

Pokemon Team Picker with Plain Text Storage:

SOFTWARE ARCHITECTURE DOCUMENTATION –

VERSION 1.0

Author:

Stephen Mills

# Table of Contents

[Table of Contents 1](#_Toc23096832)

[Table of Figures 2](#_Toc23096833)

[1. Introduction 3](#_Toc23096834)

[2. High-Level Architecture Figure 1: Authentication of the user for login 4](#_Toc23096835)

[Figure 4: changing one 5](#_Toc23096836)

[Use Case Descriptions 5](#_Toc23096837)

[2.1.1 FR-001: Upload TXT Document(s) To Server 5](#_Toc23096838)

[2.1.2 Figure 2: changing full team 6](#_Toc23096839)

[2.1.3 Use Case Summary (Activity Diagram) 8](#_Toc23096840)

[2.2 Detailed Activity Diagram 10](#_Toc23096841)

[Low-Level Architecture 12](#_Toc23096842)

# Table of Figures

Figure 1: Application Use Case Diagram. ................................................................. 5

Figure 2: Activity Diagram (Use Case Description Summary).................................... 9

Figure 3: Detailed Activity Diagram. ..................................................................... 11

Figure 4: FR-001 Sequence Diagram (Client Side). ................................................. 14

Figure 5: FR-001 Sequence Diagram (Server Side). ................................................ 15

Figure 6: FR-002 Sequence Diagram. .................................................................... 16

# Introduction

The application is a simple read and write application that writes to a series of files stores in the applications file system. The context of the application is for a user to create a team of six pokemon and store them as plain text. The user can log in with an existing account with create accounts to be added. The user will have a pre-existing team to read from and can chose to create a whole new team or to simply replace one of the pokemon. The user will also have the option to exit without saving the newly created team.

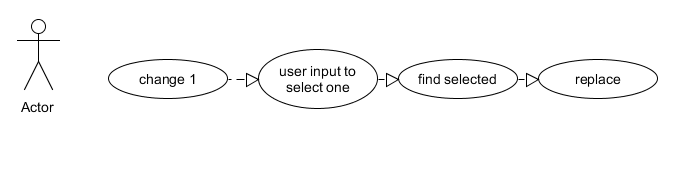
The program has a small amount of inheritance used for passing along attributes with methods to be added as the program progresses.

# A close up of a map Description automatically generatedHigh-Level Architecture Figure 1: Authentication of the user for login

A close up of text on a white background

Description automatically generatedFigure 2: Changing a full team

Figure 3: choices



## Figure 4: changing one

## Use Case Descriptions

### FR-001: Upload TXT Document(s) To Server

|  |  |
| --- | --- |
| **Use Case** | Upload TXT Document(s) To Server. |
| **Objective** | To authenticate the user |
| **Pre-Condition** | 1. User has an existing account |
| **Main Flow** | 1. The user enters a username that is case sensitive 2. The user enters a password that is case sensitive 3. The password and username are checked against existing usernames and passwords. 4. If the details are correct the user gets access to the application |
| **Alternative**  **Flow** | 1A. user fails the username  2A. the application repeats till correct username and password is supplied. |
|  | 1B. the user fails the password  2B. the application repeats till correct username and password is supplied. |
| **Post Condition** | The user has been logged in. |

**Table 1: Authentication of the user for login Use Case Description.**

### Figure 2: changing full team

|  |  |
| --- | --- |
| **Use Case** | Changing the full pokemon team |
| **Objective** | To replace the existing team with a new team |
| **Pre-Condition** | 1. The user as an existing account 2. User has logged in successfully 3. The user chooses to change their whole team |
| **Main Flow** | 1. A table is displayed with all available pokemon. 2. The user is asked to enter each pokemon they would like to add to the team. 3. The new team is created given that 6 have been chosen. |
| **Alternative**  **Flow** | 1A. The user enters less than 6 pokemon. 2A. the application repeats till 6 exactly have been selected.  1B. the user enters more than 6 pokemon.  1B. the application repeats till exactly 6 have been selected. |
| **Post Condition** | The users current team variable has been changed to a new team. |

**Table 2: Changing of a full team Use Case Description.**

**Figure 3: Chose menu**

|  |  |
| --- | --- |
| **Use Case** | User chose |
| **Objective** | To choose between the existing 5 choices of replace whole team, replace 1 member, print current team, save and exit or exit without saving. |
| **Pre-Condition** | 1. The user as an existing account 2. User has logged in successfully |
| **Main Flow** | 1. 5 options are placed for the user to choose from. 2. Option one replaces whole team. 3. Option two replaces one member. 4. Option three saves and quits. 5. Option four prints current team 6. Option five exits without saving 7. Askes for user input from 1 – 4 for options and EXIT to be entered to exit without saving. |
| **Alternative**  **Flow** | 1A. the user enters an incorrect input and the application repeats till a correct one is input. |
| **Post Condition** | To choose between the 5 different menu options. |

Figure 3: Choices

Figure 4: changing a single element in the team

|  |  |
| --- | --- |
| **Use Case** | Changing a single element in the overall team. |
| **Objective** | To replace the existing team member with a new team member |
| **Pre-Condition** | 1. The user as an existing account 2. User has logged in successfully 3. The user chooses to change a single team member |
| **Main Flow** | 1. A table is displayed the current team 2. the user is asked to choose a member from the team to replace. 3. A table of all pokemon is displayed 4. The user is asked to enter each pokemon they would like to add to the team. 5. The new team has one pokemon replaced |
| **Alternative**  **Flow** | 1A. The user enters a number that dose not exist 2A. the application will repeat till a number that is available is selected |
| **Post Condition** | The users current team variable has been changed with one replaced member |

### Use Case Summary (Activity Diagram)

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

## Detailed Activity Diagram

A screenshot of a cell phone

Description automatically generatedDetailed diagram of the whole application flow

The following diagram shows the full flow of the application from start to finish.

The application begins with the request of a login. Then moves on to display of current teams and options .three of the options loop back on the display choices while the remaining two exit the program.

# Low-Level Architecture