

Building an App (PAT)

Start with an idea, a theme, something you know a lot about e.g. formula one racing, yachting, books, security products, a school, cars, fashion, makeup, movies, celebrities, property for sale, or rent.

Break your topic into logical blocks i.e. racing car drivers, car manufactures, types of boats, clients, students, products, actors, producers, genres, authors, features, technical datasheets, etc. – whatever is logical to your theme.

1. Build a **database**; a relational database with at least three tables. Start with the collection of data. Good idea to put this into an Excel spreadsheet. (One sheet per table)
 - a. Divide the data into tables e.g. a table for drivers, a table for teams, a table for events, etc.
 - b. Normalize your tables and define the relationships between the tables.
2. **Code your classes** - a UI class that starts the program, a manager class, and several object classes (relevant to your topic). The manager class has all the methods.
3. Build a **graphical user interface (GUI)**, an app, to display, manage, and share the data with users. These GUIs call the methods in your manager class and allow you to move from GUI to GUI.
 - a. Not all users have the same view or the same rights to work with the data.
4. Build **action buttons** into your GUI to call your methods. These define instructions for automated tasks, based on user interactions or when changes in data occur.
 - a. Example: A user updates their details and clicks on the update button. This action updates the data in the database using a SQL statement.
 - b. Example: You need to move from one GUI to the next.
5. **Document** your app. A user manual. A technical manual. Print out your code. Test your program. Documents count 50%.
6. You are done. That is it.

NOTE:

Learners need to get their heads around the following concept.

There is one database in two different formats.

The first format of the database is the one you create in Ms. Access with tables. It is a **relational database** (a SQL database). This sits in secondary storage.

The second format of the database is an **object database** – this is a structure called an array of objects – there is probably more than one – an array of cars, an array of clients, an array of products, etc. These are created – our code reads in the data it needs from secondary storage into a structure. These structures sit in memory. They are dynamic and constantly changing in response to user inputs – important changes are written back to the relational database for permanent storage.

Example:

We don't sort the records in the database into alphabetical order. (*)

We do sort the objects in each array as relevant or needed.

(*) they can be sorted but this is not part of the scope of the PAT.