Софтуерна спецификация на изискванията

*на*

*Онлайн платформа за тестове- Quiz Master*

Изготвена от:

Ерика Карамучева- 2101321067

Факултет по математика и информатика

Версия 1.0

Съдържание:

1. [Въведение](#Въведение)
   1. [Цел](#Цел)
   2. [Стандарти, следвани в документа](#Стандарти)
   3. [За кого е предназначен този документ](#предназначение)
   4. [Обхват на продукта](#обхват)
   5. [Референции](#референции)
2. [Общо описание](#общо)
   1. [Обща перспектива](#перспективи)
   2. [Характеристика на продукта](#характеристика)
   3. [Видове потребители и характеристики](#потребители)
   4. [Операционна среда](#среда)
   5. [Ограничения за дизайна и реализацията](#ограничения)
   6. [Потребителска документация](#документация)
   7. [Предположения и зависимости](#зависимост)
3. [Външни изисквания към интерфейса](#външни)
   1. [Потребителски интерфейс](#потребителски)
   2. [Хардуерни интерфейси](#хардуер)
   3. [Софтуерни интерфейси](#софтуер)
   4. [Комуникационни интерфейси](#комуникационни)
4. [Функционалности на системата](#функционалности)
5. [Други нефункционални изисквания](#нефункционални)
   1. [Изисквания за производителност](#производителност)
   2. [Изисквания за безопасност](#безопасност)
   3. [Изисквания за сигурност](#сигурност)
   4. [Атрибути за качеството на софтуера](#качество)
6. [Други изисквания](#други)
7. **Въведение**
   1. **Цел**

Спецификацията на изискванията има за цел да даде детайлно описание на изискванията към софтуерния продукт „Онлайн платформа за тестове- Quiz Master“ (версия 1.0). Това е уеб базирано приложение, което позволява лесно и бързо създаване, решаване и управляване на тестове. В него всеки регистриран потребител може да създава, споделя и решава тестове. Той може да проследява напредъка си, за разлика от нерегистрираните потребители, които могат само да решават тестове, но не и да запазват и споделят своя резултат. В спецификацията се посочват архитектурните характеристики, зависимости и основните функционалности на приложението. Освен това, тя описва и ограниченията, които се налагат при неговото имплементиране и използване. Документът е създаден, за да бъде използван най- вече от екипа за разработка като препратка към клиентските изисквания и очаквания в дългосрочен план.

Софтуерната спецификация обхваща всички модули, за да даде пълна информация за функционалностите, които системата предоставя.

* 1. **Стандарти, следвани в документа**

По време на изготвянето на документа е следван шаблонът за създаване на софтуерна спецификация на изискванията (Software Requirement Specification, SRS). Началните точки представят целевите групи, общо описание и обхват на продукта. Посочени са отделните функционалности на приложението, както и зависимостите, изискванията и ограниченията, които трябва да бъдат съобразени.

Документът следва конвенция за използване на един и същи шрифт- Times New Roman (12), с цел по- добра четимост. Разстоянието между редовете е 1.15. Промяна има само в основните моменти на спецификацията.

* 1. **За кого е предназначен този документ?**

Основните целеви групи, към които се отнася спецификацията, са:

* *Мениджър на проекта-* благодарение на спецификацията, мениджърът на проекта може лесно да разбере какви са изискванията към софтуерния продукт. Тя служи като ръководство за планиране и изпълнение на проекта. Посредством спецификацията, мениджърът на проекта може да следи приоритетните модули и имплементации, които могат да бъдат направени във времето. Също така спомага и за идентифицирането на потенциални рискове, както и за планиране на мерки за тяхното управление. За мениджъра на проекта най- важните точки са Общо описание, Функционалности, Изисквания. От разбирането на всяка ключова зависимост и характеристика се определя и успеха на разработения софтуер.
* *Разработчици и тестери-* софтуерната спецификация служи за референция за изискванията за всяка функционалност и правилното й поведение при различни случаи (сценарии).
* *Маркетинг отдела-* запознаването с функционалността на софтуера е от съществено значение за маркетинговия отдел. Благодарение на нея, той може да създава маркетингови материали и стратегии, които да подчертаят предимствата на продукта за клиентите. Също така спецификацията може да спомогне за по- лесно определяне на целевата аудитория. От изключително значение за маркетинговия отдел е да са напълно запознати с Общо описание и Бъдещи насоки за развитие, за да могат да планират своята дейност.
* *Потребители*- спецификацията може да послужи и като ръководство за потребителя. Благодарение на нея, той може да разбере цялостната концепция на софтуера. По този начин се гарантира по- добро изживяване за потребителя.
  1. **Обхват на продукта**

Приложението е насочено предимно към ученици и учители. Въпреки това няма никакви ограничения и може да бъде използвано от всички. Неговата идея е да улесни работата на учителите, давайки възможност за бързо и лесно създаване на тестове, както и за възможност за проследяване на резултатите на учениците от тези тестове. Приложението подпомага и работата на учениците- дава им достъп до многобройни ресурси, както и възможност да проследят напредъка си.

Имплементацията на приложението позволява еднаквата му ефективност и бързодействие при различни натоварвания и брой потребители. И двата критерия са еднакво важни и определящи при избора на софтуера, който да се приложи в работния процес.

* 1. **Референции**

1. **Общо описание на продукта**
   1. **Перспективи за развитие**

Макар да съществуват подобни софтуерни системи, „Онлайн платформа за тестове- Quiz Master“ се отличава от останалите с интуитивен интерфейс, както и с възможностите за проследяване на напредъка. Потребителят лесно и бързо може да достъпи до всяка една от функционалностите на системата.

Чрез приложението „Quiz Master” се дава възможност на всеки потребител да създава бързо и лесно тестове, с които да тества своите познания. Тестовете биват два вида- публични и частни. Публичните тестове са видими за всички потребители в системата, независимо дали са регистрирани, или не- т.е. всеки може да ги достъпва. Единствената разлика е, че при регистрираните потребители резултатът се запазва и те могат да го споделят, да видят верните отговори на теста, както и да оставят коментар към автора при необходимост. Докато нерегистрираните потребители само виждат своя резултат. Те не могат да го запазват или споделят, без направена регистрация в системата. Частните (лични) тестове са видими само за потребителя, който ги е създал. Той може да даде достъп до тях на други потребители, само чрез предварително създаден код. Самите тестове са изградени от минимум два въпроса, като няма ограничение за техния максимален брой. Въпросите биват два вида- с отворен и с избираем отговор (броят на предложените отговори може да бъде променян от потребителя). И в двата случая има възможност за прикачване на изображение като условие/ отговор.

Текущият вариант на системата ще бъде пригоден за използване на всички налични браузъри. Започването на разработката на продукта като уеб приложение е направено с цел улеснение на потребителя- идеята е да може лесно и удобно да въвежда и редактира своите въпроси. Важно е да се отбележи, че създаването на мобилно приложение е следващата логична стъпка, в развитието на приложението. Така приложението ще бъде още по- достъпно за потребителя.

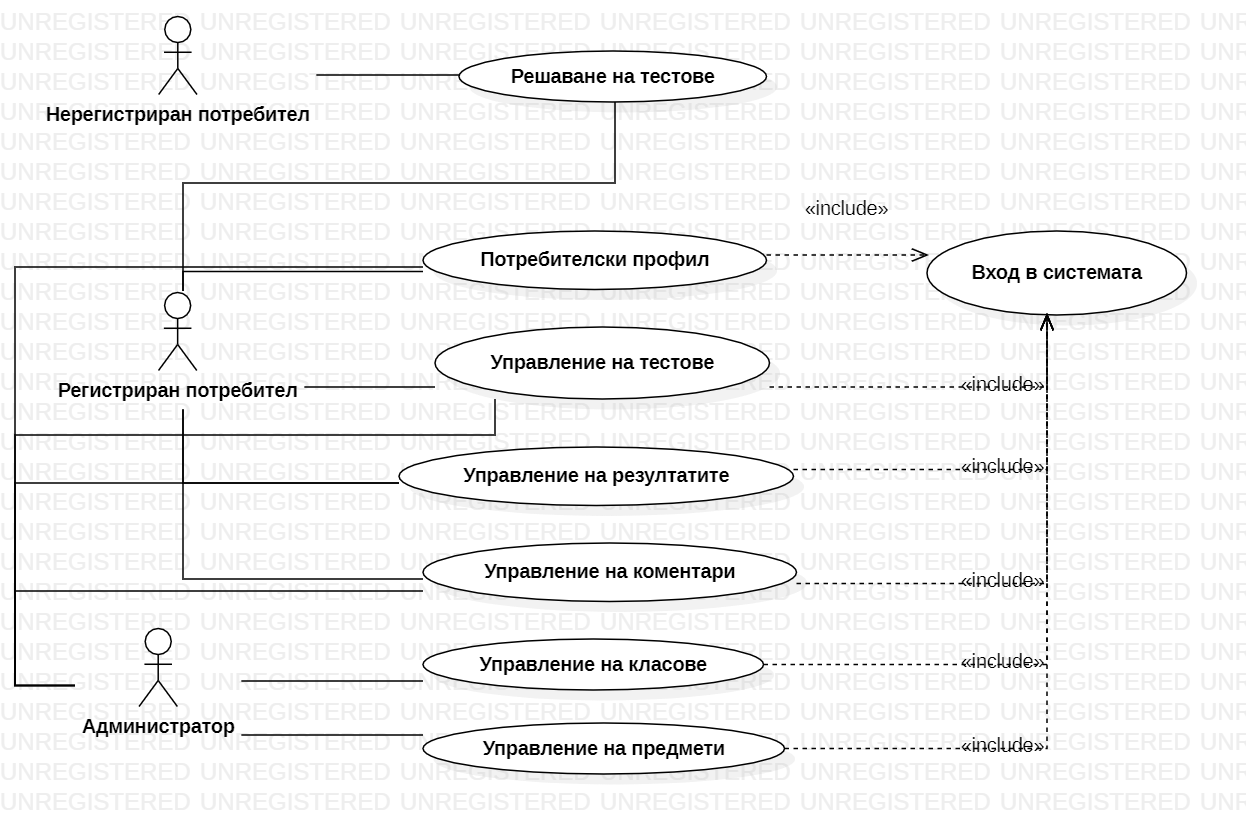
С въвеждането на мобилната версия е логично и да се въведе нова функционалност- споделяне на тестове посредством QR код. Сканирането на QR кода цели по- лесното разпространение на тестовете. Това ще изисква по- малко време и усилия в сравнение с използването на създадените от системата кодове. Разбира се, те ще останат част от бъдещите версии на приложението.

Друга функционалност, която следва да бъде въведена е и опцията за вмъкване на pdf документи и тяхното конвертиране към формата на платформата.

* 1. **Характеристика на продукта**

Основните функционалности на приложението са свързани управление и решаване на тестове и управляване на резултатите.

* *Управление на тестове-* това е една от най- значимите функционалности на приложението. Позволява на регистрираните потребители да създават, преглеждат, редактират и изтриват тестове и въпроси. Администраторите също ще имат достъп до тази функционалност. Тяхната роля е да правят промени в тестовете (изтриване, промяна на клас и др.) след докладване на грешки от страна на потребителите.
* *Решаване на тестове-* това е другата по- важна функционалност, която предоставя системата. Както регистрираните, така и нерегистрираните потребители имат достъп до нея.
* *Управление на резултатите/ проследяване на напредъка-* това е другата функционалност, която дава възможност на потребителите да виждат своите резултати и да проследяват тяхното развитие. Достъп до тази функционалност ще имат и администраторите. Тяхната роля е да извличат и да обработват информацията от гледна точка на развитието на приложението.
* *Потребителски профил-* тази функционалност дава възможност на потребителя (само за регистрираните потребители и администратори) да променя своите потребителски данни.
* *Управление на коментарите-* тази функционалност дава възможност на регистрираните потребители и администраторите да създават, редактират, четат и изтриват коментари към тестовете.
* *Управление на класовете*- благодарение на тази функционалност, администраторите ще могат да създават, преглеждат, редактират и изтриват възможния списък с класове.
* *Управление на предмети*- администраторите могат да управляват предметите, за които потребителите могат да създават тестове.



# **Фигура 1: Use Case диаграма на обобщените функционалности**

* 1. **Групи потребители и техните особености**

Основните групи, които активно използват системата са три: Регистрирани потребители, Нерегистрирани потребители и Администратори. Това разделение по „актьори“ помага за по- лесното и интуитивно използване на системата предвид отговорностите на всеки от потребителите. Освен това всяка група от потребители се характеризира със следните особености:

* *Регистрирани потребители-* това са потребителите, които имат регистрация в системата. Те имат достъп до функционалностите „Управление на тестове“ (създаване, четене, промяна и изтриване), „Решаване на тестове“, „Проследяване на напредъка/ Управление на резултатите“ (получават информация чрез различни таблици и диаграми), „Потребителски профил“ и „Управление на коментарите“. Тук едно от предимствата на регистрираните потребители е, че могат да решават частни тестове, ако потребителя, който ги е създал им предостави код. Също така, след решаването на теста на тях им се дава резултата, който може да бъде споделен. Имат достъп до верните отговори за всички тестове (след като вече са ги решили). Могат да поставят коментари под тестовете.
* *Нерегистрирани потребители-* това са потребители, които все още нямат регистрация в системата. Те имат достъп до само една функционалност, а именно „Решаване на тестове“. Тестовете, които могат да решават са изцяло публични. Те могат да проверят своя резултат, но не и да го споделят. Също така не могат да видят верните отговори за теста.
* *Администратори-* тяхната роля е свързана с цялостното управление на системата. Те се грижат за поддържката на базата, която се използва при създаването на тестовете (като класове и предмети). Имат достъп до тестовете и могат да извършват различни операции с тях. Други функционалности, до които имат досег са „Управление на коментарите“, „Управление на резултатите“ (с цел анализ на развитието на системата), както и „Потребителски профил“.
  1. **Среда на работа**

„Онлайн платформата за тестове- Quiz Master“ е Cloud базиран софтуер, който използва услугите на Microsoft- Microsoft Azure. Те осигуряват изключителна сигурност и гъвкавост при работа със системата, както и при нейното натоварване. Основните предимства, които ни предоставя Azure са:

* *Мащабируемост и гъвкавост-* едно от ключовите предимства на Microsoft Azure е неговата мащабируемост и гъвкавост. Azure позволява да се надгражда инфраструктурата нагоре или надолу, без особени усилия. Също така предоставя и широка гама от услуги, включително виртуални машини, съхранение и работа в мрежа, което позволява създаването на персонализирани решения.
* *Сигурност и съответствие-* Azure предоставя богат набор от функции и услуги, които целят защита на данните в облака. Функциите за сигурност и съответствие включват вградено разузнаване на заплахи, сертификати за съответствие и разрешена защита от заплахи.
* *Цена-* като се има на предвид, че приложението е базирано на облачна архитектура, то неговата цена ще се определя от услугите, които то използва. Това означава, че приложението ще има достъп до най- новите технологии и функции, с възможно най- малко разходи. От друга страна Azure предоставя и богат набор от инструменти и услуги, които също спомагат за оптимизация на разходите.

Хардуерът, който стои зад системата, е осигурен от Microsoft Azure и може да бъде лесно променен или подновен.

Ресурсите, които операционната система заделя и използва, са максимално оптимизирани по такъв начин, че без да бъдат прекомерно натоварвани, те могат да бъдат оптимално използвани.

* 1. **Ограничения за дизайн и имплементация**

Архитектурата на онлайн платформата за тестове „Quiz Master” е разпределена- всяка изграждаща част на софтуера работи независимо от останалите и това позволява стабилност, гъвкавост и бъдещо развитие:

* *Графично- Програмен Интерфейс- Frontend*- Ngnix- клиентската част от приложението работи на безплатен, Ngnix сървър с отворен код. Той ускорява производителността, намалява времето за чакане, предлага мащабируемост и възможност за обработка на няколко заявки едновременно. Предвид прогресивното развитите на платформата е избрана работната рамка Angular. Тя предлага широка гама от готови компоненти, осигурява множество функционалности без външни модули, както и готова архитектура, върху която могат да бъдат изградени допълнителни слоеве.
* *Приложно- Програмен Интерфейс- API*- за реализацията на бизнес логиката и работата с данни се грижи приложно- програмния интерфейс на “Quiz Master”. За неговата реализация се използва ASP.NET Core . Рамката е съвместима с различните операционни системи- Linux, MacOS, Windows и мобилни устройства (Xamarin).
* *База данни*- за изграждането на „Quiz Master” API се използва Entity Framework. Той облекчава достъпа и манипулирането на данните в базата. Предоставя високо ниво на абстракция върху нея чрез обектно- ориентиран програмен интерфейс. За управление на базата от данни се използва MS SQL Server. Тази система има много предимства- лесно въвеждане и манипулация на данните, сигурност на данните, мащабируемост и други.
  1. **Потребителска документация**

Основната част от потребителската документация се намира в самото приложение и по- конкретно в неговия потребителски интерфейс. Многобройните указания и препратки показват какво може да бъде намерено в системата, как да бъде използвано, какво би се случило, ако последва действие. Кратко ръководство с допълнителна информация ще има и в страницата „Помощ“. Тя може да бъде достъпена, от която и да е част на програмата.

Всеки потребител може да използва контактната форма в системата, за да зададе въпрос или да докладва за проблем или нередност. Също така на потребителя са предоставени телефон и имейл, с които също може да осъществи връзка с администратор.

* 1. **Зависимост от среда и особености**

*Достъп до Интернет*- единствената зависимост от потребителска гледна точка за използване на „Онлайн платформата за тестове“, е зависимостта от Интернет. Макар че софтуерът се използва от различни потребители, като всеки от тях има възможност да създава и решава тестове- т. е. имат свой собствен „дял“ в базата от данни, вариантът за локално хостване е възможен, но не препоръчителен.

* Първата и най- важна причина е, че се губи възможността да се решават тестове, създадени от друг потребител. Това до голяма степен ограничава възможностите на потребителя.
* Пестим ресурси от страна на клиентите- клиентските машини се натоварват минимално, като държи frontend частта възможно най- проста.
* Дава се възможност да се достъпи системата от всяко място- независимост от IP адрес и мрежа на свързаност. Това са сред най- важните причини облачните услуги да набират популярност.

*Външни Библиотеки*- както “Quiz Master” API, така и фронтенд частта на приложението използва голям набор от външни библиотеки с цел пестене на време и усилие при разработка на често срещани функционалности, опции и сценарии.

1. **Външни изисквания към интерфейса**
   1. **** **Потребителски интерфейс**

# Фигура 2: Начален екран

# Фигура 3: Вход в системата

# Фигура 4: Регистрация

# Фигура 5: Табло за управление

# Фигура 4: Създаване на тест

# Фигура 4: Всички тестове

# Допълнителна информация: [Figma](https://www.figma.com/file/jVeipyocf5G7pNG3HtN6tE/2101321067--%D0%9E%D0%BD%D0%BB%D0%B0%D0%B9%D0%BD-%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B0-%D0%B7%D0%B0-%D1%82%D0%B5%D1%81%D1%82%D0%BE%D0%B2%D0%B5--QuizMaster?type=design&node-id=0%3A1&mode=design&t=4qUWcnbjYCTH8M9P-1)

* 1. **Хардуер**

С помощта на избраната архитектура, платформата „Quiz Master” свежда хардуерните изисквания до минимум. Приложението е компилирано предварително, оптимизирано и готово за четене от браузърите. Единственото условие за доброто и пълноценното му използване е добрата интернет връзка. Протоколът за връзка, който трябва да се използва HTTPS. Целта е да се осигури сигурността на данните- тестовете, които създават потребителите, както и резултатите, които са получени.

* 1. **Софтуер**

Приложението ще използва няколко предварително дефинирани библиотеки, които ще се използват във frontend и backend разработките. Приложението комуникира с базата данни, за да получи актуалната информация за тестовете, въпросите и резултатите. Комуникацията между приложението и базата се състои от всички налични операции за създаване, четене, промяна и изтриване на данните. Разпределената архитектура води до наличието на няколко подсистеми, които трябва да работят стабилно, сигурно и ефективно заедно. Цялостната концепция за изграждане и начин на работа е създадена така, че да посреща оптимално нуждите на различните видове потребители (независимо от техния брой).

* 1. **Комуникационни интерфейси**

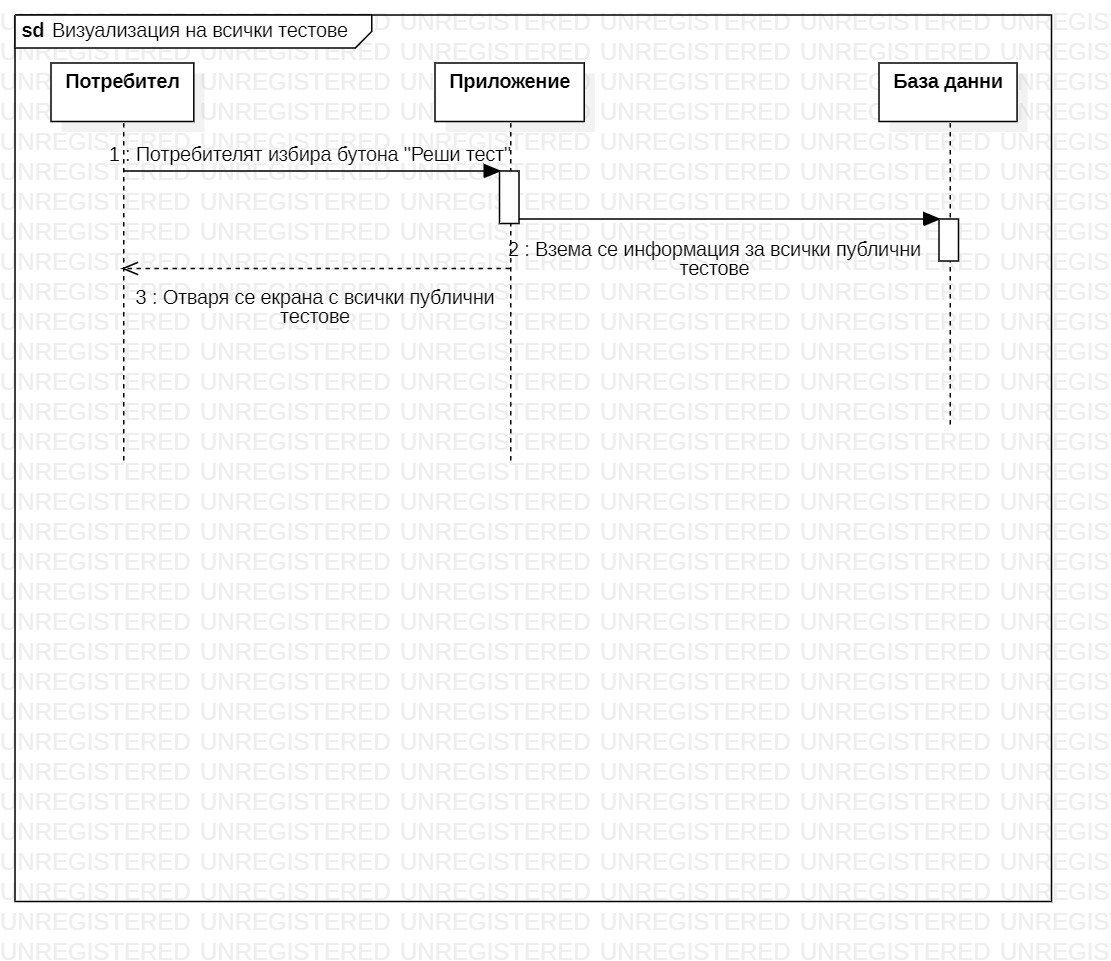
Комуникацията между отделните модули, конкурентостта и консистентността на данните е от изключително значение за разпределените архитектури. Комуникацията на приложението се осъществява на две нива:

* *Между-компонентно*- отделните части на системата се комуникират помежду си, използвайки стандартни протоколи, за да изпълнят потребителските заявки. Синхронизацията е между тях е от изключително значение за живота на приложението.
* *Между системата и компонентите*- в най- честия сценарии за използване, потребителите комуникират с frontend приложението чрез браузър. За връзката към уеб сървъра и към приложно- програмния интерфейс се използва HTTPS сертификат, който осигурява сигурността още в начална фаза на употреба.

1. **Функционалности на системата**
   1. **Визуализация на списъка с всички публични тестове**
      1. Описание и приоритет

Извеждане и екранна визуализация на всички публични тестове (достъпни и за нерегистриран потребител)

Приоритет: Висок

* + 1. Последователност от действия
    2. Предварителни изисквания

1. Потребителят e/ не е регистриран и е в началния екран на приложението.
2. Потребителят избира Реши тест.
   1. **Визуализация на списъка с всички тестове, създадени от даден потребител**

4.2.1. Описание и приоритет

Извеждане и екранна визуализация на всички тестове, създадени от даден потребител.

Приоритет: Висок

4.2.2. Последователност от действия

4.2.3. Предварителни изисквания

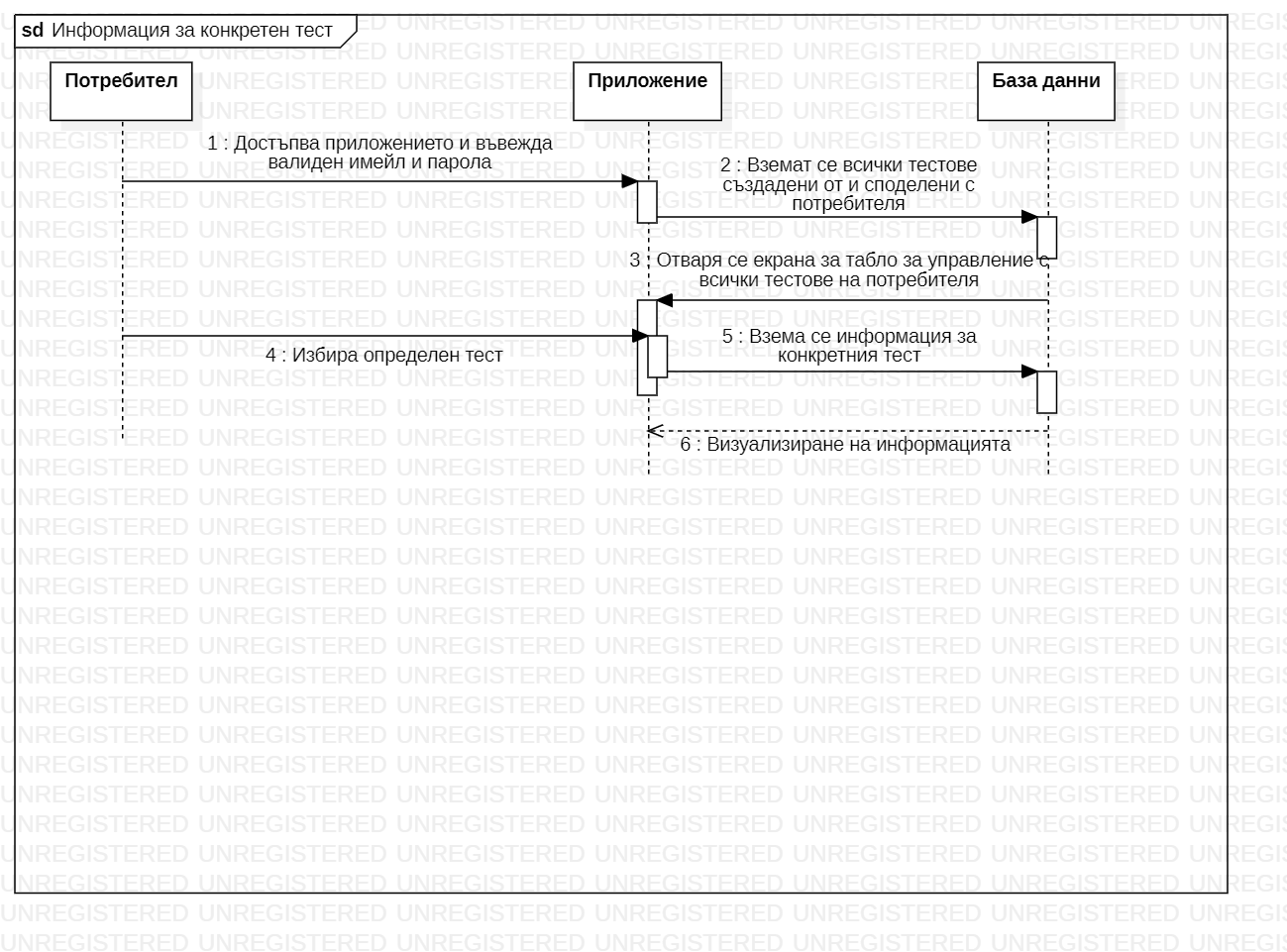
1. Потребителят да е влязъл в системата.

2. Потребителят се намира в екран, отговорен за визуализация на всички тестове създадени от и споделени с потребителя.

* 1. **Визуализация на информация за конкретен тест**
     1. Описание и приоритет

Извеждане и екранна визуализация на цялостна информация за конкретен тест.

Приоритет: Висок

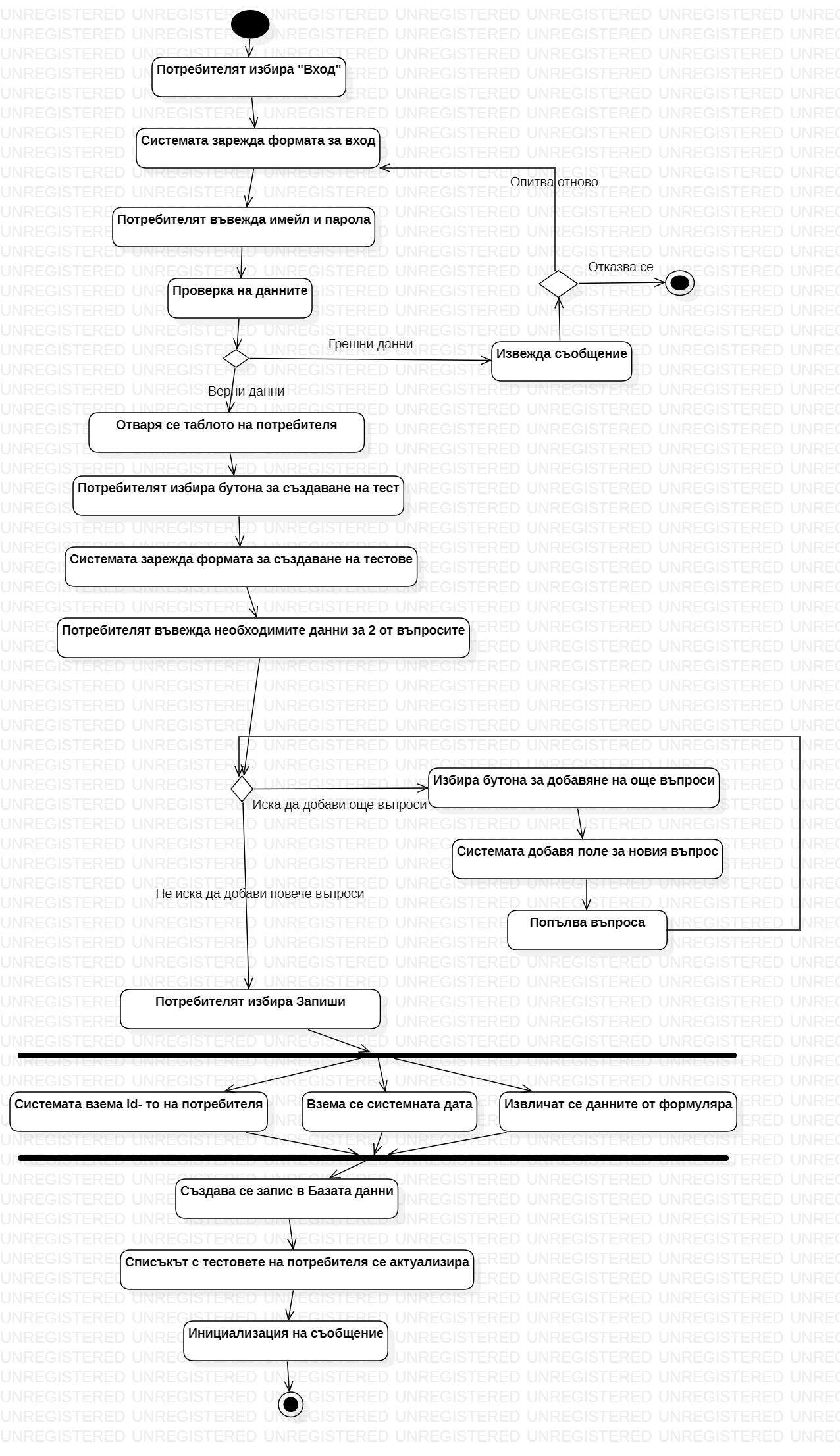
* + 1. Последователност от действия
    2. Предварителни изисквания

1. Потребителят да е влязъл в системата.
2. Потребителят се намира в екрана, отговорен за визуализация на всички тестове.
3. Потребителят избира от списъка с всички негови тестове този, за когото иска да види подробна информация.
   1. **Визуализация на формата за създаване на тест** 
      1. Описание и приоритет

Извеждане и екранна визуализация на формата за създаване на тест. Последователност от действия за създаване на тест.

Приоритет: Висок

* + 1. Последователност от действия

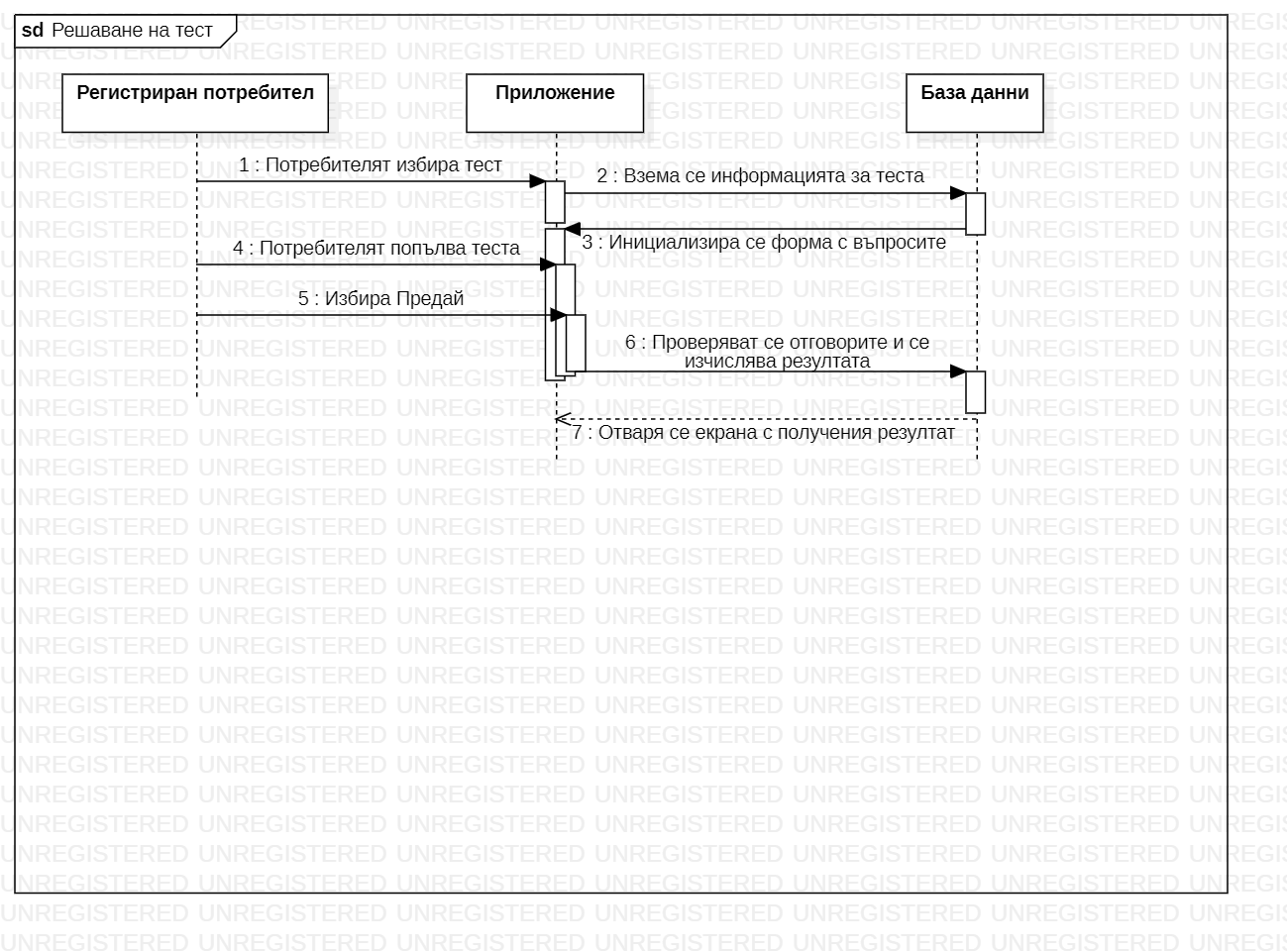


* + 1. Предварителни изисквания

1. Потребителят да е влязъл в системата.
2. Потребителят се намира в екрана, отговорен за визуализация на всички създадени от и споделени с потребителя тестове.
3. Потребителят избира бутона за създаване на нов тест.
4. Потребителят въвежда информация за поне два въпроса.
5. Добавя още въпроси и ги попълва.
6. Потребителят избира Запиши.
   1. **Решаване на тест**
      1. Описание и приоритет

Извеждане и екранна визуализация на функционалността за решаване на тест.

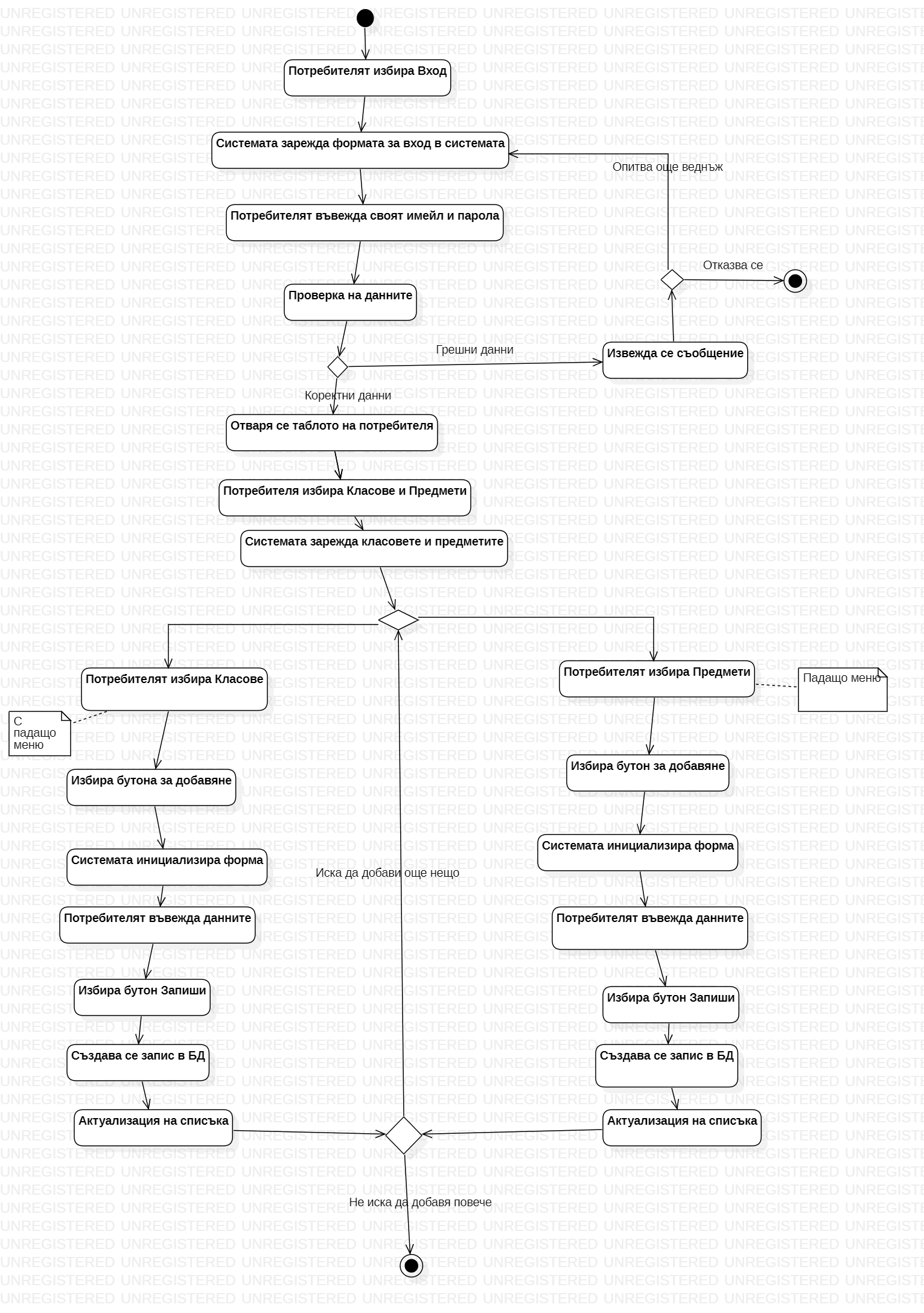
Приоритет: Висок

* + 1. Последователност от действия
    2. Предварителни изисквания

1. Потребителят да е влязъл в системата.
2. Потребителят се намира в екран, визуализиращ тестове (публични или лични).
3. Потребителят избира конкретен тест, който иска да реши.
4. Потребителят попълва теста.
5. Избира бутона Предай.
   1. **Добавяне на класове и предмети**
      1. Описание и приоритет

Извеждане на екран, визуализиращ всички възможни класове и предмети. Добавяне на още класове и предмети към системата.

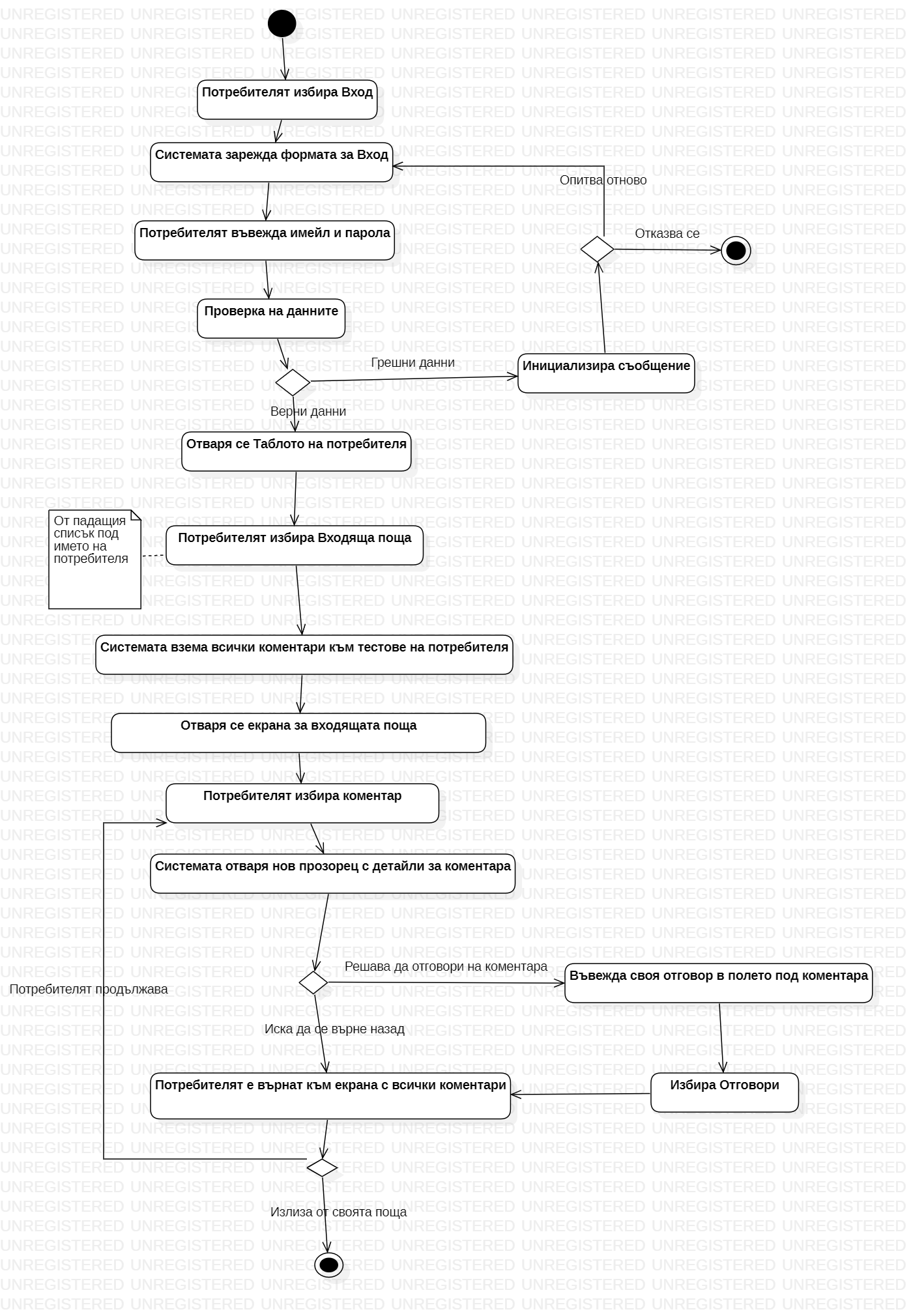
Приоритет: Висок

* + 1. Последователност от действия
    2. Предварителни изисквания

1. Потребителят да е влязъл в системата.
2. Потребителят е администратор.
3. Потребителят се намира в началния екран.
4. Потребителят избира от Класове и Предмети.
5. Потребителят отваря списъка с класове/ предмети.
6. Избира бутона за добавяне.
7. Въвежда информация във формата.
8. Избира бутон Запиши.
   1. **Визуализация на коментари към тестовете/ Входяща поща**
      1. Описание и приоритет

Извеждане на екранна визуализация, съдържаща всички коментари, свързани с даден потребител.

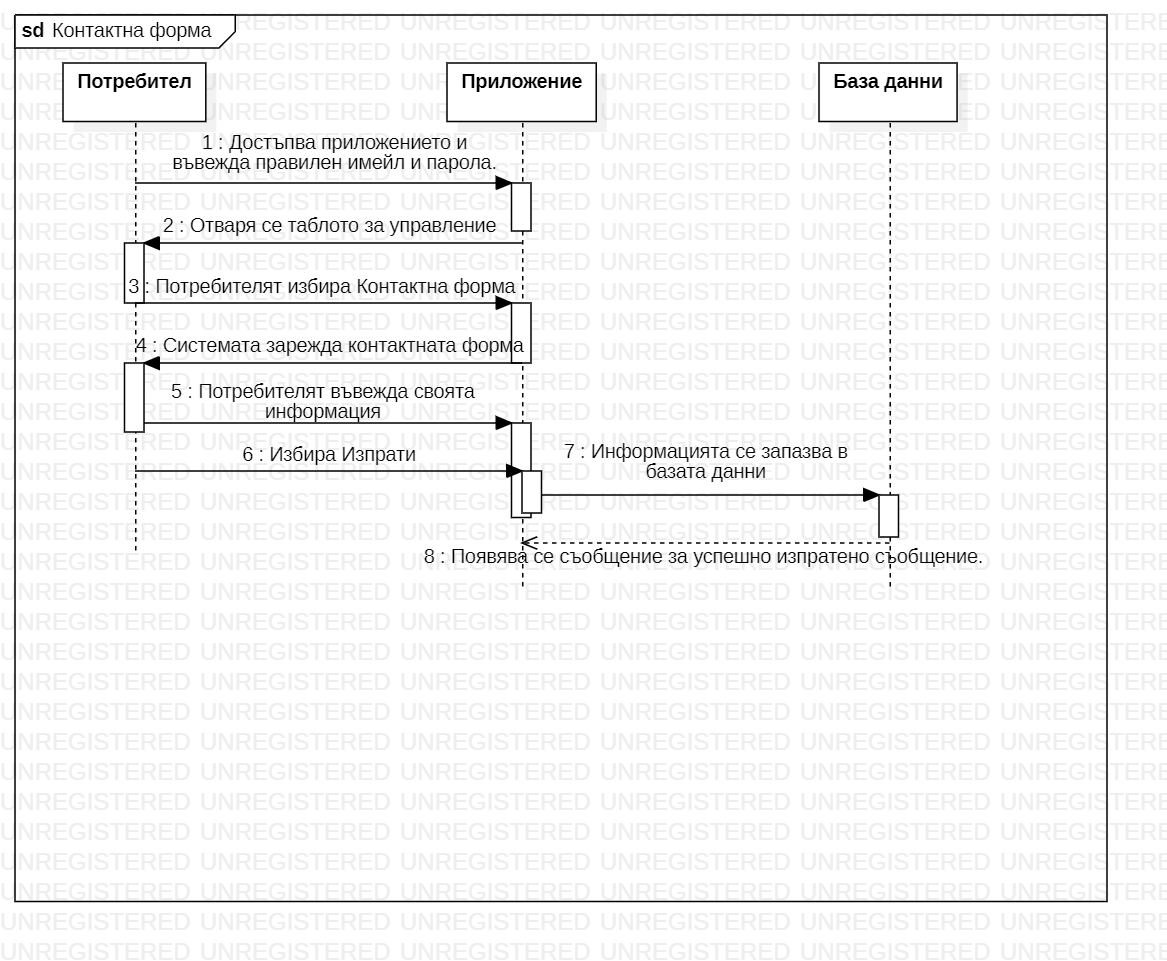
Приоритет: Среден

* + 1. Последователност от действия
    2. Предварителни изисквания

1. Потребителят да е влязъл в системата.
2. От падащото меню потребителят избира Входяща поща.
3. Потребителят избира коментар.
4. Потребителят отговаря на коментара.
5. Потребителят излиза от своята поща.
   1. **Визуализация на контактна форма и отправяне на запитване**
      1. Описание и приоритет

Извеждане и екранна визуализация на контактната форма. Попълване на формата.

Приоритет: Среден

* + 1. Последователност от действия
    2. Предварителни изисквания

1. Потребителят да е влязъл в системата.
2. Потребителят избира Контактна форма.
3. Потребителят попълва формата.
4. Избира Изпрати.
5. **Други нефункционални изисквания**
   1. **Изисквания за производителност**

Характерът на платформата създава предпоставка за сериозно увеличаване на потребителския интерес. Това поставя сериозни изисквания за критерия „производителност“. Едно от най- съществените изисквания е системата да може да поддържа работата на няколко потребителя едновременно. Те не бива да изпитват затруднения с достъпа до приложението, дори и много потребители да се опитват да го достъпят по едно и също време. Важно е времето за обработката на отделните транзакции и максимално бързото им изпълнение. Нарастването на броя клиенти и информацията, свързана с тях- тестове и резултати водят до повече изисквания на функционално ниво, следователно до по- голям брой работни файлове и по- високо ниво на сложност. С цел по- лесното администриране и решение на често срещани проблеми е нормално използването на външни библиотеки. Важно е те да не оказват негативно влияние върху потребителското изживяване по време на използването на приложението.

Зареждането на системата не бива да отнема повече от 2 секунди, като оптималният вариант е то да бъде под секунда при наличие на добра интернет връзка. Максималния допустим интервал на получаване на отговор при заявка към приложно- програмния интерфейс трябва да бъде:

* Под 100 милисекунди за прости операции като вход в системата, регистрация, добавяне на коментар, клас и предмет.
* Под 250 милисекунди за извеждане на списъци с тестове, проверката им след решаване и други. Включително и филтрация и възможност за сортиране на получените като резултат данни.
* Под 1 секунда за по- тежките операции като генериране на таблици и диаграми с резултати, извличане на въпросите към тестовете и други. В случай, че действието отнеме повече време, в потребителският интерфейс, трябва да се появи съобщение, което ясно да указва, че се изчаква отговор от приложно- програмния интерфейс и това може да отнеме малко повече време.

За бързодействието на системата и бързите изчисления в нея трябва да се грижат няколко инструмента, предлагани от Azure.

* Service Level Agreements (SLAs)- споразумението за ниво на обслужване е официално споразумение, с което Azure определя стандартите за производителност. То включва параметри като наличност, производителност, време за реакция при проблеми, както и други важни аспекти на обслужването.
* Load Testing- инструментът също може да помогне за оценяване на производителността и скалируемостта на приложението при различни натоварвания.
  1. **Изисквания за безопасност**

*Access Control Layer (ACL)* - една част от приложението е достъпна за абсолютно всички потребители, независимо дали те са регистрирани, или не. Това са функционалностите, свързани с решаването на тестове и преглед и филтриране на публични тестове. Това са действия, които не внасят промени в системата. За всички останали (такива, които предполагат някаква промяна в приложението) се изисква автентикация. За нейната реализация се използва уникален имейл адрес и парола. Системата дефинира още базирана на ролите оторизация, чрез която всеки потребител получава определен набор от функционалности в зависимост от ролята, която му е зададена. Визуалните елементи, до които потребителят няма достъп, съгласно ACL дефинициите, не биват показвани, а в случай че по някакъв начин, той успее да ги зареди, всяка Quiz Master API заявка бива проверявана за съответните привилегии преди да изпълни каквато и да била по-нататъшна операция с данните.

Други имплементации, осигуряващи безопасното използване на Quiz Master са:

* Потвърждаване на потребителския акаунт посредством имейл в определен интервал от време.
* Забраняване на пароли, които не съдържат малка буква, голяма буква и цифра.
* Спиране на достъпа до системата на всички потребители, които са маркирани като неактивни или имат наложени други ограничения.
  1. **Изисквания за сигурност**

Един от начините за осигуряване на сигурността е чрез създаване на резервно копие. За целта се използва облачната услуга Azure Backup. Избират се всички ресурси (виртуални машини, бази данни и файлови системи), които трябва да бъдат включени в резервните копия. Важно е да се отбележи, че процесът, от копирането на данните от източника до създаването на резервно копие в Azure, е криптиран и това осигурява сигурността на данните по време на трансфера. Също така услугата дава възможност за настройка на графици за автоматично изпълнение на резервните задания, което допълнително улеснява работата по разработката на приложението.

“Quiz Master” предоставя механизъм за основна оторизация чрез имейл и парола. Защитата е осигурена и от задължителния достъп до въведения имейл от страна на потребителя. Допълнителна защита осигурява и използваният механизъм за хеширане на потребителските пароли- HMACSHA256.

Подобно на всяка сигурна уеб система, Quiz Master използва Secure Sockets Layer (SSL), чрез който комуникацията между клиента (браузъра) и уеб сървъра е криптирана. Употребата на този слой е стандартна- използва се т. нар. „ръкостискане“ и сесийни ключове за всяка връзка. Публичният ключ на сървъра се използва за кодиране на генериращото се на случаен принцип число при клиента. След това само Quiz Master инфраструктурата може да го дешифрира и използва за клиентските сесии. Браузърът се грижи за разпознаването на издателя на сертификата.

* 1. **Атрибути за качеството на софтуера**

Качеството на „Quiz Master”, като софтуерна единица, е осигурено от следните критерии:

* *Достъпност*- благодарение на архитектурата си, приложението е достъпно от всяко IP по света. Не е обвързано с локални мрежи или сегменти.
* *Тестване*- за осигуряването на качеството на софтуера се провеждат два основни вида тестове:
* Ръчно- при добавяне на нова функционалност или при промяна на вече съществуваща такава.
* Автоматично- използва се тестовата рамка Jasmine, чиято роля е да създава и пуска непрекъснато регресивни тестове (regression testing). Важността й е свързана с постоянна проверка на съществуващи функционалности и осигуряването на тяхната правилна употреба. Рамката е предпочитана, защото работи добре с Angular, добре документирана е и поддържа асинхронното тестване. Синтаксисът й е ясен, което прави тестовете много по- четими и лесни за изпълнение. Също така, Microsoft Azure включва множество инструменти за тестване (напр. Azure Load Testing и други).
* *Обратна връзка от потребителя*- получаването на обратна връзка от потребителя е от ключово значение за развитието на платформата. В системата е добавена контактна форма, както и контакти за връзка с екипа по поддръжка. Контактната форма служи за получаване на запитвания и докладване на проблеми.

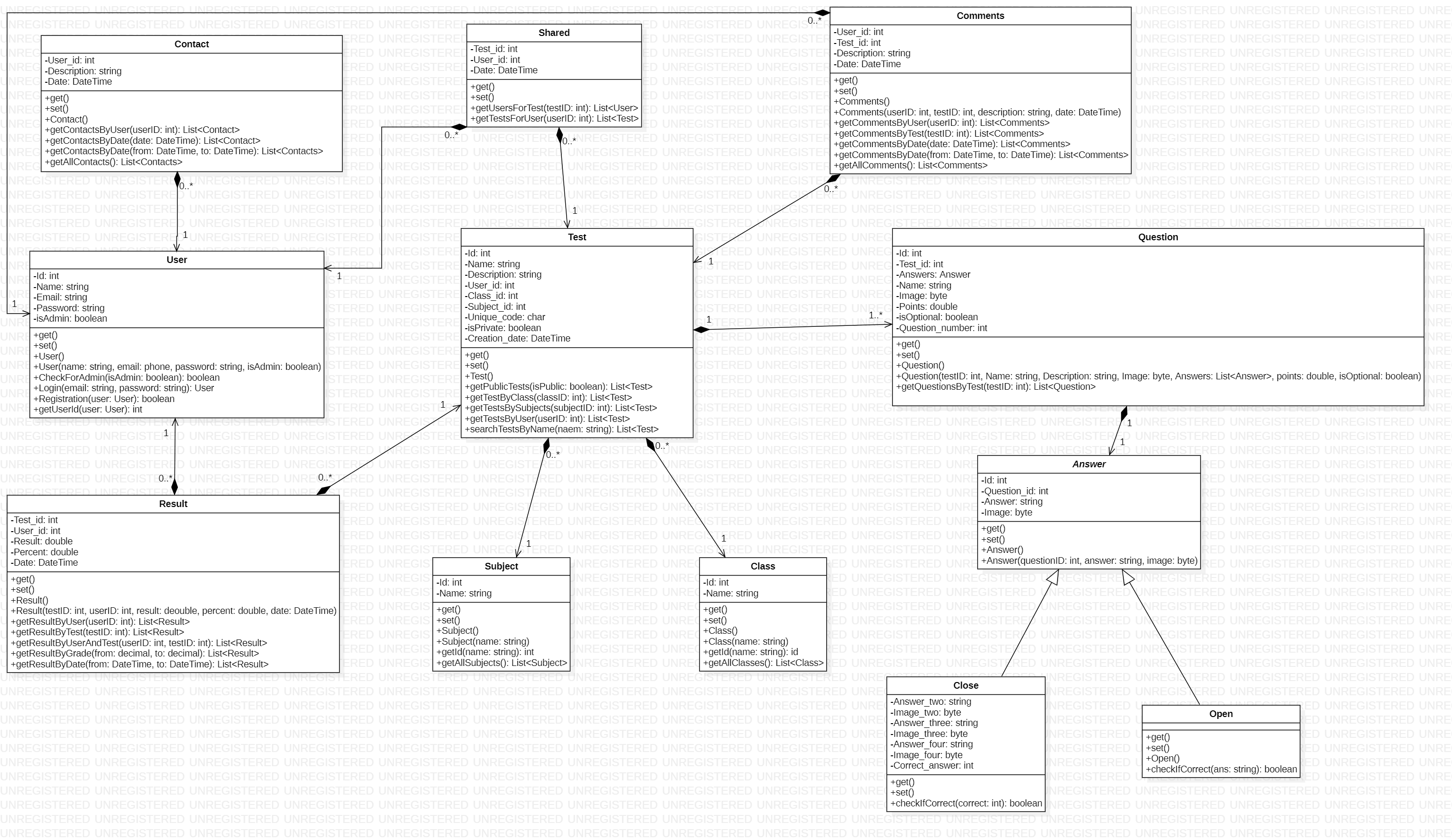
1. **Други изисквания**

***Приложение А: Терминологичен речник***

DB/ БД- база данни

Quiz Mater- името на приложението

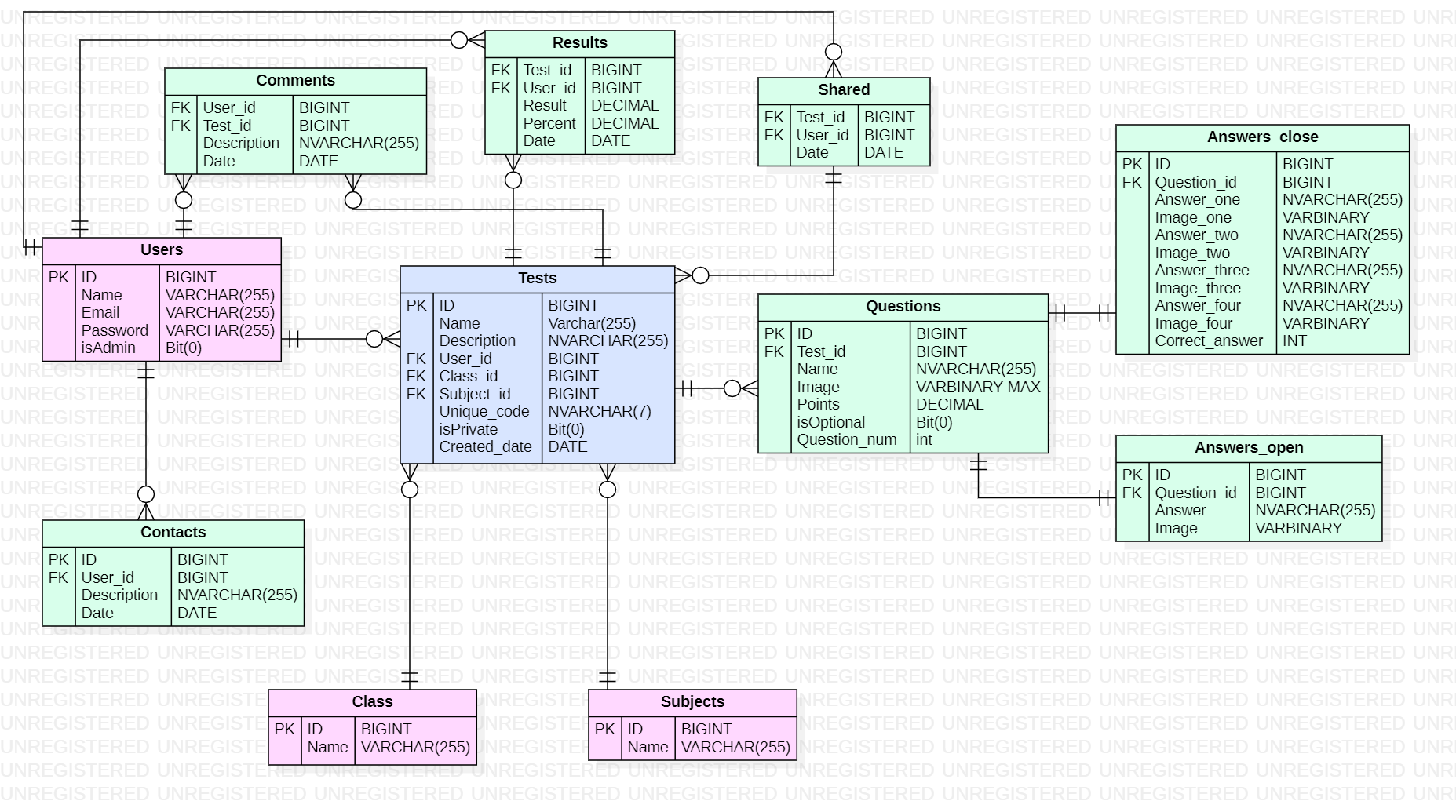
***Приложение B: Аналитични модели***



# Фигура 5: Class Diagram

Клас диаграмата се състои от 12 класа.

* Клас User- в него се съхранява информация за потребителя. Записани са и някои от основните методи, необходими за работата на приложението като функцията за вход, регистрация в системата и проверка за правата на потребителя.
* Клас Contact- съдържа основната информация, която се получава след попълване на контактната форма. С помощта на методите getContactsByDate() се дава възможност да се филтрират заявки направени в определен ден или период. Също така има възможност за филтриране по потребител (чрез неговият идентификационен номер).
* Клас Subject- съдържа името на предмета, за който е предназначен теста.
* Клас Class- съдържа името на Класа.
* Клас Test- включва информацията за всеки тест- име, описание, потребител, който е създал теста, за кой клас и предмет е предназначен, код за достъп, тип и дата на създаване. Различните методи помагат за филтрирането на тестове по клас, предмет, потребител. Реализиран е метод, с помощта на който се определя дали даден тест е публичен или не, както и възможност за търсене на тест по ключова дума.
* Клас Question- съхранява информация за теста, към който принадлежи въпроса, текста на самия въпрос, снимка, отговорите, точките, номера на въпроса в теста, както и типа. Методът getQuestionsByTest() връща списък с въпроси, които са свързани с даден тест.
* Клас Answer- абстрактен клас, който съдържа информацията, която се препокрива за двата типа отговори- отворени и затворени.
* Клас Open- съхранява отговорите на отворените въпроси.
* Клас Close- отговорен е за затворените въпроси. Методът checkIfCorrect() проверява дали избраният от потребителя отговор за даден въпрос е верен.
* Клас Comments- съхранява информацията за коментарите, които потребителят поставя под тестовете. Дава се възможност за търсене на коментари по потребител, тест, конкретна дата или период от време.
* Клас Shared- съхранява информация за тестовете, които потребителите са споделили помежду си. Методът getUsersForTest() връща списък с всички потребители, които имат достъп до определен тест. А getTestsForUser() връща всички тестове, които са били споделени с даден потребител.
* Клас Result- съхранява информация за резултатите, които потребителят е получил след решаване на даден тест. Методът getResultByUser() връща списък с всички резултати на даден потребител. GetResultByTest() връща всички резултати, свързани с определен тест. GetResultByUserAndTest() връща резултатите, които определен потребител е получил за даден тест. GetResultByGrade()- връща лист с резултати, които попадат в строго дефиниран диапазон. GetResultByDate()- резултатите от всички тестове, решени в определен период от време.



# Фигура 6: Entity Relationship Diagram

За изграждането на приложението ще са необходими 11 таблици.

* Таблица за Потребители- тя ще съдържа основната информация за потребителите на приложението- име, имейл, парола и поле, което да удостоверява дали потребителя е админ или не. Таблицата е определена като проста, предвид сравнително малкия брой на атрибутите, които се съхраняват за всеки запис, както и броя на записите.
* Таблица за Класове- тя е другата таблица, която е определена като проста. Съхранява името на класа.
* Таблица за Предмети- таблица, съдържаща предметите, към които се отнасят съответните тестове. Предвид сравнително простата си структура, таблицата е класифицирана като проста.
* Таблица за Контакти- това е таблица, която съдържа информация, която потребител е изпратил като обратна връзка към админа чрез контактната форма. Тя съдържа информация за потребителя, който я е попълнил, съдържанието (описанието) на самата форма, както и датата на отправеното запитване. Предвид структурата на таблицата, тя е класифицирана като междинна.
* Таблица за Тестове- класифицирана като сложна, тази таблица съдържа основна информация за самия тест. В нея се записва името на самия тест; допълнително описание към него (ако има такова); има външен ключ към потребителя създател, както и към класа и предмета, за които е предназначен самия тест; уникален код, който служи за споделяне на частни тестове между няколко потребителя; дали теста е частен или публичен; датата на създаване.
* Таблица за Въпроси- класифицирана като междинна, тази таблица съхранява въпросите, които са включени във всеки от тестовете. Съдържа информация за текста на въпроса, изображение, към него (ако има такова), точки, които носи, типа на въпроса- задължителен или не, номер на въпроса в теста, както и към кой тест се отнася.
* Таблица за Отговори на Затворените въпроси- тази таблица е определена като междинна. Съдържа външен ключ към въпроса, към който се отнася съответния запис, както и четирите възможни отговора, заедно с изображения за всеки от тях. Важно е да се отбележи, че само първите два отговора (т. е. отговор едно и отговор две или изображение едно и изображение две) са задължителни за попълване. Записва се и верният отговор на въпроса, под формата на цяло число от 1 до 4, което показва кой отговор е верен.
* Таблица за Отговори на Отворените въпроси- таблицата е определена като междинна. Тя включва въпроса, за който се отнася отговора, самия отговор, както и изображение, което може да бъде прикачено.
* Таблица за Резултати- таблицата е определена като междинна. Съдържа данни за теста, който е решен; потребителя, който го е решил; резултата, който е получил; процента, на вярно отговорени въпроси, както и датата на решаване на теста.
* Таблица, съхраняваща Споделените Тестове- това е таблицата, която дава възможност на потребителя да има достъп до даден тест (създаден от друг потребител), който е определен като частен. По този начин, след като веднъж един потребител даде код на друг, то вторият може да достъпва въпросника неопределен брой пъти. Също така, потребителят създал теста може да проследи, кой има достъп до него. Таблицата включва идентификатора на теста, който се споделя, потребителя, който има достъп до него, както и датата на споделяне.
* Таблица за Коментари- тази таблица е определена като междинна. Съхранява данни за потребителя, създал коментара, теста, за който е създаден, самия коментар, както и дата на добавяне.