Обобщено съдържание на архитектурен проект

Проект „Интеграция на университетски информационни системи“

Автори: Георги Лозанов, Андриан Димитров, Лиляна Иванова, Стилиян Топалов, Иван Вълков, Иван Методиев
Дата:29.10.2023 г.

Въведение

Този документ представя архитектурната схема и изгледите на проекта за автоматизиране на ръчните дейности в процеса на поддържане на системите, свързани с обучителния процес във ФПМИ, Технически университет - София
Проектът има за цел да оптимизира управлението на данните за студентите и да подобри комуникацията с тях(Moodle и Discord), чрез използването на Discord бот..

Участници:

Административен персонал: Отговаря за управлението на софтуера и внасянето на данни и за поддръжката на информационната система на ТУ-София.
Инженери за разработка и поддръжка: Отговарят за разработването и поддържането на софтуера.

Дискорд бот: Автоматизира синхронизацията на данни между Moodle и Discord.

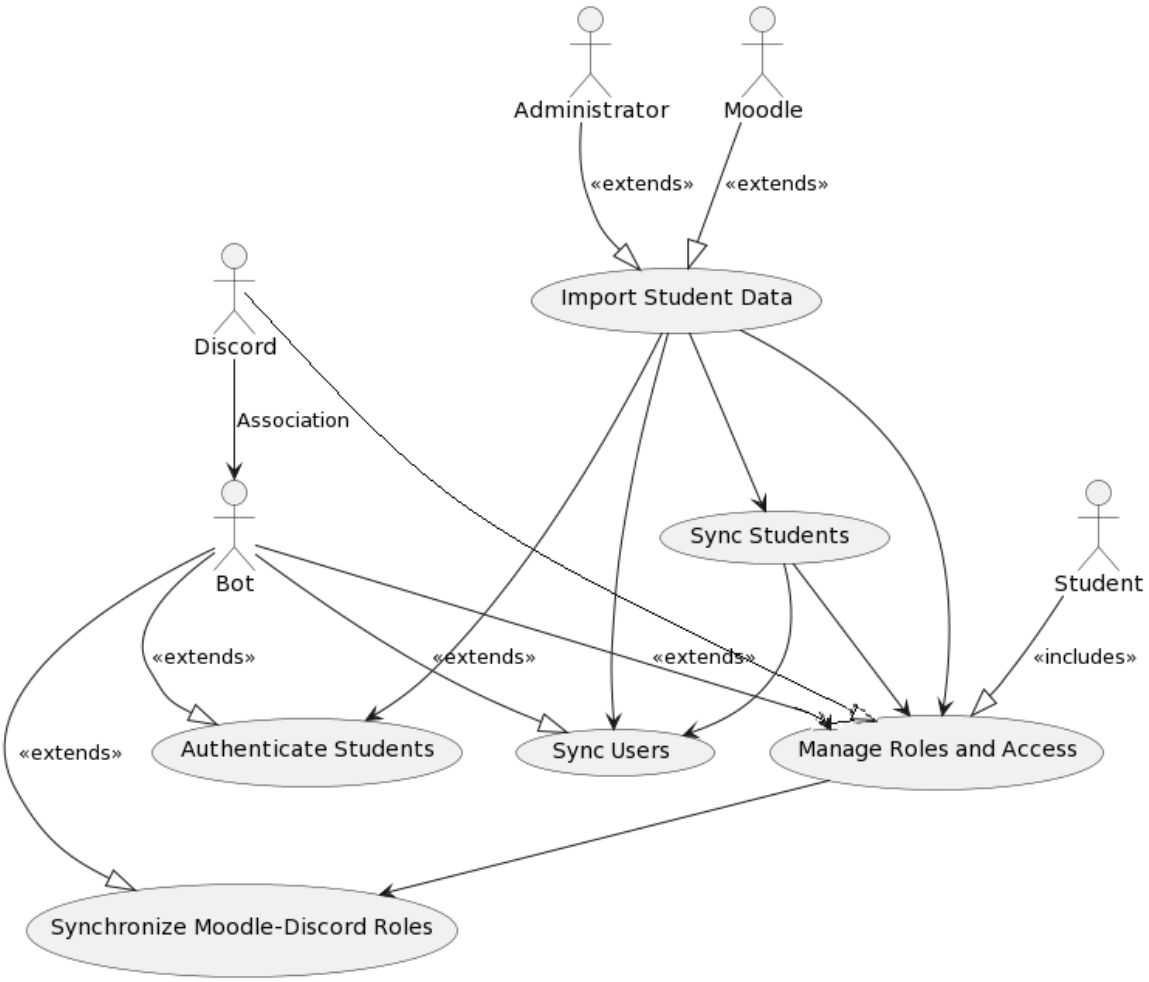
Поставя или променя роли на студентите в съответните discord сървъри.

Преподаватели: Имат достъп да правят промени по ролите на студентите (Discord) за предмети и други активности.

Студенти: Взаимодействат с Discord бота.

Архитектурните изгледи, представени в документа, описват различни аспекти на проекта, включително функционалност, структура, процеси, данни, внедряване и имплементация.

Use-case изглед:



Тази UML- Use Case Diagram илюстрира функционалността на софтуера от гледна точка на различните участници (актьори):

Администратор (Administrator):

Роля: Администраторът е отговорен за управлението на информационната система на ТУ-София и управлението на софтуера, който осигурява връзката между Moodle и Discord. Също така той има пълен достъп до функционалността на софтуера.

Възможни сценарии:

Импортира данни за студенти (Import Student Data) в системата на ТУ-София, което чрез софтуера води до синхронизация на студентите между Moodle и Discord.

Също така може да управлява ролите и достъпа на студентите (Manage Roles and Access) в Discord.

Може да инициира синхронизацията на ролите и достъпа (Synchronize Moodle-Discord Roles) чрез бота.

Мудъл (Moodle):

Роля: Представлява информационната система Moodle, където се съдържат данни за студентите и курсовете.

Възможни сценарии:

Изпраща данни за студентите към софтуера (Import Student Data).

Участва в синхронизацията на потребителите с Discord (Sync Users).

Дискорд (Discord):

Роля: Представлява платформата Discord, където студентите комуникират с преподавателите..

Възможни сценарии:

Изпраща информация за потребителите към софтуера (Import Student Data).

Участва в синхронизацията на потребителите с Moodle (Sync Users).

Дискорд Бот (Bot):

Роля: Представлява бот, който автоматизира синхронизацията между Moodle и Discord и управлява ролите на студентите в съответните Discord сървъри.

Възможни сценарии:

Изпълнява автентикация на студентите (Authenticate Students) в Discord, свързвайки ги с техните профили в Moodle.

Участва в синхронизацията на студентите (Sync Students) между Moodle и Discord.

Управлява ролите и достъпа на студентите в Discord (Manage Roles and Access) в съответствие с техните курсове и роли.

Изпълнява синхронизация на ролите и достъпа между Moodle и Discord (Synchronize Moodle-Discord Roles).

Студент (Student):

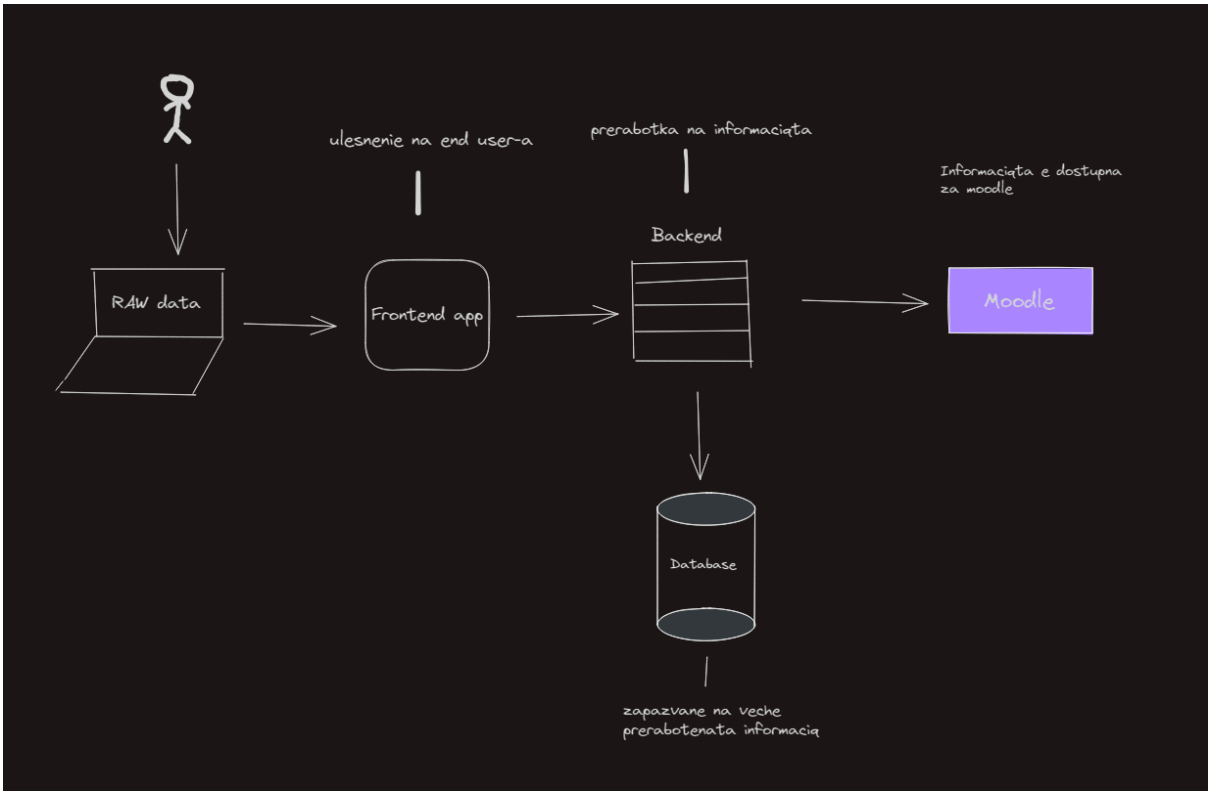
Роля: Представлява студентите, които използват платформата Discord и Moodle..

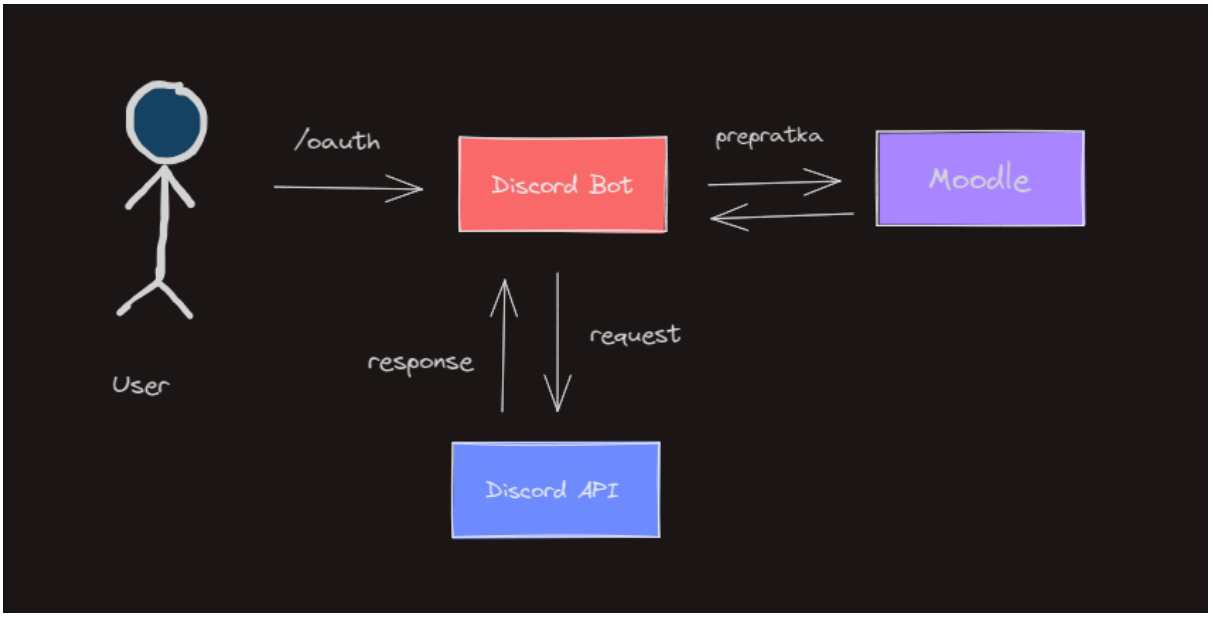
Възможни сценарии:

Участва в управлението на ролите и достъпа с бота (Manage Roles and Access), което участие включва получаване на роли за съответния курс или премахване от роли, ако този студент не е в съответния курс (прекъснал е обучението си или се е преместил в друг поток)..

Диаграмата илюстрира връзките и взаимодействията между различните участници в системата и демонстрира как всеки от тях допринася за функционирането на софтуера за управление на информационните системи и комуникацията със студентите в Discord сървъра.

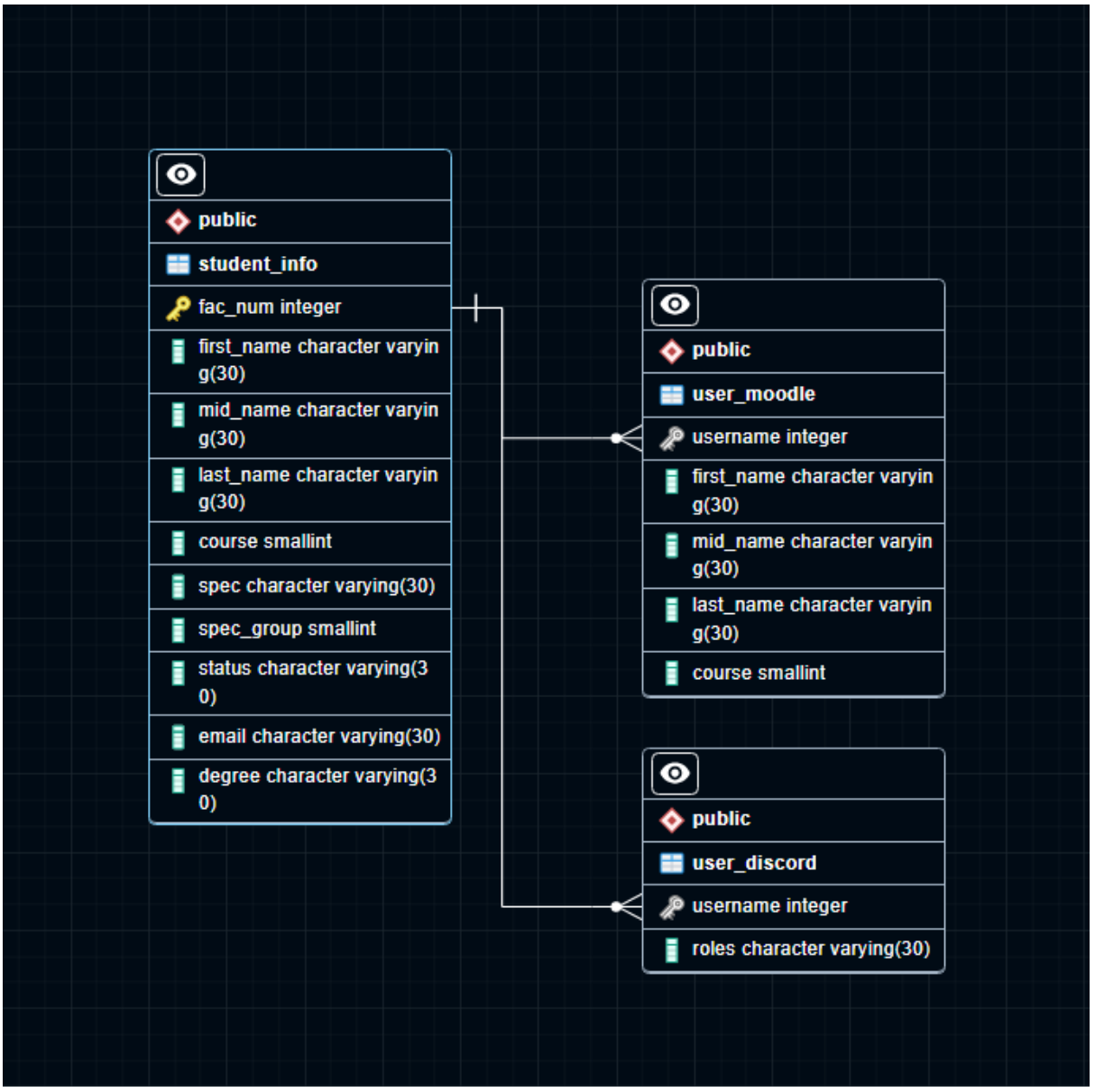
Логически изглед:



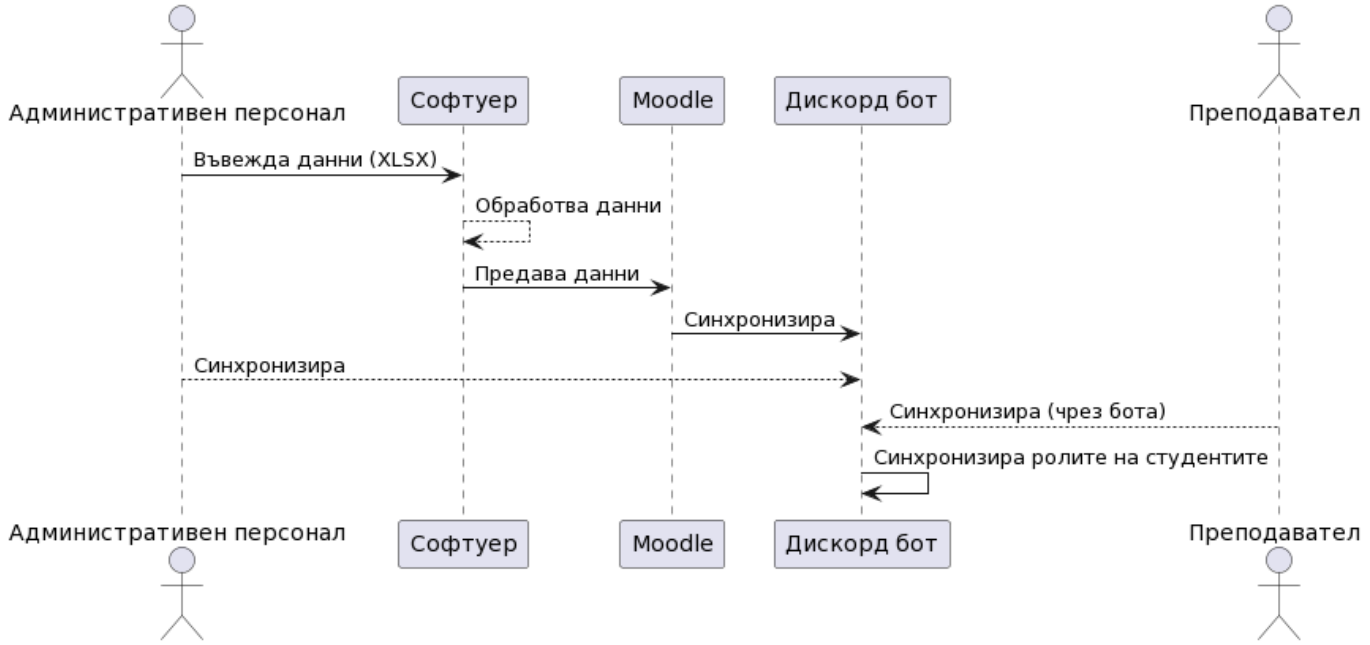


Двете диаграми представят връзките между компонентите в проекта. Първата диаграма представлява комуникацията на компонентите и потокът на действие на първата част. Втората диаграма представя комуникацията на потребителя с дискорд бота.

Изглед на данните:



Процесен изглед:



Административен персонал: Представлява администраторите или други служители, които въвеждат информацията от ТУ-София във формат XLSX. Този файл съдържа данни за студентите.

Софтуер: След като информацията бъде въведена от административния персонал, софтуерът извършва обработка на данните, преобразувайки ги в подходящ формат и генерирайки XLSX файл, който съдържа актуални данни.

Moodle: информационната система, където се съхраняват данните за студентите и курсовете. В този процес, софтуерът предава данните към Moodle, което включва актуализиране на информацията в Moodle.

Дискорд бот: Дискорд бота, който извършва синхронизация между Discord и Moodle. Ботът може да бъде иницииран чрез командите /auth и /sync.

Преподавател: Преподавателите също имат права да стартират синхронизацията между Discord и Moodle, като командата /sync.

Процесът на синхронизация между Discord и Moodle включва две стъпки:

Първо, се извършва синхронизация на потребителите между Moodle и Discord, като информацията и ролите се синхронизират между двете системи.

Второ, ботът синхронизира ролите на студентите в Discord, като настройва достъпи и роли, в зависимост от тяхната регистрация в курсовете в Moodle.

Тази диаграма представя процеса на събиране, обработка и разпределение на данни в контекста на образователна система и комуникация между различни софтуерни среди и актьори.

Изглед на внедряването :

Внедряването на системата изисква познанието за няколко конкретни хардуерни изисквания, както и софтуерни.

За хардуерната спецификация се изисква сървърна инфраструктура, която да играе като главен шаблон за съхранените данни на студентите, биде тя тип cloud или self-hosted услуга. Определената сървърна инфраструктура трябва да спазва правилните изисквания за сигурност при обмяна на данните/комуникация (сертификати от правилно CA, TLS, т.н.). Освен това трябва да се предостави, по възможност дистрибутирана, клъстеризирана, поддържаща автоматични бекъпи и миграции, с конфигурирана скалируемост на база стандартите за разработка на бази данни. Тъй като системата трябва да поддържа ролеви спецификации (ученици, преподаватели, т.н.), възможна наклонност е към релационна база данни, а при нея трябва да се осигури правилно индексиране на данните. Има нужда от хост за самия дискорд бот различен от екипен компютър, като при избора трябва също да се взимат предвид мерките за сигурност на обмена на данните.

Софтуерно се налага правилна тестова среда: тестов дискорд сървър, на който да се симулира действието на бота. Също така и тестова среда на сървърните данни, както и автоматизация на тези тестови процеси в системите на база unit и integration тестове. Дистрибутирането на приложенията за бота и сървъра може да стане на база виртуализация и/или контейнеризация чрез Docker и/или Kubernetes.

Изглед на имплементацията :

Архитектурна концепция:

Проектът за синхронизация на данни между Moodle, Discord и Дискорд бота се базира на многокомпонентна архитектура, която включва следните ключови компоненти:

Административен интерфейс: Интерфейсът, който се използва от административния персонал за въвеждане на данни и управление на процеса.

Сървърен софтуер: Сървърният софтуер, който обработва и обновява данните в системата на ТУ-София и Moodle.

Бот за Дискорд: Дискорд ботът, който осигурява комуникацията и синхронизацията между Discord и Moodle. Той е отговорен за управлението на ролите на студентите в Discord сървърите.

Слоева архитектура:

Имплементацията на проекта използва слоева архитектура, която поддържа ясно разделяне на отговорности и лесна поддръжка. Слоевете включват следните:

Потребителски интерфейс: Този слой представлява интерфейса, използван от административния персонал и преподавателите, за въвеждане на данни и управление на синхронизацията.

Бизнес логика: Тук се намират бизнес правилата и логиката на проекта. Този слой управлява обработката на данните, актуализацията на Moodle и управлението на бота за Дискорд.

Сървърна инфраструктура: Този слой представлява сървърния софтуер, който свързва сървърната инфраструктура на ТУ-София и Moodle.

Дискорд интеграция: Този слой включва всичко, свързано с Дискорд бота и синхронизацията с Discord сървърите.

Общи правила за реализация:

Всички данни се обработват и предават в структуриран формат, който е лесен за обработка и анализ.

Използват се сигурни методи за връзка и обработка на данни, за да се поддържи поверителността и цялостността на информацията.

Комуникацията между компонентите следва стандартни процедури и протоколи за да се осигури съвместимост и надеждност.

Изгледът на имплементацията осигурява ясно представяне на структурата на проекта, позволявайки на разработчиците да разберат и поддържат кода по-ефективно и да гарантират неговата правилна работа.

Нефункционални изисквания

Тук се описва как избраната архитектура реализира атрибутите на качеството чрез употреба на различни тактики.

В зависимост от реализацията могат да се разгледат атрибути на качеството като:

1. **Достъпност**

Достъпността е много важно изискване за дадения проект:

Откриване на проблеми: Ако сървърът на Discord се изключи или ботът спре работа и не може да извършва своите функции, системата трябва да бъде в

състояние да засече тези проблеми.

Възстановяване: Ако възникне проблем и системата не може да работи автоматизирано(Например, ако ботът спре работа, трябва да се рестартира системата). Трябва да може функционалността, която се извършва от Дискорд бота да може да се извършва и ръчно от потребители с такава роля. Трябва да се правят резервни копия на данните, което ще позволи да възстанови информацията в случай на смущения или загуба на данни.

Предотвратяване на проблеми:

-Сигурност: Осигуряване на защита на данните и предотвратяване на неоторизиран достъп.

-Тестване и проверки: Извършване на тестове,, за да се идентифицират и коригират проблеми преди те да се случат в реална ситуация.

2. Разширяемост

Как можем да постигнем разширяемост на един дискорд бот?
Можем да добавим нови команди или функции, които да бъдат от полза на потребителите. Също ако искаме да предоставим нови типове информация или услуги, можете да интегрираме бота с различни външни API (например, за новини и други). Можем проектираме бота да поддържа плъгини или модули, които могат да бъдат добавяни и премахвани по желание, за да се разширява функционалността му.

Поддръжката на бота също може да се счита като разширяемост, понеже той трябва да се актуализира ежедневно, за да сме сигурни, че работи с най-новата версия на дискорд.

3. Сигурност

Контрол на достъпа:

- Идентификация на потребителите и удостоверяване.
- Авторизиране на потребителите в зависимост от техните роли и отговорности.

- Ограничаване на достъпа в съответствие с дефинирани политики и роли.

Резервни копия:

- Редовни резервни копия на важни данни и конфигурации.
- Съхранение на резервни копия

Защита срещу атаки:

- Бърза реакция и възстановяване след инциденти.

Криптография:

- Криптография за защита на данни при предаване и съхранение.

Сигурността на системата, където се съхраняват данни за студенти, е от изключителна важност. Тя предпазва информацията от неоторизиран достъп и злоупотреби, поддържа поверителността и целостта на студентските данни и намалява риска от нарушения.

4. Възможност за тестване

Стрес теста на софтуера ще се проведе на етапи.

1. Преглед на софтуера и неговите функции
2. Установяване на ресурси
3. Обхват на тестване

1. Тестването ще протече по два начина, първо автоматично и след това ръчно, за да се провери дали всичко е коректно и работи по задание.

2. Тестовия обхват ще се състои от функционални тестове и тестове на изпълнението на софтуера. Функционалните тестове ще проверяват дали приложението изпълнява коректно определени функционалности. Тестовете за изпълнение от своя страна ще бъдат фокусирани върху производителността и реакцията на приложението при различни натоварвания.

5. Интероперабилност

Анализът на входно-изходните канали на информация за проекта е важна част от архитектурното проектиране и планирането на информационни системи. В случая на проекта с информационна студентска система, която използва Moodle и Discord, входните и изходните канали на информация играят ключова роля в събирането, обработката и предоставянето на информация на различните потребители. Ето анализ на тези канали:

Входни данни от административния персонал: Началото на информационния поток е въвеждането на данни за студентите в системата на ТУ-София. Административният персонал предоставя тези данни като XLSX файл. Този файл съдържа информация за студентите, техните идентификационни данни, курсове и специалности.

Обработка от софтуера: След приемане на XLSX файла, софтуерът обработва данните, включително техните структури и формати. Това включва проверка на данните, преобразуване на форматите и създаване на структури, които са пригодни за синхронизация с Moodle и Discord.

Предаване на данни към Moodle: Обработените данни се предават към Moodle, информационната система, където се съдържат данните за студентите и курсовете. Това представлява входен канал за Moodle, който приема информацията и я съхранява в собствената си база данни.

Синхронизация между Moodle и Discord: След като данните са прехвърлени към Moodle, входният канал за синхронизация между Moodle и Discord се отваря. Discord ботът и участниците като административният персонал и преподавателите могат да стартират синхронизация на потребителите и ролите между двете платформи. Този канал включва и входната команда "/auth" и "/sync," чрез които се инициира синхронизацията.

Синхронизация на ролите на студентите в Discord: Данните, свързани с ролите и достъпа на студентите в Discord, представляват изходни данни за

бота и участниците с права. Този канал позволява на бота да променя ролите на студентите в съответните Discord сървъри, като гарантира, че те отразяват техния статус и курс.

Информация за състоянието на синхронизацията и проблеми:
Входно-изходният канал включва информация за състоянието на синхронизацията, както и детайли за възникнали проблеми. Този канал може да бъде използван за мониторинг и управление на синхронизационните процеси.

6. Използваемост

Използваемостта на продукта се явява в това колко лесно се използва от ученик, учител и администратор. Като ученик е важно да може разбираемо да свърже Moodle данните си с тези на Discord, да получава известие за получени роли в дискорд, т.н. За учител или администратор - важно е лесно да въвежда входната информация за студентите в системата. Използването на лесни команди с кратки и описателни имена ще е основно изискване за бота.