



ИНТЕГРАЦИЯ НА УНИВЕРСИТЕТСКИ ИНФОРМАЦИОННИ СИСТЕМИ. АРХИТЕКТУРЕН ПРОЕКТ.

Автори: Божидар Попиванов, Мартин Петров,
Алекс Трайков, Кристиан Митев, Жоро Спасов



21 ОКТОМВРИ 2023

Съдържание

1.	Въведение	3
2.	Предназначение	3
2.1	Обхват	3
2.2	Актьори	3
3.	Архитектурен обзор.....	3
3.1	Use Case: Аутентикация на потребителите и даване на роля	3
3.2	Use Case: Мудъл система за обработка на файлове	5
3.3	Логически изглед	7
3.3.1	Дискорд бот.....	7
3.3.2	Мудъл система	8
3.4	Процесен изглед	10
3.4.1	Дискорд бот.....	10
3.4.2	Мудъл система	12
3.5	Изглед на данните.....	14
3.5.1	Дискорд бот.....	14
3.5.2	Мудъл	15
3.6	Изглед на внедряването	16
3.6.1	Дискорд бот.....	16
3.6.2	Мудъл	17
3.7	Изглед на имплементацията	18
3.7.1	Дискорд бот.....	18
3.7.2	Мудъл	19
4.	Нефункционални изисквания	20
4.1	Дискорд.....	20
4.1.1	Достъпност (Availability)	20
4.1.2	Разширяемост (Scalability):	20
4.1.3	Производителност (Performance):.....	20
4.1.4	Сигурност (Security):	20
4.1.5	Възможност за тестване (Testability):	20
4.1.6	Използваемост (Usability):	20
4.2	Мудъл.....	21
4.2.1	Достъпност (Availability)	21



4.2.2	Разширяемост (Scalability)	21
4.2.3	Производителност (Performance).....	21
4.2.4	Сигурност (Security)	21
4.2.5	Възможност за тестване (Testability)	21
4.2.6	Интероперабилност (Interoperability)	21
4.2.7	Използваемост (Usability)	22



1. Въведение

В този документ ще се запознаем с ключовите компоненти, взаимодействия и стратегии за внедряване, които са в основата на тази сложна система. От уеб интерфейса на предния край до бота Discord, задвижван от Discord.NET, и OAuth 2.0 за Moodle - всеки аспект на нашето решение е внимателно проектиран, за да осигури сигурно, мащабируемо и удобно за потребителя изживяване.

2. Предназначение

2.1 Обхват

Този документ включва анализа и дизайна на приложението, което включва:

- Анализиране на функционалните и нефункционални изисквания.
- Проектиране на архитектурата на системата, включително тази за удостоверяване на потребителите и синхронизирането на ролите.

2.2 Актьори

- Редовен потребител - това са лица, които са част от Discord сървър, където се намира ботът. Те използват приложението, за да удостоверят автентичността на своите акаунти в Moodle и потенциално да получат достъп до определени роли въз основа на своите роли в Moodle.
- Администратор - Администраторите или сървърните администратори са потребители с привилегирован достъп до сървър Discord. Администраторите имат допълнителни права и достъп до специфични за тях функции, включително командата за синхронизиране на ролите.

3. Архитектурен обзор

3.1 Use Case: Аутентикация на потребителите и даване на роля

Актьори:

- Дискорд сървър администратор
- Обикновени потребители

Предварителни условия:

- Ботът е настроен и пуснат.
- Съществуват акаунти в Мудъл за потребителите в дискорд сървър.
- Администраторът на сървър е определил разпределение на ролите в Дискорд въз основа на ролите в Мудъл.

Основен сценарии:

- Удостоверяване на потребителя:



- ✓ Член на дискорд сървъра, иска да удостовери своя Мудъл акаунт в Дискорд.
- ✓ Потребителя взаимодейства с бота, като използва /auth командата.
- ✓ Ботът подканя потребителя да въведе своите данни за Мудъл (Потребителско име и парола).
- Процес на удостоверяване на автентичността
 - ✓ Ботът инициализира потока OAuth 2.0 към Мудъл, като използва предоставените идентификационни данни на потребителя.
 - ✓ При успешна валидация Мудъл генерира токен за достъп.
 - ✓ Ботът получава този токен, който показва че потребителя е удостоверен.
- Свързване на Дискорд и Мудъл
 - ✓ Идентификаторът на потребителя в Дискорд се свързва с неговия акаунт в Мудъл в базата данни на приложението. Това позволява на бота да разпознае потребителя и да извлича данни от Мудъл за него.
- Ботът присвоява подходящата роля в Дискорд въз основа на данните от Мудъл.

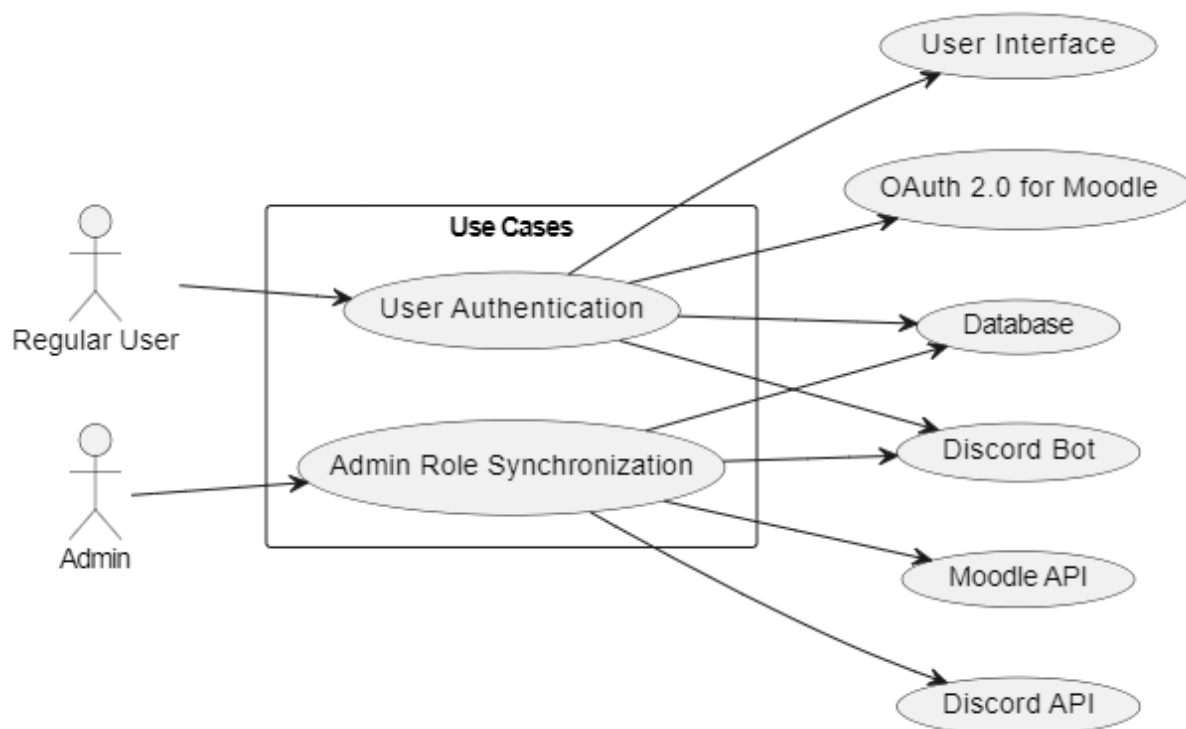
Алтернативен сценарии (обработка на грешки):

- Ако идентификационните данни на потребителя в Мудъл са невалидни, ботът информира потребителя че процесът на удостоверение е неуспешен.
- Ако потребителя вече е свързал своя акаунт, ботът го информира и не повтаря процеса.

Употреба от Администратора:

- Синхронизиране на ролите
 - ✓ Администратора инициира процеса на синхронизиране чрез командата /sync
 - ✓ Ботът извлича последните данни от Мудъл за всички потребители, със свързани акаунти.
 - ✓ Сравнява извлечените данни от Мудъл с актуалните роли в Дискорд.
- Обработка на грешки
 - ✓ Ако има грешки в процеса на синхронизиране ботът информира администратора.





3.2 Use Case: Модул система за обработка на файлове

Актьори:

- Администратор на УИСС.

Описание:

- Позволява на администратор да качва файл, съдържащ информация за студентите.

Предварително условие:

- Използва се удостоверено IP.

Пост-условие:

- Файловете се обработват, данните се съхраняват в база данни и администраторът получава файлове, които са готови за качване в Модул.

Основен процес:

- Администратора на УИСС качва файл
 - Администраторът на УИСС достъпва уеб приложението, с разрешение само за определен IP адрес.
 - Уеб приложението съдържа функционалност за качване на файл.



- Администраторът качва файл, съдържащ информацията на всички студенти, на сървъра.
- Запис на данни
 - Сървърът получава качения файл и го съхранява сигурно в база данни.
 - Данните във файла се анализират и се организират като записи за студенти в базата данни.
- Актуализация на данните за сменили специалността студенти
 - Сървърът получава качения файл и го съхранява сигурно в база данни.
 - Данните във файла се анализират и се организират като записи за студенти в базата данни.
- Скриване на данните
 - За да се поддържа поверителността и защитата на данните, сървърът скрива чувствителна информация, като лични данни, оценки и други незадължителни данни.
 - Сървърът запазва само съществената информация, необходима за актуализации в Moodle.
- Създаване на файлове
 - „NewStudents.csv” – съдържа информацията за новодобавените студенти, подходящ за качване в Мудъл.
 - „RemovedStudents.csv” – съдържа списък със студенти, които са сменили факултета/специалността и трябва да бъдат премахнати от конкретните курсове.
- Доставка на файлове
 - Уеб приложението предоставя на администратора на УИСС линкове за изтегляне на двата файла.
 - Администратора изтегля файловете за допълнителни действия.

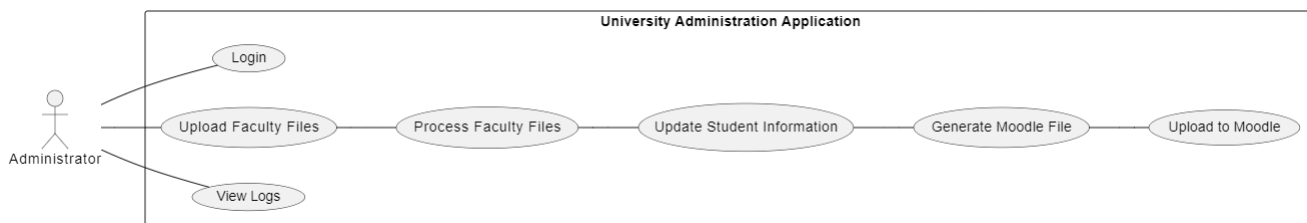
Следствие

- Администраторът на УИСС е получил необходимите файлове за качване на нови студенти и премахване на студенти от конкретни курсове.
- Информацията, подадена от администраторът на УИСС, е запазена в база данни.
- Чувствителната информация е скрита, за да се поддържа поверителността и защитата на данните.

Обработка на грешки

- Ако сървърът се сблъска с грешка по време на обработката на данните, актуализациите на базата данни или генерирането на файлове, той информира администратора на УИСС за проблема.





3.3 Логически изглед

3.3.1 Дискорд бот

Действа като посредник между Дискорд, Мудъл и базата данни на приложението. Той слуша и отговаря на команди на потребителите и администраторите. Комуникира с Дискорд АПИ и Мудъл АПИ, за да извлича и актуализира данни.

- База данни - Съхранява данните на потребителите, включително идентификаторите на Дискорд, данните на студента в Мудъл. Улеснява свързването на потребителски акаунти и извличането на данни.
- OAuth 2.0 - Удостоверя автентичността на потребителите спрямо техните идентификационни данни за Мудъл. Генерира токени за достъп при успешно удостоверяване.
- Поток на данни и контрол
 - ❖ Удостоверяване на потребителя
 - 1) Потребителя започва процеса на удостоверяване, като използват командата /auth в Дискорд.
 - 2) Ботът събира входните данни на потребителя (идентификационни данни за Мудъл).
 - 3) При успешно валидиране Мудъл генерира токън за достъп.
 - 4) Дискорд ботът комуникира с OAuth 2.0 за Мудъл, за да потвърди идентификацията данни.
 - 5) При успешно валидиране Мудъл генерира токън за достъп.
 - 6) Дискорд ботът свързва Дискорд ID на потребителя с неговия акаунт в Мудъл базата данни.
 - 7) Дискорд ботът присвоява ролята въз основа на предварително дефинирани съпоставки в базата данни.
 - ❖ Синхронизира на ролята от администратора
 - 1) Администраторите инициират процеса на синхронизация използвайки командата /sync.
 - 2) Дискорд бота извлича последните данни от Мудъл за всички свързани потребители.
 - 3) Дискорд бот сравнява данните от Мудъл със съществуващите роли в Дискорд.
 - 4) Дискорд бота актуализира ролята въз основа на данните в Мудъл.
 - 5) Дискорд бота уведомява администратора за резултатите от синхронизацията.



❖ Взаимодействия

- 1) Дискорд бота взаимодейства с АПИ на дискорд за управление на ролите на потребителите и с API на Мудъл за удостоверяване на потребителите за извличане на данни.
- 2) Дискорд бота взаимодейства с базата данни, за да съхранява данните за потребителите и ролите.
- 3) OAuth 2.0 за Мудъл взаимодейства със системата за удостоверяване на Мудъл.

❖ Контрол на достъпа въз основа на роли

- 1) Обикновените потребители могат да инициират процеса на удостоверяване чрез командата /auth.
- 2) Администраторите могат да инициират синхронизиране на ролите чрез командата /sync.

Този логически изглед илюстрира как различните компоненти в архитектурата си взаимодействат, за да удостоверяват потребителите, да управляват ролите и да синхронизират данните между Discord и Moodle.

3.3.2 Мудъл система

- Аутентикация на потребител

- User Interface

- ✓ Отговаря за представянето на информация на администраторите и за получаването на данни.
 - ✓ Състои се от страница за качване на файлове и за известия.

- Backend приложение

- ✓ Съдържа основната логика на приложението и обработката на данни.
 - ✓ Управлява потока от данни между потребителския интерфейс, компонентите за обработка на данни и външни системи.

- Компоненти за обработка на данни

- ✓ Състои се от компонентите, отговорни за обработка на факултетните файлове, интеграцията на данни и актуализацията на базата данни.
 - ✓ Включва подкомпоненти за извличане, валидиране и трансформиране на данни.
 - ✓ Осигурява правилно интегриране на данните в студентската база данни.

- База данни

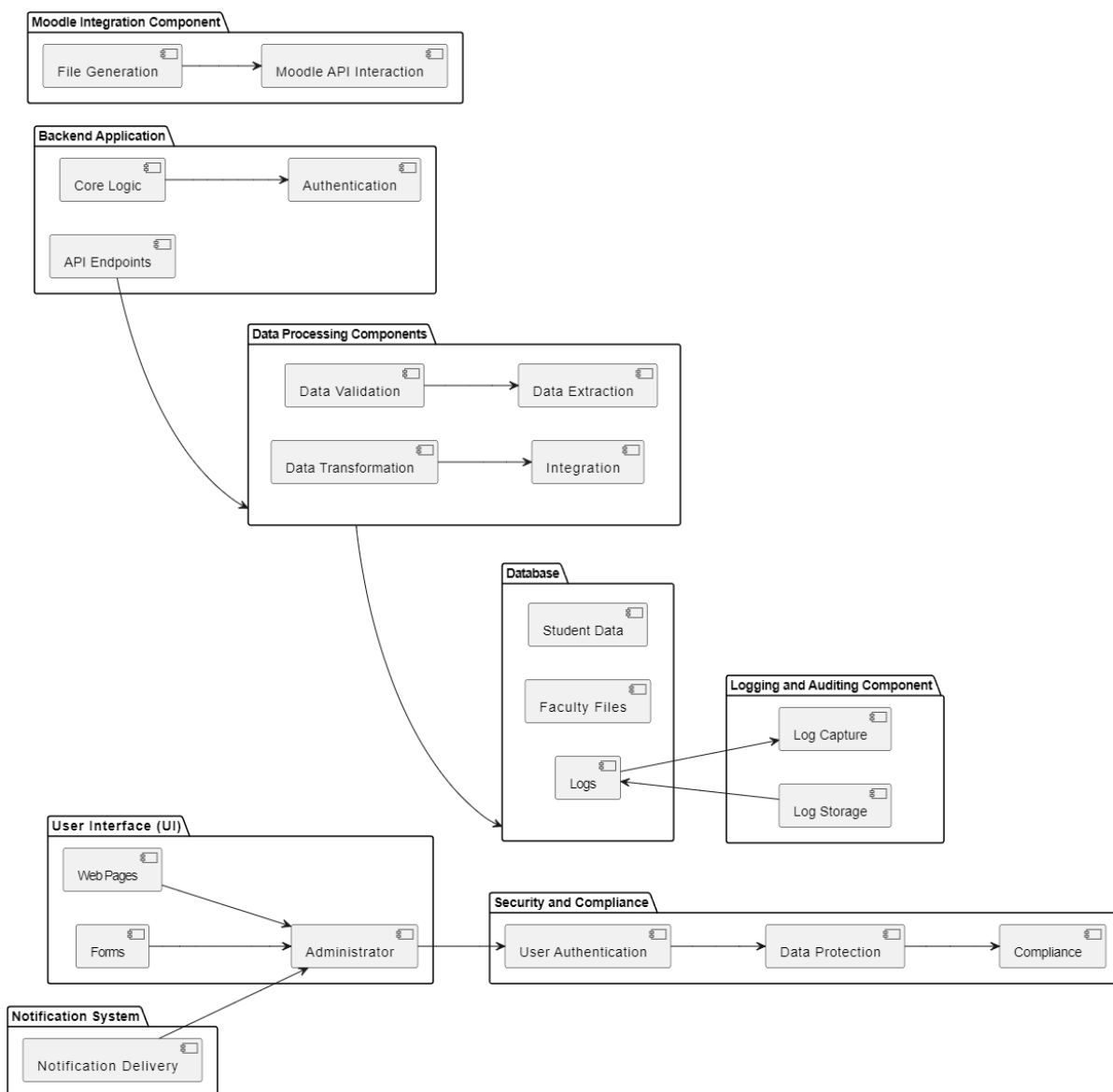
- ✓ Съхранява информация за студентите и факултетните файлове.
 - ✓ Организиране е с помоща на релационна система за бази данни (MySQL).
 - ✓ Съдържа таблици за студенти, администратори.



- Компонент за интеграция с Мудъл
 - ✓ Управлява генерирането на файлове, съвместими с Мудъл и взаимодействия с Мудъл АПИ.
 - ✓ Подготвя данните за актуализиране на информацията за студентите .
- Система за уведомяване
 - ✓ Изпраща известия до администраторите, за да ги информира за състоянието на качването и актуализирането на файлове.
 - ✓ Гарантира, че администраторите са информирани за дейностите на системата и евентуални грешки

Този логически изглед осигурява разбиране на високо ниво за това как различните компоненти в приложението са организирани и взаимодействат помежду си, за да изпълняват предвидената си функционалност. Той служи като основа за последващото проектиране и внедряване на системата.





3.4 Процесен изглед

3.4.1 Дискорд бот

Процеси:

- Процес на удостоверяване на потребителя
 - ✓ Задейства се когато потребителя изпълни /auth командата.
 - ✓ Включва взаимодействията между Дискорд бота, OAuth 2.0 за Мудъл и базата данни.
- Процес на синхронизиране на ролите на администраторите
 - ✓ Иницира се, когато администратор изпълни /sync командата в Дискорд.
 - ✓ Включва взаимодействията между Дискорд бот, Базата данни и външни АПИ (Дискорд и Мудъл).

3.4.1.1 Процес на удостоверяване на потребителя



Потребителя иницира командата /auth в Дискорд. Дискорд бота получава командата и подканва потребителя да въведе своите идентификационни данни за Мудъл чрез потребителския интерфейс. Потребителят осигурява идентификационните данни за Мудъл чрез потребителския интерфейс. Дискорд ботът изпраща данните на OAuth 2.0 на Мудъл за валидиране. OAuth 2.0 комуникира с Мудъл АПИ, за да удостовери потребителя. При успешно удостоверяване Мудъл генерира токен за достъп и го праща обратно към Дискорд бота. Ботът съхранява Дискорд ID на потребителя и свързаните с Мудъл идентификационни данни в базата данни. Дискорд ботът присвоява роля в Дискорд въз основа на данните. Предоставя се обратна връзка на потребителя.

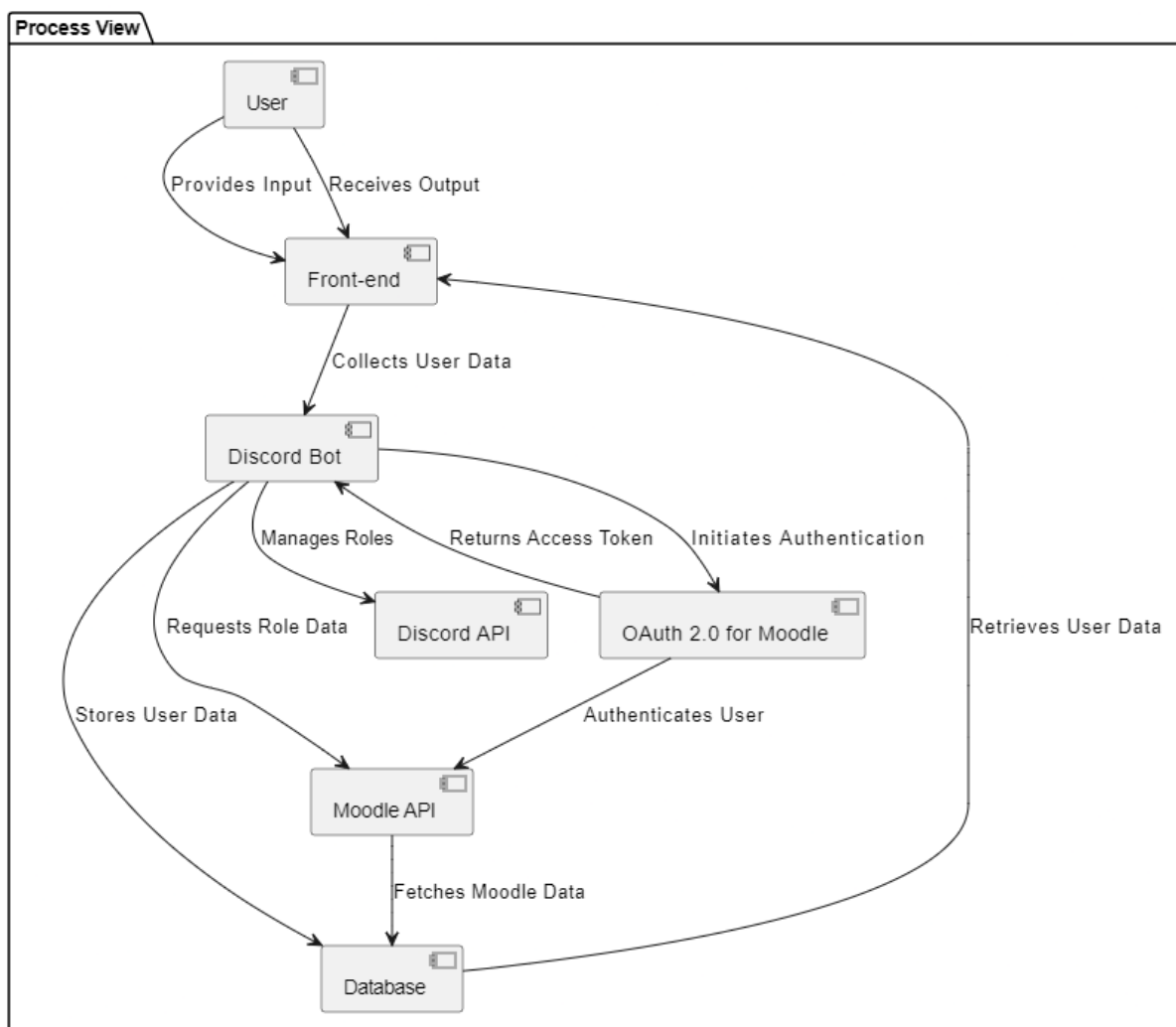
3.4.1.2 Процес на синхронизиране на ролите на администратора

Администратора иницира командата /sync. Дискорд бота извлича най-новите данни за Мудъл за всички свързани потребители с Мудъл АПИ. Ботът извлича съществуващите данни за потребителските и съпоставянето на ролите от базата данни. Сравнява данните от Мудъл със съществуващите. След това актуализира ролите в дискорд въз основа на данните. Предоставя обратна връзка на администратора по отношение на процеса на синхронизиране на ролите.

Взаимодействия: Връзка между потребителите на Мудъл и Discord

- Потребителският интерфейс взаимодейства с потребителите и предоставя средства за въвеждане на идентификационните данни на Moodle.
- Дискорд бот управлява потребителските команди, комуникира с OAuth 2.0 за Moodle и взаимодейства с базата данни за съхранение на потребителските данни.
- OAuth 2.0 за Moodle комуникира с Moodle API, за да обработва удостоверяването на потребителите.
- Discord Bot комуникира с Discord API за управление на ролите и взаимодейства с Базата данни за извличане и съхранение на данни.
- Базата данни съхранява данни за потребителите и съпоставки на роли.





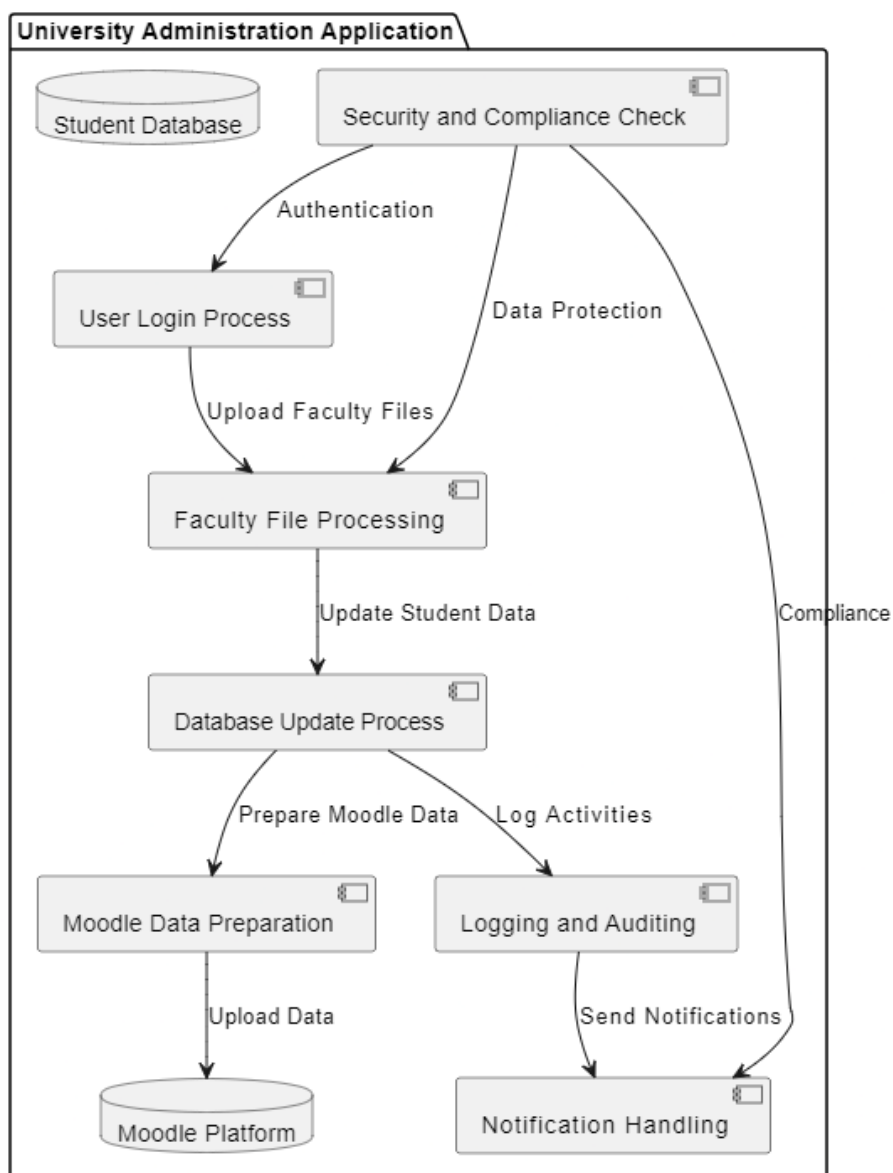
3.4.2 Мудъл система

- Обработка на файловете на факултета
 - Този процес се задейства, когато администраторите качват файлове на факултета.
 - Този процес се задейства, когато администраторите качват файлове на факултета.
 - Той включва компонентите Извличане на данни, Утвърждаване на данни и Преобразуване на данни.
 - Данните се извличат от качените файлове, валидират се за точност и се трансформират във формат, подходящ за интегриране в студентската база данни.
- Процес на актуализиране на базата данни
 - След успешна обработка на факултетните файлове се изпълнява процесът за актуализиране на базата данни.



- Той включва интегриране на трансформирани данни в студентската база данни, като информацията за студентите се актуализира въз основа на обработените факултетни файлове.
 - Този процес се организира от компонента за интеграция.
- Подготовка на данните за Moodle
 - Когато информацията за студентите е актуализирана в базата данни на приложението, администраторите могат да стартират процеса на подготовка на данни за Moodle.
 - Компонентът File Generation (Генериране на файлове) генерира файлове в необходимия формат за Moodle.
 - Moodle API Interaction (Взаимодействие с Moodle API) отговаря за осигуряване на съвместимост с Moodle и за качване на данните.
- Обработка на уведомления
 - Процесът Notification Handling се грижи за информирането на администраторите за важни системни събития. Уведомленията могат да включват успешни качвания, грешки и актуализации.
 - Компонентът Notification Delivery (Доставка на уведомления) отговаря за изпращането на тези уведомления до администраторите.





3.5 Изглед на данните

3.5.1 Дискорд бот

Изгледът на данните на архитектурата на приложението описва структурата и връзките на елементите на данните в системата.

Структура на базата данни за интеграция между Discord и Moodle:

Таблицы:

1) Users (Потребители)

- UserID: Уникален идентификатор на потребителя.
- DiscordID: ID на потребителя в Discord.
- MoodleUsername: Потребителско име в Moodle.
- AuthToken: Токен за аутентикация.



2) Roles (Роли)

- RoleID: Уникален идентификатор на ролята.
- RoleName: Име на ролята.

3) UserRoles (Потребителски Роли)

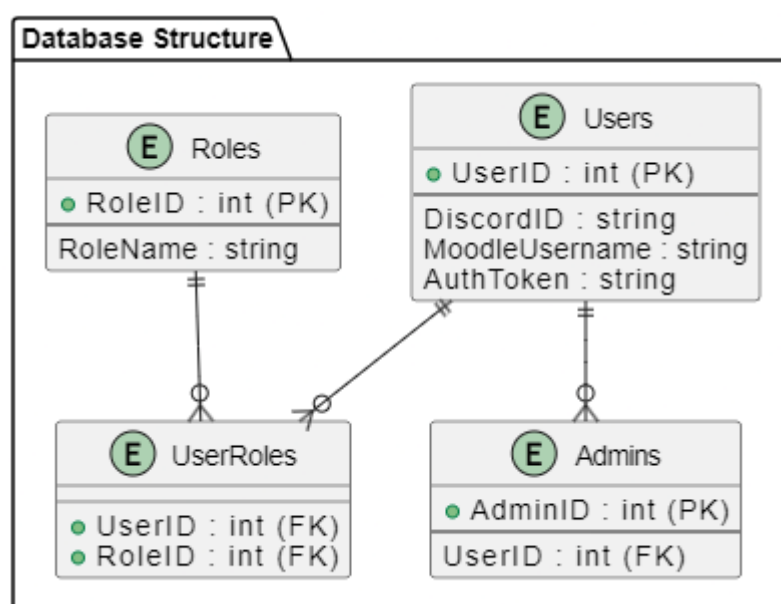
- UserID: Идентификатор на потребителя.
- RoleID: Идентификатор на ролята.

4) Admins (Администратори)

- AdminID: Уникален идентификатор на администратора.
- UserID: Идентификатор на потребителя от таблицата Users.

Връзки между таблици:

- ✓ Users и UserRoles: Отношение "много към много".
- ✓ Roles и UserRoles: Отношение "един към много".
- ✓ Users и Admins: Отношение "един към много".



3.5.2 Мудъл

Таблица: Студенти

- ID на студент: Уникален идентификатор за всеки студент.
- Име: Пълно име на студента.
- Факултет: Към кой факултет принадлежи студента.
- Специалност: Конкретната специалност на студента.
- Курс: Кой курс е студентът (напр. 1-ви, 2-ри).
- Email: Електронна поща на студента.



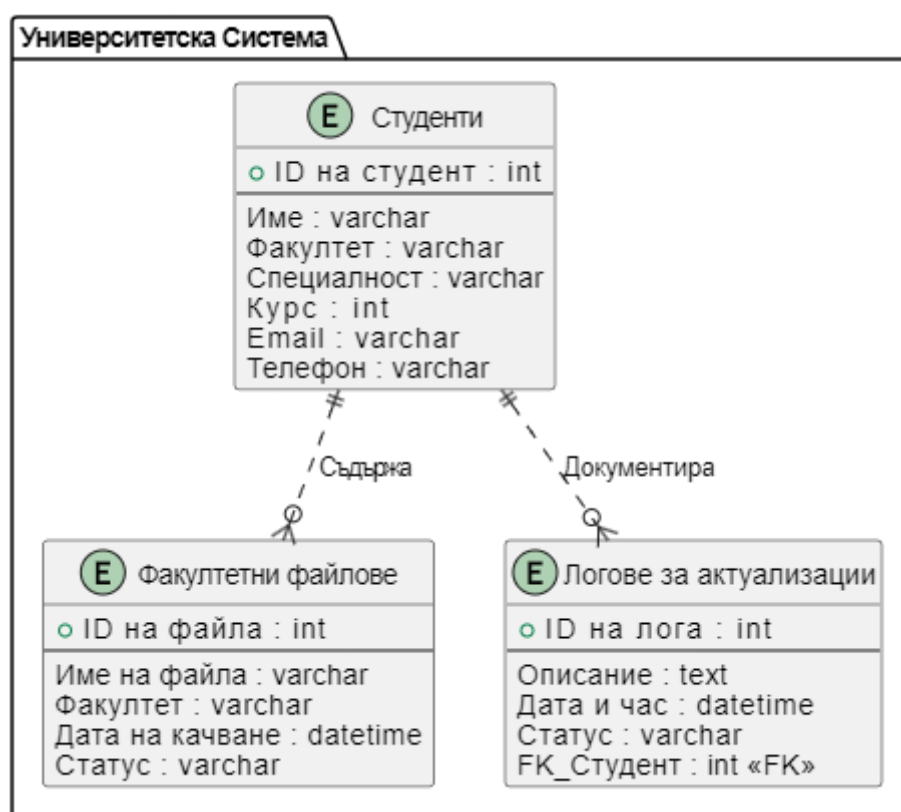
- Телефон: Телефонен номер.

Таблица: Факултетни файлове

- ID на файла: Уникален идентификатор за всеки качен файл.
- Име на файла: Името на качения файл.
- Факултет: Факултетът, към който принадлежи информацията във файла.
- Дата на качване: Дата и час на качването на файла.
- Статус: Успешно обработен, Грешка, Обработка се, и т.н.

Таблица: Логове за актуализации

- ID на лога: Уникален идентификатор за всяко действие.
- Описание: Кратко описание на извършеното действие (напр. "Актуализация на данни за студент Иван Петров").
- Дата и час: Дата и час на извършване на действието.
- Статус: Успешно, Грешка, и т.н.



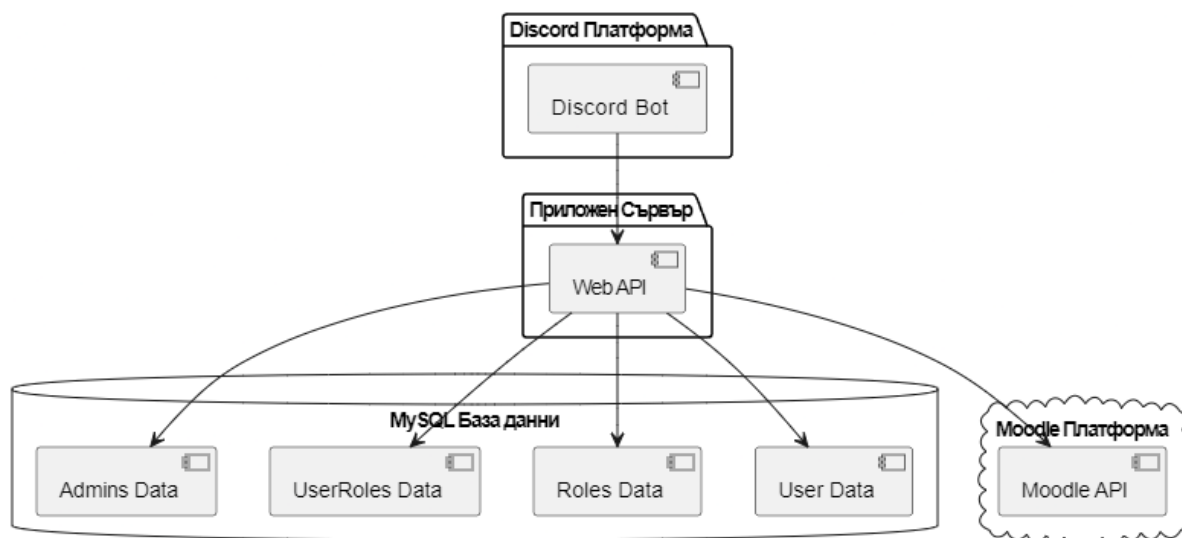
3.6 Изглед на внедряването

3.6.1 Дискорд бот



Тази платформа представлява средата, в която функционира Discord. Върху нея се изпълнява Discord Bot интеграцията. Дискорд ботът, от своя страна, предоставя потребителски команди и функционалност директно в Discord средата.

- ✓ **Приложен Сървър (Application Server):**
 - Тук се изпълнява главната логика на приложението и всички обработки.
 - **Web API:** Това е интерфейсът, който обработва командите, изпратени от дискорд бота. Чрез него се осъществява комуникацията между Discord, базата данни и Moodle. API-то може да извършва различни CRUD операции в базата данни или да изпраща и получава информация от Moodle.
- ✓ **MySQL База данни:**
 - Тук се съхраняват всички данни, свързани със системата, включително информация за потребители, роли и администратори.
 - **User Data:** Данни за потребителите, като техни Discord ID, Moodle потребителски имена и аутентикационни токени.
 - **Roles Data:** Определяне на различните роли, които потребителите могат да имат в Discord.
 - **UserRoles Data:** Свързва потребителите с техните роли, позволявайки на един потребител да има множество роли.
 - **Admins Data:** Идентификация на администраторите в системата.
- ✓ **Moodle Платформа**
 - **Moodle API:** Това е интерфейсът, чрез се взаимодейства с Moodle. Чрез него може да извличаме информация за потребителите, техните роли в Moodle и други свързани данни.



3.6.2 Модул

- ✓ **Уеб сървър** - Този сървър ще бъде отговорен за хостване на приложението. Когато потребителите достъпват системата, уеб сървърът



обработка заявките и връща съответните резултати. Компонентите са основната бизнес логика, уеб интерфейса и API-то на системата.

- ✓ База данни - Тук ще се съхраняват всички данни за студентите, информацията от качените файлове и логове за актуализации. Компоненти: Таблиците "Студенти", "Факултетни файлове", "Логове за актуализации.
- ✓ Интеграционен слой - Осигурява комуникацията между Дискорд и Moodle, като използва Moodle's API за обновяване на студентската информация. Компоненти: Модул за обработка и изпращане на данни към Moodle.
- ✓ Файлово хранилище - тук ще бъдат държани временно качените файлове, преди те да бъдат обработени от системата. Компоненти: Структурирано хранилище с папки/директории, разделени по факултети или дати.
- ✓ Интернет и мрежова сигурност - Всички комуникации между различните компоненти са защитени и криптирани, особено когато данните се трансферират в Интернет.

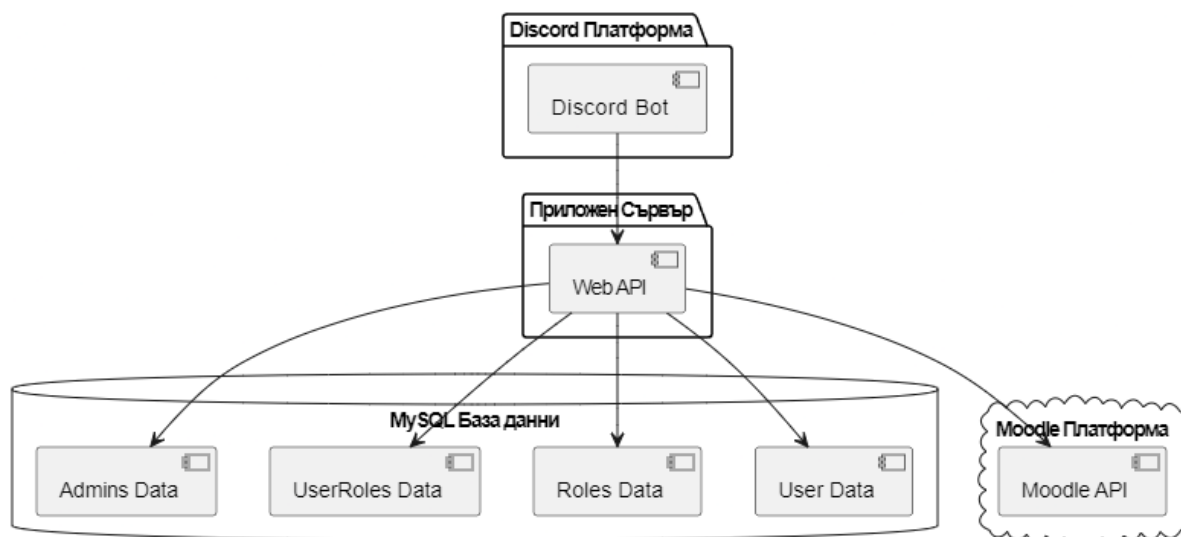
3.7 Изглед на имплементацията

3.7.1 Дискорд бот

- Discord Bot (C#):
 - ❖ Инициализация: Стартиране на бота и установяване на връзка с Discord чрез Discord API ключ.
 - ❖ Команди:
 - auth: Аутентикация на потребител чрез Moodle. Изпраща заявка до Web API, който от своя страна взаимодейства с Moodle.
 - sync: (Само за администратори) - Синхронизира ролите в Discord с тези в Moodle.
- 2. Web API (C#):
 - ❖ Ендпойнти:
 - Authenticate: Аутентикира потребител чрез Moodle и връща резултат.
 - SyncRoles: Синхронизира ролите от Moodle с тези в Discord.
 - Взаимодействие с базата данни: Използва CRUD операции за взаимодействие с MySQL базата данни.
- MySQL база данни:
 - ❖ Таблицы:
 - Users: Съхранява информация за потребителите, включително Discord ID и Moodle потребителски данни.
 - Roles: Определя ролите, които потребителите могат да имат в Discord.



- UserRoles: Свързва потребителите с техните роли.
- Admins: Идентификация на администраторите.
- Moodle API интеграция (C#):
 - Заявки:
 - ❖ GetUserData: Извлича данни за даден потребител.
 - ❖ GetUserRoles: Извлича ролите за даден потребител в Moodle.



3.7.2 Модул

- Уеб приложение
 - ✓ Потребителски интерфейс: Този модул ще бъде отговорен за всички взаимодействия с потребителите. Той ще включва компоненти за навигация, форми за качване на файлове, показване на информация и др.
 - ✓ Управление на файлове: Този модул ще обработва качените файлове от различните факултети, ще извлекат нужната информация от тях и ще я подготви за въвеждане в базата данни.
 - ✓ Бизнес логика: Тук се съдържа основната логика за обработка на студентската информация, взаимодействие с базата данни и обновяване на Moodle.
- База данни
 - ✓ Модул за взаимодействие с базата данни (DAO - Data Access Object): Този модул обезпечава комуникацията между системата и базата данни, като осигурява методи за извличане, въвеждане и обновяване на данни.
- Интеграция с Moodle
 - ✓ Moodle API клиент: Този модул ще обезпечава връзката с Moodle чрез неговото API. Чрез него системата ще изпраща обновленията към Moodle.
 - ✓ Помощни функции
 - ✓ Обработка на грешки: Отговорен за обработката и логването на грешките, които могат да възникнат по време на работа на системата.



- ✓ Логика за сигурност: Включва функционалност за удостоверяване, авторизация и други мерки за сигурност.
- Файлово хранилище
 - ✓ Управление на файлове: Обработка и съхранение на качените файлове до тяхната обработка.

4. Нефункционални изисквания

4.1 Дискорд

4.1.1 Достъпност (Availability)

- Системата трябва да има висока наличност, за да обслужва потребителите без прекъсвания. Това може да изисква използването на разпределени системи или облачни услуги с автоматично балансиране на натоварването.

4.1.2 Разширяемост (Scalability):

- Системата е проектирана така, че да може да се справя с увеличаващ се брой потребители или заявки. Това може да включва вертикално или хоризонтално мащабиране на сървъра и базата данни.

4.1.3 Производителност (Performance):

- Времето за отговор на системата след изпълнение на команда или заявка трябва да е минимално. Оптимизацията на базата данни, кешовете и алгоритмите, които се използват, е ключова.

4.1.4 Сигурност (Security):

- Всички комуникации между Discord, Web API и Moodle трябва да се криптират.
- Данните за аутентикация (пароли, токени и др.) трябва да се съхраняват в криптиран вид.
- Системата трябва да има вградени механизми срещу различни видове атаки (напр. DoS атаки, SQL инжекции).

4.1.5 Възможност за тестване (Testability):

- Кодът трябва да е написан така, че да може лесно да се тества, като се използват автоматизирани тестове, интеграционни тестове и тестове на единици.
- Интероперабилност (Interoperability):
- Системата трябва да може да взаимодейства безпроблемно с външните системи - Discord и Moodle. Това изисква използването на стандартни протоколи и формати.

4.1.6 Използваемост (Usability):



- Discord ботът трябва да има интуитивни команди и ясно описание на функционалността си.
- Възможни са и допълнителни функции като помощна команда или интерактивни графични интерфейси в Discord за по-лесно управление.

4.2 Модъл

4.2.1 Достъпност (Availability)

- Системата трябва да бъде достъпна 99,9% от времето, което означава максимум 8.76 часа прекъсвания на година.
- В случай на проблем или авария, системата трябва да може автоматично да възстанови своето функциониране в рамките на 5 минути.

4.2.2 Разширяемост (Scalability)

- Системата трябва да може да поддържа увеличение на броя потребители и обема на данните без промяна в архитектурата.
- Добавянето на нови функционалности или модули не трябва да изисква основни промени в съществуващата архитектура.

4.2.3 Производителност (Performance)

- Отговорът на заявка от системата не трябва да отнема повече от 2 секунди при стандартно натоварване.
- Системата трябва да може да обработва минимум 10 000 едновременни заявки без видимо намаляване на производителността.

4.2.4 Сигурност (Security)

- Всички данни, изпращани и получавани от системата, трябва да бъдат криптирани.
- Паролите на потребителите трябва да бъдат съхранявани в хеширана форма.
- Достъпът до системата трябва да изисква автентикация и ролева авторизация.

4.2.5 Възможност за тестване (Testability)

- Всеки модул на системата трябва да бъде проектиран така, че да може лесно да се подлага на юнит тестове.
- Системата трябва да поддържа интеграция с инструменти за автоматично тестване.

4.2.6 Интероперабилност (Interoperability)



- Системата трябва да може да комуникира безпроблемно с Moodle и други потенциални външни системи чрез стандартни протоколи и интерфейси (например RESTful API).

4.2.7 Използваемост (Usability)

- Потребителският интерфейс трябва да бъде интуитивен и лесен за използване, като изисква минимално обучение.
- Потребителите трябва да получават ясни и полезни съобщения при грешки или неправилни действия.

