DeckTechCentral

CMSC 4900 – Senior Project I

Fall 2023

Project Requirements

10/17/2023

Instructor Comments and Evaluation

Table of Contents

Abstract

The requirements document is responsible for describing and outlining the Software Engineering requirements of the project at hand, DeckTechCentral, a web-based deck list management tool for “Magic: The Gathering” (MTG). The document will begin by addressing the background, objective, and team details. It will then explore the application domain, use cases, and data sources. Finally, the document will explain the functional, nonfunctional, and documentation requirements of DeckTechCentral.

Introduction

Background

"Magic: The Gathering” is a collectible trading card game of fun-filled, strategic games to play with friends (Wizards, n.d.). The trading cards are unique character-based cards that each have their own story, tips, and statistics. The core aspect of a “Magic: The Gathering” player is making a deck of cards, also known as a deck list. The deck list is what players use to interact with other players and the game itself. Making a deck list can take up a majority of a players' time and can be a source of entertainment in itself. Because of this, there has always been a need for a quick, easy, and helpful way to store deck lists and share what cards are in them. This is the central need of the game that DeckTechCentral will attempt to solve. By providing a way to allow people to easily make deck lists, we are allowing for players’ creativities to flow and competition to work even more while optimizing their creations that they hold so dearly.

Objective

The objective of the project is to create an easy-to-use web application for people to build decks and view other user generated deck lists. To help players optimize their decks they have a multitude of ways to view the cards that are listed in it. When the user edits or creates a new list of cards the updated list is sent to the server and then saved into the database to be looked up and edited at a later date.

Team Details & Dynamics

Our goal is to have a productive environment while ensuring that everyone contributes to the project. Communication will be vital so that the team can work effectively and learn from this project. Further, each team member has a unique skillset and experience, which will need to be considered so that everyone can contribute the most. The team will help each other with any unfamiliar topics, such as new technologies. Finally, we aim to have fun during this project, just like how one would during a game of “Magic: The Gathering”. Each team member is important, and any feedback, good or bad, will always be considered.

To facilitate communication, we plan on using several methods. The simplest form will be meeting in person; we will have set meeting times as well as flexible options when needed. We will also use computer-assisted tools (CASE) including Discord as a messaging platform and GitHub for version control. These will all ensure open communication where nobody feels left out.

While the entire team will be involved in some way at every step, each team member has agreed to lead a phase of the project based on their individual skills and strengths:

Team Member	Major	Project Phase
Christian Messmer	Computer Science	Implementation
Paul Shriner	Computer Science	Analysis
Adir Turgeman	Computer Science	Presentation
Luke Vukovich	Computer Science	Design

Application Domain

Project Context

The application domain of DeckTechCentral is a subset of all internet users. The subset and domain can be described as users who are active or interested in MTG and would like to implement or view custom deck lists. The domain can act as a creative platform for MTG users.

Initial Business Model

Operating Environment

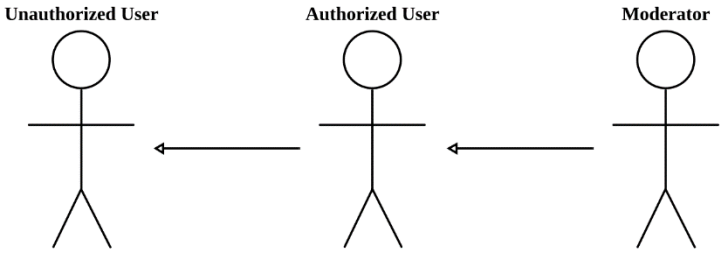
The Front-End user interface will take the form of a clean and simple website. The website will be operational on standard internet browsers and mobile devices. The Back-End - consisting of a remotely hosted database - will communicate with the Front-End to send and receive data rapidly and efficiently.

Description of Data Sources

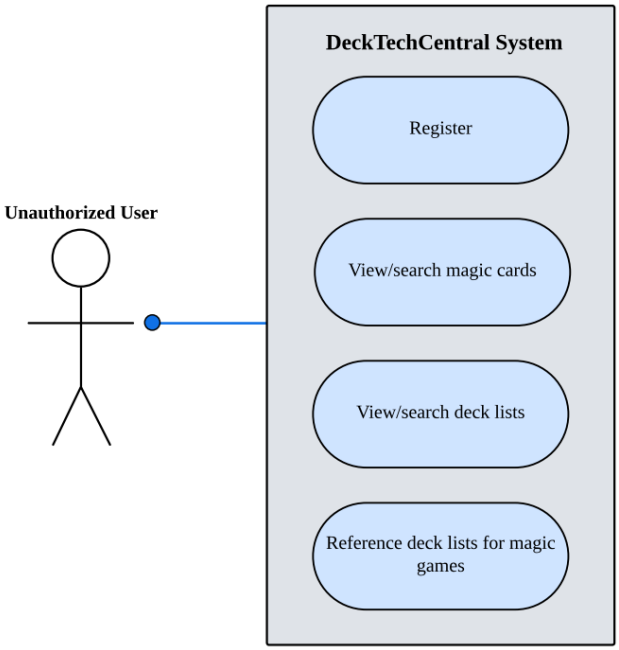
The backbone of DeckTechCentral is the main database. The database will be very user dependent. Users submit data through the website’s Front-End, such as by registering for an

account or creating a deck list. All submitted data is processed and stored in the main database. When the data needs accessed, such as a user logging into their account, a request is sent to the database and the data is sent back to the website. To source all MTG cards, a reliable API will be needed to reference cards and all static deck list assets. To help reduce the amount of requests made to the outside data source, it would be optimal to store a weekly backup of data we would need to regularly access. This design would allow for more dependence on the hosted database and less on an outside source.

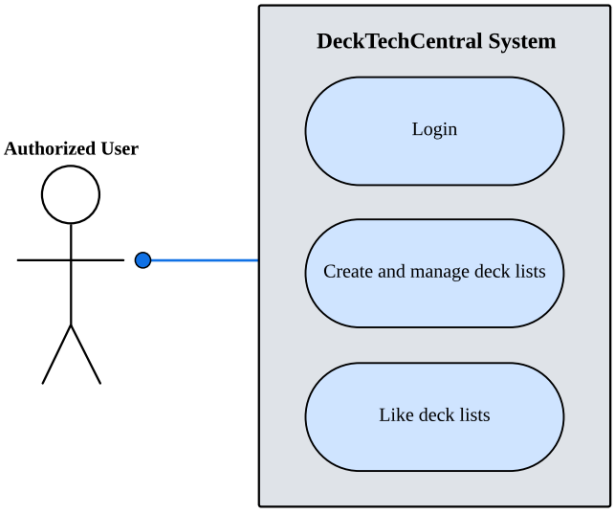
Use Case Diagrams



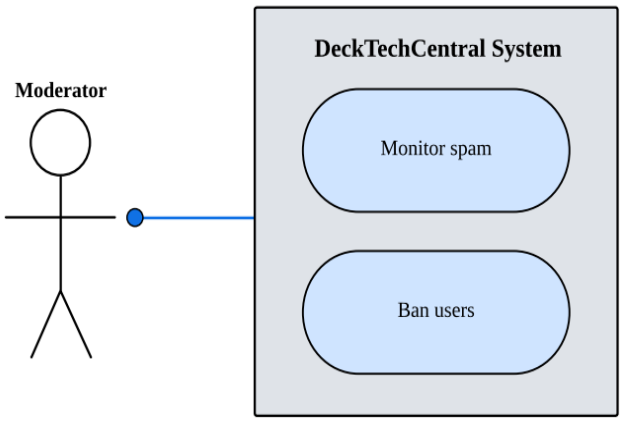
User privileges can be represented by the inheritance diagram above.



Unauthorized Users do not inherit any user group and is considered to have *read access*. This group, while being prohibited from deck list creation, does not need an account to view cards and deck lists. *Read access* also allows this group to reference deck lists for in-game activities. *Unauthorzed Users* can register using the website to be promoted to an *Authorized User*.



Assuming proper registration, *Authorized Users* would have the ability to log into their personal account. *Authorized Users*, along with inheriting all privileges from *Unauthorized Users*, are also granted the ability to manage and create deck lists. Users in this group will also have access to the website's deck list review system.



Moderators assume the privileges of all user groups and act as a protection layer for the website. They are responsible for monitoring any spam and banning *Authorized Users* if needed.

Initial Requirements

Our project is going to be a web application that will follow the MVC model. The model will be a database containing the deck lists and users, the view will be a React Typescript web app, and the controller will be a .NET restful API that will handle authentication, data creation, and data gathering.

Functional Requirements

1. The user can send a login or register request to begin consuming the website as an authorized user.
2. Unauthorized users can view public deck lists made by users of the website.
3. Authorized users can create and manage their deck lists.
4. Authorized users can view deck lists that they have access to.
5. Users can search for deck lists based on certain criteria.
6. Users can rate cards in their deck and receive statistics about their deck based on that data.
7. Moderators can ban users if spam or other malicious activity is suspected.

Nonfunctional Requirements

1. Optimize database and other data source requests to meet the required response time.
2. Proper documentation throughout each phase to ensure proper design, implementation records, and explanations.

3. Hold user information to highest standards and only store required user information. Secure information at all costs.
4. Back-End server will be running in the cloud to have maximum availability, performance, and scalability.
5. Minimize number of outside API calls by storing received data for periods of time.
6. A user-friendly Front-End design that is easy to understand and navigate for both new and current users.

Documentation

Proposal Document

The document that provides a brief introduction and description of the project.

Requirements Document

This document, which goes more in detail about the goals and development requirements of the project.

Specification Document

The document which outlines in detail what will be needed to realize the requirements of the project.

Design Document

The document which fully describes the project and specifics needed for implementation.

Additional Documentation

Documentation will be an essential part of the project to ensure that all team members understand the details of the project and that anyone else who works on the project in the future could pick up on the team's work. In addition to the four documents described, other documents such as a code commenting style reference sheet may be created as needed.

Testing and Revisions

This document was created by all team members using a shared Microsoft Word document. Members would draft portions outside of the document, then rewrite it inside the document. Each portion was proofread several times to ensure that the work is of high quality. A similar method will be used for future documentation in the project.

For code writing, Git and GitHub will be used for version control and documentation, ensuring high quality work for the code itself. Git has several functions including branches and pull requests, allowing for team members to review code and adjust if necessary. GitHub is an online platform for Git where the code repository and changes will be stored.

Appendix

Technical Glossary

.NET

.NET is a free, cross-platform, open-source developer platform for building many different types of applications (Microsoft, n.d.).

API

Abbreviation for “Application Programming Interface”; set of rules for communication with a service (IBM, n.d.).

Authorized User

Has the same capabilities as an Unauthorized User but can also create and review deck lists.

Back-End

Handles data and requests from the user. (freeCodeCamp, 2022).

CASE

Abbreviation for “Computer Aided Software Engineering”; the implementation of computer facilitated tools in software development (GeekForGeeks, 2023).

Creatures

A type of Magic card; creatures are the main way to attack and defend in a game (Wizards, n.d.).

Database

Organizes and stores an application’s resources in a way that is easily accessible and maintainable (freeCodeCamp, 2022).

Deck list

Set of Magic cards the player utilizes throughout the game. In a real-life game, this would be a physical deck. In this project, it will be a list. Decks have a minimum of 40 cards and have different types of cards including lands and creatures (DiceBreaker, 2019).

Discord

A communication platform that allows setting up separate communities (servers) for members to communicate in (Discord, n.d.).

Front-End

What the user sees and interacts with (freeCodeCamp, 2022).

Git

A version control system (GitHub, n.d.).

GitHub

A platform for hosting and managing Git repositories (GitHub, n.d.).

Lands

A type of Magic card; lands produce mana, which the player can use to play other cards (Wizards, n.d.).

Magic

See *Magic: The Gathering.*

Magic: The Gathering

Trading card game created by Richard Garfield in 1993 and published by Wizards of the Coast (MTG Fandom, n.d.).

Mana

On a Magic card, mana appears on the top right of the card. Mana is used to play other cards and is categorized into colors. For example, playing a card could require two mana of any color as well as one mana of a specific color, totaling three mana (Wizards, n.d.).

Moderator

Has the same capabilities as an Authorized User but can also take action against malicious activity. This includes, but is not limited to, spam or toxicity.

MTG

See *Magic: The Gathering.*

MVC Model

Architectural/design pattern that separates an application into three components: Model (data logic), View (UI logic), and Controller (connection between View and Model) (GeekForGeeks, 2023).

Repository

In the context of Git and GitHub, it is the collection of files and folders associated with a project (GitHub, n.d.).

Restful API

API using the “REST” architecture, which imposes conditions on how an API should work. These conditions include the following: uniform interface, statelessness, layered system, cacheability, and code on demand (Amazon, n.d.).

Unauthorized User

A user who has not registered with the service. These users can view publicly available deck lists but can’t create a deck list or review someone else’s deck list.

UML (Unified Modeling Language)

A general-purpose modeling language. The main aim of UML is to define a standard way to visualize the way a system has been designed (GeekForGeeks, 2023).

Version control

Allows tracking history of changes to a repository and recovery of an older version (GitHub, n.d.).

Website

A group of World Wide Web pages usually containing hyperlinks to each other and made available online by an individual, company, educational institution, government, or organization (Merriam-Webster, n.d.)

Team Details

The current revision of the requirements document was prepared by the team. The team takes on the responsibility of revising the document and preparing all necessary material for the project at hand. The team utilized Discord and other public meeting areas to successfully meet online and in person. Contributions to the document at hand are outlined below.

Paul Shriner was responsible for the document design and layout, including elements like the table of contents. This helped produce a successful writing environment. Along with document design, Paul completed documentation for parts of the Introduction, Documentation, Glossary, and References. Paul reviewed the document several times and made revisions as needed.

Luke Vukovich designed all UML diagrams to aid in the representation of the use cases of the project. He designed and explained each use case to aid in the understanding of the application domain. After reviewing his work, each team member approved of the design he established. Luke was responsible for documenting the Application Domain and Initial Business Model.

Christian Messmer designed an early draft of the nonfunctional and functional requirements. He took on this role due to his experience with “Magic: The Gathering” and other known MTG deck list management websites. After the team reviewed and approved, Christian implemented and documented all necessary requirements for the project. He also aided in the completion of the Introduction and use cases.

Adir Turgeman acted as the approval layer for all aspects of the requirements document. Before any documentation was finalized, Adir was responsible for a final review and approval.

He added many terms to the Glossary and ensured that all required references were included and formatted.

Workflow Authentication

I, Christian Messmer, attest that I executed the functions listed within the team details section of the document. Also, I agree with all the information stated within the requirements document.

Christian Messmer		October 17, 2023
Printed Name	Signature	Date

I, Paul Shriner, attest that I executed the functions listed within the team details section of the document. Also, I agree with all the information stated within the requirements document.

Paul Shriner		October 17, 2023
Printed Name	Signature	Date

I, Adir Turgeman, attest that I executed the functions listed within the team details section of the document. Also, I agree with all the information stated within the requirements document.

Adir Turgeman		October 17, 2023
Printed Name	Signature	Date

I, Luke Vukovich, attest that I executed the functions listed within the team details section of the document. Also, I agree with all the information stated within the requirements document.

Luke Vukovich		October 17, 2023
Printed Name	Signature	Date

Report from the Writing Center

Review writing, grammar, and layout of a Computer Science Senior project paper. Upon completion, I took note of a few sentence construction errors and assisted on how to modify them. Additionally, I also took note of a few punctuation errors, mainly placement and use of commas. Highlighted the changes that needed to be made. Overall, the paper is well written and organized.

References

Amazon. (n.d.). *What is RESTful API?*. AWS. https://aws.amazon.com/what-is/restful-api/

Coles, J. (2019, December 18). *How to build a Magic: The Gathering deck for beginners*. Dicebreaker. https://www.dicebreaker.com/series/magic-the-gathering/how-to/build-a-deck-for-beginners

Fandom, Inc. (n.d.). *Magic: The Gathering*. MTG Wiki. https://mtg.fandom.com/wiki/Magic:_The_Gathering/

GeeksforGeeks. (2023, January 3). *Computer Aided Software Engineering (CASE)*. GeeksforGeeks. https://www.geeksforgeeks.org/computer-aided-software-engineering-case/

GeeksforGeeks. (2023, August 29). *MVC Framework Introduction*. GeeksforGeeks. https://www.geeksforgeeks.org/mvc-framework-introduction/

GeeksforGeeks. (2023, September 18). *Unified Modeling Language (UML) | An Introduction*. GeeksforGeeks. https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/

Get started with GitHub documentation. GitHub Docs. (n.d.). https://docs.github.com/en/get-started/

How to Play: Magic: The Gathering. MAGIC. (n.d.). https://magic.wizards.com/en/how-to-play

Lemonaki, D. (2022, March 18). *Frontend VS Backend – What's the Difference?* freeCodeCamp.org. https://www.freecodecamp.org/news/frontend-vs-backend-whats-the-difference/

Merriam-Webster. (n.d.). *Website Definition & Meaning*. Merriam-Webster. https://www.merriam-webster.com/dictionary/website

What is .NET? an open-source developer platform. Microsoft. (n.d.). https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet

What is an Application Programming Interface (API)?. IBM. (n.d.). https://www.ibm.com/topics/api

Wizards. (n.d.). *What is MTG: Magic: The Gathering*. MTG. https://magic.wizards.com/en/intro

Your Place to Talk and Hang Out. Discord. (n.d.). https://discord.com/