

СОФИЙСКИ УНИВЕРСИТЕТ
„СВ. КЛИМЕНТ ОХРИДСКИ“

ФАКУЛТЕТ ПО
МАТЕМАТИКА И ИНФОРМАТИКА



SOFIA UNIVERSITY
ST. KLIMENT OHRIDSKI

FACULTY OF
MATHEMATICS AND INFORMATICS

Проект

по

Софтуерни архитектури и разработка на софтуер

на тема

Информационна система за публикуване на конкурси и подаване на оферти за участие

Изготвен от:

Надежда Панделиева, 62468

Станислава Иванова, 62463

Съдържание

1. Въведение	4
1.1 Обща информация за текущия документ	4
1.1.1 Предназначение на документа	4
1.1.2 Описание на използваните структури на архитектурата	4
1.1.2.1 Декомпозиция на модулите	4
1.1.2.2 Структура на процесите	4
1.1.2.3 Структура на внедряването	4
1.1.3 Структура на документа	5
1.2 Общи сведения за системата	5
1.3 Терминологичен речник	5
2. Декомпозиция на модулите	5
2.1 Общ вид на декомпозиция на модули за системата	5
2.2 Контекстна диаграма	6
2.3 Потребител	7
2.3.1 Предназначение на модула	7
2.3.2 Основни отговорности на модула	8
2.3.3 Описание на интерфейсите на модула	8
2.3.3.1 Регистрация	8
2.3.3.2 Логин	8
2.3.3.3 Профили	9
2.3.3.4 Връзка със сървър	11
2.4 Сървър	12
2.4.1 Предназначение на модула	12
2.4.2 Основни отговорности на модула	12
2.4.3 Описание на интерфейсите на модула	13
2.4.3.1 Административни дейности	13
2.4.3.2 Сигурност	13
2.4.3.3 Нотификация по имейл	14
2.4.3.4 Връзка с потребител	14
2.4.3.5 Връзка с база данни	15
2.4.3.6 Връзка с НАП, търговски регистър	15
2.5 База данни	16
2.5.1 Предназначение на модула	16
2.5.2 Основни отговорности на модула	16
2.5.3 Описание на интерфейсите на модула	16
3. Описание на допълнителните структури	17
3.1 Структура на процесите	17
3.1.1 Първично представяне	17
3.1.2 Описание на елементите и връзките	20
3.1.3 Описание на обкръжението	20

3.1.4 Описание на възможните вариации	20
3.2 Структура на внедряването	21
3.2.1 Първично представяне	21
3.2.2 Описание на елементите и връзките	21
3.2.3 Описание на обкръжението	22
3.2.4 Описание на възможните вариации	22
4. Архитектурна обосновка	22
4.1 Архитектурен стил	22
4.2 Архитектурни драйвери	22

1. Въведение

1.1 Обща информация за текущия документ

1.1.1 Предназначение на документа

Документът цели да представи архитектурата на система за публикуване на състезания и конкурси и подаване на оферти за участие. Полезен е за всички заинтересовани лица – както разработчиците на системата, така и за клиентите ѝ.

1.1.2 Описание на използваните структури на архитектурата

1.1.2.1 Декомпозиция на модулите

Тази структура показва разделянето на системата на модули и подмодули, както и съответните връзки между тях. Основните модули са Потребител, Сървър, База данни. Модулът “Потребител” предоставя потребителския интерфейс. “Сървър” съдържа логиката за функционалностите на системата и обособява тези, които са достъпни само за администраторите. Също така осъществява и връзката с външните услуги. Модулът “База данни” съхранява данните на системата.

1.1.2.2 Структура на процесите

Структурата на процесите показва основните процеси в системата и как те се свързват помежду си чрез комуникационни канали. Тя би била полезна за определянето на производителността на системата, защото чрез нея ще може да се открива къде има нужда от подобрение в комуникацията. Също така, тази структура ще е от полза за разработчиците, QA и тестерите в осигуряването на по-голяма надеждност и сигурност. Може да служи и като основа за разработката на in-app tutorial, който да представи на потребителя начина на използване на системата.

1.1.2.3 Структура на внедряването

Структурата на внедряването показва как софтуерът се разполага върху хардуера, връзките с външните системи и определя необходимите за реализацията на системата хардуерни ресурси. Всеки модул от декомпозицията е разпределен на различно устройство, като някои от модулите са репликирани на допълнителни хардуерни устройства. Може да бъде полезна за всички заинтересовани лица.

1.1.3 Структура на документа

Документът се разделя на 4 секции:

Секция 1: Въведение - обща информация за документа и системата.

Секция 2: Представяне на структурата “Декомпозиция на модулите” и описание на всеки от модулите.

Секция 3: Представяне на допълнителните структури - “Структура на процесите” и “Структура на внедряването”.

Секция 4: Архитектурна обосновка - архитектурен стил и драйвери.

1.2 Общи сведения за системата

Системата е предназначена за публикуване на конкурси и състезания, които включват поръчки с точно определени условия, и съответно за подаване на оферти от желаещи да кандидатстват за изпълнители по обявите. Предоставя възможност за безплатно регистриране на профил, лесно електронно подаване на документи за участие, публикуване на обяви за публични конкурси и стипендии. Всяка дейност се проверява, за да може да бъде гарантирана сигурност. При голяма натовареност системата има за цел да осигури висока надеждност и изправност така, че да не се допусне срив и обработката на заявки да не повлияе на взаимодействието с потребителите.

1.3 Терминологичен речник

- *ЕИК (Единен идентификационен код)* - уникален идентификационен номер, задължителен за търговците, вписани в търговския регистър.
- *In-app tutorial* - ръководство, предназначено за нови потребители или хора, които имат нужда да се запознаят с работата на дадена система.

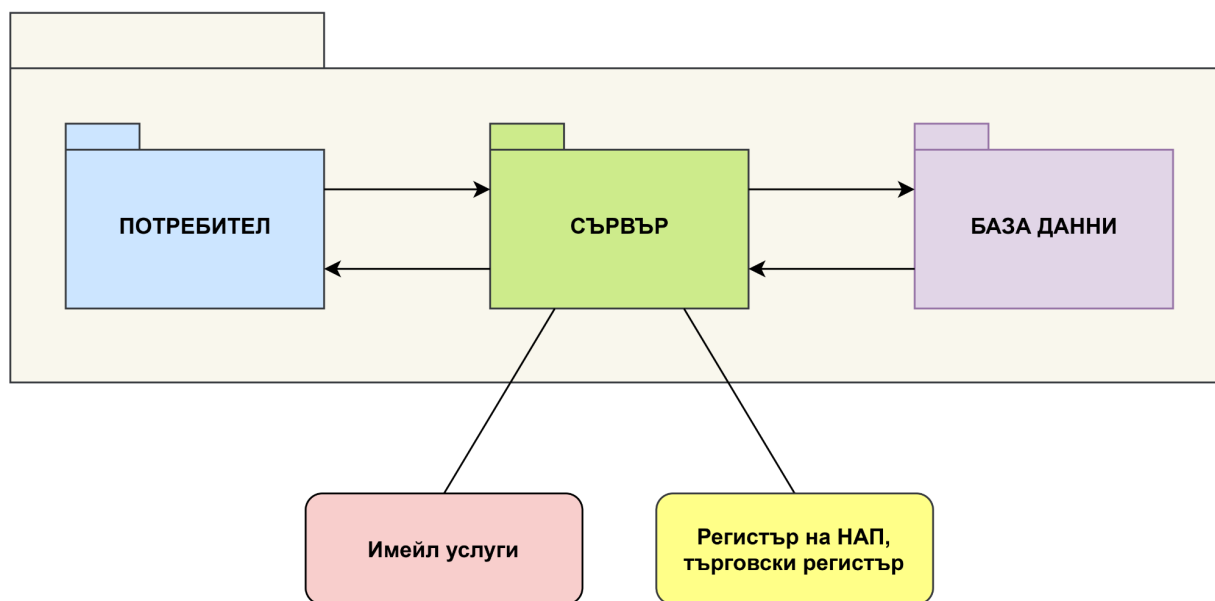
2. Декомпозиция на модулите

2.1 Общ вид на декомпозиция на модули за системата



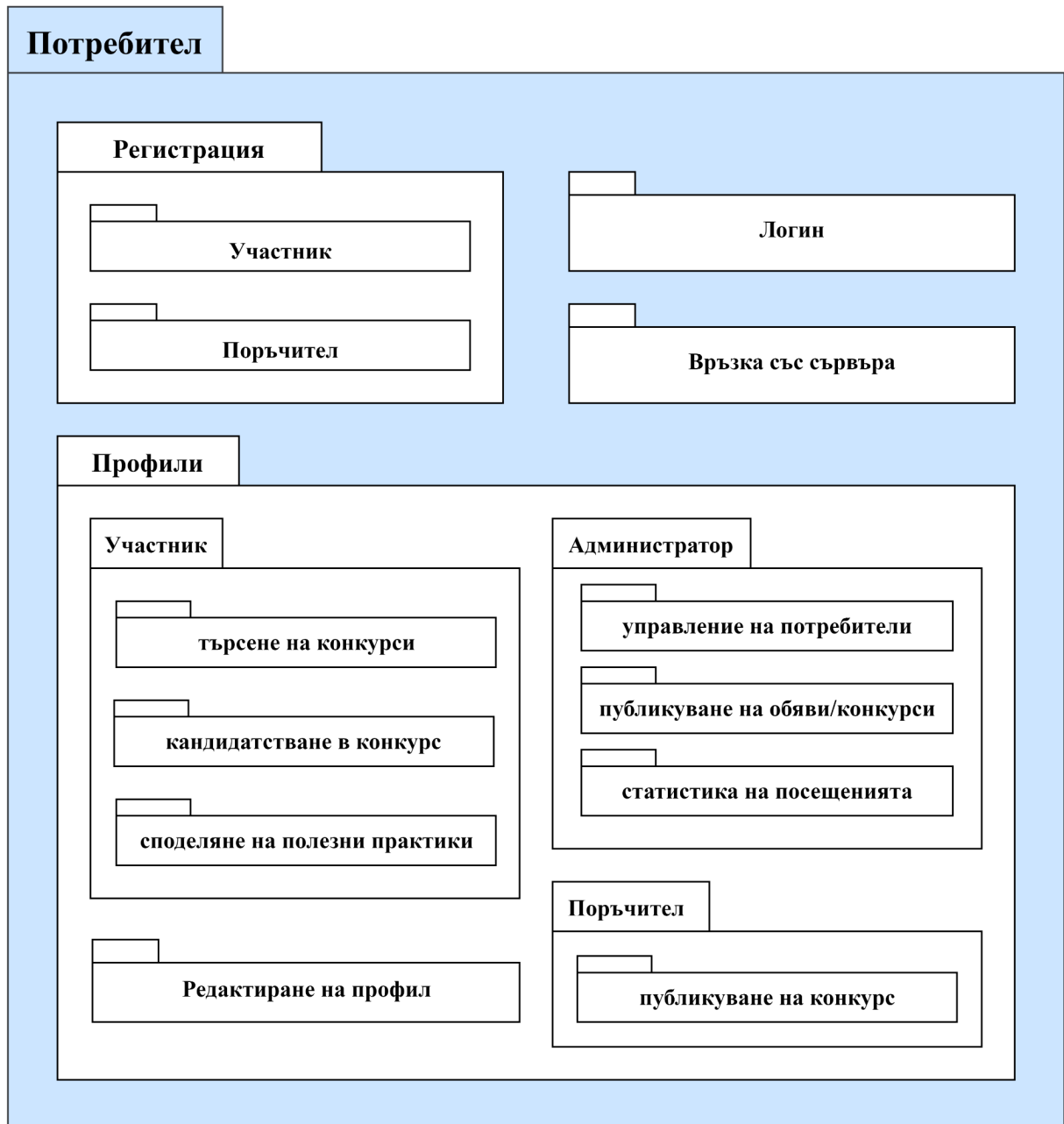
Системата е разделена на модули, като всеки модул представлява отделно логическо ядро. Представена е връзката между различните модули, т.е. как всеки от тях си комуникира с останалите. Всеки модул се състои от подмодули и класове, които отговарят за изпълнението на различни задачи в системата.

2.2 Контекстна диаграма



Диаграмата показва комуникацията на модулите с различни външни системи, т.е. как се осъществява връзката на системата с външния свят.

2.3 Потребител



2.3.1 Предназначение на модула

Модулът контролира взаимодействието на системата с нейните потребители посредством различни интерфейси на функционалностите, които зависят от ролята на потребителите в системата.

2.3.2 Основни отговорности на модула

Модулът има за цел да представи интерфейс за потребителите и техните права като осигурява качествена връзка със сървъра на системата.

2.3.3 Описание на интерфейсите на модула

2.3.3.1 Регистрация

(администраторите имат генерирани имейл и парола от системата, те нямат нужда от регистрация)

- **Участник**

- **RegisterAsApplicant(email: string, password: string)**

Входни данни: имейл, парола

Изходни данни: съобщение за успешно създаден профил

Грешки и изключения: при невалидни данни се изкарва съобщение за съответната грешка и се дава възможност за повторно въвеждане

Зависимост от други елементи: данните се проверяват от администратора и се изпращат до сървъра, който от своя страна ги записва в базата данни

- **Поръчител**

- **RegisterAsGuarantor(email: string, password: string, EIK: string)**

Входни данни: имейл, парола, ЕИК

Изходни данни: съобщение за успешно създаден профил

Грешки и изключения: при невалидни данни се изкарва съобщение за съответната грешка и се дава възможност за повторно въвеждане

Зависимост от други елементи: данните се проверяват от администратора (ЕИК се проверява чрез заявка към външните за системата търговски регистри) и се изпращат до сървъра, който от своя страна ги записва в базата данни

2.3.3.2 Логин

- **Login(email: string, password: string, EIK: string)**

Входни данни: имейл, парола и при необходимост ЕИК

Изходни данни: съобщение за успешно влизане в системата

Грешки и изключения: влизането в системата се отхвърля при въведени невалидни данни

Зависимост от други елементи: "Връзка със сървъра"

Пример за употреба: Потребител въвежда имейл и парола (и ЕИК, ако е поръчител). Те се изпращат към сървър, който изпраща заявка към базата данни да провери дали информацията за профила е валидна. Ако данните са коректни, се изпраща съобщение за успешно влизане в системата.

2.3.3.3 Профили

- **Участник**

- **Търсене на конкурси**

- **FindContest(keyword: string)**

- Входни данни:** ключова дума/критерии

- Изходни данни:** извеждане на конкурс според търсена дума

- Грешки и изключения:** при невалиден идентификатор се изкарва съобщение за грешка

- Зависимост от други елементи:** “Връзка със сървър”

- Пример за употреба:** В полето за търсене се въвежда желаната ключова дума или категория, по която да се търси конкурс, и при съвпадение се извежда търсената информация.

- **Кандидатстване в конкурс**

- **ApplyForContest(fileName: string)**

- Входни данни:** документ в PDF формат с нужната информация, свързана с кандидатурата

- Изходни данни:** успешно подадени документи, чиито данни се записват в базата

- Грешки и изключения:** при несъответствие с правилата на конкурса или неправилно попълнени данни, се извежда съобщение за грешка

- Зависимост от други елементи:** “Връзка със сървър”

- Пример за употреба:** Участник изпраща заявка за кандидатстване, която се изпраща към сървър, където се проверява дали подадената информацията е валидна според изискванията на конкурса. Това става посредством заявка към базата данни. Ако данните са коректни, се изпраща съобщение за успешно добавена кандидатура в системата.

- **Success story**

- **PublishStory(story: vector<string>)**
Входни данни: история от потребител
Изходни данни: успешно публикувана история

- **Поръчител**

- Публикуване на конкурс

- **AddContest(content: vector<string>)**
Входни данни: съдържание и изисквания на конкурса
Изходни данни: успешно публикуван конкурс
Ограничения при употреба: публикуването се случва само след проверката и одобрението на администратора за изрядност на данните
Зависимост от други елементи: “Връзка със сървъра”
Грешки и изключения: не са спазени изискванията за формалност

- **Администратор**

- Управление на потребители

- **ManageProfiles(userInfo: vector<string>)**
Входни данни: приемат се данните на потребителите за проверка
Изходни данни: успешно добавена или обновена информация
Ограничения при употреба: изпраща се заявка към сървъра за проверка на данните
Зависимост от други елементи: “Връзка със сървъра”

- Публикуване на обяви/конкурси

- **PublishContests(content: vector<string>)**
Входни данни: конкурс от поръчител или външна система
Изходни данни: добавена обява към системата
Ограничения при употреба: изпраща се заявка към сървъра за проверка на данните
Зависимост от други елементи: “Връзка със сървъра”

- Статистика на посещенията

- **ShowStatistics(filter: string)**
Входни данни: заявка за статистика по показател
Изходни данни: представяне на желаната информация
Зависимост от други елементи: “Профили”, “Връзка със сървъра”

- Редактиране на профил

- **EditProfile(newInfo: vector<string>)**

- Входни данни:** нова информация, която трябва да се запише

- Изходни данни:** съобщение за успешно направени промени

- Грешки и изключения:** при невалидна нова информация се извежда съобщение за съответната грешка

- Зависимост от други елементи:** администраторът проверява валидността на подадените параметри и записва новите данни за дадения потребител в базата данни

2.3.3.4 Връзка със сървъра

- **SendQuery(query: Query)**

- Входни данни:** заявка, която се препраща към сървъра

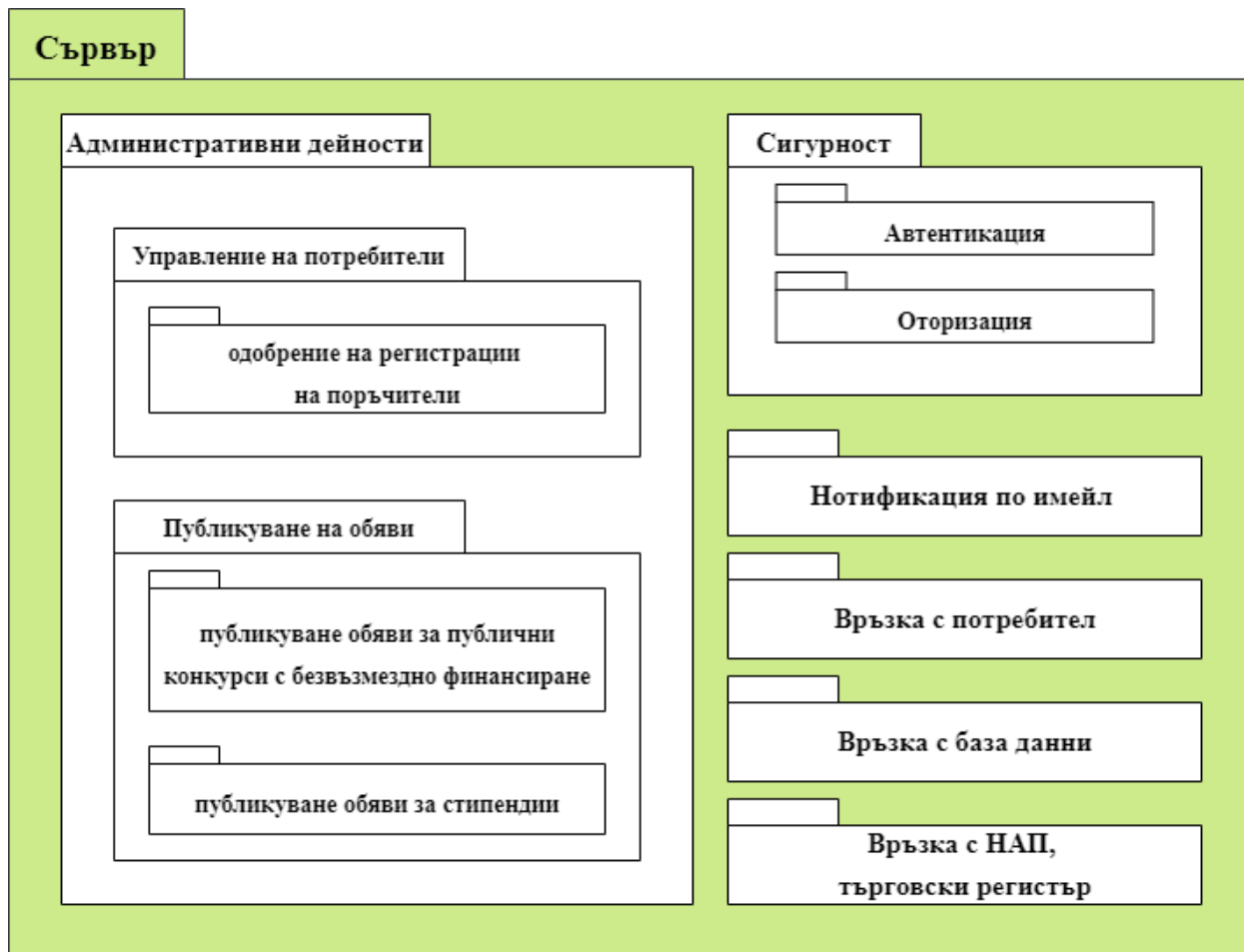
- Изходни данни:** отговор на съответната заявка

- Грешки и изключения:** при неналичност на сървъра се връща съобщение за грешка

- Зависимост от други елементи:** подмодулите на "Потребител", „Сървър“

- Пример за употреба:** Получава заявка от даден модул и я препраща към сървъра. След като се обработи, връща отговора към съответния модул, който е изпратил заявката.

2.4 Сървър



2.4.1 Предназначение на модула

Модулът съдържа логиката за функционалностите на системата, осигурява нейната сигурност и осъществява връзката с останалите модули и базата данни. Това се реализира чрез приемане на заявки за обработване от отделните модули, а след това се връща отговор след запитване до базата данни.

2.4.2 Основни отговорности на модула

Модулът трябва да предоставя основни функции като защита на данните, които да бъдат предпазени от недобронамерен достъп, да се свързва с базата данни и външни услуги, за да получава нужната информация.

2.4.3 Описание на интерфейсите на модула

2.4.3.1 Административни дейности

- **Управление на потребители**
 - **Одобрение на регистрации на поръчители**
 - **ValidateGuarantor(EIK: string)**
Входни данни: ЕИК на поръчител
Изходни данни: съобщение за успешно/неуспешно одобрение на съответния поръчител, което се изпраща към “Връзка с потребител”
Зависимост от други елементи: “Връзка с НАП, търговски регистър”, “Връзка с базата данни”, „Връзка с потребител“
Пример за употреба: Получава се ЕИК на поръчител. След това се изпраща заявка към “Връзка с НАП, търговски регистър”, за да се потвърди състоятелността му. Ако бъде открит, се изпраща още една заявка към “Връзка с базата данни”, за да се провери дали даденото лице не се е регистрирало вече. При въведен невалиден ЕИК се прекъсва процеса на регистрация на поръчителя.
- **Публикуване на обяви**
 - **Публикуване обяви за публични конкурси с безвъзмездно финансиране**
 - **PublishContests(content: vector<string>)**
Входни данни: обява за публичен конкурс
Изходни данни: добавена обява към системата
 - **Публикуване обяви за стипендии**
 - **PublishScholarshipAd(content: vector<string>)**
Входни данни: обява за стипендии
Изходни данни: добавена обява към системата

2.4.3.2 Сигурност

- **Автентикация**
 - **AuthenticateUser(email: string, password: string, EIK: string)**
Входни данни: имейл, парола и при необходимост ЕИК.

Исходни данни: изпраща се съобщение към „Връзка с потребител“ като отговор за това дали потребителят е този, за когото се представя

Зависимост от други елементи: „Връзка с база данни“, „Оторизация“

Пример за употреба: Получава се информация за даден профил, изпраща се заявка към базата данни с цел да се провери дали потребителят е този, за когото се представя. Ако данните са коректни, то се предават за оторизиране.

- **Оторизация**

- **AuthorizeUser(email: string, password: string)**

Входни данни: получават се данните на профил, чиито права трябва да се проверят

Исходни данни: списък с правата на потребителя на системата.

Зависимост от други елементи: „Автентикация“, „Връзка с потребител“, „Връзка с база данни“

Пример за употреба: Получава се информация за профила, чиито права трябва да се определят. За тази цел се изпраща заявка до базата данни. Полученият резултат се препраща към модула „Потребител“ чрез „Връзка с потребител“.

2.4.3.3 Нотификация по имейл

- **NotifyByEmail(email: string, message: string)**

Входни данни: администраторите получават по имейл съобщение за грешка в системата

Исходни данни: връща се отговор за неутрализиране на проблема

Зависимост от други елементи: външния източник „Имейл услуги“ и всички модули и подмодули в системата

Пример за употреба: При настъпване на критични събития в системата администраторите получават имейл, с който се известяват, че има грешка или загуба на връзка с някои от модулите, при което те трябва да се заемат и възможно най-бързо да се справят с настъпилния проблем.

2.4.3.4 Връзка с потребител

- **ConnectWithUser(query: Query)**

Входни данни: заявка към някой подмодул от сървъра, както и информация от кой подмодул е подадена

Изходни данни: отговор на заявката, който се изпраща до модула

Грешки и изключения: процесът се прекратява при невъзможност за свързване с потребителя

Зависимост от други елементи: „Връзка със сървъра“ от „Потребител“

Пример за употреба: Получава се заявка от потребител, която се препраща към съответния модул, който трябва да я обработи. След това се връща отговор обратно към „Връзка със сървъра“.

2.4.3.5 Връзка с база данни

- **ConnectWithDatabase(query: Query)**

Входни данни: заявка, която се препраща към базата данни, както и информация от кой подмодул е подадена

Изходни данни: отговор на заявката, който се изпраща до модула

Грешки и изключения: изчакване при неналичност на базата данни

Пример за употреба: Получава се заявка и се препраща към базата данни, за да се обработи. Връща се отговор и се препраща към модула, който я е изпратил.

2.4.3.6 Връзка с НАП, търговски регистър

- **ConnectWithExternalSystems(query: Query)**

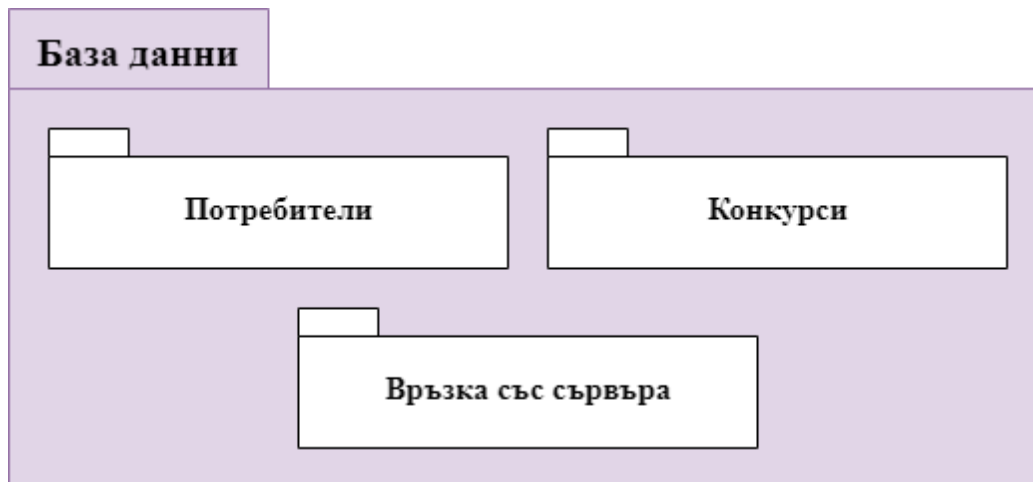
Входни данни: заявка, която се препраща към външните системи (търговските регистри)

Изходни данни: отговор на заявката, който се изпраща до съответния модул

Грешки и изключения: връща се съобщение за грешка при неналичност на системите

Пример за употреба: Получава се заявка, препраща се към регистър на НАП или търговски регистър, за да се обработи. След това се връща отговор на съответния модул.

2.5 База данни



2.5.1 Предназначение на модула

Този модул се грижи за съхраняването и извличането на данните от системата. Базата изпълнява заявки в конкретната реализация (SQL, Oracle, ...). Тя съхранява данните за потребителите и конкурсите така, че да са по-бързо достъпни.

2.5.2 Основни отговорности на модула

Модулът има за цел да съхранява личната информация на потребителите в системата, както и тази за конкурсите, които са публикувани (запазва се и информация за това от кого е публикуван конкурса и колко души са кандидатствали).

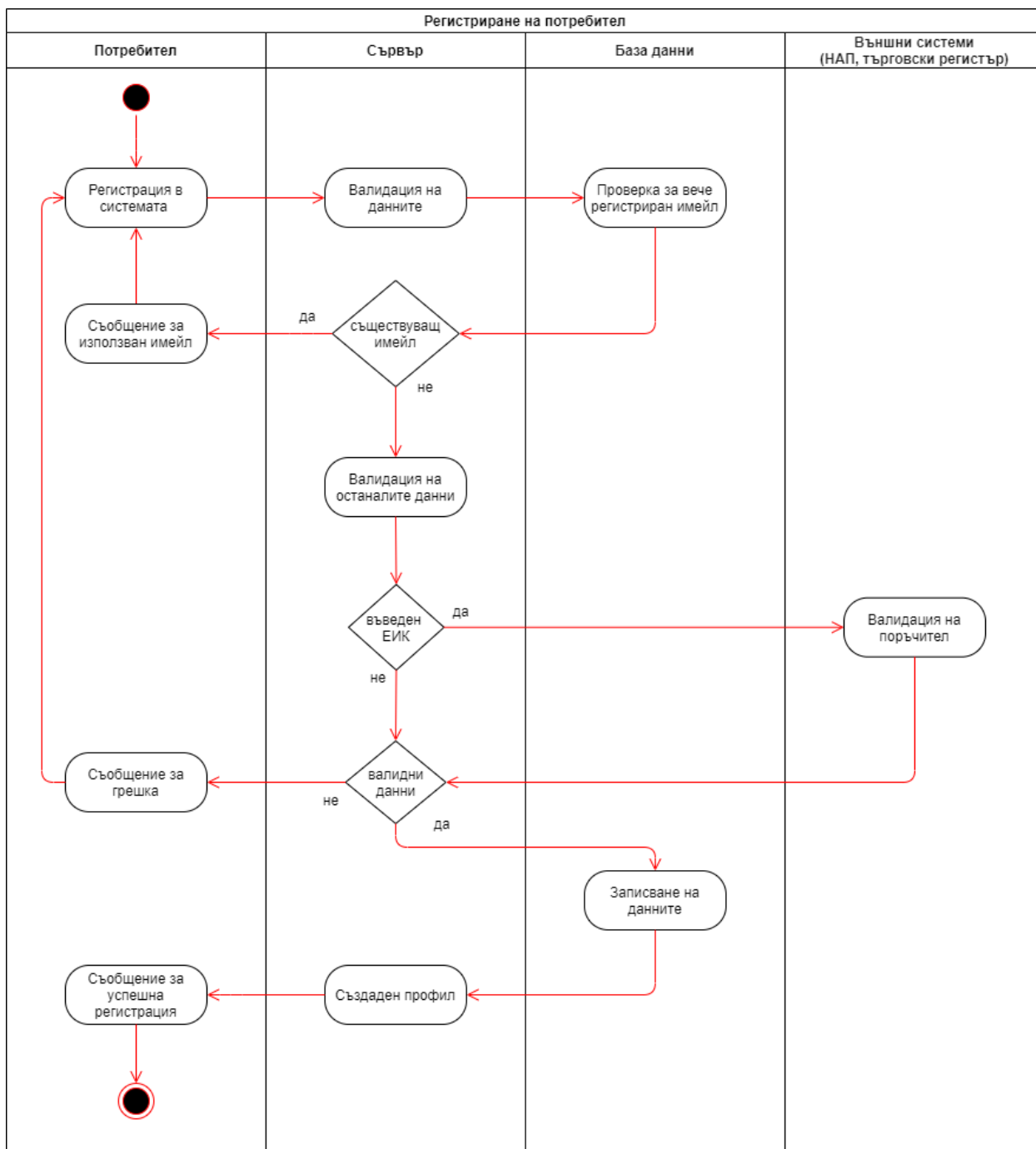
2.5.3 Описание на интерфейсите на модула

- Връзка със сървъра
 - **AcceptQuery(query: Query)**
Входни данни: заявка, която се получава от сървъра
Изходни данни: отговор на заявката към съответния модул
Зависимост от други елементи: „Сървър“
Пример за употреба: Получава се заявка от сървъра, обработва се и след това се връща отговор към съответния модул.

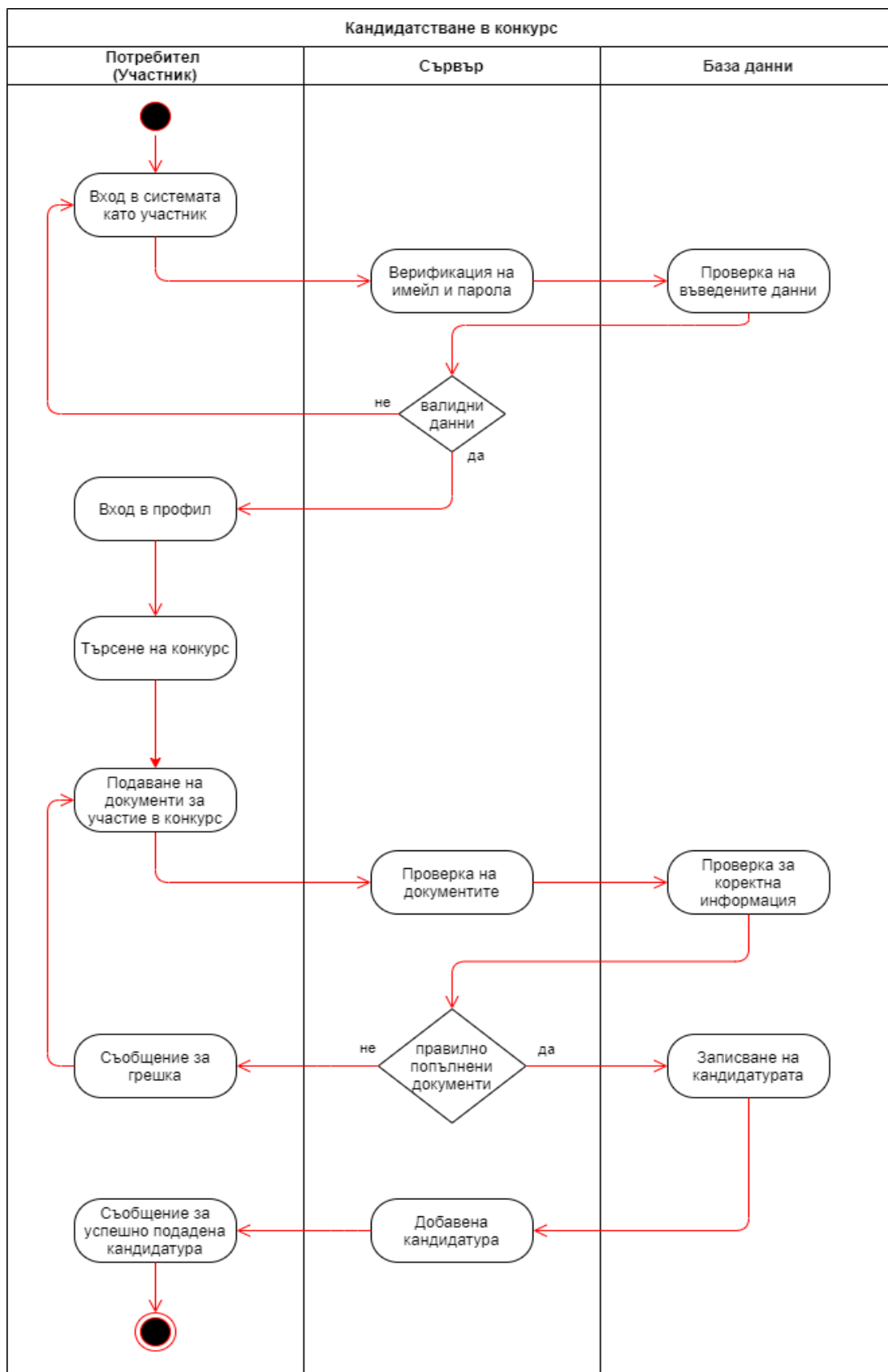
3. Описание на допълнителните структури

3.1 Структура на процесите

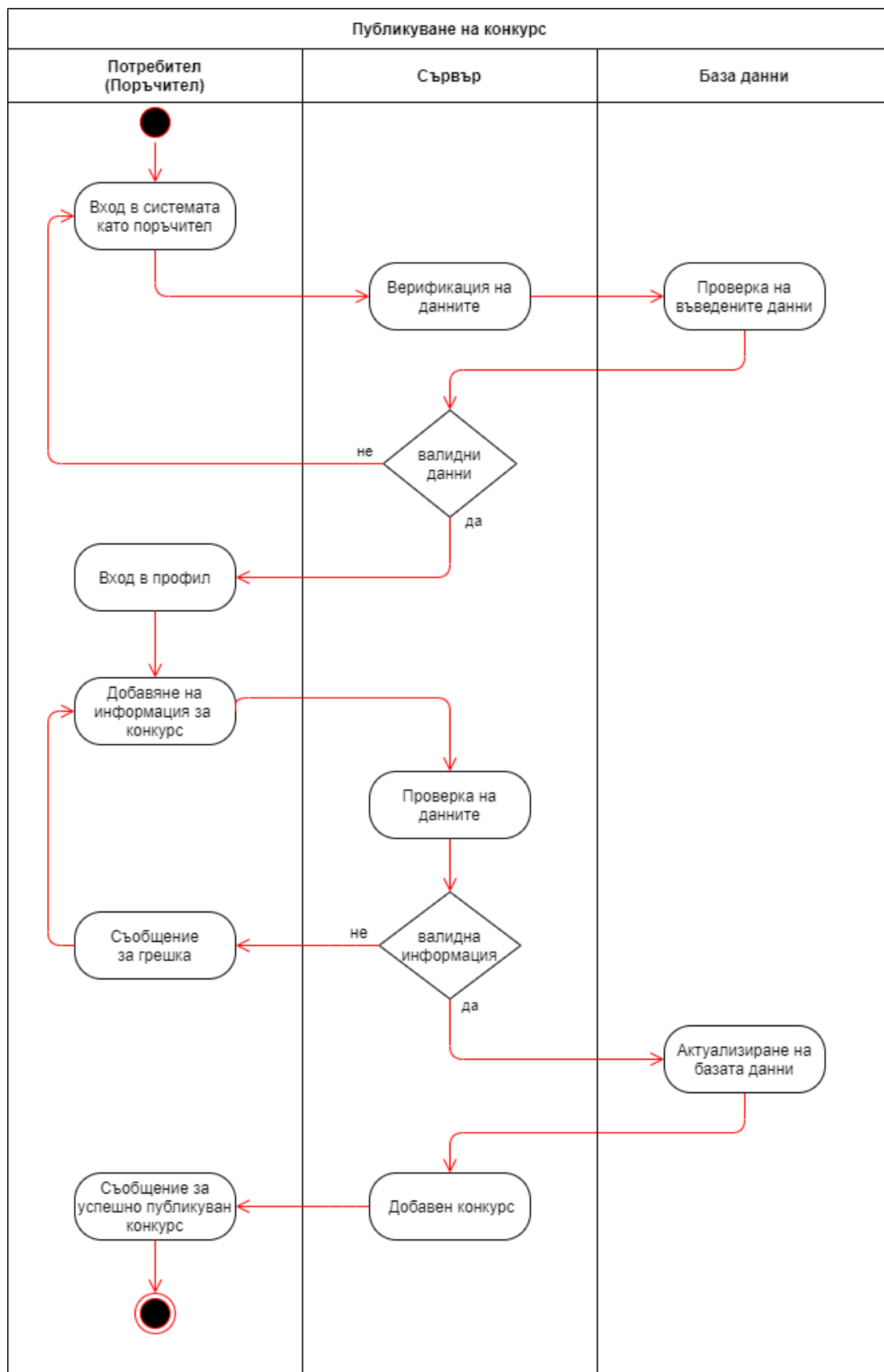
3.1.1 Първично представяне



Фиг. 1



Фиг. 2



Фиг. 3

3.1.2 Описание на елементите и връзките

На първата диаграма (фиг. 1) е показан процесът на регистрация на потребител. Валидират се въведените данни, проверява се в базата данни за вече съществуващ профил с посочения имейл. Ако съществува, се позволява въвеждане на нов, различен имейл. Ако имейлът не е регистриран, се валидират и останалите данни. Поръчителите въвеждат и ЕИК, чрез който се проверява състоятелността им със заявка до външните системи (НАП, търговски регистър). Ако всички данни и документи са валидни и изрядни, се записват в базата данни и се изпраща съобщение до потребителя за успешно създаване на профил.

Втората диаграма (фиг. 2) представя процеса на кандидатстване в конкурс. Регистрираният участник се вписва в системата и след валидация на въведените данни, се осигурява достъп до профила му. След търсене и успешно намиране на желан конкурс, потребителят подава електронно своите документи за участие. Извършва се проверка за съответствие с правилата на конкурса и при правилно попълнени документи, информацията се записва в базата данни. Процесът приключва със съобщение до участника за успешно подадена кандидатура.

Третата диаграма (фиг. 3) показва процеса на публикуване на нов конкурс. След одобрена регистрация, поръчителят се вписва в системата, въведените данни се верифицират и ако са валидни, се влиза успешно в профила. Добавя се информация за новия конкурс - ако тя е некоректна или непълна, се изпраща съобщение за грешка и се позволява повторно въвеждане. Валидната информация се записва в базата данни и се връща съобщение до поръчителя за успешно регистриран конкурс.

3.1.3 Описание на обкръжението

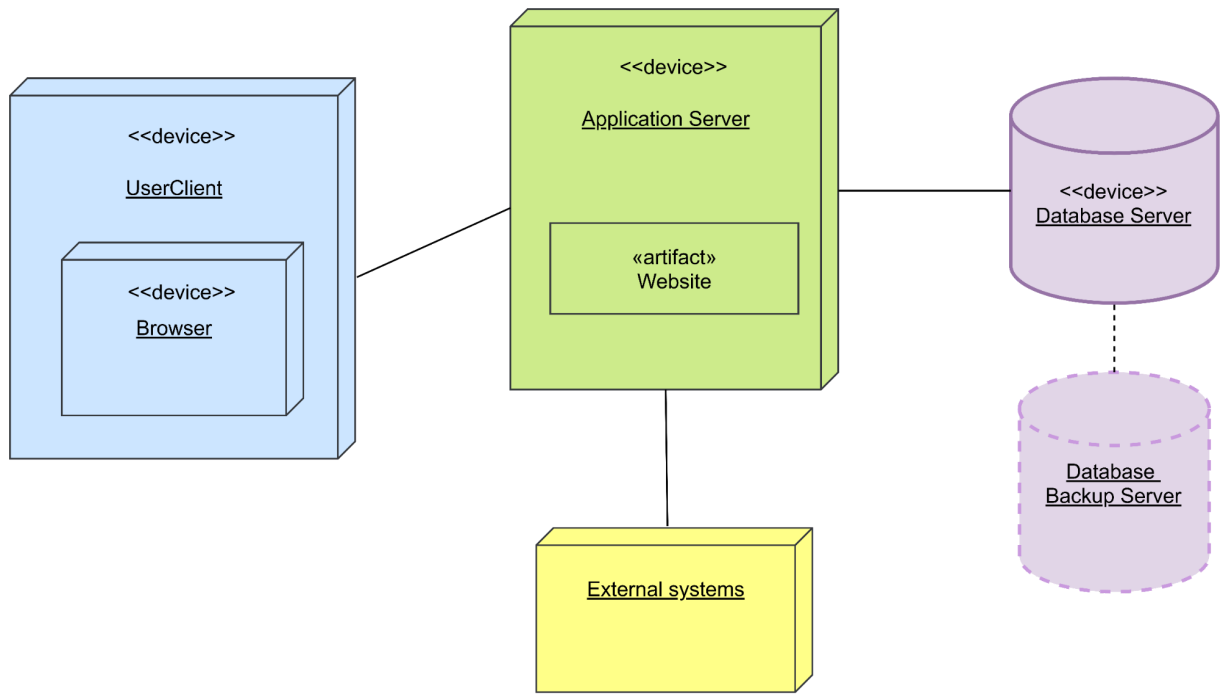
За да се удостовери коректността и изправността на подадените данни от поръчителите, и след това да им се дадат правата да публикуват конкурси, е нужно системата да има връзка с външни услуги (НАП, търговски регистър), които служат за проверка на състоятелността им.

3.1.4 Описание на възможните вариации

Възможност за бъдещо развитие на системата – обвързване с външни системи за социални мрежи, които да се използват за популяризиране на платформата.

3.2 Структура на внедряването

3.2.1 Първично представяне



3.2.2 Описание на елементите и връзките

UserClient е устройството, чрез което потребителят достъпва платформата, като използва уеб браузър. Предоставя потребителския интерфейс, чрез който се изпращат заявките към главния сървър.

Application server съдържа основната логика за функционалностите на системата и отговаря за сигурността (верификация на потребители, конкурси, данни и др.). Свързва се директно с базата данни и комуникира с външните системи.

В Database server се съхранява цялата база от данни на системата. Обособен е отделно с цел предпазване на данните от външни въздействия и от откази. Изискването за наличност налага на базата данни да е налична постоянно. Затова ще бъде дублирана, така че когато някоя инстанция спре да работи, работният поток да продължи безпроблемно.

В External systems са отделени външните системи, с които се свързва сървъра. Така системата се освобождава от отговорността за тяхната поддръжка и улеснява добавянето на нови услуги.

3.2.3 Описание на обкръжението

Потребителската част изпраща заявки към главния сървър, той се свързва с базата данни, за да ги обработи. Освен това може да се свързва и с външни ресурси, откъдето да взима допълнителна информация за конкурси.

3.2.4 Описание на възможните вариации

При разрастване на платформата главният сървър също може да бъде дублиран, за да се обработват успоредно повече заявки. Възможно е интегрирането на други външни системи.

4. Архитектурна обосновка

4.1 Архитектурен стил

Използвана е трислойна архитектура - клиент, сървър, база данни. Всеки модул и съответните му подмодули отговарят за различна функционалност. Модулите позволяват модификации и замяна, които не влияят и не водят до необходимост от внасяне на промени и в останалите части на системата. Разпределението на отговорностите осигурява по-лесна поддръжка. По-лесно може да се осигури добра защита, защото потребителите нямат пряк достъп до базата данни.

4.2 Архитектурни драйвери

1. Системата трябва да поддържа няколко различни вида потребители.

Важно за системата е да могат да се разграничават различните видове потребители, всеки от които има различни права. Основните групи потребители и техните основни функции са:

- **Администратори**
 - Публикува обяви за стипендии.
 - Публикува обяви за публични конкурси с безвъзмездно финансиране.

- Управлява потребителите.
- Одобрява регистрации на поръчители.
- Отчита статистика на посещенията.
- **Участници**
 - Търси и заявява участие в обявени конкурси.
 - Споделя полезни практики.
- **Поръчители**
 - Публикува конкурси и състезания с поръчки.

В модула “Потребител” е обособен подмодулът “Профили” (**2.3.3.3**), който изпълнява ролята на обвивен модул, в който се разполагат всички видове потребители, които ще ползват системата. Така се постига разделение на различните функционалности, до които те ще имат достъп.

2. Електронно подаване на документи за участие в конкурси

Като една от основните идеи на системата, изключително важно за нея е да осигури лесен и безпроблемен начин за подаване на документи за участие в конкурс, които се приемат от сървъра. След като участник потърси и избере конкурс, в който желае да се включи, трябва да подаде електронно своите документи в PDF формат (**2.3.3.3 - Кандидатстване в конкурс**), а на тях ще се прави автоматично проверка за формално и административно съответствие с правилата на конкурса (напр. дали всички документи са подадени и правилно попълнени).

3. Използване на данни от външни системи

За да се провери състоятелността на поръчителите, системата трябва да комуникира с външни системи (като НАП, търговски регистър), от които ще се взима информация за всички лица и фирми. Администраторите правят справки със съответните нормативни документи и след това одобряват регистрациите им. Актуализирането и проверката на тези услуги е от съществено значение за платформата, за да не се достига до нежелани последици при неизрядност. Връзката с външните източници се осигурява чрез специализиран за това подмодул „Връзка с НАП, търговски регистър” (**2.4.3.6**) в „Сървър”.

4. Сигурност

Сигурността е основна качествена характеристика, поради голямото количество лични данни. Трябва да се осигури добра защита на личните данни на потребителите, невъзможност за изгубване и неоторизиран достъп, подмяна на съдържанието или изтриването на обява, нерегламентирано публикуване на конкурс и т.н. За тази цел са предприети няколко тактики - при влизане в системата се прави автентикация (**2.4.3.2** -

Автентикация). Следва оторизация - проверка дали потребителят има достъп до определени ресурси (**2.4.3.2 - Оторизация**). За конфиденциалност на данните се използва криптиране. За допълнителна сигурност само сървърът се свързва директно с базата данни.

5. Използваемост

За да се подобри работата с платформата, е необходим интуитивен потребителски интерфейс. Намирането на различни конкурси, обяви и истории може да се улесни чрез предоставяне на възможност за търсене с опция за филтриране на резултатите по различни критерии в зависимост от интересите на потребителите (**2.3.3.3 - Търсене на конкурси**).

6. Надеждност

Важно качествено изискване за системата е да се осигури висока надеждност, като за това отговарят администраторите. Тази характеристика се осигурява чрез допълнителен подмодул "Нотификация по имейл" в "Сървър" (**2.4.3.3**). Админите се известяват чрез имейл за евентуално настъпили проблеми в системата, свързани с нарушаване на връзката между отделни модули и подмодули.

7. Изправност, Наличност

Системата трябва да може да се справя с трафика, който се увеличава към крайните дати на конкурсите. Трябва да осигури на потребителите, че обработването на техните заявки няма да отнеме много по-дълго от обикновеното време. За изправност се прилага активен излишък - дублира се базата данни, за да се осигури непрекъсната наличност, а при необходимост и сървъра, за да може да се обработват по-голям брой заявки и да продължи нормално протичането на работата без грешки или откази (**3.2**).