful colours!):

Activity: Brainstorm requirements for the prediction app

Identify requirements for the prediction app.

	Paralympics prediction app
Target audience	UK school pupils aged 16 to 18. These students are used to using mobile phones and web based apps.
App overview	A web app that includes a machine learning prediction algorithm to predict country medal results in future paralympic games. Historical medal rankings will be included in the app. The app will use the paralympics dataset medal rankings; and a machine learning model trained on the dataset.
Scenario	Students have been asked to learn about the medal rankings in previous paralympic games, and to make predictions for the next paralympic games including:

Next activity (6-03-document-requirements.md)

3. Document requirements

Approaches to documenting requirements

Select a natural language structured format to formally record the requirements. The two options likely to be of most use for this course are:

- User stories
- Functional and non-function requirements

Consider if there is a format that better suite the nature of your project/product. For example, User Stories would be expected if you are following an overall Agile approach to the project.

User story format

Most agile methods use the user story format.

The basic user story template is:

As a _role_, I want _goal_ so that _benefit_.

Acceptance criteria, or tests, are often added to the user story to clarify the definition. This is one way of adding non-functional requirements (or 'constraints') to user stories.

For example:

As a website user, I want search functionality to be available on all pages so that I can search for books using keywords.
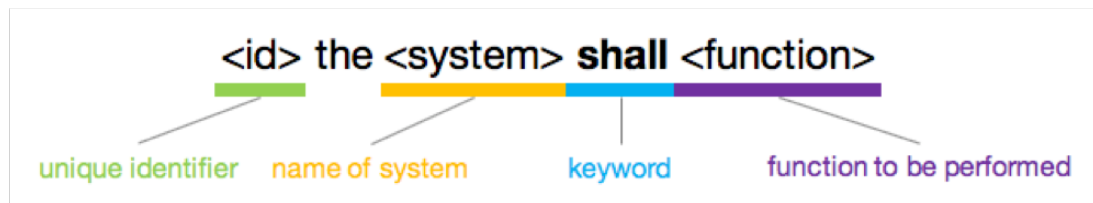
Acceptance criteria:

1. Search box should accept alphanumeric values
2. Search results should display 10 items per page
3. System responds to all search requests within 2 seconds of receiving the request

---

User stories are not typically classified as functional/non-functional. In some cases the non-functional requirements may be covered by the acceptance criteria; other times you will see user stories written for the non-functional needs (https://www.mountaingoatsoftware.com/blog/non-functional-requirements-as-user-stories) such as:

> "As a customer, I want to be able to run your product on all versions of Windows from Windows 95 on."

Functional and non-functional requirements

0-1. Natural language specification

Each requirement should focus on a single distinct feature or behaviour and should be written in the same uniform sentence structure. They should not be too vague or abstract; too general or imprecise; or include implementation information. e.g. "32 The ATM system shall validate the PIN"

You will typically see these listed in a table format, and they may be grouped in some way e.g.

ELECTION MAINTENANCE

RQ16	The EVS shall email reminders to users who haven't yet voted 2 days before the election end date.	Functional	Election maintenance	Could Have
RQ17	The EVS shall display a list of candidates for each category and briefly outline their manifestos and store details in the election database.	Functional	Election maintenance	Must Have
RQ18	The EVS shall display a list of categories for each election and store details in the election database.	Functional	Election maintenance	Must Have
RQ19	The EVS shall support the insertion of candidate photographs.	Functional	Election maintenance	Could Have
RQ20	The EVS shall automatically start and stop an election based on a predefined date range.	Functional	Election maintenance	Should Have
RQ21	The EVS shall email an invitation to vote to all relevant voters once the election has been setup.	Functional	Election maintenance	Should Have

0-1. Functional requirements

RQ3	The EVS shall use an Internet browser as its user interface.	Non-Functional	General	Must Have
RQ4	The EVS shall support the ALL versions of Internet Explorer and Netscape browsers.	Non-Functional	General	Must Have
RQ5	The EVS shall be written using standard Java to run on different operating systems (e.g. Linux and Windows).	Non-Functional	General	Must Have

0-1. Non functional requirements

Document the quiz app requirements as user stories

The following is a potential list of user stories for the quiz app based on the mindmap.

Ref	User story	MoSCoW Categorisation
1.	As a system administrator I want to make updates to the paralympic games database so that the data remains accurate and up to date.	
2.	As a system administrator, I want to make sure that only a teacher can edit questions and quizzes, and access student scores.	
3.	As a student I want to browse information about paralympic games so that I can see what is available.	
4.	As a student I want to search for information so that I can answer quiz questions.	
5.	As a student I want to view statistical information about the paralympics for each event and across events so that I can find information for my project.	
6.	As a student I want to see charts that show trends over time in the events data and be able to narrow the charts to specific fields or date ranges so that I can find information for my project.	
7.	As a student I want to complete a quiz so that I can assess my learning.	
8.	As a student I want to submit my quiz score so that I can participate in the competition to see who scores the highest.	
9.	As a teacher I want to add and edit questions so that these can be added to quizes.	
10.	As a teacher I want to add and edit quizzes, including adding questions to them, so that I can offer my students formative assessment.	
11.	As a teacher I want to access students' scores to see which student gained the highest score.	
12.	As a student or teacher I want to be able to access the quiz app from a PC or mobile phone so that I can work at home or in school.	
13.	As a student or teacher I want to be able to use the app through a web browser so that I do not have to install an application on my device.	

Activity: write user stories for the medal standings app.

Adding detail to user stories

User stories are typically written with just enough detail and act as placeholders for later conversations.

Detail may be added following these conversations, usually just before the 'sprint' or 'iteration' in which the user story will be developed. Such detail is also likely to be useful when you later start testing.

This further detail can be added using 'acceptance criteria' or 'constraints' to the user story.

Activity: Add detail to a few of the following user stories.

Ref	User story	Constraints
1.	As a system administrator I want to make updates to the paralympic games database so that the data remains accurate and up to date.	Updates will be requested once per day. A request will be triggered in the quiz app and will use the paralympics API to look for changes since the last date of update. Any changes will be updated in the database. The last update date will be stored.
2.	As a system administrator, I want to make sure that only a teacher can edit questions and quizzes, and access student scores.	Authentication must use the school's single sign on method.
3.	As a student I want to browse information about paralympic games so that I can see what is available.	
4.	As a student I want to search for information so that I can answer quiz questions.	
5.	As a student I want to view statistical information about the paralympics for each event and across events so that I can find information for my project.	
6.	As a student I want to see charts that show trends over time in the events data and be able to narrow the charts to specific fields or date ranges so that I can find information for my project.	
7.	As a student I want to complete a quiz so that I can assess my learning.	
8.	As a student I want to submit my quiz score so that I can participate in the competition to see who scores the highest.	
9.	As a teacher I want to add and edit questions so that these can be added to quizes.	
10.	As a teacher I want to add and edit quizzes, including adding questions to them, so that I can offer my students formative assessment.	
11.	As a teacher I want to access students' scores to see which student gained the highest score.	
12.	As a student or teacher I want to be able to access the quiz app from a PC or mobile phone so that I can work at home or in school.	

Ref	User story	Constraints
13.	As a student or teacher I want to be able to use the app through a web browser so that I do not have to install an application on my device.	

4. Requirements prioritisation

Why prioritise?

Once you have your initial list of documented requirements, decide how to prioritise them.

Further, the number of requirements exceeds the number of features that can be implemented within the given time and available resources (people, funding, etc.). Prioritizing requirements helps teams focus on the most important features first. Not all requirements will deliver the same impact or benefit. Typically, those that generate the most business value are given a higher priority.

Business value is not the only factor considered, risk, technical complexity, and other factors can also be considered.

Prioritisation methods

There are numerous techniques for prioritising techniques such as:

- MoSCoW (https://www.lucidchart.com/blog/introduction-to-moscow-prioritization)
- Forced pair ranking (https://www.lynda.com/Project-Management-tutorials/Forced-ranking-prioritization/471658/585113-4.html)
- Story mapping (https://jpattonassociates.com/the-new-backlog/)
- 100 points / $100 method (http://www.modernanalyst.com/Careers/InterviewQuestions/tabid/128/ID/2122/How-is-the-100-point-method-used-to-prioritize-requirements.aspx)
- Priority poker (http://www.uxforthemasses.com/priority-poker/)

MoSCoW prioritisation

MoSCoW is a widely used approach, though other techniques may be more relevant for user stories such as forced pair ranking.

The MoSCoW technique (https://www.agilebusiness.org/dsdm-project-framework/moscow-prioririsation.html) uses a simple categorisation for requirements:

*M*ust have: These requirements that are absolutely critical to the product/project's success.

*S*hould have: These requirements are also important, but may not be as time-sensitive or vital as the "must have" requirements.

*Co*uld have: These requirements are nice to have and would make a great addition to the project, but are not critical. If there's time, consider adding these in.

*W*on't have for now: The value of these requirements is sufficiently low compared to the time, energy, or budget needed. They could be considered at a later time.

There are limitations to using MoSCoW. While MoSCoW helps with organizing the categories by importance, it doesn't provide a way to determine priorities within each of the categories. Relying only on the MoSCoW method, you could unintentionally ignore critical dependencies and the interrelationship between features.

Activity: Prioritise the user stories

Consider each of the user stories and assign the most appropriate categorisation (Must, Should, Could, Won't).

Ref	User story	MoSCoW Categorisation
1.	As a system administrator I want to make updates to the paralympic games database so that the data remains accurate and up to date.	
2.	As a system administrator, I want to make sure that only a teacher can edit questions and quizzes, and access student scores.	
3.	As a student I want to browse information about paralympic games so that I can see what is available.	
4.	As a student I want to search for information so that I can answer quiz questions.	
5.	As a student I want to view statistical information about the paralympics for each event and across events so that I can find information for my project.	
6.	As a student I want to see charts that show trends over time in the events data and be able to narrow the charts to specific fields or date ranges so that I can find information for my project.	
7.	As a student I want to complete a quiz so that I can assess my learning.	
8.	As a student I want to submit my quiz score so that I can participate in the competition to see who scores the highest.	
9.	As a teacher I want to add and edit questions so that these can be added to quizes.	
10.	As a teacher I want to add and edit quizzes, including adding questions to them, so that I can offer my students formative assessment.	
11.	As a teacher I want to access students' scores to see which student gained the highest score.	
12.	As a student or teacher I want to be able to access the quiz app from a PC or mobile phone so that I can work at home or in school.	
13.	As a student or teacher I want to be able to use the app through a web browser so that I do not have to install an application on my device.	

Look at your prioritised list. Of those in the 'Must' which would you work on first?

There is no 'correct' answer. Prioritisation is a judgement in a given set of circumstances by one or more people.
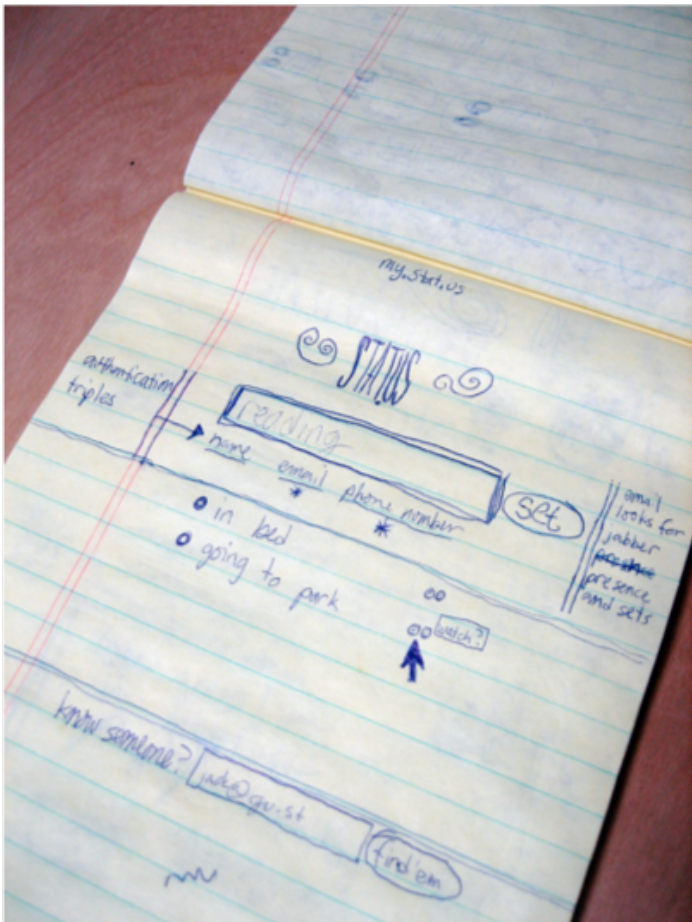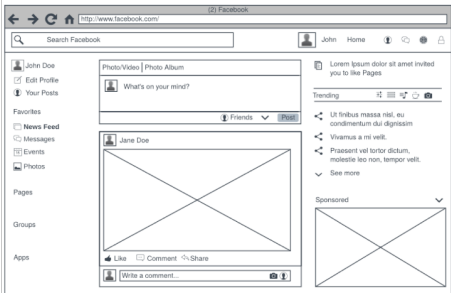
6. Drawing wireframes

What is a wireframe?

A wireframe is a low fidelity design of an application that allows you to focus on basic layout and interaction of an interface independent of styling. It is typically used for early prototyping in web applications, and can be useful for:

- visualising layout and structure
- clarifying functionality
- testing ideas before investing in detailed designs
- communicating ideas between designers, developer, clients, and other stakeholders

It provides an outline of the structure and layout and presents the main information and application flows.

The following examples show both hand drawn and digital wireframes:

Hand-drawn	Digital

Why create a wireframe?

"A picture is worth a thousand words", wireframes clearly communicate design ideas to stakeholders and teammates to facilitate decision-making. Wireframes can provide a reference point for the functionality that will need to be built.

They are quick to create and change so you can explore ideas with minimal effort. Keeping it lo-fi (sketch) helps avoid a focus on visual design.

Some use wireframes to help capture and/or refine requirements; others develop them after the requirements are documented. Whenever you start there is likely to be some iteration.

Which screen size(s) and orientation are you designing for?

Consider your target user group and design for the screen size they are most likely to use. For example, designing for a user audience primarily using a phone would imply a different layout to a target audience primarily using a laptop or desktop PC (not least as one is portrait and the others landscape in orientation).

Statcounter.com (https://gs.statcounter.com/screen-resolution-stats) provides statistics for the most commonly used screen sizes.
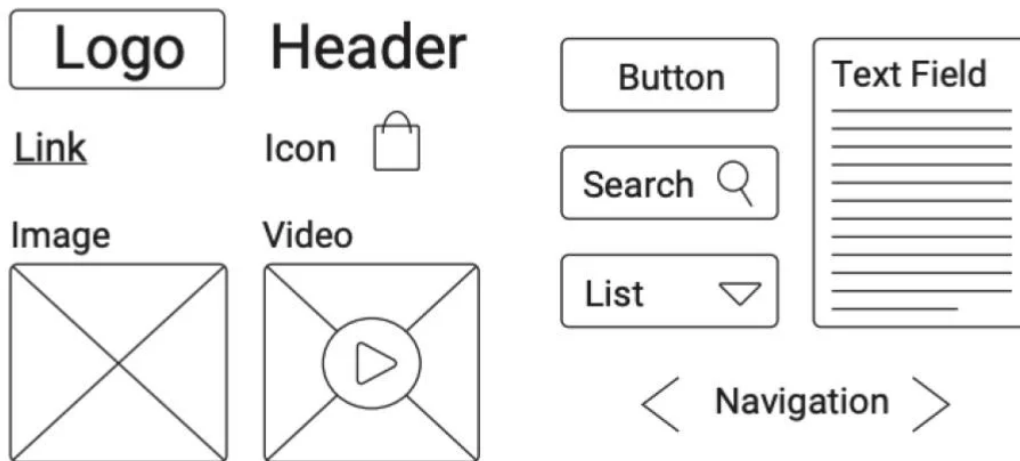
How to create wireframes

There is no single format or approach to creating wireframes.

This is a generalised approach that is influenced by Alex Coleman's article How to create web application wireframes (https://selftaughtcoders.com/how-to-create-web-application-wireframes/) .

1. Start with your requirements (e.g. user stories), and/or if you have them use cases. Try to write down the key goals the user has when using your web app, e.g.

   - Create and manage user account
     - Create an account
     - Update account e.g. change password
     - Login
     - Logout
   - Analyse data using chart X
     - View chart X
     - Tailor chart X
   - Manage data (e.g. in the tutorial example project this might be managing the data associated with a paralympic event)
     - Search for an existing data entry and view it
     - Create a new data entry
     - Update/edit an existing data entry

2. For each of the above sketch out a potential wireframe of the page (or area of the page if a single page app design is used for a dashboard style app).

3. Optionally, consider how the user may navigate from one page/feature to the next. A flow chart may be useful here.

Wireframe tools

Wireframes use generic shapes such as:

0-1. wireframe-elements.png

You can hand draw wireframes.

There are numerous online wireframe tools. They usually require sign up and if you use these, ensure you save and download your designs in an image format (e.g., png, jpd, pdf) before the trial expires.
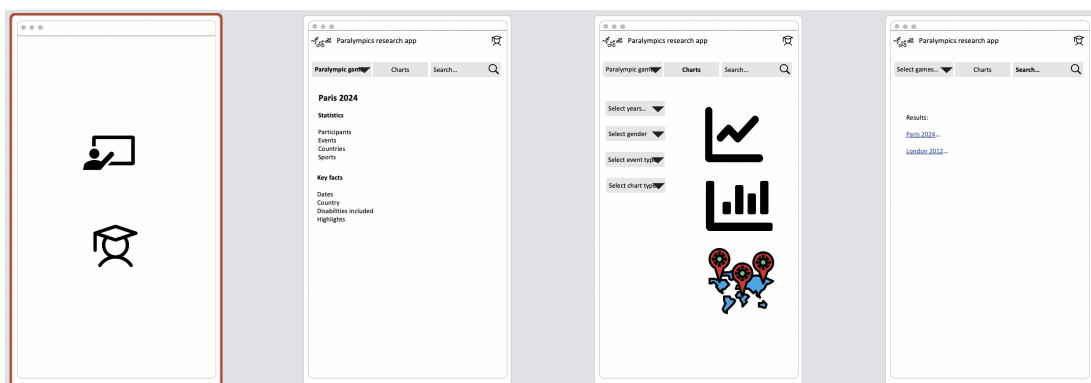
Most basic shapes can also be created in PowerPoint.

While not design tools, if your wireframes contain charts, and you want to understand the options, these chart chooser tools may help. Note that choosing and designing charts is not required for the coursework.

- data visualisation catalogue (https://datavizcatalogue.com/index.html)
- interactive chart chooser (https://depictdatastudio.com/charts/)
- data to viz (https://www.data-to-viz.com/)

Activity: draw wireframes

The following attempt to sketch the screens for the paralympic games fact pages, the chart selector and the search.

0-1. Paralympics app wireframes

Draw your own wireframes for the student user stories related to taking a quiz and saving their score. There is a very basic wireframe PowerPoint file in the '/docs/6_requirements/extra' folder.

Do your wireframes highlight any new user stories that need to be added? Or any constraints that need to be added to the existing user stories?