MMA BOT

BY : SHANE MILLER

(SRS) (CEN 3073)

(GITHUB.COM/SHANEMILLER1?TAB=REPOSITORIES)

CONTENTS

1. Introduction

1.1 Purpose

The program that I'm working on would allow users to place bets on MMA matches. The program being developed will benefit the company by having several unique features such as passive database collection with API, sleek Ui and a contact page. These alternatives will be an improvement to my sponsor's website. The software will collect results autonomously, then be shown to users of the website. This is more convenient for managing the website and can increase productivity.

1.2 Scope

Product Name

- MMA BOT:

  Stands for Mixed Martial Arts bot short for robot.

Overview

The system running on JavaFX would work in relation to the sponsor's website. That would entail UFC scores, results and MMA fighters. This bot program is part of a larger system, the bot is the brains of the larger system, and without it nothing else would work.

Goals

- Automatic database collection:

  Having a fast database will ensure user satisfaction, this will mean less buff/loading on the website. Can be tested through website traffic/heavy load test.

- Easy to use, Sleek and modern UI:

  The website is built on Wix, allowing developers to have many options in customization, many studies show that the end user will appreciate.

- Contact Page:

  Allows the software engineer to fixed programs, add requirements if needed, and plan ideas to implement within a timeframe.

- Increase website revenue

  Increasing Revenue will support sponsors and stakeholder's ability to keep the website running. Additionally, it's a way to increase morality and motivation, and makes the Software Engineer credible to future projects. Revenue can be measured on Wix interface.

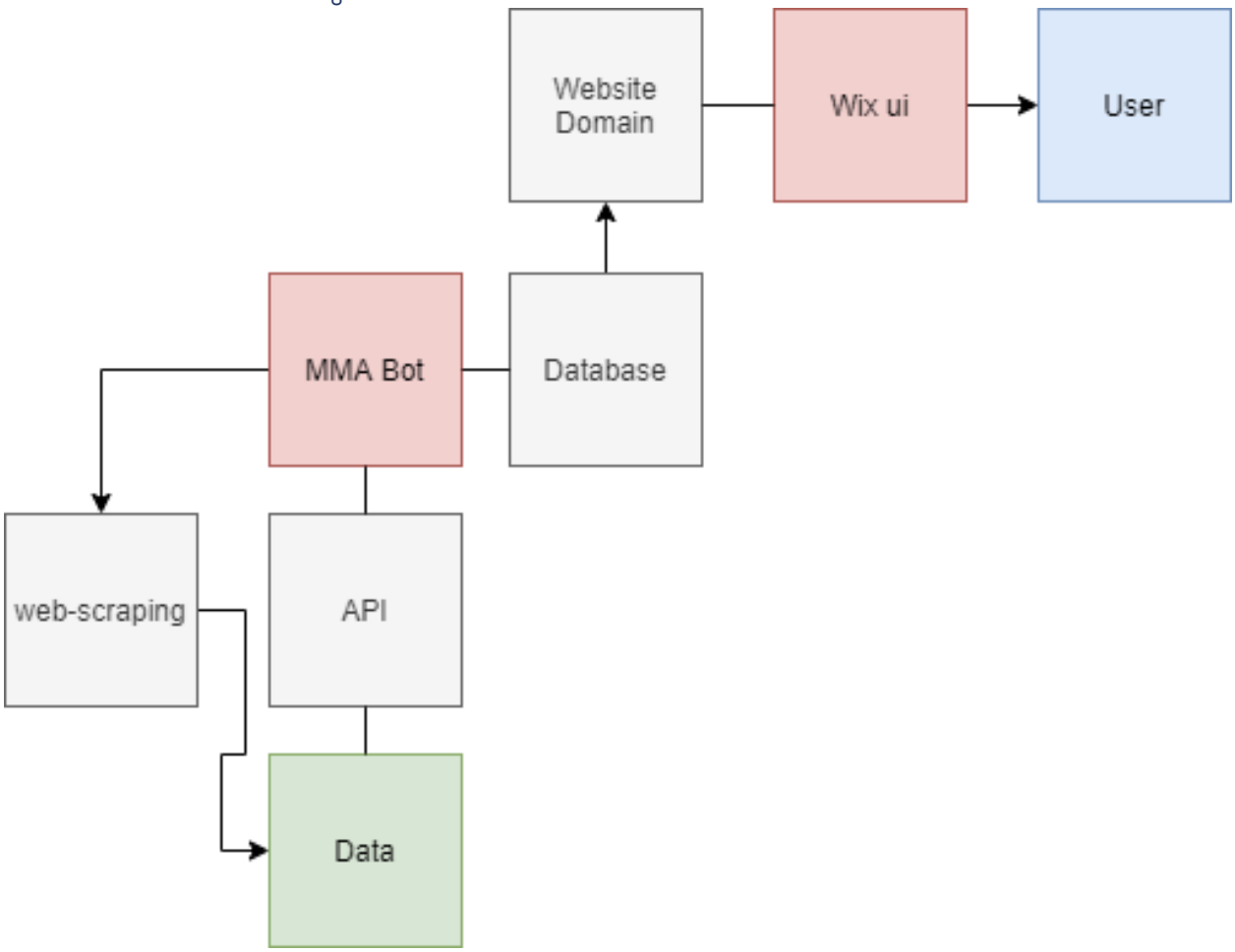
Out of Scope

For later releases, focus towards a database management system that the administrator of the website could use. Also planning on features to implement that can increase website revenue (this would be planned with the use of backlog and roadmap on Jira). In this release there is no application programming interface, so that will be worked on in a later release along with improving the UI. A contact page for users to send feedback and can be used to debug future problems as well as API integration.

1.3 Product overview

1.3.1 Product perspective

Software Context Model: Figure 1.



This bot program is part of a larger system as can be seen in the block diagram. The bot is the brains of the larger system, and without it nothing else would work. As you can see the bot is connected to several interfaces, this entails the website domain, WIX, and the end-user. A context diagram shows the relationships of a system. This generally has a main system in the middle and around it are arrows to represent how information flowed in the system. This is beneficial because it allows developers to get a better understanding of the relationships in the system. Another benefit can be, by making a level 0 data flow diagram you can easily find flaws in the system.

Constraints

Regulatory requirements and policies:

- Regulations for web-scrapping makes it not necessarily legal, it mostly it depends on the website's TOS if you can use their information or not; and the end use of the program.

- APIs are legal.
- Online Sports betting is not legal in Florida, pending approval still.

Hardware limitations:

- The program would run on a personal computer (not cloud based) so steady internet connection is necessary.
- Computer that the program is running on should not be a potato.

Interfaces to other applications

- Compatibility between software
- Java heavy focused program

parallel operation

- Main program and the website should communicate with each other within 10 seconds of latency.

Audit functions

- Make the program reliable" no way for the user to crash the program".

Control functions

- Must have on, off functions to the main program.

Quality requirements

- Program will be tested before release

Safety and security considerations

- Epilepsy protection
- Server based checking

1.3.2 Product functions

For my hypothetical product it would be an AI bot program. Here is a diagram of how it would perform in a real scenario (refer to Figure 1.). The major function that the bot would perform would be gather data from the internet to show on my sponsor's website. That would entail UFC scores, results and MMA fighters. Then it would be most efficient to store that data in a database so there won't be any duplicate data or client-side corruption. I say that because this program would have to not only bring in data, but it would have to be secure enough for a betting system on a website. It would be a good to have a log system in place too to show operations being performed and store time operation was done. Storing time would also be important to show users when they bet on a match.

1.3.3 User characteristics

The general characteristics of the intended groups of users of the software would include users of the website. These users or "customers" would not directly interact with the program but are passively viewing data from the program on the website. Their characteristic is mainly usability and can be at any education level. However, the backend maintenance personnel of the website could actually interact "physically", with the program to run diagnostics. These personnel could be classified as DMA, or database administrators and would have at least an associate degree in computer science to operate the program. The personnel could also have technical expertise in the field and not need necessarily a degree to operate the program if they are higher up in the company.

1.3.4 Limitations

The limitations of the program currently are not using an API, however, the plan is to use API to eliminate existing issues that are making the program run slow and more complicated than it needs to be. Also implementing angular which is a web assistant UI program running off HTML and JavaScript; looking into that as another alternative but very new to it.

1.4 Definitions

- DBMS:
  Database management system.
- DBA:
  Database administrator.
- API:
  Application programming interface.
- MMA:
  Mixed martial arts.

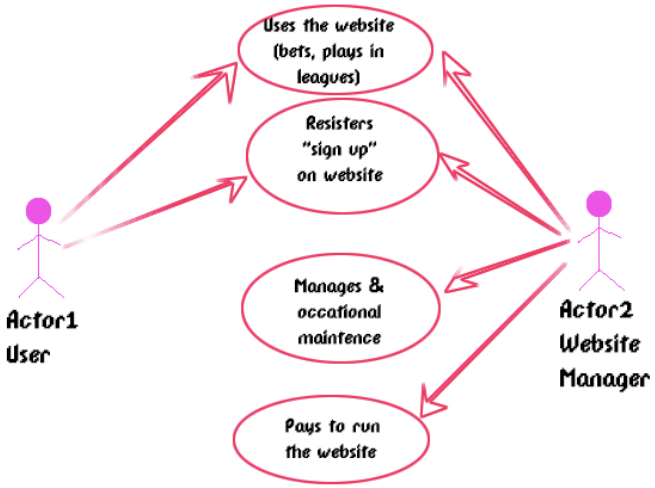
2. References

- Cover art on title page: Mrdestructiod art is an emote on popular streaming website twitch.
- MMA API by NodeJS: (github.com/valish/mma-api)
- Betting online regulations in Florida: (gamblingsites.com/online-gambling-jurisdictions/us/florida)
- Context diagram made on Draw.io
- Sponsor's Website: (mmafanstasy.com)

3. Specific requirements

Use Case: Figure 2

Here is a representation of a use case and how some actions are handled on the website. On the left is a user, they can register as a first-time user to set up their account and "Play on the



website". The relationship describing that actor 1 can do these tasks is shown with an arrow. On the right is actor 2, the manager. They can also use the website if they would like, and can also create an account too. However, they are the only ones who can manage the website and pay the domain website to keep the website operational.

Key	Summary	Description	(Issue) Type	Linked Issues	Priority	Labels
BA-11	As an administrator, I want the ability to see a schema of daily records, so that there is a way to see all operations of the day	Requirement Involved in Sprint 1	Story	n/a	High	Epic
BA-13	"As a user, I want to be able to add friends, so that I can check what my social connections are betting on"	Requirement Involved in Sprint 1	Story	n/a	Low	Epic
BA-2	The system shall show data to the user automatically without the use of manually entering data behind the scenes; information should be shown within 100ms-1000ms of the user entering into the website.	Requirement Involved in Sprint 1	Business Rule	BA1	Medium	Epic
BA-5	The system shall be able to be paused if there are a problem and information then entered manually if need be.	Requirement Involved in Sprint 2	Business Rule	BA1	High	Epic
BA-3	The system shall have an option to allow the owner/maintainer to manually check on information pertaining to the	Requirement Involved in Sprint 2	Story	n/a	Medium	Epic

	database with an administration login.					
BA-10	"As an administrator, the ability to have an option to turn off the program(into manual mode), in the case of an emergency "	Requirement Involved in Sprint 2	Story	n/a	High	Epic
BA-14	"As a user, I want to get notifications of upcoming leagues, so that I can keep track of tasks"	Requirement Involved in Sprint 2	Story	n/a	Medium	Epic
BA-12	"As a user, I want a page so I can check my bet history so that I have a record of previous bets"	Requirement Involved in Sprint 2	Story	n/a	Medium	Epic
BA-6	The website was made with Wix, so determining how to implement a program will be a first for me. Also there might be performance and security constraints we run into in the future.	Requirement Involved in Sprint 2	Story	n/a	Medium	Epic
BA-7	We are trying to focus our efforts on usability, and efficiency, to not overload the servers and to	Requirement Involved in Sprint 2	Story	n/a	Medium	Epic

	allow for a user-friendly UI.					
BA-8	A system created with java to web scrap MMA results, which is then stored in a database to be used and displayed on a website.	Requirement Involved in Sprint 2	Story	n/a	Medium	Epic

4. Verification

Key	Summary	Verification	Approach
BA-15	REQ 1 The system will have an off state, in off state, no data will be transferred to the website.	When program is off new data will not be updated onto website.	Adding slider button on interface of program to have (on-off)
BA-16	REQ 2 The system will have an on state, while on the system will perform task such as using the API and interacting with interfaces.	When program is on new data will be sent to the website.	Adding slider button on interface of program to have (on-off)
BA-17	REQ 3 The system will have a maintenance at least once a week, during this time the system will be in the off state and will be debugged.	DBA will have a recording sheet to fill out every week and higher up will have to verify.	Can be approached using excel tables.
BA-18	REQ 4 The system will update information to the domain website and Wix within 10 seconds.	Transmission rate can be read in the logs.	Implement logs into program.

BA-19	REQ 5 The system will interact with the database run by H2, to store data from the public API.	Check database to see if correct information in being stored	Go to open sourced API from GitHub and implement into Java FX workspace.
BA-20	REQ 6 The system will delete information from the database if that specific information has not been using a 5 years.	Data will be deleted	Create method to clear tables with date > 5 years of creation.
BA-21	REQ 7 The system will allow for expansion to memory, if the database is running low on storage.	DBA or personnel running maintenance on program, can check disk usage,	Adding more storage when storage gets filled.
BA-22	REQ 8 The system can be run on a cloud server if needed.	DBA or personnel running maintenance can transfer program to cloud server.	When security risks or other judgement allows, research cloud servers then pick one.
BA-23	REQ 9 The system will have DDOS protection.	Run a DDOS program and test website.	Install anti-DDOS open source plugin.
BA-24	REQ 10 The system will be secure using Caesar Cipher algorithms.	Test Caesar Cipher encryption.	Find open sourced Caesar Cipher algorithm / change to own unique.

5. Appendices

5.1 Assumptions and dependencies

Having software that can collect results and data autonomously would be more convenient to the person managing the website. I would assume this would allow more time to the developer of the website to improve the website elsewhere. We assume the person running the website has staff eligible to be DMA's. We also assume this product would be completed during a duration of a university spring semester. Assume all funding is taken care of. Assume pc/server will handle DDOS, high server load, the program itself, has window 10, pc has latest drivers and hardware. We also assume all dependencies are compatible and will work with one another. This includes open-sourced programs, JavaFX, Wix, scene builder, cloud servers.

5.2 Acronyms and abbreviations

- DBMS:
  Database management system.
- DBA:
  Database administrator.
- API:
  Application programming interface.
- MMA:
  Mixed martial arts.

- Open Source Software:
  Open-source software is a type of computer software in which source code is released under a license in which the copyright holder grants users the rights to study, change, and distribute the software to anyone and for any purpose.

- DDOS:
  DDoS is short for Distributed Denial of Service. DDoS is a type of DOS attack where multiple compromised systems, which are often infected with a Trojan, are used to target a single system causing a Denial of Service (DoS) attack.