AH SDD Project: Game Assist Tool

Contents

[Analysis 1](#_Toc114821436)

[Purpose 1](#_Toc114821437)

[Pre-Development User Survey 2](#_Toc114821438)

[Constraints 2](#_Toc114821439)

[UML Use Case Diagram 3](#_Toc114821440)

[Requirements Specification 3](#_Toc114821441)

[End-User Requirements 3](#_Toc114821442)

[Functional Requirements 3](#_Toc114821443)

[Personas 3](#_Toc114821444)

[User Stories 4](#_Toc114821445)

[User Scenarios 4](#_Toc114821446)

[Project Plan – Initial estimated version 4](#_Toc114821447)

[Project Plan – Final version 6](#_Toc114821448)

[Design 8](#_Toc114821449)

# Analysis

## Purpose

* Currently there is no maintained game assist tool which records match data on the game: Magic: The Gathering Online (MTGO). The game is played by two players, using decks of cards, in a best of three format (called a match). This leaves users to record this match data manually on an excel document, which is very time consuming, or not record the data at all. This is a problem as the game rewards the player for winning matches, so the players are incentivised to increase their win rate (% of winning a game). This leads to a highly competitive environment, where players spend lots of money on the game to maximise their win rate. So, if players can get huge volume of match data, which accurately displays their overall, and, trending win rate and useful additional statistics, they can improve their gameplay, which would hopefully increase their win rate. Which, as I’ve already outlined, they are incentivised to do.
* The end users are people who play Magic: The Gathering Online (MTGO), they are technically adept (as they play an online video game). This means they should be capable of installing and using a game assist tool. Their age range is between 16-40. The end users also spend lots of money on the game, as they typically want to maximise their win rate. Most end-users live in Europe and North America.
* The app is built on Electron so that an interface can be easily created using HTML, CSS and JavaScript. The app will display all currently collated data, which can be manipulated by the end-user, and graphs of the data. The collated data and graphs can be downloaded by the end-user. On the app, the user can press a button which runs the main processes of the data collection algorithm. This main process navigates to MTGO’s file system, where it gets the users’ match logs (stored in .dat files). The app then reads all information within the logs, which it categorises and stores in a local sqlite3 database. It may require the end-user’s input to clarify data which the match logs leave out, this is conditional on the basis that one of the players don’t concede on the final game. The app also uses the Selenium web driver to get additional information (deck names), to store in the database.
* The project meets these requirements:
  + Is an SDD project integrated with a local database.
  + The project will use a sort algorithm, object-oriented and procedural programming, an array of records (specifically dictionaries and lists, as python doesn’t have arrays), create a local database and update, select, delete records (based on what the end-user selects) from that database.

## Pre-Development User Survey

* This survey was carried out to gauge potential end-user interest in the app, whether design decisions should be followed and assumptions about potential end-users were accurate.
* The user survey confirmed that there was an interest in an MTGO game assist tool among potential end-users, and so proves the app is viable.
* See Appendix 1 for more information.

## Constraints

* Technical Constraints
  + The app will run on Windows 7 or later.
  + The app itself does not take up lots of space, however, the databases have no limit on how large it can be (this is down to the user’s discretion).
  + Python’s performance is relatively slow, however, it is very sufficient for this app.
  + All the necessary resources currently exist for the project to be completed.
* Business Constraints
  + The scheduling and timescales must be met, otherwise the app will be incomplete.
  + There is no budget for the app.
  + There is a single person working on the app.
  + There are no costs associated with the project as it uses free software.
  + The app can be further developed which could allow it to follow the freemium model, so the end-user could pay for more features.
  + The app will take 80 hours to complete.
  + There are no legal issues with development of the app: all web scraping is within current precedent (site doesn’t have a robots.txt, so web scraping isn’t disallowed, and the web driver doesn’t login/create an account) and no personal data is stored (so GDPR isn’t in violation).

## UML Use Case Diagram

### Actors

### Use case

### Relationships

## Requirements Specification

### End-User Requirements

The end user will expect the interface to display:

* Graphs
  + Pie Charts, Scatter Graphs, etc.
* Comparison
  + Possible comparisons include: Past win rate Vs Current win rate,

### Functional Requirements

* Inputs
* Processes
  + Outputs

## Personas

### Sean Greaves

* A 23-year-old man who lives in America, on their own in a house. They’ve being playing MTGO for 4 years and are very committed to the game. They have a middling income job at Specsavers and are highly competitive as they are a high-performance swimming athlete in their spare time.

### Winnie Manning

* An 18-year-old woman who lives in the UK, in a flat (with 2 flatmates). They are in their first year of university. They’ve been playing MTGO for 2 years and are fairly committed to the game. They have a part-time job as a barista at a local indie coffee shop. They read the newspaper and always make sure to complete the sudoku, which they enjoy finishing. They also have game night with their flatmates every Saturday night and are quite competitive when it’s Monopoly that night.

## User Stories

* As a (user type) I want to

## User Scenarios

## Project Plan – Initial estimated version

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | | Start Date | End date | Resources Required |
| Analysis (27/08/22 – 19/10/22) | | | | |
| 1. | Description of problem | 27/08/22 | 29/08/22 |  |
| 2. | Pre-Development user survey | 31/08/22 | 05/09/22 | Using Survey Monkey |
| 3. | Constraints | 20/09/22 | 21/09/22 |  |
| 4. | UML Use Case diagram | 22/09/22 | 27/09/22 | Using Lucidchart |
| 5. | Create persona, user stories and senarios | 01/10/22 | 05/9/22 |  |
| 6. | Requirements specification: end-user req. | 07/10/22 | 08/10/22 |  |
| 7. | Requirements specification: functional req. | 09/10/22 | 17/10/22 |  |
| 8. | Project plan | 17/10/22 | 19/10/22 | Using Gantt Project |
| Design (20/10/22 – 21/10/22) | | | | |
| 9. | Pseudocode design | 20/10/22 | 22/10/22 |  |
| 10. | UML class diagram | 24/10/22 | 27/10/22 |  |
| 11. | Project Design | 27/10/22 | 30/10/22 |  |
| 12. | User-Interface Design | 27/10/22 | 31/10/22 |  |
| Implementation (01/11/22 – 10/12/22) | | | | |
| 13. | Implementation | 01/11/22 | 10/12/22 |  |
| 14. | Research and development of new skills | 01/11/22 | 05/12/22 |  |
| 15. | Log of ongoing testing | 01/11/22 | 10/12/22 |  |
| End Testing (12/12/22 – 23/12/22) | | | | |
| 16. | Final Test Plan | 12/12/22 | 16/12/22 | Using pytest |
| 17. | Requirements Testing | 17/12/22 | 18/12/22 |  |
| 18. | Post-Development user survey |  |  | Using Survey Monkey |
| 18. | Testing with personas and test cases | 20/12/22 | 23/12/22 |  |
| Evaluation (05/01/23 – 07/01/23) | | | | |
| 19. | Evaluation Report | 05/01/23 | 07/01/23 |  |

## Project Plan – Final version

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | | Start Date | End date | Resources Required |
| Analysis | | | | |
| 1. | Description of problem | 27/08/22 | 29/08/22 |  |
| 2. | Pre-Development user survey | 31/08/22 | 05/09/22 | Survey monkey – create account |
| 3. | Constraints | 20/09/22 | 21/09/22 |  |
| 4. | UML Use Case diagram |  |  | Using Lucidchart |
| 5. | Create persona, user stories and senarios |  |  |  |
| 6. | Requirements specification: end-user req. |  |  |  |
| 7. | Requirements specification: functional req. |  |  |  |
| 8. | Project plan |  |  | Using Gantt Project |
| Design | | | | |
| 9. | Pseudocode design |  |  |  |
| 10. | UML class diagram |  |  |  |
| 11. | Project Design |  |  |  |
| 12. | User-Interface Design |  |  |  |
| Implementation | | | | |
| 13. | Implementation |  |  |  |
| 14. | Research and development of new skills |  |  |  |
| 15. | Log of ongoing testing |  |  |  |
| End Testing | | | | |
| 16. | Final Test Plan |  |  | Using pytest |
| 17. | Requirements Testing |  |  |  |
| 18. | Post-Development user survey |  |  |  |
| 19. | Testing with personas and test cases | 20/12/22 |  |  |
| Evaluation | | | | |
| 20. | Evaluation Report | 05/01/23 | 07/01/23 |  |

# Design

## Pseudocode Design

## UML Class Design

## Project Design

## User-Interface Design

Implementation

End Testing

Evaluation

Appendix 1 – User Survey Results

Q1

Chart, bar chart

Description automatically generated

Results – Most potential end-users are under 18 to 24, so

Q3

Graphical user interface, text, application, email, website

Description automatically generated

Numerical Results – 5, 50+, 34, 1, 23, 1, 4, 6, 14, 2, 0, 0, 10, 10, 12, 10, 10, 31, 17, 18, 24, 16

Mean – 14 (2 s.f.)

Results – Most potential end-users play videogames a fair amount.

Q3

Chart

Description automatically generated

Results – Most potential end-users would want the app to display feedback on their game.

Q4Chart

Description automatically generated

Results – Most potential end-users would want the app to try and help them improve.

Q5

Chart, bar chart

Description automatically generated

Results – Most potential end-users want a combination of ways to display information stored by the app.

Q6

Chart, bar chart

Description automatically generated

Results – Cost is a barrier of entry for most potential end-users, this confirms that the app should be free, to enable as many users to use it as possible.

Q7

Chart, bar chart

Description automatically generated

Results – Most potential end-users would be interest in installing the finished app.

Appendix 2

Appendix 3