

Assignment	Assigned	Deadline
Pokédex T-Level Software Development SD31	16 Jan 2024	15 Mar 2024

Learning Objectives

- Analyse a problem
- Design a solution
- Build and test a working solution
- Evaluate the solution against the scenario

Support

These resources may help you:

[Python Programming Examples - GeeksforGeeks](#)

[How to Create Flowcharts in Draw.io](#)

[Test Plans - Isaac Computer Science](#)

[Python Tkinter Tutorial - GeeksforGeeks](#)

Instructions

- Use Python and Tkinter
- Internet access is allowed for research purposes
- Work submitted must be your own



ASSIGNMENT

Scenario

An online Pokémon fan club would like to make a basic version of a pokedex, with some functionality of the Pokémon PCs.

- **Individual users should be able to login**
- **Each user should have their own collection as defined in the table below**
- **Each user can only own a full party of 6 Pokémon at a time, with the ability to switch specific Pokémon.**
- **If the user's party is full, taking a new one should replace a Pokémon of the user's choice.**

Username	Password	Poké1	Poké2	Poké3	Poké4	Poké5	Poké6
Custom Name							
Ash							
Misty							
Brock							

When searching for Pokémon:

- **Can search individually, by ID or name, returns first result**
- **Can search by type, or other tags, returns first 10 results**
- **Multiple result searches should display each Pokémon's default front facing sprite.**
- **For individual Pokémon search, display an image of the specified Pokémon's "Official Artwork"**

Pandas should be used to store and manipulate data of users and Pokémon.

The data received and processed must come from the pokedex API: <https://pokeapi.co>

Tkinter must be used to create the UI for your program, using standard UI elements (Colour palette, element positions, scale etc.).

URLs for individual Pokémon must either parse the name or ID of the respective Pokémon, such as either of the following:

<https://pokeapi.co/api/v2/pokemon/ditto>

<https://pokeapi.co/api/v2/pokemon/132>

Additional functionality that the program should try to achieve:

- The option to make a new user should be available
- User's should be able to edit their name
- Each user should have their own password
- A user logged into their own account should be able to delete their own account.

All data should be stored in a csv file titled "UserData.csv".



ASSIGNMENT

Task 1: Analysis

1hr recommended

You should include an introduction summarising the overall problem.

The problem should be broken down into sub-problems. You should write a description of each sub-problem you identify and explain your selection of sub-problems.

State any assumptions you have made.

Save your work in the Report folder as a document called Analysis.

Task 2: Design

3hrs recommended

Algorithms

Design algorithms, using pseudo-code or flowcharts, that show a logical solution to each sub-problem.

You should include inputs, processes, outputs, validation checks and the programming constructs that you will use when you produce your program.

You should show how the algorithms will link together and lead to an overall solution.

EXTENSION: Using CAD software of your choice, design wireframe mock-ups of how your program will look, consider the UI and UX, colours, fonts etc.

Save your algorithms in the Report folder as a document called Design.

Initial test plan

You should complete the relevant sections of the test plan template provided to produce an initial test plan that will demonstrate your strategy for testing your solution.

Save your initial test plan in the Report folder as a document called TestPlan.

**Save a copy of TestPlan in the Report folder as a document called Debugging.
This will be used in Tasks 3 and 4.**

Task 3: Implementation

8hrs recommended

You should translate your design into a program. Ensure that your program is clear and easy to understand.

Add the results of any tests carried out during the implementation stage to the Debugging document.

Save the updated Debugging document in the Report folder.

Create a subfolder called Implementation in the Report folder.



ASSIGNMENT

Save your source code and all the files required to execute the program, including any external files, in the subfolder.

Task 4: Testing, Refining, and Evaluation

2hrs recommended

You should complete the Debugging document by adding any further tests carried out at this stage, including the results of retesting following the correction of any errors.

Save the completed Debugging document.

Evaluate your solution by explaining how well your program meets each of the requirements that you identified in your analysis and describing any refinements that you made to your program during design and implementation.

Save your evaluation in the Report folder as a document called Evaluation.

EXTENSION: Expansion and Documentation

If you complete the project before the deadline, create a copy and explore how the application can be expanded to make further use of the Pandas and Matplotlib libraries for data analysis and plotting.

Create a subfolder called Extension in the Report folder.

Save your copied source code and all the files required to execute the program, including any external files, in the subfolder.

Document the development process to explain any changes made from Task 3, discuss why and how the changes were made and include annotated screenshots where applicable.

Save your document in the Extension folder as a document called Development Documentation.

Mark Scheme

Analysis

Excellent	Good	Requires Improvement	Unacceptable
The project demonstrates a deep understanding of the requirements, including feasibility analysis and consideration of user needs.	The project adequately analyses the requirements with some areas that could be further refined or clarified.	The project's analysis is limited, lacking clarity, completeness, or consideration of user needs.	The project's analysis is missing or inadequate, leading to a lack of direction.

Design

Excellent	Good	Requires Improvement	Unacceptable
The project includes well-documented algorithms and a clear design plan using both pseudocode and flowcharts, showcasing a logical and comprehensive solution.	The project presents effective algorithms for most sub-problems, demonstrating a logical solution with clear documentation using both pseudocode and flowcharts.	The project's algorithm design is lacking in detail or coherence, with some sub-problems not adequately addressed or unclear connections between algorithms.	The project lacks clear algorithm design, making it challenging to understand the logical solution for sub-problems or how these solutions combine into an overall solution.

Implementation

Excellent	Good	Requires Improvement	Unacceptable
<p>The project meets all the specified requirements of the brief and functions flawlessly, providing a comprehensive solution.</p> <p>The project has a well-designed, intuitive, and visually appealing user interface.</p> <p>It effectively handles all possible errors and exceptions, providing informative error messages and ensuring graceful degradation.</p>	<p>The project meets most of the requirements and functions effectively, with minor issues or missing features.</p> <p>The project has a decent user interface that is reasonably intuitive and visually presentable, with some room for improvement.</p> <p>It adequately handles most expected errors and exceptions, providing useful error messages.</p>	<p>The project partially meets the requirements of the brief and has some functional issues or missing features.</p> <p>The project's user interface lacks design elements, usability, or aesthetics, and requires significant improvements.</p> <p>It also lacks proper error handling in some areas, leading to unexpected behaviour or cryptic error messages.</p>	<p>The project does not meet the specified requirements of the brief and lacks basic functionality.</p> <p>The project has a poorly designed and unintuitive user interface that hampers user experience and usability.</p> <p>It has inadequate or no error handling, resulting in frequent crashes, undefined behaviour, or unclear error messages.</p>



ASSIGNMENT

Testing, Refining, and Evaluation			
Excellent	Good	Requires Improvement	Unacceptable
<p>The project includes comprehensive testing plans and effectively evaluates the software against the specified requirements.</p> <p>It demonstrates thorough testing, including unit, integration, and user acceptance testing, with clear documentation of test cases and results.</p>	<p>The project incorporates testing and evaluation processes but may lack comprehensive coverage or clear documentation of testing efforts.</p> <p>It demonstrates some level of testing, but there is room for improvement.</p>	<p>The project lacks a structured testing and evaluation approach, resulting in limited or inconsistent testing efforts.</p> <p>Test cases and results may be incomplete or unclear.</p>	<p>The project lacks any meaningful testing and evaluation, leaving the software's functionality unverified and unvalidated.</p>

Overall Grade:	
----------------	--