

ИНТЕГРАЦИЯ НА УНИВЕРСИТЕТСКИ ИНФОРМАЦИОННИ СИСТЕМИ. АРХИТЕКТУРЕН ПРОЕКТ.

Автори: Божидар Попиванов, Мартин Петров,

Алекс Трайков, Кристиян Митев, Жоро Спасов



5 Януари 2024

|  |  |  |
| --- | --- | --- |
| **Развитие на документа** | | |
| **Редакция** | **Дата** | **Описание на изменението** |
| 0 | 21.10.2023 | Първоначално издание |
| 1 | 05.01.2024 | Актуализация на базата данни и диаграми |
|  |  |  |
|  |  |  |

Съдържание

[1. Въведение 4](#_Toc155435304)

[2. Предназначение 4](#_Toc155435305)

[2.1 Обхват 4](#_Toc155435306)

[2.2 Актьори 4](#_Toc155435307)

[3. Архитектурен обзор 4](#_Toc155435308)

[3.1 Use Case: Аутентикация на потребителите и даване на роля 4](#_Toc155435309)

[3.2 Use Case: Уеб приложение за обработка на файлове за Мудъл 6](#_Toc155435310)

[3.3 Логически изглед 7](#_Toc155435311)

[3.3.1 Дискорд бот 7](#_Toc155435312)

[3.3.2 Мудъл система 8](#_Toc155435313)

[3.4 Процесен изглед 10](#_Toc155435314)

[3.4.1 Дискорд бот 10](#_Toc155435315)

[3.4.2 Уеб приложение 12](#_Toc155435316)

[3.5 Изглед на данните 13](#_Toc155435317)

[3.5.1 Дискорд бот 13](#_Toc155435318)

[3.6 Изглед на внедряването 15](#_Toc155435319)

[3.6.1 Дискорд бот 15](#_Toc155435320)

[3.6.2 Мудъл 16](#_Toc155435321)

[3.7 Изглед на имплементацията 16](#_Toc155435322)

[3.7.1 Дискорд бот 16](#_Toc155435323)

[3.7.2 Мудъл 17](#_Toc155435324)

[4. Нефункционални изисквания 18](#_Toc155435325)

[4.1 Дискорд 18](#_Toc155435326)

[4.1.1 Достъпност (Availability) 18](#_Toc155435327)

[4.1.2 Разширяемост (Scalability): 18](#_Toc155435328)

[4.1.3 Производителност (Performance): 18](#_Toc155435329)

[4.1.4 Сигурност (Security): 18](#_Toc155435330)

[4.1.5 Възможност за тестване (Testability): 18](#_Toc155435331)

[4.1.6 Използваемост (Usability): 18](#_Toc155435332)

[4.2 Мудъл 19](#_Toc155435333)

[4.2.1 Достъпност (Availability) 19](#_Toc155435334)

[4.2.2 Разширяемост (Scalability) 19](#_Toc155435335)

[4.2.3 Производителност (Performance) 19](#_Toc155435336)

[4.2.4 Сигурност (Security) 19](#_Toc155435337)

[4.2.5 Възможност за тестване (Testability) 19](#_Toc155435338)

[4.2.6 Интероперабилност (Interoperability) 19](#_Toc155435339)

[4.2.7 Използваемост (Usability) 19](#_Toc155435340)

# 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 до Discord бота, задвижван от Discord.JS, и OAuth 2.0 за Moodle - всеки аспект на нашето решение е внимателно проектиран, за да осигури сигурно, мащабируемо и удобно за потребителя изживяване.

# Предназначение

## 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.

## Актьори

* Редовен потребител - това са лица, които са част от Discord сървъра, 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 потребителския им профил в Moodle.
* Администратор - Администраторите или сървърните администратори са потребители с привилегирован достъп до дискорд сървъра. Администраторите имат допълнителни права и достъп до специфични за тях функции, включително командата за синхронизиране на ролите.

# Архитектурен обзор

## Use Case: Аутентикация на потребителите и даване на роля

Актьори:

* + Дискорд сървър администратор
  + Обикновени потребители

Предварителни условия:

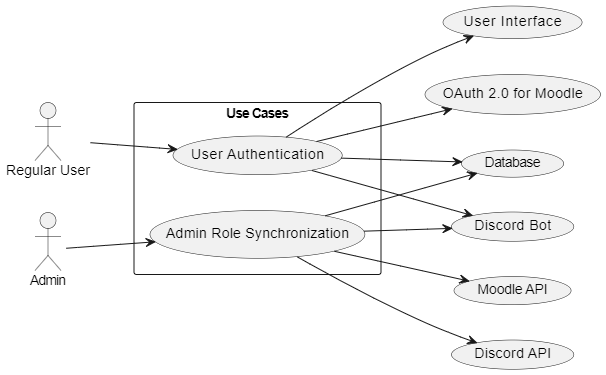
* + Ботът е насторен и пуснат.
  + Съществуват акаунти в Мудъл за потребителите в дискорд сървъра.
  + Администраторът на сървъра е определил разпределение на ролите в Дискорд въз основа на курсовете в Мудъл.

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 изпраща линк на потребителя да въведе своите данни за вход в Мудъл (потребителско име и парола).
  + Процес на удостоверяване на автентичността
    - Ботът получава данните за потребителя, след успешен вход в Мудъл.
  + Свързване на Дискорд и Мудъл
* Ботът присвоява подходящата роля в Дискорд въз основа на данните от УИСС и Мудъл.

Употреба от Администратора:

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



## Use Case: Уеб приложение за обработка на файлове за Мудъл

Актьори:

* + - Администратор на УИСС.

Описание:

* + - Позволява на админстратор да качва файл(ове), съдържащ(и) информация за студентите.

Предварително условие:

* + - Използва се удостоверено IP.

Пост-условие:

* + - Файловете се обработват в бекенд приложението, след това администраторът получава списък от промените, които ще се извършват и може да се избере одобряване или прекъсване на операцията.

Основен процес:

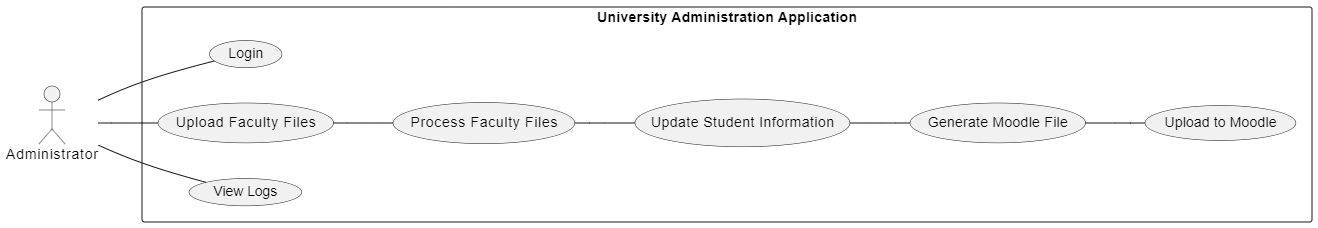
* + - Администраторът на УИСС качва файл
      * Администраторът на УИСС достъпва уеб приложението, с разрешение само за определен IP адрес.
      * Уеб приложението съдържа функционалност за качване на файл.
      * Администраторът качва файл, съдържащ информацията за студентите от определен кохорт(и), на сървъра.
    - Запис на данни
      * Данните във файла се анализират и се организират като записи за студенти в базата данни след одобрение от страна на администратора.
    - Актуализация на данните за сменили специалността студенти
      * Данните във файла се анализират и се организират като записи за студенти в базата данни след одобрение от страна на администратора.
    - Скриване на данните
      * Сървърът запазва само съществената информация, необходима за актуализации на данните в Moodle.
    - Създаване на файлове
      * „NewStudents.csv” – съдържа информацията за новодобавените студенти, подходящ за качване в уеб приложението. В последствие студентите се добавят в Мудъл.
      * „RemovedStudents.csv” – съдържа списък със студенти, които са сменили факултета/специалността и трябва да бъдат премахнати от конкретните курсове в Мудъл.

Следствие

* + - * Администраторът на УИСС е получил необходимите файлове за качване на нови студенти и премахване на студенти от конкретни курсове.
      * Информацията, подадена от администратора на УИСС, е запазена в база данни.

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.



## Логически изглед

### 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, Мудъл АПИ и УИСС, за да извлича и актуализира данни.

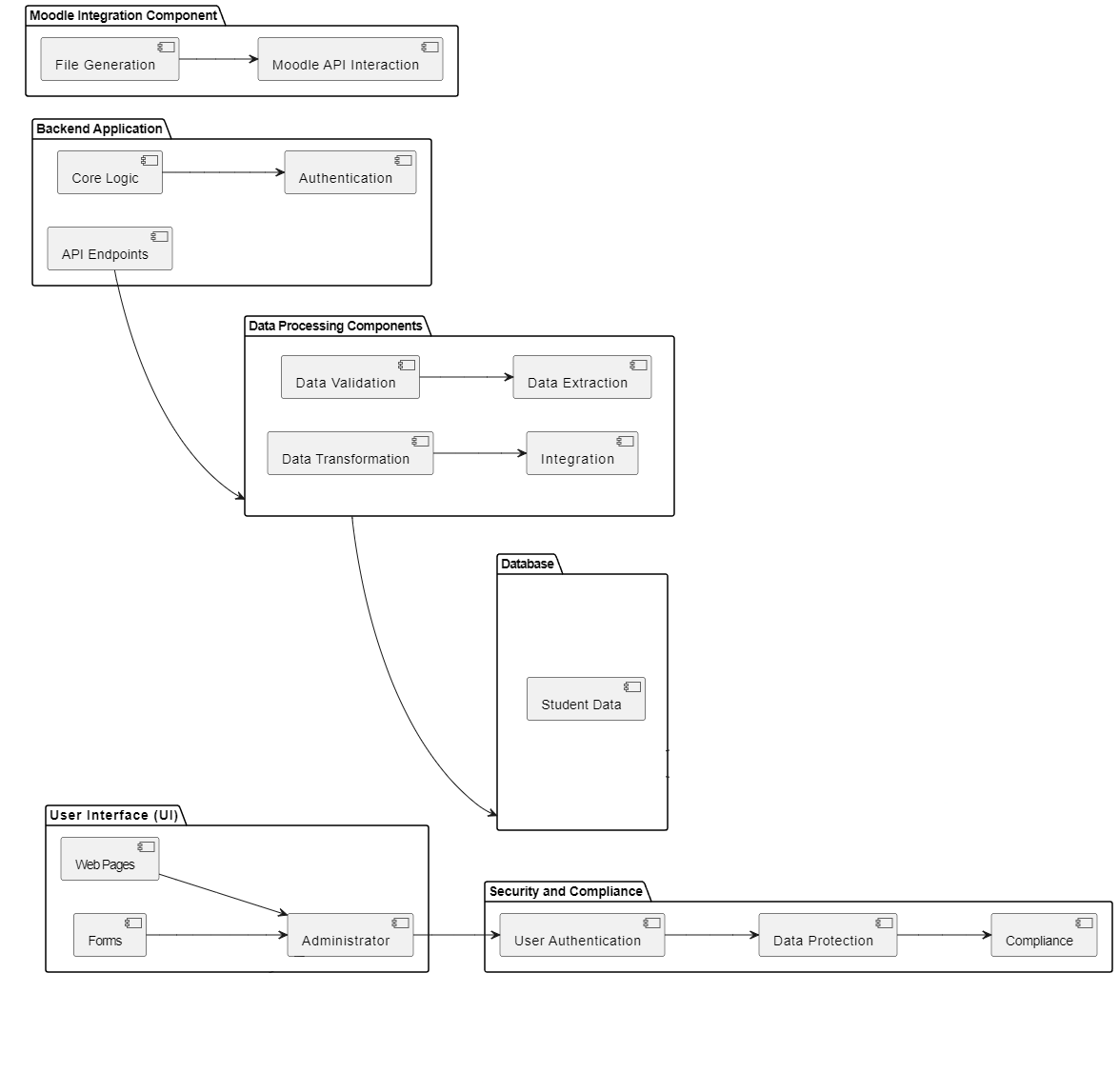
* База данни - Съхранява данните на потребителите, включително идентификаторите за Дискорд, данните на студента в Мудъл. Служи като резервно копие на данните.
* OAuth 2.0 - Удостоверява автентичността на потребителите спрямо техните идентфикационни данни за Мудъл. Генерира токени за достъп при успешно удостоверяване.
* Поток на данни и контрол
  + Удостоверяване на потребителя
    1. Потребителят започва процеса на удостоверяване, като изпозлва командата /auth в Дискорд.
    2. При успешно валидиране Мудъл генерира токън за достъп.
    3. Дискорд ботът свързва Дискорд ID с факултетния номер на студента и ИД-то на сървъра в базата данни.
    4. Дискорд ботът присвоява ролите въз основа на получените данни за студента от УИСС.
  + Синхронизиране на ролите от администратора
    1. Администраторите инициират процеса на синхронизация използвайки командата /sync.
    2. Дискорд ботът извлича последните данни от УИСС за всички свързани потребители.
    3. Дискорд ботът сравнява данните от УИСС със съществуващите роли в Дискорд.
    4. Дискорд бота актуализира ролите въз основа на данните в УИСС.
    5. Дискорд ботът уведомява администратора за резултатите от синхронизацията.
  + Взаимодействия
    1. Дискорд ботът взаимодейства с АПИ на Дискорд за управление на ролите на потребителите и с АПИ на УИСС за удостоверяване на потребителите за извличане на данни.
    2. 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.

### 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.
  + 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 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.
    - Организирана е с помощта на релационна система за бази данни (MySQL).
    - Съдържа таблици за студенти, данни за дискорд сървърите и връзката между дискорд идентификатора на студент и сървъра, към който принадлежи.
  + Компонент за интеграция с Мудъл
    - Управлява генерирането на файлове, съвместими с Мудъл и взаймодества с Мудъл АПИ.
    - Подготвя данните за актуализиране на информацията за студентите.
  + Система за уведомяване
    - Информира администратора за състоянията при качването и актуализирането на файлове.
    - Гарантира, че администраторите са информирани за дейностите на системата и за 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.



## Процесен изглед

### 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 УИСС.
  + Процес на синхронизиране на ролите от администраторите
    - Инициира се, когато администратор изпълни /sync командата в Дискорд.
    - Включва взаимодействията между Дискорд бот, Мудъл и УИСС.

#### Процес на удостоверяване на потребителя

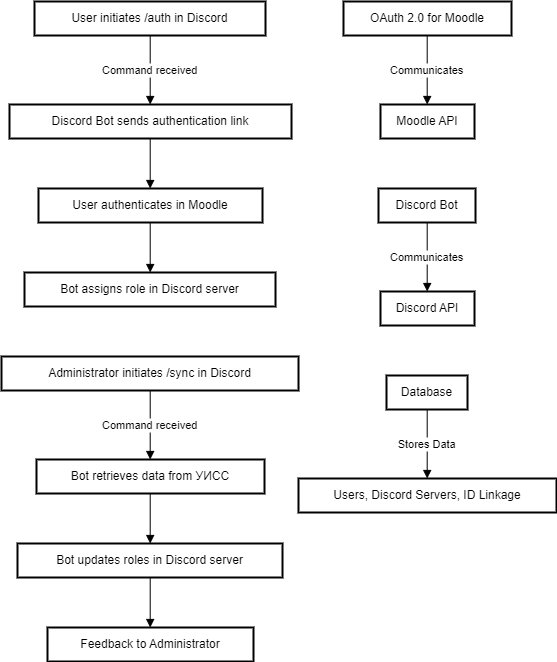
Потребителят иницира командата /auth в Дискорд. Дискорд ботът получава командата и изпраща линк на потребителя, който при избирането му, го препраща в Мудъл. Там той въвежда своите идентефикационни данни (потребителско име и парола). При успешно удостоверяване на потребителя в Мудъл, Дискорд ботът му дава нужната роля за достъп до индивидуални канали в дискорд сървъра, свързани със съответния курс.

#### Процес на синхронизиране на ролите от администратора

Администраторът инициира командата /sync. Дискорд ботът извлича най-новите данни за всички потребители от УИСС. След това актуализира ролите в дискорд сървъра въз основа на получените данни. Предоставя обратна връзка на администратора по отношение на процеса на синхронизиране на ролите.

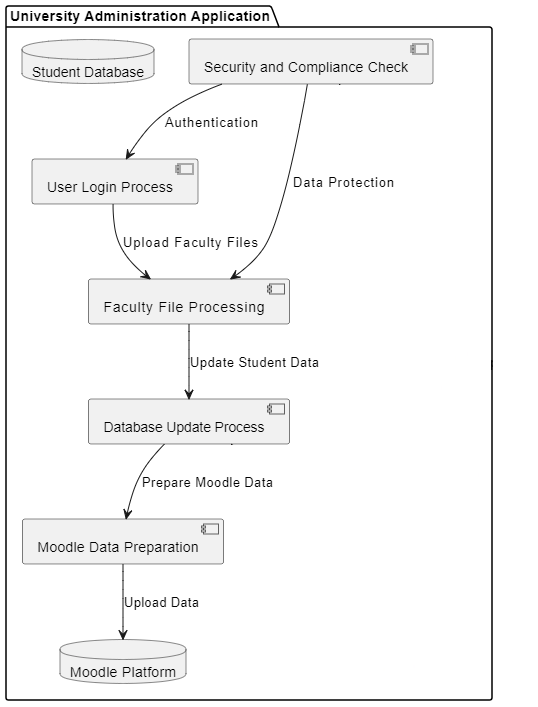
**Взаимодействия: Връзка между потребителите на Мудъл и Discord**

* + OAuth 2.0 за Moodle комуникира с Moodle API, за да обработва удостоверяването на потребителите.
  + Дискорд ботът комуникира с Discord API за управление на ролите.
  + Базата данни съхранява данни за потребителите, дискорд сървърите и връзката дискорд ид – студент ид.



### Уеб приложение

* Обработка на файловете за студентите
  + Този процес се задейства, когато администраторите качват файлове за студенти.
  + 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 файловете за студентите и одобрението им от старана на администратора на УИСС, се изпълнява процесът за актуализиране на базата данни.
  + Той включва интегриране на трансформираните данни в студентската база данни, като информацията за студентите се актуализира въз основа на обработените файлове.
  + Този процес се организира от компонента за интеграция.
* Подготовка на данните за Moodle
  + След като файловете бъдат одобрени за качване в Мудъл от администратора, то се пристъпва към тяхното качване и актуализиране на информацията.



## Изглед на данните

### 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:

**Таблици:**

1. **Students (Студенти)**

* Id: Уникален идентификатор на потребителя.
* Username: Това е факултетният номер на потребителя.
* FirstName: Името на потребителя.
* LastName: Фамилията на потребителя.
* Email: Имейлът на потребителя.

1. **Guilddata (Имената на Дискорд сървърите)**

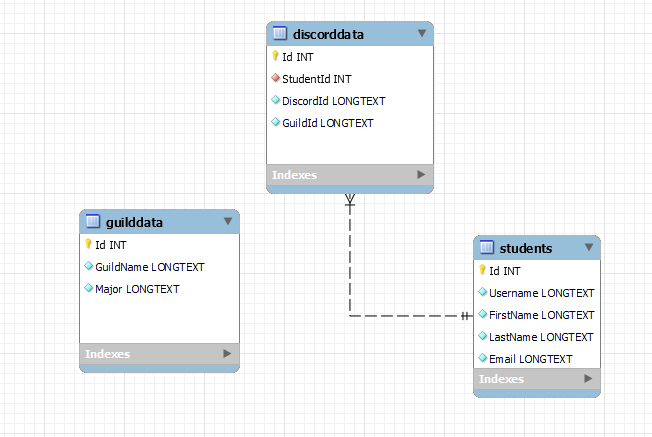
* Id: Уникален номер на сървъра в Дискорд.
* GuildName: Името на сървъра в Дискорд.
* Major: Специалността, която се взима от УИСС.

1. **Discorddata (Таблица, съдържаща информация за Дискорд ИД-тата на студентите и това в кой сървър са)**

* Id: Уникален номер на запис в таблицата.
* StudentId: Номер на студент от таблица Students.
* DiscordId: Номер на ИД-то на потребителя в Дискорд.
* GuildId: Номер на Дискорд сървър.

**Връзки между таблици:**

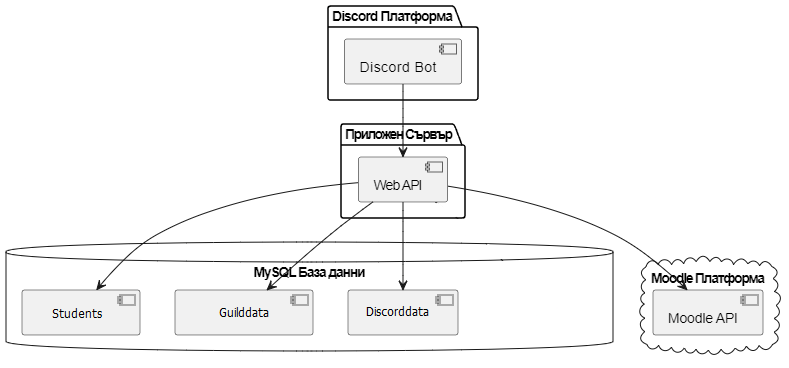
* students и discorddata: Отношение "едно към много".



## Изглед на внедряването

### 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.

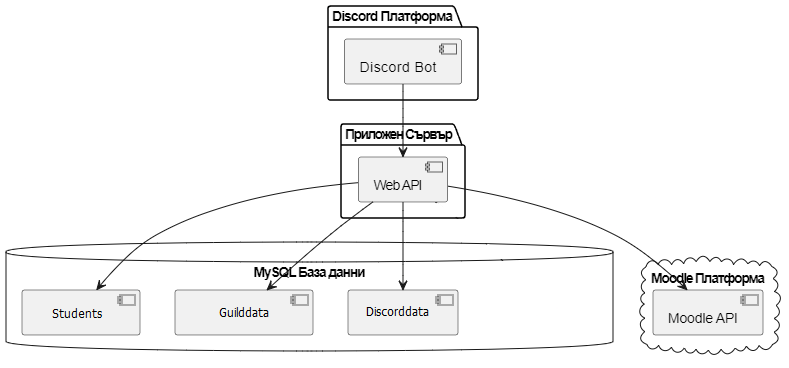
* Приложен Сървър (Application Server):
  + Тук се изпълнява главната логика на приложението и всички обработки.
  + Web API: Това е интерфейсът, който обработва командите, изпратени от дискорд бота. Чрез него се осъществява комуникацията между Discord, базата данни и УИСС. API-то може да извършва различни CRUD операции в базата данни или да изпраща и получава информация от Moodle.
* MySQL База данни:
  + Тук се съхраняват всички данни, свързани със системата – потребители и сървъри.
  + Students: Данни за потребителите, като техните факултетни номера, имена и имейл в ТУ.
  + Guilddata: Таблица, в която се изобразява отношението между името на сървъра в Дискорд и специалността, която се извлича от УИСС чрез HTML.
  + Discorddata: Свързва потребителите с техните Дискорд ИД-та и сървърите, в които са авторизирани. Позволява един потребител да има повече от един профил в даден сървър.
* Moodle Платформа
  + Moodle API: Това е интерфейсът, чрез който се взаимодейства с Moodle. Чрез него може да извличаме информация за потребителите, техните роли в Moodle и други свързани данни.

### Мудъл

* Уеб сървър - Този сървър ще бъде отговорен за хостване на приложението. Когато потребителите достъпват системата, уеб сървърът обработва 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 и сървърите, в които са се авторизирали. Компоненти: Таблиците "Students", "Guilddata", "Discorddata”.
  + Интеграционен слой - Осигурява комуникацията между Дискорд и Moodle, като използва Moodle 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: HTML страница, в която се качват файловете и се преглеждат от администратор.
  + Интернет и мрежова сигурност - Всички комуникации между различните компоненти са защитени и криптирани, особено когато данните се трансферират в Интернет.

## Изглед на имплементацията

### Дискорд бот

* Discord Bot (JavaScript):
  + 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 УИСС.
* 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:
  + Таблици:
    - Students: Съхранява потребителски данни – факултетен номер, имена и имейл.
    - Guilddata: Тук се съдържа информация за името на сървъра в Дискорд и името на специалността, която се взема от УИСС.
    - Discorddata: Свързва потребителите с техните Дискорд ИД-та и сървърите, в които са авторизирани. Позволява един потребител да има повече от един профил в сървър.
* Moodle API интеграция (C#):

### Мудъл

* 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 извлича 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 и авторизация.
* Файлово хранилище
  + Управление на файлове: Обработка и съхранение на качените файлове до тяхната обработка.

# Нефункционални изисквания

## Дискорд

### Достъпност (Availability)

* + Системата трябва да има висока наличност, за да обслужва потребителите без прекъсвания.

### Разширяемост (Scalability):

* + Системата е проектирана така, че да може да се справя с увеличаващ се брой потребители или заявки.

### 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.

### 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.

### 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.

### Използваемост (Usability):

* + Discord ботът трябва да има интуитивни команди и ясно описание на функционалността си.

## Мудъл

### 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.

### 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.

### 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.

### 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.

### 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.

### Интероперабилност (Interoperability)

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

### 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.